Mehak Gupta

301311972

Computing Assignment 1

Part 1

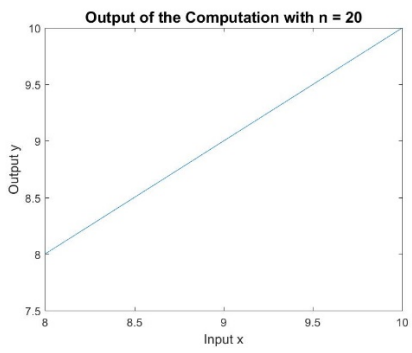
cos (0) = 1

cos (π /2.0) = 6.1232 x 10^-17

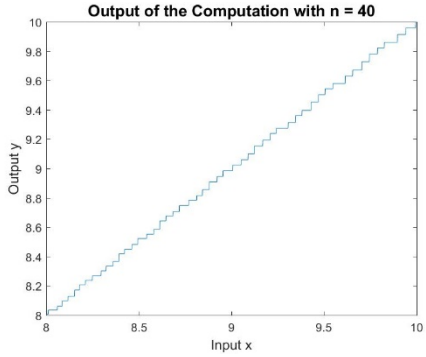
Cos (0) in MATLAB gives us the very expected result whereas cos (π /2.0) gives us a number definitely very close to 0 but not exactly zero as expected. This is because there are some floating-point errors which occur due to the approximated value of pi.

Part 2

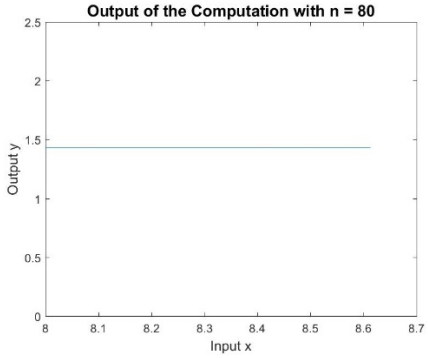
a)



(A)



(B)



(C)

b) The smallest value of n after which the result of the finite precision computation begins to differ from exact arithmetic computation is 38. As in part(a) A fig. we can see the exact arithmetic computation indicated by the straight line which tends to get zig zag when it strikes 38 and increases thereafter as shown in fig B until n=50

c) For large n, we see the straight horizontal line in the graph that stays exactly the same after n strikes 50 which clearly indicates the limiting behavior for large n. The numbers approximately as shown in Fig C are x = 8.62 and y = 1.45 when n=80. We see these particular numbers because of the error. For x= 8.62, y should be 8.62 according to the exact arithmetic computation. Continuous increase in the error make the results inconsistent for large n which makes the value of Y go lower to 1.45.

d) Computing in floating point arithmetic leads us to these results because we are constantly approximating values and rounding, chopping or ignoring some parts of it while computing our values for x and y. These values as expected are pretty low so don't seem to affect until n = 37 or 38 but after that, the error values definitely seem to get really large affecting the results for x and y as we can see in the graphs in part (a) for different values of n.

CODE USED FOR THE ASSIGNMENT

The same code with the modification of n value as 40 and 80 to produce the graphs shown in part a of question 2.

```
% MACM 316 - Homework 1
% Floating Point Arithmetic with logarithm function
% Description: Performs n-fold logarithm followed by n-fold exponentiation
% File name: FloatPt_log.m

clear
%format long; %this changes the number of decimal digits that DISPLAY
n=20;
st=0.001; % Define a stepsize
x=8:st:10; % x is a row vector of numbers between 1.3 and 2.3 of increments st
y=x;

for i=1:n
    y=4*log(y);
end
%y(1) %this is how you print the 1st element of y to the screen
%y(11)
%y(101)
%y(1001)
%pause %this stops your program until you press a key
for i=1:n
    y=exp(y/4.0);
end
%y(1)
%y(11)
%y(101)
%y(1001)
% Plot the output y versus the input x
plot(x,y) %you can change the title and axis labels in this manner
title(['Output of the Computation with n = ' num2str(n)],'fontsize',14)
xlabel(['Input x'],'fontsize',12)
ylabel(['Output y'],'fontsize',12)
```