Due Friday, March 18th at 11:59pm

Problem 1

In real life we are never given a differential equation to solve. We must find the differential equation based on what we know about the process and physical laws. Let's look at the tank problem that we did in Week 8, but this time with a slight twist.

Two tanks with capacities of 10 liters initially contain 2 grams of salt and 1 liter of water. Water containing 1 g/L of salt enters the first tank at a rate of 2 L/hour, and the well-mixed solution flows out of the first tank into the second tank at a rate equivalent to the volume of the first (i.e., V_1 L/Hr). None of the brine flows out of the second tank.

To model the process follow the steps in the lecture for each tank separately. Notice that the rate for the second tank will depend on the first.

(1) Model the problem for the volume in liters as an IVP(s) and by using `ode45` (for MATLAB) or `solve_ivp` (for Python) solve the IVP (ODE + IC) for the volume of water in the tanks. Solve from time $t = 0$ to $t = 6$ with $\Delta t = 0.01$. Save the volume of the first tank at each time as a 601×1 column vector named `A1` and the volume of the second tank at each time as a 601×1 column vector named `A2`.

(2) At what time to the nearest integer does the water start overflowing (hint: use the `round` function)? Save this integer value as `A3`. And which tank does it overflow from (1 or 2)? Save this integer value as `A4`.

(3) Model the problem for the amount of salt in grams as an IVP(s) and by using `ode45` (for MATLAB) or `solve_ivp` (for Python) solve the IVP (ODE + IC) for the amount of salt in the tanks. Solve from time $t = 0$ to $t = 6$ with $\Delta t = 0.01$. Save the amount of salt of the first tank at each time as a 601×1 column vector named `A5` and the amount of salt of the second tank at each time as a 601×1 column vector named `A6`.

(4) What is the salt concentration when tank that overflowed in part (b) is completely full of brine? Save this salt concentration (remember it's amount/volume) as `A7`.

Problem 2

Suppose the initial probability of finding a quantum particle on the line from $x = -1$ to $x = 1$ is a compact Gaussian. The particle has equal probability of going to the left or to the right, and has the ability to leave the region. What is the probability that the particle is located at $x = 0.5$ at $t = 1$?

This can be solved via the following PDE:

$$\frac{\partial P}{\partial t} = \frac{\partial^2 P}{\partial x^2}; \quad P(t=0,x) = \exp\left(1 - \frac{1}{1-x^2}\right); \quad P(t, x=-1) = P(t, x=1) = 0. \tag{1}$$

We will use the Crank-Nicolson scheme to numerically solve the problem.

(1) Following the finite difference process we used in lecture, use a second order difference scheme to approximate P_{xx} and rewrite this as a linear system of equations $A\mathbf{x} = \mathbf{b}$. Here the matrix A will be the usual tridiagonal matrix from finite differences, and the vector $\mathbf{b}$ will change at each timestep. Use $\Delta t = 0.01$ and $\Delta x = 0.01$. Save a copy of A in a variable named `A8`. Save a copy of $\mathbf{b}$ as a 199×1 column vector after the first iteration in a variable named `A9`.

Solve this linear system at each time `t = [0:0.01:1]` using Gaussian elimination (the backslash operator in MATLAB or the `solve` function in python). Save a copy of $\mathbf{b}$ as a 199×1 column vector after the last iteration in a variable named `A10`.

Save the probability of finding the particle at $x = 0.5$ at $t = 1$ as `A11`. Find the exact probability of finding the particle at $x = 0.5$ at $t = \infty$ and save it as `A12` (Hint: do you see a pattern as you iterate the problem?)

Problem 3

Download the files `CP5_M1.mat`, `CP5_M2.mat`, `CP5_M3.mat` from our Canvas page. Then load the files onto MATLAB with the following commands:

```
load CP10_M1.mat
load CP10_M2.mat
load CP10_M3.mat
```

or Python with the following commands:

After running these commands the variables M1, M2, and M3 will be defined. It is your task to figure out which one represents each of U, S and V. [Hint: You should be able to see a clear picture]. After you find which matrices are U, S, and V, let `data` $= USV^T$.

(1) Calculate how many megabytes it would take to store the matrix `data`. That is, calculate how many entries there are in this matrix and multiply by `8 / 1e6`. Save your answer in a variable named `A13`.

(2) Find the number of singular values that capture more than 99% of the information in this image. That is, find the smallest value of k such that

$$E_k = (m * k + k + n * k)/(m * n) > 0.99,$$

where m is the number of rows and n is the number of columns of `data`. Save the number you found for k in a variable named `A14`.

(3) Calculate how many megabytes it would take to store this new matrix the same way you did in part (1). Save your answer in a variable named `A15`.

(4) Use the Week 10 coding lecture to guide you in displaying the image that this SVD encodes using the `uint8` and `imshow` commands on MATLAB. Save the integer corresponding to the word that best describes this image as `A16`.

1) porter, 2) pig, 3) profit, 4) quiet, 5) quicksand, 6) cast, 7) quince, 8) fang, 9) aftermath, 10) experience, 11) ticket, 12) linen, 13) creator, 14) look, 15) trouble, 16) teeth, 17) dog, 18) arm, 19) basin, 20) toad, 21) dinner, 22) position, 23) snail, 24) snake, 25) alarm, 26) ocean, 27) act, 28) argument, 29) cable, 30) cattle, 31) ink, 32) reward, 33) pump, 34) university, 35) church, 36) kittens, 37) government, 38) umbrella, 39) doll, 40) secretary, 41) spring, 42) bird

It may also be instructive to reconstruct the image using 1,2,3,... of the singular values and see how many are really needed to fully capture the image. Since this part is subjective, it will not be part of the submission.