Numerical Methods – Spring '18

PA #2

Created By: Clarisa Leu-Rodriguez

Dr. Christopher Willett

Programming Assignment #2

Problem 1:

A method of estimating $\sqrt{a}$ using a sequential approximation:

$$x_0 = a, \quad x_{i+1} = \frac{x_i + \frac{a}{x_i}}{2}$$

Write a MATLAB function that inputs a positive constant a and a tolerance ϵ and returns the estimate of $\sqrt{a}$ that has a relative error less than ϵ.

My Solution:

leurodriguez1.m

```
%% Problem 1

%% Main Function
% A function  that inputs a positive constant a and tolerance
%epsilon. Returns the estimate of the square root of a that has a relative
%error less than epsilon.

function y = leurodriguez1(a,epsilon)
if a <0||epsilon<0
    disp('Sorry, try again! a>0 & epsilon>0.');
elseif a==0
    y=0;
else
    oldapprox=a;
    %Relative Error%
    presentapprox=(oldapprox+(a./oldapprox))./2;
    while abs((oldapprox-presentapprox)./oldapprox)>epsilon
        oldapprox=presentapprox;
        presentapprox=(oldapprox+(a./oldapprox))./2;
    end
    y=presentapprox;
end
```

Problem 2:

Create a MATLAB function that inputs two row vectors of numbers, A and B, and returns a row vector of three things. **See comments in* ***leurodriguez2.m****

My Solution:

leurodriguez2.m

```
%% Problem 2

%% Main Function
%A function that inputs two row arrays, A & B, & returns a row vector
%containing: Row vector returned = [bvals, primes, avals] bvals = The
%number of elements in B > mean of A primenums = A row vector that contains
%the prime numbers larger than or equal to the minimum of the elements of A
%& less than the maximum of the elements in B avals = Sum of the prime
%numbers greater than the mean of A or less than or equal to the max of B

function [bvals,primenums,avals]=leurodriguez2(A,B)
disp('Running leurodriguez2.m...')

%Delegated to subfunction
bvals=part1(A,B);


%Part 2 and 3
vals=primes(max(B));
j=1;
res=[];
res2=[];
counter=length(vals);
while counter>0
    if vals(j)>=min(A)
        res=[res vals(j)];
    end
    if vals(j)>mean(A)
        res2=[res2 vals(j)];
    end
    counter=counter-1;
    j=j+1;
end

disp('Almost done hoss...')
summed=sum(res2);
primenums=res;
avals=summed;

%Call Display
display(bvals,primenums,avals)
```

```
%Call Display
display(bvals,primenums,avals)
end

%% A subfunction for Part 1
function y = part1(A,B)
j=1;n=0;
counter=length(B);
while counter>0
    if B(j)>mean(A)
        n=n+1;
    end
    j=j+1;
    counter=counter-1;
end
y=n;
end

%% A subfunction for display
function y=display(bvals,primenums,avals)
string1='Values of Array B greater than the mean of the elements of A: ';
string2='Prime numbers larger than or equal to the minimum of the elements of A & less than the maximum of the elements of B: ';
string3='Sum of prime numbers greater than the mean of A or less than or equal to the max of B: ';

disp(string1)
disp(bvals)
disp(string2)
disp(primenums)
disp(string3)
disp(avals)
disp('Row Vector:')
disp([bvals,primenums,avals])

end
```

leurodriguez2_test.m

```
%% Problem 2
%A test script for problem 2

%% Test1
disp('Running tester script for problem 2...')
disp('Running the first test...')
A=[1,3,4,5,6,3,2];
B=[4,5,6,4,2,6,3];
disp('A =')
disp(A)
disp('B =')
disp(B)
[bvals,primenums,avals]=leurodriguez2(A,B)
disp('----------------------------------------------')

%% Test 2
disp('Runing the second test...')
A=[6,59,20];
B=[4,5,7,8,4,22];
disp('A =')
disp(A)
disp('B =')
disp(B)
[bvals,primenums,avals]=leurodriguez2(A,B)
disp('----------------------------------------------')


%% Test 3
disp('Runing the third test...')
A=[3,4,5,6,4,200];
B=[3,4,5,20,4];
disp('A =')
disp(A)
disp('B =')
disp(B)
[bvals,primenums,avals]=leurodriguez2(A,B)
disp('Done running tester script!')
```

Problem 3:

From Lab 2, we wrote a function that returned the graph of population levels for a logistic population of n generations given a constant of k and constant n.

Write a routine to plot the various population levels for a fixed value of k as a function of k. Write a MATLAB function that accepts the starting k value, the ending k values, and a step size h for the values of k and returns a plot where the horizontal axis is the range of k values between k1 and k2 and the vertical axis is different population levels for a logistic population with a growth constant k for $80 \leq p_n \leq 120$ generations.

My Solution:

leurodriguez3.m

```
%% Problem 3

%% Main Function
%A function that inputs an initial growth rate k1, a final growth rate k2,
%and a step size h. Returns a plot of the range of k values between k1 and
%k2 vs the different long term population levels of a logistic population
%occuring between 80 and 120 generations.

function y = leurodriguez3(k1,k2,h)
disp('Running leurodriguez3.m...')
popvals=[logistic(k1,120)];
counter=k1;
kcur=k1;
kvals=k1:h:k2;
counter=k1;

while counter<k2
    counter=counter+h;
    kcur=counter;
    pcur=logistic(kcur,120);
    popvals=[popvals;pcur];
end

plotgraph(kvals,popvals);
end

%% A subfunction that returns the population carrying capacity of a population over 80 to 120 generations with a growth rate k in a vector
function y = logistic(k,n)
pold=.5;
popvals=[pold];
counter=0;
while(counter<n)
    counter=counter+1;
    pnew=k.*pold.*(1-pold);
    pold=pnew;
    popvals=[popvals pnew];
end
res = popvals(80:120);
y=res;
end
```

```
%% A subfunction to plot the graph
function y = plotgraph(kvals,popvals)
disp('Plotting...')
disp('Almost done...')
plot(kvals,popvals,'.')
title('Population Levels For Logistic Populations')
xlabel('Growth Rate Constant -  k')
ylabel('Percent of Carrying Capacity of Population from generations 80 to 120')
end
```

Outputs: