MATH 131: Numerical Methods for scientists and engineers - Assignment 3

MATLAB Grader Homework Assignment 3 due by 11:45 PM, October 23, 2019. Test your answers on MATLAB Grader or Live Script on MATLAB.

1. Consider the function $f(x) = (x^2 - 3x + 2)\arctan x$. Estimate the derivative of the function at the point $x = 0$ using each of the three formulas listed below for all of the following values of h: $h = 10^{-n}$, $n = 1, 3, 6$. Make a `loglog` plot of your error verses h (use the same plot for the three methods).

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h} \qquad \text{(forward difference)}$$
$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0 - h)}{2h} \qquad \text{(3 pt centered difference)}$$
$$f'(x_0) \approx \frac{-f(x_0 + 2h) + 8f(x_0 + h) - 8f(x_0 - h) + f(x_0 - 2h)}{12h} \qquad \text{(5 pt centered difference)}$$

   For $n = 1, 3, 6$ call `FD1`, `FD3`, `FD6` the result obtain with the forward difference, `CD31`, `CD33`, `CD36` the result obtain with the 3 point centered difference, and `CD51`, `CD53`, `CD56` the result obtain with the 5 point centered difference. Plot your error for all three methods on the same graph. Explain your findings. Is the error decreasing with h? Why or why not? Which method has the largest error? Which method has the smallest error? Why? Write those answers in comment using `%`.

2. Create a function called `composite_trapezoid` that inputs a function, a pair of endpoints, a, b, and a number n of subintervals, and outputs the approximation to the integral of f from a to b using the composite trapezoid rule. Your function header should look like this

```
function I = composite_trapezoid(f,a,b,n)
```

   Use that function to compute the integral $I = \int_0^\pi \cos\left(\frac{\pi t^2}{2}\right) dt$, for the number of points $n = 10^3, 10^5$. Call `I1` and `I2` the obtained results. The exact answer to this integral can be computed using the Fresnel cosine integral in Matlab: **`fresnelc`**`(x)` $= \int_0^x \cos\left(\frac{\pi t^2}{2}\right) dt$. *Compute the value on Matlab and copy the result with the LONG format*. Compute the error in your computation for each n and for each method. Call `Err1` and `Err2` those errors. Make a `loglog` plot of error verses the number of points for both results on the same plot. Comment on the result using `%`.

3. Create a function called `composite_simpsons` that inputs a function, a pair of endpoints, a, b, and a number n, and outputs the approximation to the integral of f from a to b using the Composite Simpson's rule on $n + 1$ points. Your function header should look like this

```
function I = composite_simpsons(f,a,b,n)
```

   Use that function to compute the integral $I = \int_0^\pi \cos\left(\frac{\pi t^2}{2}\right) dt$, for the number of points $n = 10^3, 10^5$. Call `I3` and `I4` the obtained results. The exact answer to this integral can be computed using the Fresnel cosine integral in Matlab: **`fresnelc`**`(x)` $= \int_0^x \cos\left(\frac{\pi t^2}{2}\right) dt$. *Cody Coursework might not recognize this function, compute the value on Matlab and copy the result with the LONG format*. Compute the error in your computation for each n and for each method. Call `Err3` and `Err4` those errors. Make a `loglog` plot of error verses the number of points. Comment on your results: which method works best between Trapezoid and Simpson's ? Which method works the worst? Why? Write your answer as a comment using `%`.