Homework and ICE Answers and Analysis

Liu Yazhuo

Homework 1

Problem 1:

1. Calculate the monthly payment (P) for a loan using the following formula: $$P(N) = \frac{rL(1+r/12)^{12N}}{12\{(1+r/12)^{12N}-1\}}$$

where N is the number of years used to pay back the loan, r is the interest rate, and L is the loan amount. Set r to 15%, L to $50,000, and vary N from .5 to 20 years. If you enter your formula correctly P(20) = 658.39. Make sure to add comments explaining what the formula is doing.

2. Plot the monthly payment vs. the number of years (make sure you have enough data points to make a smooth curve).
3. Use the "text" command to print your name on the plot. Search MATLAB's help files for information if needed.

```
%% problem1
r = 0.15;                   % interest rate in Percentage
L = 50000;                  % loan amount in Dollar
N = (6:240)/12;             % number of years
% Calculate the monthly payment (P) for a loan
P = r*L*(1+r/12).^(12*N)./(12*((1+r/12).^(12*N)-1));

% test for the value of P(20) is correct or not
if abs(P(1,end)-658.39) <= 0.1
else
    fprintf('Wrong answer')
end

plot(N,P);
hold on
text(10,6000,'Name');      % print my name on the figure
title('the monthly payment P vs. the number of years N');
xlabel('number of years');
ylabel('monthly payment');
```

Due to the computer storage accuracy, it is difficult for two numbers to be absolutely equal, so they are often judged to be equal by the absolute value of the difference being less than a certain threshold.
(*eps* or other value)

An **anonymous function**（匿名函数） is a very simple, one-line function. The advantage of an anonymous function is that it does not have to be stored in an M-file. This can greatly simplify programs, since often calculations are very simple, and the use of anonymous functions reduces the number of M-files necessary for a program. Anonymous functions can be created in the Command Window or in any script. The syntax for an anonymous function is:

fnhandle = @ (arguments) functionbody

where fnhandle stores the function handle; it is essentially a way of referring to the function. The handle is assigned to this name using the @ operator. The arguments, in parentheses, correspond to the argument(s) that are passed to the function, just like any other kind of function. The functionbody is the body of the function, which is any valid MATLAB.

Try not to define symbolic functions and then substitute numerical values, because symbolic functions are very slow to operate while running.

$$P(N) = \frac{rL(1+r/12)^{12N}}{12\{(1+r/12)^{12N}-1\}}$$

$$\Downarrow$$

```
P =@(N) r*L*(1+r/12).^(12*N)./(12*((1+r/12).^(12*N)-1));
```

Problem 2:

Create another section to do the following. Add a comment at the end of each line detailing what each line does. Make sure command outputs are not suppressed so outputs are published in your final document (as well as to the command window).

a. Create the matrix A

b. assign the first row of A to a vector called x1

c. assign the last 2 rows of A to an array called y

d. assign the even-numbered columns of A to an array called B

e. assign the transpose of A (i.e. turns it into a 4-by-3 array) to C

f. compute the reciprocal of each element of A

g. change the number in column 2, row 3 of A to 100.

$$A = \begin{bmatrix} 20 & 4 & 2 & 6 \\ 6 & 37 & 2 & 3 \\ 8 & 5 & 9 & 9 \end{bmatrix}$$

```
%% problem2
A = [20 4 2 6 ; 6 37 2 3 ; 8 5 9 9]
x1 = A(1,:)                    %assign the first row of A to x1
y = A(end-1:end,:)             %assign the last 2 rows of A to y
B = A(:,2:2:end)               %assign the even-numbered columns of A to B
C = A'                         %assign the transpose of A to C
1./A                           %compute the reciprocal of each element of A
A(3,2) = 100                   %change the number in column 2, row 3 of A to 100
```

Problem 3:

a. Create an empty figure.

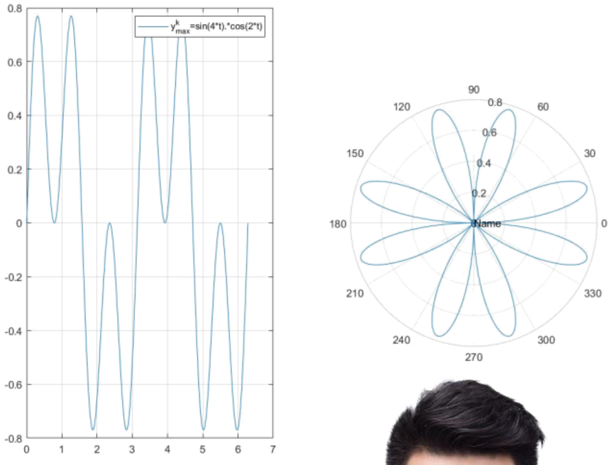
b. Plot $y_{max}^{k} = \sin(4t)\cos(2t)$ for $0 \le t \le 2\pi$ using subplots to generate normal and polar plots. Make sure there is enough data points to make a smooth plot.

c. Add a legend with the formula plotted with correctly formatted subscripts and superscripts if applicable to the xy plot.

d. Turn on the grid lines for the xy plot.

e. Add your name to the polar plot at the origin (0,0) using the text function

f. If done correctly your figure should look like this



```
%% Problem 1
t = linspace(0,2*pi,500);
y = sin(4*t).*cos(2*t);

figure();
subplot(1,2,1);
plot(t,y);
grid on;
axis([0,7,-0.8,0.8]);
legend('y^{k}_{max}=sin(4t).*cos(2t)','Location','NorthEast’);

subplot(1,2,2);
polarplot(t,y);
text(0,0,'Name’);
hold off;
```

- For the **legend** function, we usually set the value of its “Location" parameter to “Best".
- Note that in the input parameters of the **polarplot** function, *theta* is the first and *rho* is the last.

Problem 4:

Write code, in the following order, to convert inches into both centimeters and mm.

a. Prompt the user to enter a number.

b. Using **fprintf()**, output a string, using a complete sentence that contains:

The number the user just entered

The number converted to cm (there are 2.54 cm/in)

c. Using **disp()**, output a string, using a complete sentence that contains:

The number the user just entered

The number converted to mm

d. Make sure ALL the numbers have 2 numbers after the decimal point.

e. When run, the command window may look like :

Enter a number: 5

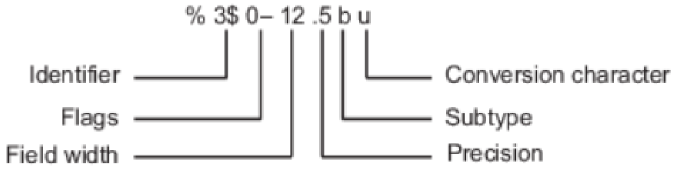
5.00 inches is 12.70 cm

5.00 inches is also 127.00 mm

```
%% Problem 2
num = input('Enter a number: ');
numCM = 2.54*num;
fprintf('%.2f inches is %.2f cm\n',num,numCM);
numMM = numCM*10;
answer = [num2str(num,'%.2f'),' inches is also ',num2str(numMM,'%.2f'),' mm'];
disp(answer);
```

conversion characters

值类型	转换	详细信息
整数，有符号	%d 或 %i	以 10 为底
整数，无符号	%u	以 10 为底
	%o	以 8 为底（八进制）
	%x	以 16 为底（十六进制），小写字母 a - f
	%X	与 %x 相同，大写字母 A - F
浮点数	%f	定点记数法（使用精度操作符指定小数点后的位数）。
	%e	指数记数法，如 3.141593e+00（使用精度操作符指定小数点后的位数）。
	%E	与 %e 相同，但为大写，如 3.141593E+00（使用精度操作符指定小数点后的位数）。
	%g	%e 或 %f 的更紧凑形式，没有尾随零（使用精度操作符指定有效数字位数）。
	%G	%E 或 %f 的更紧凑形式，没有尾随零（使用精度操作符指定有效数字位数）。
字符或字符串	%c	单一字符
	%s	字符向量或字符串数组。输出文本的类型与 formatSpec 的类型相同。



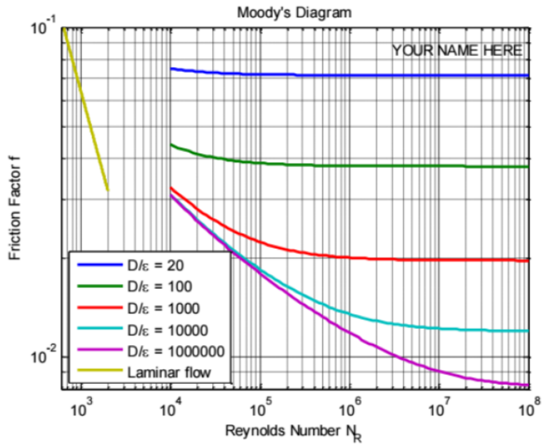
Value Type	Conversion	Details
Integer, signed	%d or %i	Base 10 values
	%ld or %li	64-bit base 10 values
	%hd or %hi	16-bit base 10 values
Integer, unsigned	%u	Base 10
	%o	Base 8 (octal)
	%x	Base 16 (hexadecimal), lowercase letters a–f
	%X	Same as %x, uppercase letters A–F
	%lu %lo %lx or %lX	64-bit values, base 10, 8, or 16
	%hu %ho %hx or %hX	16-bit values, base 10, 8, or 16
Floating-point number	%f	Fixed-point notation
	%e	Exponential notation, such as3.141593e+00
	%E	Same as %e, but uppercase, such as3.141593E+00
	%g	The more compact of %e or %f, with no trailing zeros
	%G	The more compact of %E or %f, with no trailing zeros
	%bx or %bX %bo %bu	Double-precision hexadecimal, octal, or decimal value Example: %bx prints pi as 400921fb54442d18
	%tx or %tX %to %tu	Single-precision hexadecimal, octal, or decimal value Example: %tx prints pi as 40490fdb
Characters	%c	Single character
	%s	String of characters

This is a more complete form.

Moody's diagram as shown on the next page is a famous plot used to determine the effect internal friction (surface roughness) has on fluids flowing in pipes. The equation below has been developed by Jain and Swamee for the friction factor (f) for turbulent pipe flow.

$$f = \frac{0.25}{\left[\log\left(\frac{1}{3.7(D/\varepsilon)} + \frac{5.74}{N_R^{0.9}}\right)\right]^2}$$

a. Plot the friction factor following function for values Reynolds number (N_R) between 10^4 and 10^8 for D/ε values of 20, 100, 1000, 10,000, and 100,000. Your plot should look similar to example below. A couple of hints:
   a. Notice that both axes are log scaled
   b. log( ) in MATLAB does not do what you think it does
   c. The logspace( ) function may be a helpful in generating data that is evenly spaced on log-scaled axes (similar to linspace( )).
b. Add a line for $f = 64/N_R$ for smooth pipes to the same plot (be sure to match the N_R range shown in the figure).
c. Add a complete title and x and y axis labels (including any of the subscripts or superscripts).
d. Add a legend with the actual Greek character epsilon for each trace (e.g. D/ε = 1000, etc.). It can be in any position on the figure.
e. Adjust the axis limits so it looks like the figure on the last page.
f. Use the "text" command to print your name anywhere on the plot.

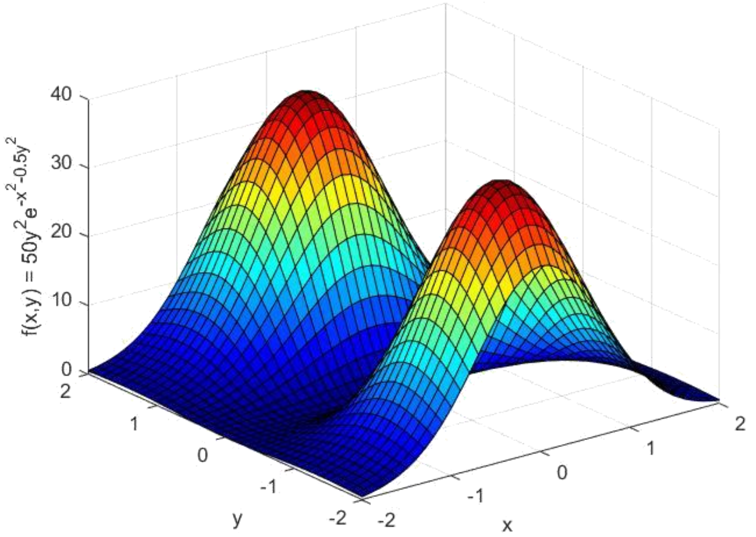


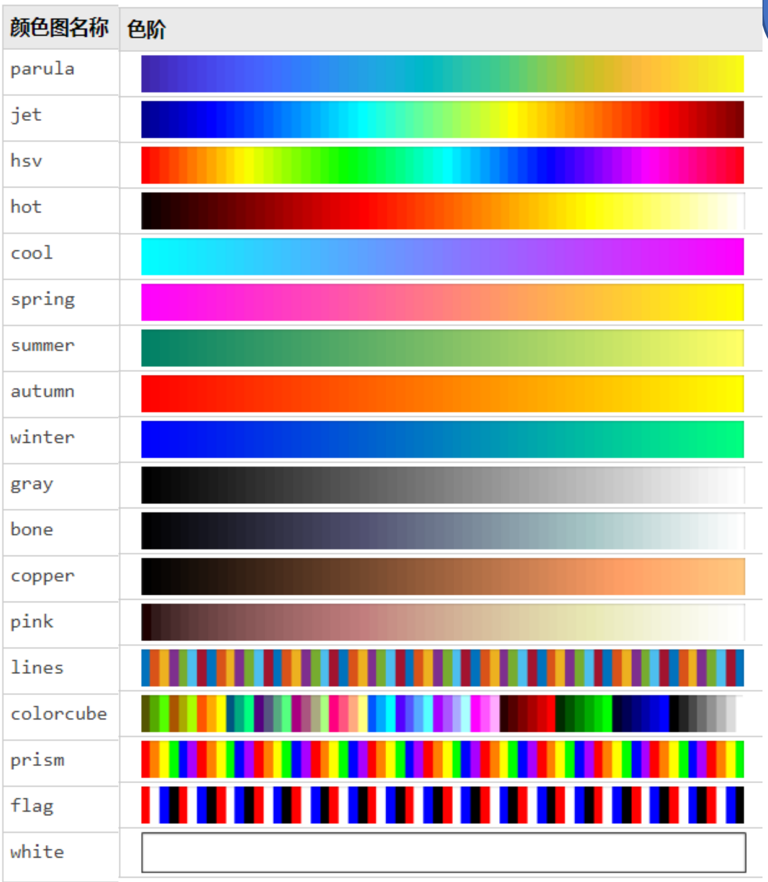

```
%% problem 3
figure()

NR1 = logspace(4,8,1000);
f = @ (De,NR)
0.25./(log10(1/(3.7*De)+5.74./NR.^0.9)).^2;
for x = [20,100,1000,10000,100000]
    loglog(NR1,f(x,NR1),'linewidth',2);
    hold on;
end

NR2=600:1:2000;
fLaminar = 64./NR2;
loglog(NR2,fLaminar,'linewidth',2);
grid on;
title('Moody''s Diagram');
xlabel('Reynolds Number N_R');
ylabel('Friction Factor f');
legend('D/\epsilon = 20','D/\epsilon = 100',...
    'D/\epsilon = 1000','D/\epsilon = 10000',...
    'D/\epsilon = 100000','Laminar
flow','Location','southwest');
xlim([600,10^8]);
text(10^7.5,0.09,'Name');
hold off;
```

Problem 1:

1. Plot $f(x,y) = 50y^2e^{-x^2-0.5y^2}$ for $-2 \le (x,y) \le 2$. Use the step size of 0.1 for x and y.
2. Put a descriptive title and axis labels on the figure as shown. Include your name in the title.
3. If you plot it correctly, it will look like the figure below except that you will need to a different title.

```
%% problem 4
[xgrid,ygrid] = meshgrid(-2:0.1:2,-2:0.1:2);
z = 50*ygrid.^2.*exp(-xgrid.^2-0.5*ygrid.^2);
figure();
surf(xgrid,ygrid,z);
xlabel('x');ylabel('y');
zlabel('f(x,y) = 50y^{2}e^{-x^{2}-0.5y^{2}}');
title("Who's 3D ploting");
colormap jet;
```





Here, the last line of code is the most important, which is to adjust the color of the image to match the color in the example diagram.

MATLAB在创建每一个图形对象时，都会给该对象分配一个唯一确定的值，称其为图形对象句柄。

（1）计算机屏幕句柄默认为0。

（2）图形窗口对象的句柄值为一正整数，并显示在窗口标题栏中。

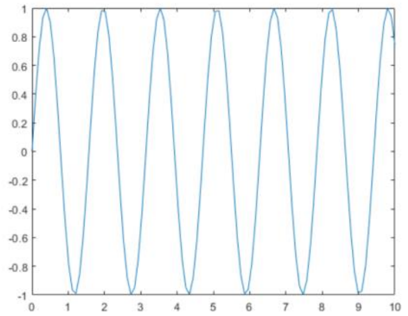
（3）其他图形对象的句柄为浮点数。

Function	Handle
gcf	获取当前图形窗口句柄
gca	获取当前坐标轴的句柄
gco	获取最近被选中的图形对象的句柄
findobj	按照指定的属性来获取图形对象的句柄

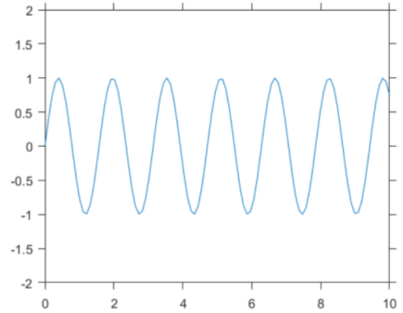
也可以通过get和set函数对句柄进行操作，例如

```
ans=get(gca,'XLabel’);
set(ans,'String','Values of X axis');
```

```
x = linspace(0,10); y = sin(4*x); plot(x,y)
```




```
ax = gca; % current axes
ax.FontSize = 12; ax.TickDir = 'out’;
ax.TickLength = [0.02 0.02];
ax.YLim = [-2 2];
```



如果你希望自己画出来的图像大方美观，那么操作句柄将会是你必不可少的技能。这里只是简要介绍，具体练习还希望大家能够多查一查帮助文档，对着里面的实例练习。

ICE for week 2
Solutions and Analysis

Problem 1 Write a function called even_index that takes a matrix, M, as input argument and returns a matrix that contains only those elements of M that are in even rows and columns.

```
function a=even_index(M)
    [rows, cols]=size(M);
    a=M(2:2:rows,2:2:cols);
end
```

Test code

```
M=rand(6,5) % create a matrix M
a=even_index(M) % call the function even_index
```

Problem 2 Write a function called flip_it that has one input argument, a row vector v, and one output argument, a row vector w that is of the same length as v. The vector w contains all the elements of v, but in the exact opposite order. For example, is v is equal to [1 2 3] then w must be equal to [3 2 1]. You are not allowed to use the built-in function flip.

```
%% Problem 2
function w=flip_it(v)
    [rows,cols]=size(v); % get the size of v
     % creat the vector w has all v element but in opposite order
     for cnt=1:cols
         w(cols-cnt+1)=v(cnt);
     end
end
```

Test code

```
% Problem 2
v=rand(1,8)% creat the vector v
w=flip_it(v) % call the function flip_it
```

Problem 3 Write a function called top_right that takes two inputs: a matrix N and a scalar non-negative integer n, in that order, where each dimension of N is greater than or equal to n. The function returns the n-by-n square subarray of N located at the top right corner of N.

```
%% Problem 3
function subArray=top_right(N,n)
    [rows,cols]=size(N); % get the rows and columns of N
    subArray=N(1:n,cols-n+1:cols);
end
```

Test code

```
N=rand(5,5) % create matrix N
n=3 % create value of n
subarray=top_right(N,n) % call the top_right
```

Problem 4 Write a function called peri_sum that computes the sum of the elements of an input matrix A that are on the "perimeter" of A. In other words, it adds together the elements that are in the first and last rows and columns. Note that the smallest dimension of A is at least 2, but you do not need to check this. Hint: do not double count any elements!

```
function addElement=peri_sum(A)
    [rows, cols]=size(A); % get the rows and columns of A
    addElement = 0;
    for i = 1: cols
        if i == 1 || i == cols
            addElement = addElement + ones(1,rows) * A(:,i);
        else
            addElement = addElement + A(1,i) + A(end,i);
        end
    end
end
```

See the red code.

Test code

```
% Problem 4
A=rand(4,5) % creat matrix A and
addElement=peri_sum(A) % call peri_sum
```

Problem 5 The power series for sin(x) is given by

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots, \qquad -\infty < x < \infty$$

The script on the right will compute this power series for a given value of x.
What causes the loop to terminate? How accurate is the series for $x=\pi/2$? How many terms were needed?

```
function s=powersin(x)
s=0;
t=x;
n=1;
while s+t~=s
  s=s+t;
  t=-x.^2/((n+1)*(n+2)).*t;
  n=n+2;
end
```

I will not give you the exact answer of this question, but I will introduce the *eps* function in detail.

First of all, *eps* is a function that represents the minimum precision with which a number can be distinguished. By default, it represents half the distance from 1 to its next float number, and is exactly equal to the distance between a float number with a maximum less than 1 and a float number with a minimum greater than 1.

```
>> eps
ans =
   2.2204e-16
```

```
>> 1 + eps*0.6 == 1
ans =
  logical
  0
```

```
>> 1 + eps*0.4 == 1
ans =
  logical
  1
```

In this problem, s + t ~= s means t > 0.5*eps(s)

eps(N) represents relative precision, it is not the smallest number, nor is it a fixed number, it will increase as the N grows.

Problem 6 A ball is dropped from a height h of 2 meters. The velocity when it strikes the floor is given by $v^2=2gh$ and rebounds with a velocity that is 85% of the impact velocity. The ball then rebounds to a height of $h=v^2/2g$. What is the height after the 8th bounce?

```
%% Problem 6
h = 2 % create initial height
Rebound = 8 % create value of rebound
Finalheight = reboundHeight(h,rebound)

function height = reboundHeight(h,rebound)
    for cnt = 1:rebound
        h = h * 0.85^2;
    end
    height = h;
end
```

Problem 7 The ideal gas law is given by $P = \frac{nRT}{V}$

The van der Waals equation corrects for high pressure effects and is given by $P = \frac{nRT}{V-nb} - \frac{n^2 a}{V^2}$

Plot pressure vs. volume for n=1, T=300K, R=0.08206 L-atm/mol-K, a=1.39 L^2-atm/mol^2, and b=0.039 L/mol.
Use 0.08<V<6 liters

```
function VolPrePlot
    n=1;
    T=300;
    R=0.08206;
    a=1.39;
    b=0.039;
    V=linspace(0.08,6,100);
    P=n*R*T./(V-n*b)-n*n*a./(V.*V);
    plot(V,P);
    xlabel('Volumn');
    ylabel('Pressure');
end
```

Problem 8

a. generate a random sized array of random numbers using x = 10*rand(ceil(10*rand)+2,1)

b. Use “for” loop to add up all the values in the array and assign the result to the variable mysum.

for example if the array is x=[1 1 1 1 1 1 1 1 1 2], then the sum would be mysum = 11

c. Check your answer using the build-in MATLAB sum() function by adding the following code snipper to the end of your script

```
if mysum==sum(x)
    disp('Congratulations!!,you did it right’)
    load handel; sound(y,Fs)
else
    fprintf('Sorry, %.2f ~= %.2f. Please try again.\n',mysum,sum(x))
end
```

d. Repeat but use a “while” loop this time

There are many ways to sum vectors or matrices. In mathematics,

$$[1 \quad \cdots \quad 1]\begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} = a_1 + \cdots + a_n,$$

$$[1 \quad \cdots \quad 1]\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = a_{11} + \cdots + a_{mn}$$

```
function praticeForLoop
    x=10*rand(ceil(10*rand)+2,1);
    [rows,cols] = size(x); totalElement = rows*cols;
    mysum1 = 0;
    % add all element by for-loop
    for addfor = 1:totalElement
        mysum1 = mysum1 + x(addfor);
    end
    % check
    if mysum1 == sum(x)
        disp('Congratulations!!,you did it right')
           load handel; sound(y,Fs)
    else
        fprintf('Sorry, %.2f ~= %.2f. Please try again.\n',mysum1,sum(x))
    end
    % add all element by while loop
    addwhile=1; % reset the count value
    mysum2=0;
    while addwhile <= totalElement
        mysum2 = mysum2 + x(addwhile);
        addwhile = addwhile+1;
    end
    % check again
    if mysum==sum(x)
        disp('Congratulations!!,you did it right')
           load handel; sound(y,Fs)
    else
        fprintf('Sorry, %.2f ~= %.2f. Please try again.\n',mysum,sum(x))
    end
end
```

Homework 2
Solutions and Analysis

Problem 1 Consider the array A, Compute the array B by computing the square roots of all the elements of A whose value is no less than 0 and adding 50 to each element that is negative.

$$A = \begin{bmatrix} 0 & -1 & 4 \\ 9 & -14 & 25 \\ -34 & 49 & 64 \end{bmatrix}$$

```
A=[0 -1 4; 9 -14 25; -34 49 64]; % set the values of A
[rows,cols]=size(A); % get the size of A
edge=rows*cols; % set the edge for loop
for iCont =1 : edge
    if A(iCont) < 0
        A(iCont) = A(iCont)+50;
    else
        A(iCont) =sqrt(A(iCont));
    end
end
```

Problem 2 Determine how long it will take to accumulate at least $10,000 in a bank account if you deposit $500 initially and $500 at the end of each year, if the account pays 5 percent annual interest

```
years = 0;
totalMoney = 500;
while totalMoney < 10000
    years = years + 1;
    totalMoney = totalMoney *(1+0.05);
    totalMoney = totalMoney + 500;
end
fprintf('I need %d years to accumulate $10000\n', years)
```

Problem 3 The price, in dollars, of a certain stock over a 10-day period is given in the following array.

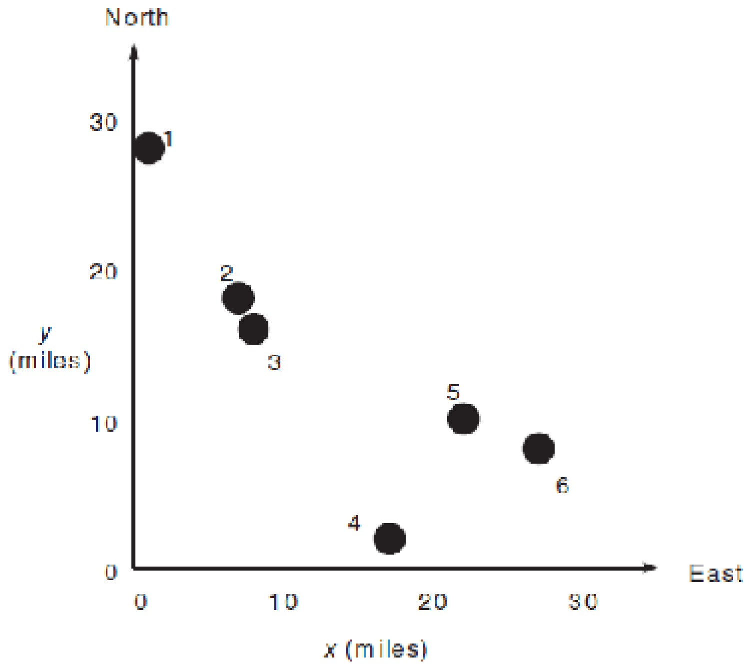
Price=[19,18,22,21,25,19,17,21, 27, 29]

Suppose you owned 1000 shares at the start of the 10-day period, and you bought 100 shares every day the price was below \$20 and sold 100 shares every day the price was above \$25. Write a script to compute (a) the amount you spent in buying shares, (b) the amount you received from the sale of shares, (c) the total number of shares you own after the 10th day, and (d) the net increase in the worth of your portfolio.

```
Price=[19,18,22,21,25,19,17,21,27,29];
stock=1000; moneyToBuy=0; profit=0; day=1;
while day <= 10
    if Price(day) < 20
        stock = stock + 100;
        moneyToBuy = moneyToBuy + Price(day)*100;
    elseif Price(day) > 25
        stock = stock - 100;
        profit = profit + Price(day)*100;
    end
    day=day+1;
end
      % print the answer
      fprintf('\n')
      fprintf('the amount spent in buying shares is %.2f\n', moneyToBuy)
      fprintf('the amount received from sale shares is %.2f\n', profit)
      fprintf('the total number of shares i own is %.2f\n', stock)
      fprintf('the net increase in the worth of my portfolio is %.2f\n', profit-moneyToBuy)
```

Problem 4 A company wants to locate a distribution center that will serve six of its major customers in a 30x30 mi area. The locations of the customers relative to the southwest corner of the area are given in the following table in terms of (*x*,*y*) coordinates (the *x* direction is east; the *y* direction is north). Also given is the volume in tons per week that must be delivered from the distribution center to each customer. Then weekly delivery cost c_i for customer i depends on the volume V_i and the distance d_i from the distribution center. For simplicity we will assume that this distance is the straight-line distance. (This assumes that the road network is dense.) The weekly cost is given by $c_i = 0.5 d_i V_i$, $i = 1, \ldots, 6$, Find the location of the distribution center (to the nearest mile) that minimizes the total weekly cost to service all six customers.

Customer	*x* location (mi)	*y* location (mi)	Volume (tons/week)
1	1	28	3
2	7	18	7
3	8	16	4
4	17	2	5
5	22	10	2
6	27	8	6

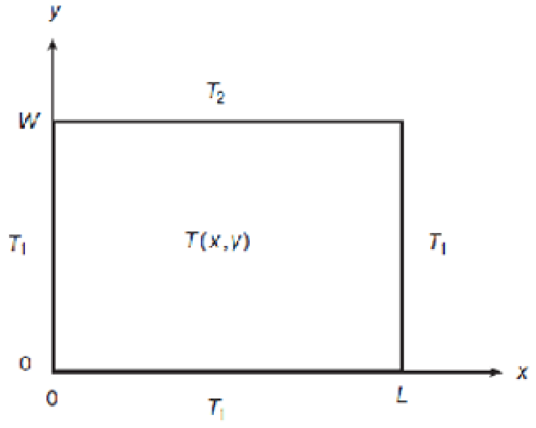


```
%% Problem 4
[xgrid,ygrid] = meshgrid(0:30,0:30);
customer = [1,28,3 ; 7,18,7 ; 8 16 4 ; 17 2 5 ; 22 10 2 ; 27 8 6];
pay = cost(xgrid,ygrid,customer);
minpay = min(min(pay));
[row,col] = find(minpay == pay);
location = [xgrid(row,col),ygrid(row,col)]

function pay = cost(xgrid,ygrid,customer)
    pay = zeros(size(xgrid,1),size(xgrid,2));
    for i = 1 : size(customer,1)
        distance = sqrt( (xgrid - customer(i,1)).^2 + (ygrid - customer(i,2)).^2 );
        pay = pay + 0.5 * distance * customer(i,3);
    end
end
```

Problem 5 Many applications require us to know the temperature distribution in an object. For example, this information is important for controlling the material properties, such as hardness, when cooling an object formed from molten metal. In a heat-transfer course, the following description of the temperature distribution in a rectangular metal plate is often derived. The temperature is held constant at T_1 on three sides and at T_2 on the fourth side. The temperature as a function of the *xy* coordinates shown is given by

$$T(x,y) = (T_2 - T_1)\,w(x,y) + T_1$$

$$w(x,y) = \frac{2}{\pi}\sum_{n\ \mathrm{odd}}^{\infty} \frac{2}{n}\sin\left(\frac{n\pi x}{L}\right)\frac{\sinh(n\pi y/L)}{\sinh(n\pi W/L)}$$



Use the following data: T_1=70°F, T_2=200°F, and W=L=2ft.

a) The terms in the preceding series become smaller in magnitude as *n* increases. Write a program to verify this fact for n=1, …., 19 for the center of the plate (x=y=1)

b) Using x=y=1 , write a program to determine how many terms are required in the series to produce a temperature calculation that is accurate to within 1 percent. (That is, for what value of *n* will the addition of the next term in the series produce a change in *T* of less than 1 percent?) Use your physical insight to determine whether this answer gives the correct temperature at the center of the plate.

c) Modify the program from part b to compute the temperatures in the plate; use a spacing of 0.2 for both x and y.

```
%% problem 5
T1=70; T2=200; W=2; L=2;

% (a)
w =@(ns,x,y)(2/pi) * 2./ns .* sin(n*pi.*x/L) .* sinh(n*pi.*y/L) ./ sinh(n*pi*W/L);
ns = 1:2:19;
plot(ns, abs(w(ns,ones(size(n)),ones(size(n)))) );
xlabel('n');
ylabel('w(1,1)');

% (b)
T = @(w) (T2-T1) * w + T1;
i = 1; w_sum = 0;
while true
    w_sum = w_sum + w(i,1,1)
    w_next_sum = w_sum + w(i+2,1,1)
    difference = abs(T(w_next_sum) - T(w_sum))/abs(T(w_sum))
    if difference <= 0.01
        break
    else
        i = i+2;
    end
end
fprintf('%d terms are required to produce an accuracy within 1%% when x = %d, y = %d.\n',(i+1)/2,1,1);
```

```
% (c)
figure(2);
[X,Y] = meshgrid(0:0.2:W,0:0.2:L);
W_position = w(ones(size(X)),X,Y);
Z = T(w(ones(size(X)),X,Y));
mesh(X,Y,Z)
```

Problem 6 On January 1st, Cindy opens a savings account and deposits $10,000. At the end of every month, she deposits $1000 more into the account for the next 12 months (starting January 31). At the end of each month (before the $1000 deposit), interest is calculated and added to her balance. The monthly interest rate varies depending on the account balance at the time interest is calculated.

Balance ($)	Interest
B≤15,000	1%
15,000<B≤20,000	1.5%
B>20,000	2%

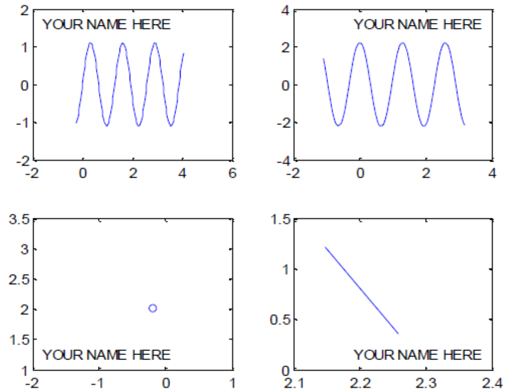
Write a script that displays in the command window, for each of the 12 months, **with informative headers**, the:

1. Number of the current month
2. Interest rate for current month as a percentage (e.g. 1.5 and not 0.015)
3. total amount of interest earned that month (with two decimal places)
4. New balance (with two decimal places)
5. Total interest earned (running total of the cumulative interest earned from the opening deposit)

```
%% problem 6
clear all; clc
balance = 10000;
interest = 0;
rate = 0;
total = 0;

for i = 1:12
    if balance <= 15000
        rate = 1;
    elseif balance>15000 &&balance <= 20000
        rate = 1.5;
    else
        rate = 2;
    end
    interest = rate * balance / 100;
    balance = balance + interest + 1000;
    total = total + interest;
    fprintf('%d\t%.1f\t$%.2f\t$%.2f\t$%.2f\n',i,rate,interest,balance,total)
end
```

Problem 7 Place your name in the right corner as shown in the figure. For this assignment, the code for the plots are given and you are expected to create a user defined function that places your name in the corner of any plot.

```
%%
% This MATLAB script tests a user-defined function that places a name
% on randomly sized plots.

%% Generate randomized data to plot
clear all;clc;

% x data
xmin = (-10) + (10-(-10)).*rand; %Generate random number between -10
and 10
xrange = 2 + (5-2).*rand; %Generate random number between 2 and 5
xmax = xmin + xrange;
numPts = 150; %Number of data points
x = linspace(xmin,xmax,numPts);
x2 = x-0.2*xrange;

% y data
Amp = 0.5 + (2-0.5).*rand; %Generate random amplitude between 0.5 and 2
Freq = 0.5 + (1.5-0.5).*rand; %Generate random freq between 0.5 and 1.5
y = Amp*sin(2*pi*Freq*x);
y2 = 2*Amp*cos(2*pi*Freq*x2);

%% Plot data and test your function
r = 2; %number of subplot rows
c = 2; %number of subplot columns

subplot(r,c,1)
plot(x,y)
%*****************************
%TYPE THE NAME OF YOUR FUNCTION HERE TO PUT YOUR NAME IN THE UPPER LEFT
%*****************************
subplot(r,c,2)
plot(x2,y2)
%*****************************
%TYPE THE NAME OF YOUR FUNCTION HERE TO PUT YOUR NAME IN THE UPPER
RIGHT
%*****************************

subplot(r,c,3)
plot(-5*rand,3*rand,'o')
%*****************************
%TYPE THE NAME OF YOUR FUNCTION HERE TO PUT YOUR NAME IN THE LOWER LEFT
%*****************************

subplot(r,c,4)
plot([5*rand 5*rand],[2*rand,6*rand])
%*****************************
%TYPE THE NAME OF YOUR FUNCTION HERE TO PUT YOUR NAME IN THE LOWER
RIGHT
%*****************************
```



```
%% Problem 7
xmin = -10 + 20.*rand;
xrange = 2 + 3.*rand;
xmax = xmin + xrange;
numPts = 150;
x = linspace(xmin,xmax,numPts);
x2 = x - 0.2*xrange;

Amp = 0.5 + 1.5.*rand;
Freq = 0.5 + rand;
y = Amp*sin(2*pi*Freq*x);
y2 = 2*Amp*cos(2*pi*Freq*x2);

r = 2;
c = 2;

subplot(r,c,1)
plot(x,y)
xl = xlim;
yl = ylim;
text(xl(1)+0.05*(xl(2)-xl(1)),yl(2)+0.07*(yl(1)-yl(2)),'Mio','FontSize',14)

subplot(r,c,2)
plot(x2,y2)
xl = xlim;
yl = ylim;
text(xl(2)+0.2*(xl(1)-xl(2)),yl(2)+0.07*(yl(1)-yl(2)),'Mio','FontSize',14)

subplot(r,c,3)
plot(-5*rand,3*rand,'o')
xl = xlim;
yl = ylim;
text(xl(1)+0.05*(xl(2)-xl(1)),yl(1)+0.1*(yl(2)-yl(1)),'Mio','FontSize',14)

subplot(r,c,4)
plot([5*rand 5*rand],[2*rand,6*rand])
xl = xlim;
yl = ylim;
text(xl(2)+0.2*(xl(1)-xl(2)),yl(1)+0.1*(yl(2)-yl(1)),'Mio','FontSize',14)
```

ICE for curve fitting Solutions and Analysis

Problem 1 The temperature of the ground at a depth x for surface temperature T_s and initial temperature T_i is given as

$$\frac{T - T_s}{T_i - T_s} = \operatorname{erf}\left(\frac{x}{2\sqrt{\alpha t}}\right)$$

How deep should a water main be buried if we want to keep the water from freezing if the surface is at -15°C for 60 days?

t=60*24*3600

T_s=-15 C

T_i=20 C

T=0 C

$\alpha=1.38*10^{-7}\ m^2/s$

```
%% problem 1
ts = -15; ti =20; at = 60*24*3600*1.38*10^-7;
y = @(x) ((ti-ts)*erf(x/(2*sqrt(at))) + ts)
Ans1_1 = fzero(y,1) %tries to find a point x near 1 where y=0

x = 0:0.1:6;
T = (ti-ts)*erf(x/(2*sqrt(at))) + ts;
plot(x,T)
xlabel('X')
ylabel('T')
grid on
```

```
Ans1_1 =

    0.6770
```

Problem 2 Find first two positive values of β that solve this equation for L=4.2m.

$$1 + \cosh(\beta L)\cos(\beta L) = 0$$

For $EI = 21000\ N \cdot m^2$ and $\rho = 0.53\ kg/m$, calculate the frequencies from

$$\omega = \beta^2 \sqrt{\frac{EI}{\rho}}$$

```
%% Problem2
ei = 21000;
f = 0.53
l = 4.2;
y = @(x) (1 + cosh(x*l).*cos(x*l))
x = -10:0.01:10;
plot(x,y(x))
xlabel('X')
ylabel('Y')
grid on
root1 = fzero(y,0.4)
root2 = fzero(y,1)
w1 = (root1^2)*sqrt(ei/f)
w2 = (root2^2)*sqrt(ei/f)
```

```
root1 =

    0.4465


root2 =

    1.1176


w1 =

   39.6756


w2 =

  248.6430
```

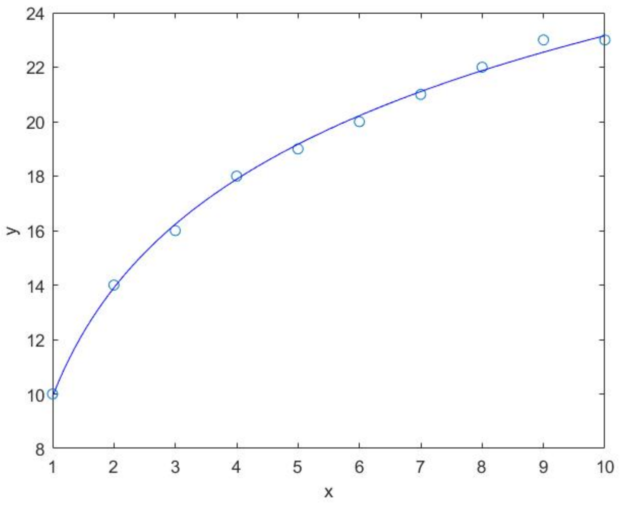
Problem 3 The following function is linear in the parameters a_1 and a_2.

$$y(x) = a_1 + a_2 \ln x$$

Use least-squares regression with the following data to estimate the values of a_1 and a_2. Use the curve to estimate the values of y at x=2.5 and at x=11.

x	1	2	3	4	5	6	7	8	9	10
y	10	14	16	18	19	20	21	22	23	23

```
%% problem 3
close all;% clear all the code,close all figures
clear all;
clc;
% init the value
x=[1:10];
y=[10,14,16,18,19,20,21,22,23,23];
% plot the point
plot(x,y,'o')
hold on
A=[length(x) sum(log(x)); sum(log(x)) sum(log(x).^2)]; % calculate matrix A
B=[sum(y); sum(log(x).*y)]; % calculate matrix B
b=A\B % slove the equation

x1=linspace(x(1),x(end),100);
plot(x1,b(2).*log(x1)+b(1),'b')
xlabel('x')
ylabel('y')
fprintf('f(2.5)=%.4f, f(11)=%.4f',b(1)+b(2)*log(2.5),b(1)+b(2)*log(11))
```




```
b =

    9.9123
    5.7518

f(2.5)=15.1826, f(11)=23.7044
```

Problem 4 Chemists and engineers must be able to predict the changes in chemical concentration in a reaction. A model used many single-reactant processes is

$$\text{Rate of change of concentration} = -kC^n$$

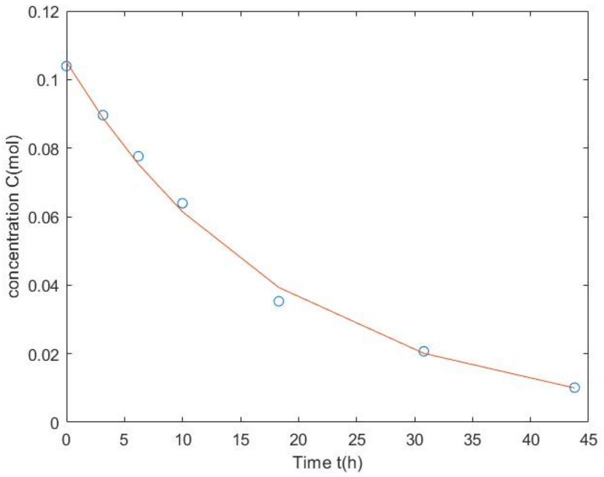
where C is the chemical concentration and k is the rate constant. The following data describe the reaction

$$(CH_3)_3CBr + H_2O \rightarrow (CH_3)_3COH + HBr$$

Use these data to obtain a least-squares fit to estimate the value of k

Time t (h)	C(mol of $(CH3)_3$ CBr/L)
0	0.1039
3.15	0.0896
6.20	0.0776
10.0	0.0639
18.3	0.0353
30.8	0.0207
43.8	0.0101

```
%% problem 4
t=[0,3.15,6.20,10,18.3,30.8,43.8];
C=[0.1039,0.0896,0.0776,0.0639,0.0353,0.0207,0.0101];
p = polyfit(t,log(C),1);
k = -p(1);
c0 = exp(p(2));
c = c0 * exp(-k*t);
J1 = sum((c-C).^2);
S1 = sum((c-mean(C)).^2);
r = 1 - J1/S1;
plot(t,C,'o',t,c)
xlabel('Time t(h)')
ylabel('concentration C(mol)')
fprintf('C(t)=%.2f*e^(-%.2ft)\n',k,c0)
fprintf('The value of k is %.2f \n',k)
```




```
C(t)=0.05*e^(-0.11t)
The value of k is 0.05
```

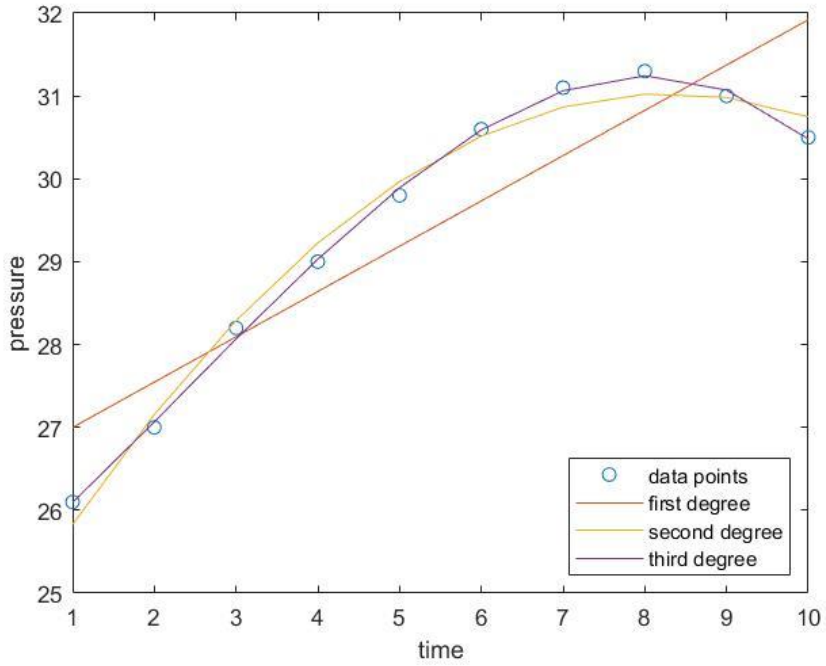
Remember to consider the initial value of the ODE

Problem 5 The following represents pressure samples, in pounds per square inch (psi), taken in a fuel line once every second for 10 sec.

- Fit a first-degree polynomial, a second-degree polynomial, and a third-degree polynomial to these data. Plot the curve fits along with the data points
- Use the results from part a to predict the pressure at t = 11 sec. Explain which curve fit gives the most reliable prediction. Consider the coefficients of determination and the residuals for each fit in making your decision.

Time (sec)	Pressure (psi)	Time (sec)	Pressure (psi)
1	26.1	6	30.6
2	27.0	7	31.1
3	28.2	8	31.3
4	29.0	9	31.0
5	29.8	10	30.5

```
the 3 degree polynimial is reliable
f(11)= 29.4100
```



```
%% problem 5
time=[1:10];
press=[26.1, 27.0, 28.2, 29.0, 29.8, 30.6, 31.1, 31.3, 31.0, 30.5];
% plot point
plot(time,press,'o'); hold on
% first-degree
p1=polyfit(time,press,1); y1=@(time)p1(1)*time+p1(2); plot(time,y1(time))
% second degree
p2=polyfit(time,press,2); y2=@(time)p2(1)*(time.^2)+p2(2)*time+p2(3); plot(time,y2(time))
% thrid degree
p=polyfit(time,press,3); y3=@(time)p(1)*(time.^3)+p(2)*(time.^2)+p(3)*time+p(4);
plot(time,y3(time))
% put legend and label
xlabel('time')
ylabel('pressure')
legend('data points','first degree','second degree','third degree','Location','southeast')
% calculate residuals
resid(1)=sum((y1(time)-press).^2);
resid(2)=sum((y2(time)-press).^2);
resid(3)=sum((y3(time)-press).^2);
% find the min
ind=find(resid==min(resid));
fprintf('the %d degree polynimial is reliable\n',ind)
fprintf('f(11)= %.4f',eval(['y',num2str(ind),'(11)']))
```