Homework 05

General comment: Matlab has built in numerical integration methods. For example `quad` is one of them. You might check validity of your implementations with answers produced by this algorithm.

Of course it is always better when you do it vs the analytically calculated integral.

Problem 1 (5 points)

Implement the rectangle numerical integration method. Call you function `rectInt(f,a,b,N)`, where a and b limits of integration, N the number of points, and f is handle to the function. Do not forget to test it!

Problem 2 (5 points)

Implement the trapezoidal numerical integration method. Call you function `trapezInt(f, a,b,N)`, where a and b limits of integration, N the number of points, and f is handle to the function. Do not forget to test it!

Problem 3 (5 points)

Implement the Simpson numerical integration method. Call you function `simpsonInt(f, a,b,N)`, where a and b limits of integration, N the number of points, and f is handle to the function. Remember about special form of N=2k+1. Do not forget to test it!

Problem 4 (5 points)

Implement the Monte-Carlo numerical integration method. Call you function `montecarloInt(f, a,b,N)`, where a and b limits of integration, N the number of points, and f is handle to the function. Do not forget to test it!

Problem 5 (5 points)

Plot the absolute error of integration of the above 4 methods vs different number of points N. Try to do it from small N=3 to N=10^6. Use `loglog` plotting function for better representation (make sure that you have enough points in all areas of the plot). Why error start to grow with larger N? Does it grows for all methods? For your tests calculate

$$\int_0^{10} [\exp(-x) + (x/1000)^3] dx$$