Homework (50 pts) ……………………

<u>(Chapter 6 –Lent book) Answer and submit in one script file the</u>.

<u>Write the program and outputs as directed in the questions below:</u>

Q2- (5 pts)- Decaying circle

Q4- (5 pts)- Lissajous

Q7- (10 pts)- Epicycles

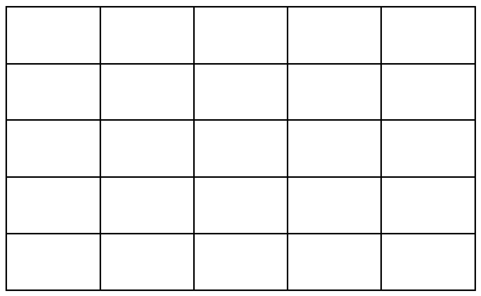
Q8- (10 pts)- A random walk in two dimensions

Q9- (10 pts)- Vertical motion and energy

Q10-(10 pts)- Two dimensional billiards

Q7(5 pts):

Write a meshgrid command to create the grid below with the following coordination: the left and right lower- nodes are (1, 10) and (1,15) respectively, while the left and right upper - nodes are (6,10) and (6, 15). Draw the out put as mesh plot and show the output of the meshgrid command



Q8 (5pts):

Rotate a polygon ABCD a quarter, half, full and 5 cycles around the origin if A(0,0), B(4, 1), C(2, -4), D(-1, -3).

Q9 (10 pts):

What is quiver function? Create a meshgrid for a vector [-3,3], then compute the vector field R (include both Rx and Ry). Plot the quiver image if the velocity of the position vector (V) = |R|. sin(Φ). Make any necessary assumptions. Read page 243-245 of the text book for more explanation (attached below).

Q10- (10 pts)

Using dumbbell program, rotate three spheres connected as triangle. Make any necessary assumption.

PROGRAMMING PROBLEMS

1. Consider a (very small) class with the following students and their GPA's.

Student name	GPA
Alfonso Bedoya	3.43
Tonya Harding	2.77
Warren Harding	2.30
Warren Piece	3.25

Write a MATLAB program, `ProcessStudents.m`, that constructs `students`, a one-dimensional array of structures with fields named `firstname`, `lastname`, and `gpa`. It should then loop through each element in the array and call a function `DisplayStudentRecord(theStudent)` (which you also need to write) that displays data for each student on the screen. This function should take a single student structure as an argument.

```
function DisplayStudentRecord(theStudent)
% function Display_StudentRecord(theStudent)
% displays theStudent.firstname, theStudent.lastname,
%          and theStudent.gpa
```

2. Write a MATLAB program, `ShowPoem.m`, that constructs `poemlines`, a one-dimensional cell array of strings. Use an appropriate brief poem or part of a poem of your choosing. It should call a function (which you also need to write) `DisplayCellText` that displays the poem on the screen. The function should take a single-cell array of strings as an argument.

```
function DisplayCellText(castring)
% function DisplayCellText(castring)
%      input castring is a cell-array of string
%      the function displays each line of castring
%        to form a block of text on the screen
```

3. Magic 8Ball. A Magic 8Ball uses an icosahedral die floating in a dark liquid to randomly produce an answer to a yes-or-no question. Write a program `MagicEightBall.m` that simulates this. Use a cell array of strings to store the twenty possible responses shown as follows and randomly select an answer to display to the user.

It is certain	It is decidedly so	Without a doubt
Yes—definitely	You may rely on it	As I see it, yes
Most likely	Outlook good	Yes
Signs point to yes	Reply hazy, try again	Ask again later
Better not tell you now	Cannot predict now	Concentrate and ask again
Don't count on it	My reply is no	My sources say no
Outlook not so good	Very doubtful	

4. **Break into words.** Write a function `wordsca=breakIntoWords(s)` that returns a cell array of strings containing the words in the given string. Words are here defined as any set of non-blank characters separated by blanks. You may want to use the built-in function `strtrim` that removes leading and trailing white space.
5. **Reverse words.** Write a function `sout=reverseWords(s)` that returns a string containing the words in the original string in reverse order, separated by blanks.
6. **Scramble words.** Write a function `sout=scrambleWords(s)` that returns a string containing the words in the original string in random order, separated by blanks. (Use the function `randperm`.)

```
% demoSurfaces
%     Author:  Khan Noonien Singh

%% set parameters
Nx=60;
xmin=-3;
xmax=+3

%% make grid
x=linspace(xmin, xmax, Nx);
y=linspace(xmin, xmax, Nx);
[X, Y]=meshgrid(x, y);

%% calculate function z=f(x, y)
Z=(X.*Y).*exp(-(X.*X+Y.*Y));

%% plot surface
figure(1)
mesh(X, Y, Z, ones(Nx))
xlabel('x')
ylabel('y')
zlabel('z')

figure(2)
surfl(X, Y, Z)
colormap gray;
shading interp
xlabel('x')
ylabel('y')

figure(3)
contour(X, Y, Z, 25)
xlabel('x')
ylabel('y')

figure(4)
pcolor(X, Y, Z)
colormap gray;
shading interp
xlabel('x')
ylabel('y')
```

13.4 Plotting vector fields

A vector field in two dimensions defines both a direction and a magnitude at each point in the plane. The vector field $\vec{v}(x, y)$ can be defined by its two Cartesian components, $v_x(x, y)$

and $v_y(x, y)$. To visualize the vector field, one typically constructs a rectangular mesh with `meshgrid`, and then computes each component of the vector field. The `quiver` command creates an array of arrows, one at each mesh point, which indicate the direction and relative magnitude of the vector field. The syntax is:

```
quiver(X, Y, Vx, Vy)
```

`Vx` and `Vy` are arrays of the same size as `X` and `Y` containing the horizontal and vertical components of the vector at each mesh point.

As an example, for each point in the plane define the vector $\vec{r}$ from the origin to the point, it's magnitude r, and the unit vector $\hat{r}$.

$$\vec{r} = (x, y) = r\hat{r}$$

where

$$r = |\vec{r}| = \sqrt{x^2 + y^2} \quad \text{and} \quad \hat{r} = \vec{r}/|r|$$

The unit vector $\hat{\theta}$ is at each point perpendicular to $\vec{r}$ in the counterclockwise direction.

$$\hat{\theta} = (-y, x)/r$$

We define a velocity vector at each point that points in the $\hat{\theta}$ direction and whose magnitude is a function of distance from the origin.

$$\begin{aligned} v_m &= \sin(2\pi r/\lambda) r e^{-r/a} \\ \vec{v}(x, y) &= v_m \hat{\theta} \end{aligned}$$

The following program computes this vector field and plots the results with `quiver`, as shown in Figure 13.8.

```
% DemoQuiver.m
%   Author: Heino Vanderjuice
%% set parameters
Nx=30;
a=1.5;
lambda=5;
```

```
%% define grid
x=linspace(-4, 4, Nx);
[X, Y]=meshgrid(x, x);
```

```
%% compute vector field
Rmag=sqrt(X.*X+Y.*Y);
ThetaHatx=-Y./Rmag;
ThetaHaty=X./Rmag;
```

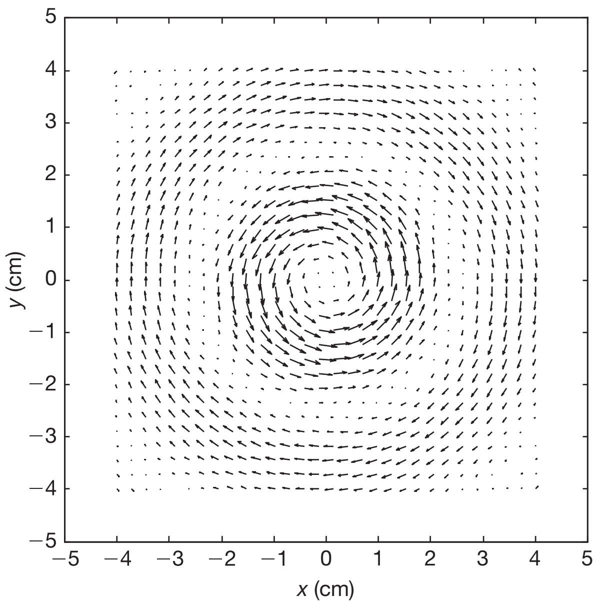


FIGURE 13.8 The vector field calculated by `DemoQuiver` and imaged with the `quiver` command.

```
Vmag=sin(2*pi*Rmag/lambda).*Rmag.*exp(-Rmag/a);
Vx=Vmag.*ThetaHatx;
Vy=Vmag.*ThetaHaty;
```

```
%% plot vector field
quiver(X, Y, Vx, Vy, 'k');
axis square
xlabel('x(cm)')
ylabel('y(cm)')
```

13.5 Working with images

Importing and manipulating bit-mapped images

Bit-mapped images represent a two-dimensional image by an array of numbers indicting the color or intensity of light in each pixel. A rectangular array of pixels form an approximate representation of the spatial variation in intensity and/or color over the plane of the image. A simple grayscale image represents only the light intensity, so a 150×300 image is represented by a matrix of numbers with 150 rows and 300 columns. So-called RGB color images use an $N \times M \times 3$ array, with each of the three $N \times M$ planes representing a different color: red, green, and blue.