Math 458, Fall 2022, USC

Instructor: Steven Heilman

Final Exam, Part 2

This exam contains 5 pages (including this cover page) and 4 problems. Enter all requested information on the top of this page.

You may use your books, notes, or a calculator on this exam. You may **not** use the internet or any internet-enabled device. You **cannot** consult with any person, other than yourself, when completing the exam.

The following rules apply:

- This exam is due at 11:59PM PST, December 10, 2022 (31 hours to complete the exam).
- Your final submission of the exam will be a **single** Matlab .m file, submitted in blackboard, under the Assignments tab. (No zip files, no folders, just a single .m file.) This file should be named LastnameFirstname.m . For example, if I submitted the final, the filename would be HeilmanSteven.m .
- **Comment your code** to a reasonable extent, so it is clear what you are doing.
- The desired output of each problem is a **single Matlab figure**.
- When your submitted Matlab file is run in Matlab, it should produce 4 **distinct** figures. You can use distinct `figure` commands to create different figures.
- If you need to make multiple plots on one figure, use the **`hold on`** command, and/or the `subplot` command. Good luck![a]

Problem	Points	Score
1	10	
2	10	
3	10	
4	10	
Total:	40	

- If your Matlab code does not produce an output as specified here and below, you will receive severe point deductions.
- Lateness will be severely penalized.
- If you do not follow the rules on this page correctly, your grade could be severely penalized.
- I am not answering any questions less than 12 hours before the due date.

1. (10 points) Create Matlab code to compute the following functions as precisely as possible (i.e. with the least amount of numerical errors) on the desired domains.

	function	domain D	subplot command
(a)	$f(x) = \sqrt{x + \frac{1}{x}} - \sqrt{x - \frac{1}{x}}$	$[10^5, 10^5 + 1]$	`subplot(2,2,1)`
(b)	$f(x) = 1 + \cos(x)$	$[\pi - 3 \cdot 10^{-8}, \pi + 3 \cdot 10^{-8}]$	`subplot(2,2,2)`
(c)	$f(x) = \frac{e^x \sin^2(x)}{1 - \cos(x)}$	$[10\pi - 10^{-7}, 10\pi + 10^{-7}]$	`subplot(2,2,3)`
(d)	$f(x) = \frac{\sin(\cos(x))}{\frac{\pi}{2} - x}$	$[\frac{\pi}{2} - 10^{-8}, \frac{\pi}{2} + 10^{-8}]$	`subplot(2,2,4)`

In the figure your code generates for this problem, you should create four plots in one figure, using the subplot command. In each subplot, plot the exact formula that appears in the corresponding row in the above table for f as a blue curve, and then plot your "better" formula for f as a red curve.

The domain of your plot should use 1000 equally spaced points.

The range of plotted values (i.e. the values specified in the `axis` command that specifies the range of values in the y axis) should be $[\min_{x \in D} f(x), \max_{x \in D} f(x)]$ for each plot (where you use the exact formula written above for f to determine the max and min values.)

2. (10 points) In this question we will model a bouncing ball dropped on a flat surface. Let $y(t)$ denote the vertical position of a rubber ball at time $t \geq 0$. At time $t = 0$, we have $y(0) = 1$ and $y'(0) = 0$, so that the ball is held stationary at 1 meter above the ground, and it is then dropped at time $t = 0$. As the ball is falling, it is under the influence of gravity, so that $y''(t) = -9.80665$ for any time t such that the ball does not hit the floor.

   Let $0 < k < 1$. Before the ball hits the floor, it is moving downwards, so its velocity is negative ($y'(t) < 0$ for any times t right before the ball hits the floor). The moment right after the ball hits the floor, its velocity becomes $-k$ times the velocity it had before it hit the floor. (For example, if $k = .9$, after bouncing, the ball has 90% of the speed it had before it hit the ground.) Put another way, the collision with the ground results in a speed loss proportional to the previous speed of the ball.

   So far, we have neglected air friction. In order to incorporate air friction, we replace the assumption that $y''(t) = -9.80665$ with the assumption that $y''(t) = -9.80665 - cy'(t)$ for some constant $c \geq 0$, for any time t where the its first and second derivative exist. (Here $c = 0$ corresponds to no air friction and $c > 0$ corresponds to air friction.)

   - Using `subplot(2,2,1)`, plot the bouncing ball's position $y(t)$ from time $t = 0$ to time $t = 10$, when $k = .9$ and $c = 0$. The vertical axis should have values from 0 to 1. The horizontal axis should be time t. In this same plot, also plot the bouncing ball when $k = .9$ and $c = 1$.
   - Using `subplot(2,2,2)`, plot the bouncing ball's position $y(t)$ from time $t = 0$ to time $t = 10$, when $k = .8$ and $c = 0$. The vertical axis should have values from 0 to 1. The horizontal axis should be time t. In this same plot, also plot the bouncing ball when $k = .8$ and $c = 1$.
   - Using `subplot(2,2,3)`, plot the bouncing ball's position $y(t)$ from time $t = 0$ to time $t = 30$, when $k = .9$ and $c = 0$ on the moon (so that the acceleration due to gravity is -1.625 instead of -9.80665). The vertical axis should have values from 0 to 1. The horizontal axis should be time t.
   - Using `subplot(2,2,4)`, plot the bouncing ball's position $y(t)$ from time $t = 0$ to time $t = 30$, when $k = .8$ and $c = 0$ on the moon (so that the acceleration due to gravity is -1.625 instead of -9.80665). The vertical axis should have values from 0 to 1. The horizontal axis should be time t.

   When you generate these plots, you can use whatever differential equation solution method you want, though you should be mindful that your solution method should produce a realistic outcome. For example, you should choose a step size that leads to realistic plots.

3. (10 points) Let $y_0 \in \mathbf{R}$. Consider the initial value problem (IVP)

$$\begin{cases} y'(t) = (y(t))^2 - t, & \forall\, 0 \le t \le 4 \\ y(0) = y_0. \end{cases}$$

(i) Using `subplot(2,2,1)`, plot solutions to the initial value problem using a second-order Euler's method with a step size of $h = 4/1000$. On this plot, you should plot all solutions for a range of values of y_0 from -1 to 1 in increments of .01. (So in total the plot should display 201 different curves.) In your plot, adjust the axes with the command `axis([0,4,-2,3])`.

Let $y_h(4)$ denote the computed output of your program that approximates the exact value of $y(4)$ when you run your Euler's method solver with a step size of h. Use the quantity $|y_h(4) - y_{2h}(4)|$ as an estimate for the error $|y_h(4) - y(4)|$. How small do you have to take h in order to guarantee that your error estimate is smaller than 10^{-4}, over the range of initial values you considered previously? (This is a rhetorical question, you do not have to provide an answer to this question.) Judging from your experimental observations, is it true that $|y_h(4) - y(4)| < 10^{-4}$ when $h = 4/1000$ for all y_0 values that you chose in part (i)? If this is true, enter the title `yes` into `subplot(2,2,1)`. Otherwise, enter the title `no` into `subplot(2,2,1)`.

(ii) You should observe that there is one particular choice $z \in [-1, 1]$ for y_0 such that solutions of the IVP with $y_0 > z$ behave much differently than solutions of the IVP with $y_0 < z$. Find the value of z within five decimal places of accuracy. (Hint: there is more than one way to do this. One relatively quick way is to solve the ODE by reversing time.) Using `subplot(2,2,2)`, plot the solution y of the IVP when $y_0 = z$, and in the title of this plot, report the value of z to five decimal places of accuracy. In your plot, adjust the axes with the command `axis([0,4,-2,3])`.

(iii) Using `subplot(2,2,3)`, give a brief text description of how you found y_0 in part (ii). To input this text in the subplot, use the commands

```
subplot(2,2,3)
text(0,0,{'my explanation appears here','perhaps need','multiple lines'})
axis([0 1 -1 1])
```

4. (10 points) Consider the three-dimensional boundary value problem.

$$\begin{cases} \begin{pmatrix} \frac{d^2}{dt^2}x(t) \\ \frac{d^2}{dt^2}y(t) \\ \frac{d^2}{dt^2}z(t) \end{pmatrix} = \begin{pmatrix} x(t) - z(t) \\ y(t) - 2z(t) \\ x(t) + y(t) - 3z(t) \end{pmatrix}, & \forall\, t \geq 0. \\ \begin{pmatrix} x(0) \\ y(0) \\ z(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\ \begin{pmatrix} x(T) \\ y(T) \\ z(T) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, & \text{for some } T > 0. \end{cases}$$

- Solve this problem using whatever method you wish to use. (If you use an ODE solver make sure to use a step size h such that $h \leq 1/100$.) (Your solution does not have to take the exact value $(1, 1, 1)$ at time T. It is acceptable if you find a solution that is reasonably close to $(1, 1, 1)$.)
- When you solve this problem, record the value of T you used.
- Plot the solution you found using the `plot3` command (i.e. plot the path $(x(t), y(t), z(t)) \in \mathbf{R}^3$ where $0 \leq t \leq T$). Report the value of T you found in the title of the figure. Make sure to label the axes accordingly using `xlabel('x')`, `ylabel('y')` and `zlabel('z')`. Also format the axes with `axis([0 1 0 1 0 1])`.

(You should not use the `subplot` command in this problem. The plot you generate should be a normal sized figure with no subplots.)