**Introduction to Scientific Computation, Homework #1**

**Due by 0 am on Friday 5/10/2024**

**Problem 1 (10 pts):** If C and F are Celsius and Fahrenheit temperatures, respectively, the formula for conversion from Celsius to Fahrenheit is F = 9C/5 + 32.

1. Write a script that will ask you for the Celsius temperature and display the Fahrenheit equivalent with some sort of comment, such as

The Fahrenheit temperature is:...

Try it out on the following Celsius temperatures (answers in parentheses): 0 (32), 100 (212), −40 (−40!), 37 (normal human temperature: 98.6).

**Problem 1(a) Solution**

function F = FtoC(C)

F = 9\*C/5 + 32;

disp(“The Fahrenheit temperature is:...”+F)

end

1. Change the script to use vectors and array operations to compute the Fahrenheit equivalents of Celsius temperatures ranging from 20℃ to 30℃ in steps of 1℃, and display them in two columns with a heading, like this：

Celsius Fahrenheit

20.00 68.00

21.00 69.80

...

30.00 86.00

**Problem 1(b) Solution**

C\_arr = 20:30;

F\_arr = 9\*C\_arr/5 + 32;

array = [C\_arr',F\_arr'];

disp(" Celsius Fahrenheit")

disp(array)

**Problem 2 (10 pts):** Set up a matrix (table) with degrees in the first column from 0 to 360 in steps of 30, sines in the second column, and cosines in the third column.

**Problem 2 Solution**

deg\_arr = 0:30:360;

rad\_arr = deg\_arr/360\*2\*pi;

sin\_arr = sin(rad\_arr);

cos\_arr = cos(rad\_arr);

matrix = [deg\_arr' sin\_arr' cos\_arr'];

disp(matrix);

**Problem 3 (20 pts):** Develop a structure plan for the solution to two simultaneous linear equations (i.e., the equations of two straight lines). Your algorithm must be able to handle all possible situations; that is, lines intersecting, parallel, or coincident. Write a program to implement your algorithm, and test it on some equations for which you know the solutions, such as

x + y = 3

2x − y = 3

(x = 2, y = 1). Hint: Begin by deriving an algebraic formula for the solution to the system:

ax + by = c

dx + ey = f

The program should input the coefficients a, b, c, d, e, and f.

**Problem 3 Solution**

function lines(a,b,c,d,e,f)

if a\*e == b\*d

if a\*f == c\*d

sprintf('The lines are coincident, the solutions is (x,y) satisfying %f x +%f y = %f',a,b,c)

else

disp('The lines are parallel')

end

else

x = (c \* e - b \* f) / (a \* e - b \* d);

y = (a \* f - c \* d) / (a \* e - b \* d);

sprintf("x = %f , y = %f",x,y)

end

**Problem 4 (20 pts):** Consider the following system of linear equations



Solve this system of equations using three syntactically distinct methods in a single script file (m file). The arrays corresponding to the solutions of *x* should be named xSol1, xSol2, and xSol3 for solution methods 1, 2, and 3, respectively. What are the values of *x*1 through *x*5 to three significant figures?

**Problem 4 Solution**

% Define the coefficient matrix A

A = [-23 -18 2 -19 -15;

-1 -16 10 1 3;

-15 -23 1 8 7;

-19 7 2 19 -4;

-15 -11 -3 -11 -15];

% Define a column vector b

b = [22;-21;-20;18;-17;];

% Solve the system of equations using 9 syntactically different

% methods

xsol1 = inv(A)\*b;

xsol2 = A\b;

xsol3 = linsolve(A, b);

The solution to the system to three significant digits was found to be:

*x*1 = -3.44

*x*2 = 3.34

*x*3 = 2.01

*x*4 = -3.10

*x*5 =4.00

**Problem 5 (20 pts):** Hypocycloids, epicycloids, epitrochoids and hypotrochoids are all fancy names for curves generated when a circle rotates about another circle (a.k.a. roulettes). You can find interesting descriptions and .gifs for all of these roulettes on Wikipedia. You will write a short MATLAB function with the following function declaration:

function spirograph(R,r,d)

You function will generate curves of the following form



You should begin by creating an array called theta that has values between 0 and 10π in steps of no more than 0.001 radians. You can do this using MATLAB’s colon operator or the linspace command. You will then compute values for *x* and *y* using array operations, addition, subtraction, the cosine function, etc. This should all be achieved without loops. Once you have the two arrays, x and y, your function should plot the curve using the command plot(x,y). Look at the help pages for the plot command to determine the visualization options this command offers and how you can go about changing the color or style of the plotted line.

Include pretty plots for the following values of *R*, *r*, and *d* in your assignment submission.

|  |  |  |  |
| --- | --- | --- | --- |
| **Plot number** | ***R*** | ***r*** | ***d*** |
| 1 | 5 | 1 | 0.4 |
| 2 | 12 | -1 | 1.5 |
| 3 | 7 | -1 | 1 |

**Problem 5 Solution**

function spirograph(R,r,d)

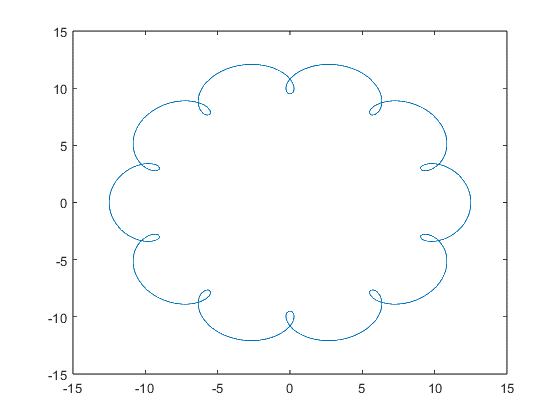
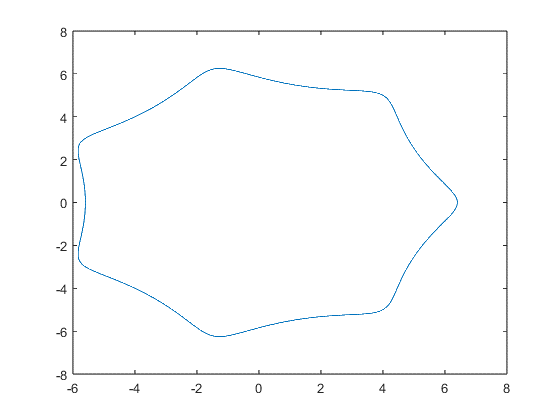
theta = 0:0.001:10\*pi;

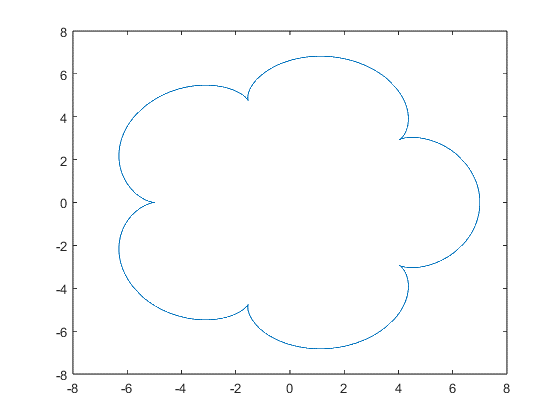
x = (R+r)\*cos(theta)+d\*cos((R+r)/r\*theta);

y = (R+r)\*sin(theta)-d\*sin((R+r)/r\*theta);

plot(x,y)

Plots corresponding to the three cases above are shown below.





**Problem 6 (20 pts):** Fibonacci sequences often appear in nature. See the following link for a description of Fibonacci numbers: <http://en.wikipedia.org/wiki/Fibonacci_number>

The Fibonacci sequence is described by the relationship



where 2≤*n*<*N* with *N* being some integer. The initial values of the Fibonnaci sequence are

 and 

The Fibonnaci sequence for *N* = 6 is: 0, 1, 1, 2, 3, 5, 8.

You are to write a program that returns a vector of the first *N* Fibonacci numbers with the following function declaration:

function nums = fib(N)

% fib(N) returns a list of the first N Fibonacci Numbers.

% N must be an integer.

What is the Fibonacci sequence for *N* = 20?

**Problem 6 Solution**

function nums = fib(N)

function num = fibonacci(N)

if N == 0

num = 0;

elseif N == 1

num = 1;

else

num = fibonacci(N-2) + fibonacci(N-1);

end

end

temp = [];

for i = 0:N

temp(end+1) = fibonacci(i);

end

nums = temp;

end

The Fibonacci sequence for N=20 is

0 1 1 2 3 5 8 13 21 34 55 89

144 233 377 610 987 1597 2584 4181 6765