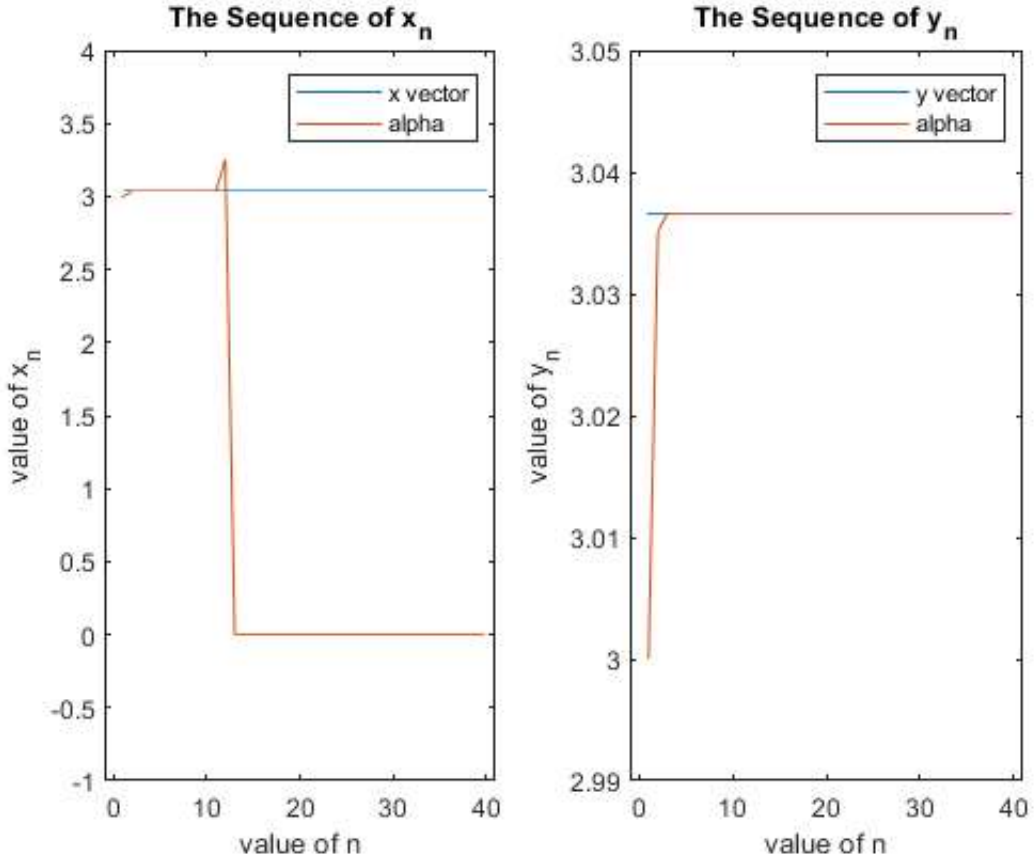
```
%1.i
x = zeros(1,40);
xn = 3;
n=1;  %Initial Conditions
while n~= 41 %Started a while loop to iterate untill n = 41 so that it calculates from 1 to 4
0 for n
    x(1,n) = xn;%assigning values to the nth place in the vector x
    xn = 5^n *sqrt(4*(2- sqrt(4-(xn/5^n)^2)));%calculating the next itteration in the sequenc
e and assigning it to xn
    n = n+1;%keeping track of irterations
end
fprintf("1.i) the vector x is \n")
fprintf('%1.4f ',x)%formating and printing the vector keeping an appropriate precision

%1.ii
naxis = (1:40);
constant = ones(1,40)*3.0366;%setting up other 1D arrays for plotting
subplot(1,2,1)  %I went for a subplot as i like the direct comparison
plot(naxis,constant,naxis,x),axis([-1,41,-1,4]), legend('x vector','alpha'),xlabel('value of
n'),ylabel('value of x_n'), title('The Sequence of x_n')%plotting the first vector x against
the constant

%1.v
y = zeros(1,40); %See question 1.i for the comments on these lines of working as it is the sa
me code again
yn = 3;
n = 1;
 while n ~= 41
     y(1,n) = yn;
     yn = (4*yn)/(sqrt(4*(2+ sqrt(4-(yn/5^n)^2))));
     n = n+1;
 end
fprintf("\n\n1.v) the vector y is \n")%Slightly different formating so everything prints to c
onsole nicely spaced
fprintf('%1.4f ',y)

%1.vi
naxis = (1:40); %See question 1.ii
constant = ones(1,40)*3.0366;
subplot(1,2,2)
plot(naxis,constant,naxis,y),axis([-1,41,2.99,3.05]), legend('y vector','alpha'),xlabel('valu
e of n'),ylabel('value of y_n'),title('The Sequence of y_n')

%2.a
xn = 2;
n = 0;
test = true;
x = zeros(1);%Setting up initial conditions
while test == true %my logic tests are within the loop as i like the flexibility this affords
 when coding at the expense of a clean look
    xn2 = (16*xn^5 + 1)/(20*xn^4 - 1);%assigning the value of the next itiration to a seperat
e value for comparison test
    n = n+1;%tracking number of iteraions
    x(1,n)=xn;%assigning value to vector for plotting
    xn = xn2;%resetting value for next loop
```

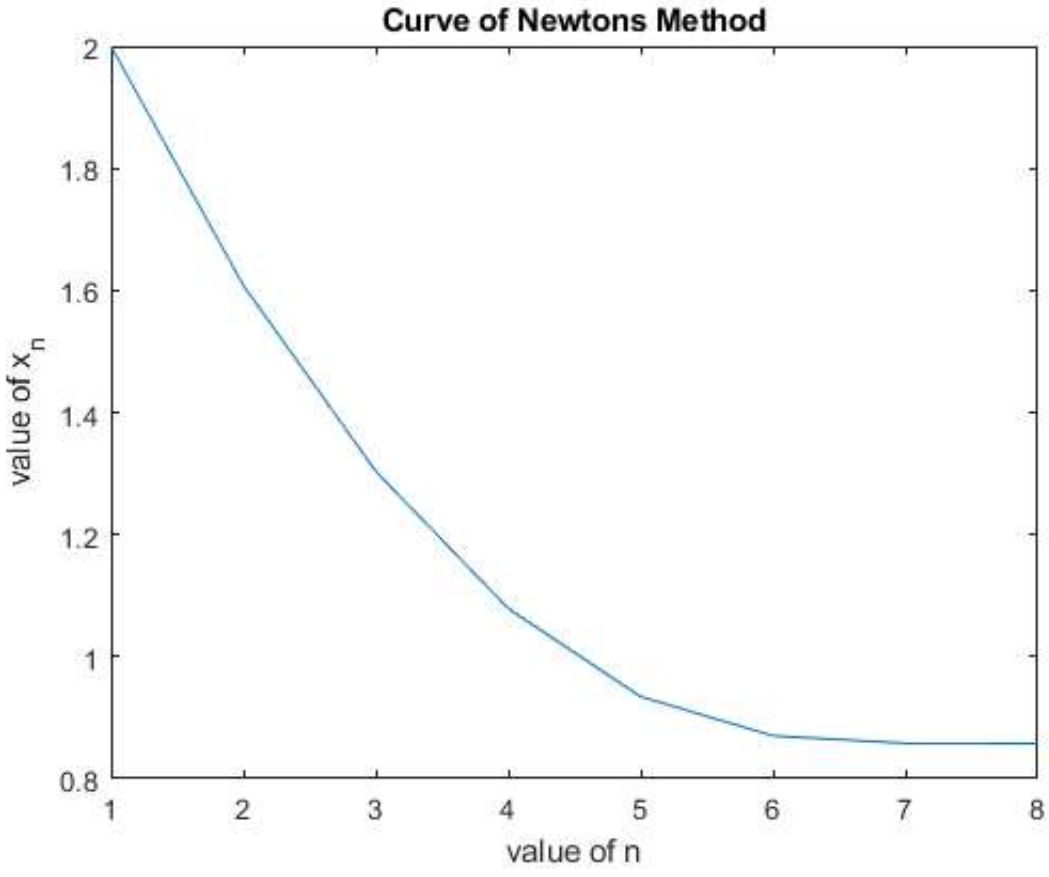
```
    if ((xn - xn2)^2) < 10^(-20)%testing first endpoint condition, i did not use abs here as
i thought this would be more elegant but in retrospect it probably wasnt
        if ((4*xn2^5 - xn2 -1)^2) < 10^(-20)%testing second enpoint condition
            test = false;%setting test varible to false so that the next while loop will end
the looping
        end
    end
    if n == 1000%testing if the sequence likely isnt converging
        test = false;%ending loop
    end
end
fprintf("\n\n2.a)the final value of x is ")
fprintf('%f \n',x(1,n))%formating and printing to console, this could have been done in one l
ine but i was originally going to do it differntly and im a lazy coder
figure(2)%switching to my next figure as im creating a seperate graph
naxis = (1:n);%setting up graph varibles
plot(naxis,x),xlabel('value of n'),ylabel('value of x_n'),title("Curve of Newtons Method") %J
ust plotting
```

```
1.i) the vector x is
3.0000 3.0352 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0369 3.0385 3.2539 0.0000 0.
0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00
00 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
```

```
1.v) the vector y is
3.0000 3.0352 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.
0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.03
66 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366 3.0366
```

```
2.a)the final value of x is 0.857804
```


The Sequence of x_n
x vector
alpha
value of x_n
value of n
The Sequence of y_n
y vector
alpha
value of y_n
value of n


Curve of Newtons Method
value of x_n
value of n

Published with MATLAB® R2019b