



## EN40: Dynamics and Vibrations

### Homework 1:

### Mupad and MATLAB practice

Due 12:00 noon Friday February 1 (online submission)

School of Engineering  
Brown University

- Your solution to this homework should consist of two files:
  1. A commented MATLAB Live Script stored as a .mlx file
  2. A commented MATLAB function stored as a .m file
- Please submit the assignment electronically on the EN40 canvas website. You can log into canvas at <http://brown.edu/it/canvas/> (the login link is near the top right of the page). You can find some instructions on Canvas use (from CIT) at <https://ithelp.brown.edu/kb/articles/canvas-for-students>

### Part 1: Write a Matlab ‘Live Script’ to solve the following problems.

1. Find all the solutions to the simultaneous equations

$$x^2 + y^2 + z^2 = 4 \quad x + xy + z = 1 \quad x - y - z = 1$$

2. Plot the function

$$P(x, \sigma) = \frac{1}{\pi x} \left[ \frac{\sigma}{\log(x)^2 + \sigma^2} \right]$$

(log denotes a natural log) in the range  $0.01 < x < 3$ , for  $\sigma = 0.1$ ,  $\sigma = 0.5$ ,  $\sigma = 1$  (on the same plot). ( $P$  is the ‘log-Cauchy’ distribution function - it is used in modeling statistics associated with HIV infections and [population statistics](#). A few points of credit are awarded for making the plot look nice.

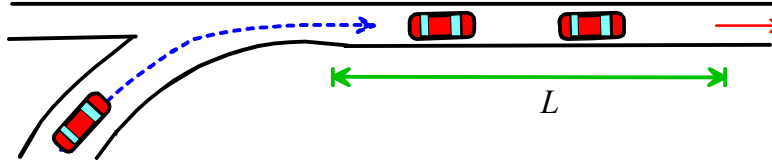
3. Evaluate the integral

$$\int_0^x P(\xi, \sigma) d\xi$$

Assume that  $\sigma > 0$  and  $\sigma$  is real.

4. Find the value of  $x$  that maximizes  $P(x, \sigma = 0.5)$  (you can find potential maxima by differentiating  $P$  with respect to  $x$ ; then solving  $dP/dx = 0$ . MATLAB will give you two solutions. Your plots in problem 3 should help you see whether the solutions correspond to maxima, minima, or inflections).

5. Find the maximum value of  $P(x, \sigma = 0.5)$  (use your answer to 4 and the ‘subs’ function)



6. The differential equation

$$\frac{dN}{dt} = N_0 \left( 1 - N(t) \frac{c}{L} \right) - \frac{N(t)}{L} V_0 \left( 1 - N(t) \frac{c}{L} \right)$$

estimates the total number of cars on a length of (one way) road.  $N_0$  is the number of cars per second arriving to enter the road;  $V_0$  is the speed limit;  $L$  is the length of the road, and  $c$  is the average length of a car (the first term is the number of cars entering the road; the second is the number leaving the road).

6.1 Use the 'dsolve' function to solve the differential equation for  $N$  as a function of time. Use the initial condition  $N(t) = 0$   $t = 0$

6.2 Plot graphs (on the same figure) of  $N(t)$  for  $L=500\text{m}$ ;  $c=3\text{m}$ ,  $N_0 = 1/10 \text{ s}^{-1}$  for (i)  $V_0 = 11\text{m/s}$   $V_0 = 20\text{m/s}$   $V_0 = 45\text{m/s}$ , for a time interval of  $0 < t < 300$  sec.

**Part 2: Please solve the remaining problems using MATLAB** (write your code in a matlab .m file). You should make your MATLAB (.m) file a function, so that when the file is executed, it will solve all the homework problems. For example:

```
function EN40_Homework1_2019
    code that will solve problems 7-11
end
function B = update_matrix(A)
... Code
end
```

(You might find the solutions to homework 1, 2009-2018 helpful, if you get stuck – you could use the 2018 HW as a template, if you like.)

7. Using a loop, or dot notation, or the 'linspace' function, create a vector  $t$  of 501 equally spaced points between  $-\pi$  and  $\pi$

8. Using the solution to problem 8, create two vectors  $x$  and  $y$  that contain values of the function

$$x = C(t) \quad y = S(t) \quad -\pi < t < \pi$$

where  $C$  and  $S$  are the '[Fresnel integrals](#)'. You don't need to do the integrals yourself -  $C$  and  $S$  are hard-coded in MATLAB as the functions `fresnelc()` and `fresnels()`; you can use them just like `sin()` and `cos()`.

Hence, plot a graph of  $y-v-x$ . (The curve is called the 'Cornu Spiral'. You will get a few points for making your plot look nice!)

9. A grayscale image in MATLAB is stored as a matrix  $M$ . Each entry in the matrix specifies the color of a pixel in the image: if  $M(i,j)=1$  the pixel is white; if  $M(i,j)=0$  the pixel is black, and if  $0 < M(i,j) < 1$  the pixel is gray. The indices  $i,j$  identify the position of the pixel, so for example  $M(1,1)$  is the pixel at the top left corner of the image.  $M(10,1)$  is the pixel 10 rows down on the left hand side of the image; and  $M(1,10)$  is the pixel 10 columns to the right at the top of the image.

You can display a matrix using the 'imshow' command, e.g.

```
imshow(M, 'Initialmagnification', 200)
```

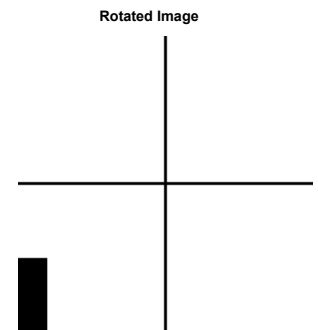
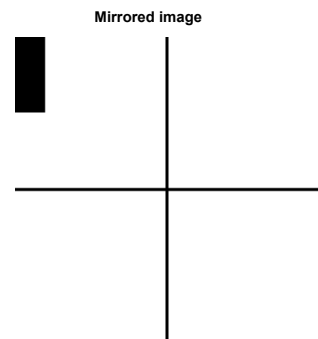
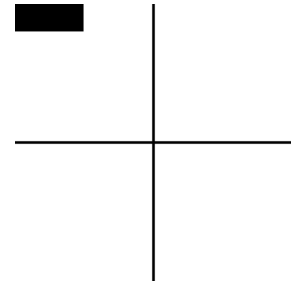
(The 'Initialmagnification' is optional, it just makes the figure larger)

9.1 Create a 101x101 matrix that will produce the image shown in the picture and display it. The black rectangle is 10 pixels high and 25 pixels long. The following commands will be useful:

- (a)  $M = \text{ones}(100)$  creates a 100x100 matrix with 1 in every entry of the matrix
- (b)  $M(1:10,1:10) = 0$  (eg) will set the top 10x10 block of a matrix to zero (try it and see what happens!)
- (c)  $M(:,1) = 0$  will set the first column of a matrix to zero (the  $:$  is shorthand for all the rows)

9.2 Using your solution to 9.1, create a new image that is a mirror image of the first one about the diagonal (you need to switch rows and columns of the matrix – remember there is a matlab command that does this)

9.3 Create and display a new image that rotates the first one counterclockwise by 90 degrees. Write your code so it will rotate any image, rather than just changing the entries in the matrix by hand. (Hint: the pixel at  $(i,j)$  in the first image will be located at  $(102-j,i)$  in the new one. You can create the new matrix by using two nested loops)



10. The goal of this problem is to code a simple version of a two-dimensional ‘[Potts model](#),’ which is used in physics and engineering to predict how the structure of various different kinds of material evolve with time. Applications range from studies of ferromagnetism, to motion of interfaces in polycrystals.



One version of the model idealizes a material as a rectangular grid of cells, which can each adopt one of two possible states (in magnetism, eg, these represent positive or negative spin). These are stored in an  $n \times n$  matrix  $A$  with elements  $A(i,j)$  either  $-1$  or  $+1$ . The simulation starts with a random distribution of  $\pm 1$  in the cells, which then flip their states so as to lower the total energy of the system, as described below. By repeatedly flipping the states of the cells, the simulation predicts how magnetic domains or microstructures form in the material.

10.1 Write a MATLAB function of the form  $B = \text{update\_matrix}(A)$  that takes as input a square  $n \times n$  matrix  $A$  with elements  $A(i,j)$  either  $-1$  or  $+1$ ; and returns a second matrix  $B$  (with the same size as  $A$ ).  $B$  should be calculated inside your function using the following steps:

- (i) Set  $B=A$
- (ii) Find the number of rows  $n$  and columns  $m$  of  $A$  (use the ‘size’ function)
- (iii) Repeat the following calculation  $n^2$  times (use a loop):
  - (a) Generate two random numbers  $i$  and  $j$  between 2 and  $n-1$ . You can do this with (eg for  $i$ , use something similar for  $j$ )  
 $i = \text{randi}(n-2)+1;$

$$(b) \text{ For the current value of } i,j, \text{ calculate } E = \left[ 1 - A(i,j) \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} A(k,l) \right]$$

(you could use the matlab ‘sum’ function to calculate  $E$  – but read the manual to be sure how it works, it doesn’t work the way you would expect. Or you could use two nested loops over the values of  $k$  and  $l$  (this is a bit slower but works fine), or if you hate loops just type in all 9 terms in the double sum by hand). In the physics of the Potts model,  $E$  represents the energy of the cell located at  $i,j$  – it has a low energy if all the neighbors have the same sign as the one at  $i,j$ , and a high energy if they all have a different sign.

- (c) Calculate  $B(i,j)$  from the following formula (use a conditional statement)

$$B(i,j) = \begin{cases} A(i,j) & E < 0 \\ -A(i,j) & E \geq 0 \end{cases}$$

10.2 Test your code using the following lines of script (which should be in your main homework function). You should see an animation of the microstructure forming a set of domains. You can compare your predictions with [these images of magnetic domains](#).

```
M = 2*randi(2,101)-3;
M(:,1) = 0;
M(1,:) = 0;
M(101,:) = 0;
M(:,101) = 0;
figure
for step=1:100
    M = update_matrix(M);
    imshow((M+1)./2, 'Initialmagnification',200)
    pause(0.1)
end
```

11. The ‘[Higgins-Selkov](#)’ equations are a simple model of the chemical reactions involved in [glycolysis](#) (the metabolic pathway that generates energy from glucose). The equations were intended to explain experimental observations of repetitive fluctuations in the concentrations of the metabolites. They assume that the concentrations of fructose-6-phosphate  $S$  and ADP  $P$  are given by

$$\frac{dS}{dt} = v_0 - k_1 SP^2$$

$$\frac{dP}{dt} = k_1 SP^2 - k_2 P$$

where  $v_0$  is the rate of production of  $S$  (fructose-6 phosphate);  $k_1$  is a reaction constant that specifies the rate of the autocatalytic reaction of  $S$  and  $P$  to produce more  $P$  (ADP), and  $k_2$  is a reaction constant that controls the rate of reaction of ADP in the next step of the glycolysis reaction.

11.1 Write a MATLAB function of the form `dw/dt = glycolysis(t,w,v0,k1,k2)` function that computes the vector of time derivatives  $d\mathbf{w}/dt = [dS/dt; dP/dt]$  given a value of  $t$  and the current values of  $\mathbf{w} = [S; P]$  and the values of the constants  $v_0, k_1, k_2$ .

11.2 Hence, use the MATLAB `ode45` function to calculate and plot the variation of  $S(t), P(t)$  with time.

Use the following parameters:

- Initial conditions  $S = 1, P = 0.8$  (mols/cc) at time  $t=0$ ;
- Time interval  $0 < t < 100$  min
- Plot  $P$  and  $S$  as a function of time for  $v_0 = 1$  mols/cc/min,  $k_1 = 1$  mols/cc/min,  $k_2 = 1.01$  1/min, for a time interval  $0 < t < 100$  min. You should see the concentrations fluctuating.
- On a separate figure, Plot  $P$  and  $S$  for  $k_2 = 1.075$  1/min. For this reaction rate there are no fluctuations.