Tasksheet 5:

Task 1. Implementation of the Implicit Euler Method.

In class we considered the approximate solution of initial value problems of the form

$$\frac{dx}{dt} = f(t, x(t))$$

with initial condition $x_t)0) = x_0$ via the Implicit Euler method. Implement this method for application to the Logistic equation

$$\frac{dP}{dt} = \alpha\ P - \beta\ P^2$$

with $P(0) = P_0$. Your code should accept as input

- the growth constant, $\alpha > 0$,
- the limiting constant, $\beta > 0$, and
- the initial value, P_0, to start the simulation.

Note: Use your shared library to solve the root finding problem that arises in the algorithm. You should start by using your Newton method code to solve the problem at each step. You will need to provide the Newton method with the derivative of the right hand side of the ordinary differential equation.

Test your codes for some given values of the parameters:

- test 1: $\alpha = 0.2$, \beta=0.0005, and $P_0 = 10.0$
- test 2: $\alpha = 0.01$, \beta=0.0005, and $P_0 = 10.0$
- test 3: $\alpha = 2.0$, \beta=0.0005, and $P_0 = 10.0$

Task 2. An Analytic Solution of the Logistics Equation.

Provide the details of the derivation of an analytic solution for the Logistic equation

$$\frac{dP}{dt} = \alpha\, P - \beta\, P^2$$

for the parameters α, β, and P_0. Write a routine/code that will produce the analytic solution value at any point, $t \geq 0$. You will need the routine to compare the explicit and implicit Euler methods to a numerical approximation below.

Task 3. Comparison of the Solutions

Provide a comparison of the explicit and implicit Euler methods applied to the logistic equations. Use matplotlib.py to produce a graph that includes the analytic solution of the Logistic equation for the methods and the analytic solution of the problem. Use your code from previous tasks to perform the work. Discuss the graphical results you present in working on this task.

Task 4. Numerical Integration - The Trapezoid Method

Write a code that implements the composite trapezoid method for computing the definite integral

$$\int_a^b f(x)\, dx$$

The code should have as input the function, $f(x)$, the endpoints of the interval of integration, a and b, and the number of intervals to use in the approximation. The output should be a single real number, double or float, depending on your choice of coding language. Test your code on the following problem

$$\int_0^{\frac{\pi}{4}} e^{-x^2}\, dx$$

with $n = 2, 4, 8, 16$. Indicate the value the sequence if converging to.

Task 5. Simpson's Rule and a Computational Convergence Study

Repeat the same work as in Task 4 for Simpson's rule. In addition, perform a computational convergence rate for Simpson's rule. Do this by using more intervals in the approximation. That is, use a sequence of values for $h = \frac{1}{4}, \frac{1}{8}, \frac{1}{16} \ldots$ for 8 different sizes. Fit the data to the log-log plot of the results using the error term

$$|error| \leq \ C \ h^p$$

to determine an estimate of the accuracy of the method.