

Introduction to Scientific Computing, Homework #5
Due by 12 pm on Thursday Jun. 2, 2022

Problem 1 [20 Pts]: Calculate symbolic operations of the following functions using MATLAB. Include the code use to generate the output and the output.

- i) $\frac{d(f(x,y))}{dx}$ where $f(x,y) = \cos(x)\sin(y) + y^3 \log(x)$
- ii) $\text{div}(F)$ with respect to (x,y,z) where $F(x,y,z) = (2\cos^2(x), \sin^3(y), 2x^2 + y^2 + 4z^2)$
- iii) ∇f where $f(x,y,z) = x^2 + 2y^2 + 4z^2$
- iv) $\int f(x)dx$ where $f(x) = x^2 - 5x$
- v) $\int f(x)dx$ where $f(x) = x^{(2-n)} - 5x$
- vi) $\iiint f(x,y,z)dx dy dz$ where $f(x,y,z) = \sin(x)\cos(y)\tan(z) + y\cos(x) + x\sin(y)$ (triple integration)

Problem 2 [20 Pts]: Consider the following function.

$$f(x) = \sin(x) + \cos^2\left(x^2 - \frac{\pi}{7}\right)$$

- i) Plot $f(x)$ using `ezplot` over the domain $-5\pi \leq x \leq 5\pi$. Modify the linestyle of the plot using code. Include axis titles.
- ii) Use MATLAB's built-in trapezoidal integration function (`trapz`) to integrate the area under $f(x)$ over the domain $-5\pi \leq x \leq 5\pi$. Vary the number of integration divisions from 5 to 500 by steps of one and produce a plot of integrated area as a function of the divisions used during integration. Make sure your plot is pleasing.
- iii) Integrate $f(x)$ over the domain $-5\pi \leq x \leq 5\pi$ using the numerical integration function `integral`. How does the result returned in (ii) compare to that of (iii)? How many divisions are necessary in (ii) to converge to the solution in (iii)?

Problem 3 [20 pts]: You are to create a function that takes in a vector of x points and a vector of y points and fits the data with the polynomial of the lowest degree, n_{crit} , that meets or exceeds a specified coefficient of determination (aka, R squared or R^2). Your function will have the declaration

```
function [R2out,pOrder,pCoeff] = discoverDegree(x,y,R2crit)
```

where x and y are vectors of length $1 \times L$ or $L \times 1$, $R2_{\text{crit}}$ is the critical coefficient of determination that must be met by the fit, $R2_{\text{out}}$ is the actual coefficient of determination returned by the polynomial fit of degree n_{crit} , $pOrder$ is equal to n_{crit} , and $pCoeff$ is a vector of the coefficients for n_{crit} of the form $[p_n, p_{n-1}, p_{n-2}, \dots, p_0]$.

You cannot use pre-written MATLAB commands to determine R^2 . Instead, the coefficient of determination should be calculated from the fit and data according to the formula

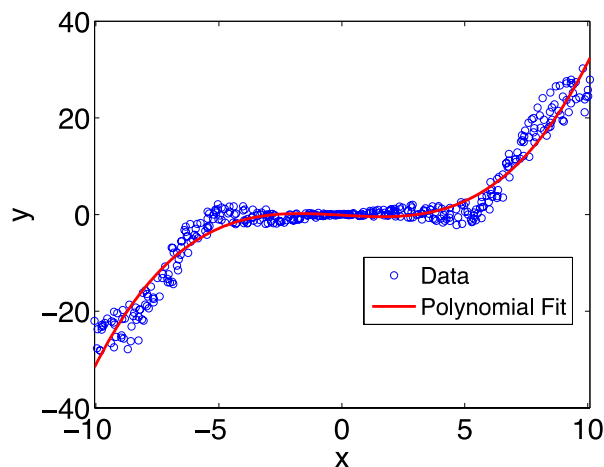
$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2}$$

where y_i are the data set values at locations x_i , f_i are the values of the fit function corresponding to each x_i , and

$$\bar{y} = \frac{1}{L} \sum_{i=1}^L y_i$$

You will find x and y data to test your function on the assignment callout in the file `poly.mat`.

Create a script that calls your `discoverDegree` function to test your function on the data contained in file `poly.mat`. Produce a pretty plot like the one shown below with a fit determined by `discoverDegree` for $R^2 \geq 0.85$. Include a legend and x/y labels in your plot.



Include your function, script, and plot in your .docx submission.

Answer the following questions:

- What polynomial degree is necessary to achieve $R^2 \geq 0.85$ for the data in `poly.mat`.
- What polynomial degree is necessary to achieve $R^2 \geq 0.95$ for the data in `poly.mat`.

Problem 4 [20 pts]: You had been playing a lot of cornhole (<http://en.wikipedia.org/wiki/Cornhole>) before each NCAA game. However, you were a horrible shot! You realize that you could improve your performance next season by calculating the velocity with which you need to throw your bean bag to land inside the hole. The perfect part is – you have already developed a simulation of a projectile for problem 4 of homework 4. This function simply needs to be adapted to the cornhole game and run through an algorithm to determine the proper throwing velocity.

You will implement a golden section search to determine the optimal initial x -component of velocity to get within the tolerance `tol` of the cornhole. You will assume that the cornhole is some positive distance in x from your starting position of $(x,y) = (0,0)$. Your search function will return the optimal x velocity and have the following function declaration.

```
function vxOpt = goldCornhole(g,c,x0,y0,vxa,vxb,vy,tstep,dis,tol)
```

where g is the gravitational constant (9.8 m/s^2), c is the coefficient of damping (taken to be 0.2), vxa is the lower bound of your x velocity, vxb is the upper bound of your x velocity, vy is the y component of velocity, $tstep$ is the time step of the dynamic simulation, dis is the distance to the cornhole, and tol is the maximum difference between where the throw lands and the cornhole location to be considered a successful shot.

You will have to slightly modify your `basketball()` function for use in these simulations. In particular, your simulation should run until the projectile (your bean bag) hits the ground. Your new function declaration will be of the following form.

```
function disX = projectileSim(g,c,x0,y0,vx0,vy0,tstep)
```

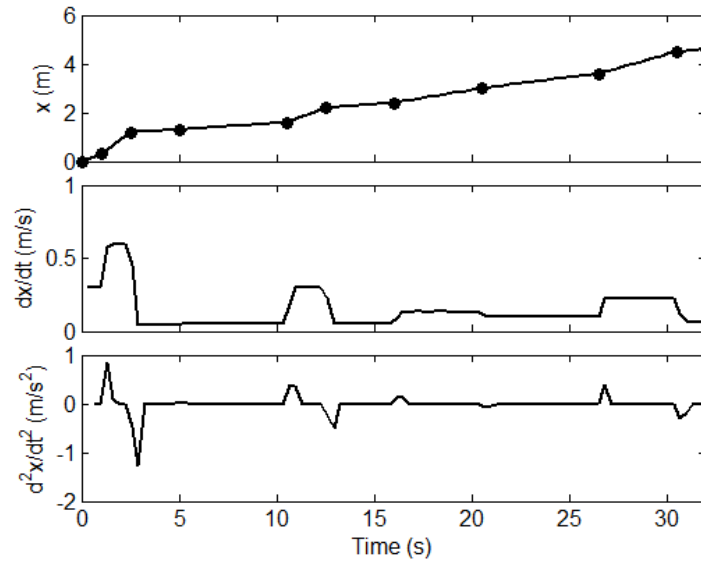
where $vx0$ is the initial x component of velocity, $vy0$ is the initial y component of velocity, and $disX$ is the distance the projectile has traversed from the location of the throw to its landing spot.

Problem 5 [20 Pts]: You have entered a turtle race with your pet turtle Speedy. You have been working with Speedy to increase his speed using positive vocal encouragement (“go Speedy”) and strawberries placed along the race course. You want to determine if your strawberry placement could be adjusted to improve Speedy’s performance.

In order to do this, you collected the following time (seconds) and position (meters) data from Speedy’s last race in which you placed the majority of the strawberries at the 1-meter mark of the race.

```
t = [0, 1, 2.5, 5.0, 10.5, 12.5, 16, 20.5, 26.5, 30.5, 32]
x = [0, 0.3, 1.2, 1.3, 1.6, 2.2, 2.4, 3.0, 3.6, 4.5, 4.6]
```

You decide to analyze and plot this data using MATLAB. To do this, you create a script that interpolates the race data to 100 equally spaced points from the beginning to ending time of data collection, calculates the velocity and acceleration of Speedy using the interpolated position values, and plots the collected position data, interpolated position, velocity, and acceleration in a format like that shown below. Multiple axes like that shown below should appear on a single figure, all plot commands should be coded at the script level, and the command `subplot()` should not be used. Measured data should be plotted as data markers and the interpolated data as connected lines.



You will perform analysis on interpolated data using (1) linear and (2) spline interpolation.

You will use structures / fields to keep track of the data sets you need to create. For instance, the measured data points might be placed in variables `t.raw` and `x.raw` and the velocity based on linearly interpolated data in the variable `dx.interp.linear`.

Include a copy of your plots on the .docx submission