**ME 318M Homework #8**

Name: Jingsi Zhou

UT EID: jz24729

Section Number: 17460

**Problem 1:**

MATLAB Code:

function value = f(x, y)

value = x^2 + y^2 - 16;

function value = g(x,y)

value = exp(x\*1/3) + (1/3)\*y - 1;

function out = Jacobian(x\_guess, y\_guess)

partial\_circle\_y = 2\*y\_guess;

partial\_circle\_x = 2\*x\_guess;

partial\_curve\_y = 1/3;

partial\_curve\_x = (1/3)\*exp(x\_guess\*1/3);

out = [partial\_circle\_x, partial\_circle\_y; partial\_curve\_x, partial\_curve\_y];

function [out, count] = VectNewR(x\_guess, y\_guess)

x\_0 = x\_guess;

y\_0 = y\_guess;

count = 0;

while (sqrt((f(x\_0, y\_0))^2 + (g(x\_0, y\_0))^2) >= 10e-7) || (sqrt((x\_0 - x\_nm1)^2 + (y\_0 - y\_nm1)^2) >= 10e-6)

count = count + 1;

A = [x\_0; y\_0];

J = Jacobian(x\_0, y\_0);

F = [f(x\_0, y\_0); g(x\_0, y\_0)];

var\_New = A - (inv(J))\*F;

x\_nm1 = x\_0;

y\_nm1 = y\_0;

x\_0 = var\_New(1);

y\_0 = var\_New(2);

end

out = [x\_0; y\_0];

count = count;

Command Window:

>> [out, count]= VectNewR(4,-4)

out =

**2.2343**

**-3.3178**

count =

5

I wrote a function to implement the Vectorial Newton-Raphson method. To do this, I wrote separate functions for the two given curves and a function to compute the Jacobian given an input. Then, the next x-value and y-value was found using the formula from class. The intersection of these two curves was found to be at x = 2.2343 and y = -3.3178.

**Problem 2:**

1. Not sure what this question was asking so I just linearized the given equations. However, I could’ve also written this question as [JacobianMatrix(x\_0, y\_0)]\*[x\_1 – x\_0; y\_1 – y\_0] = - [equation1(x\_0, y\_0); equation2(x\_0, y\_0)]

Diagram, schematic

Description automatically generated

Text, letter

Description automatically generated

MATLAB code:

function value = f(x, y)

value = (x^2/186^2) + (y^2)/(300^2-186^2) - 1;

function value = g(x,y)

value = (y - 500)^2/(279^2) + (x-300)^2/(500^2-279^2) - 1;

function out = Jacobian(x\_guess, y\_guess)

partial\_f\_y = -y\_guess/27702;

partial\_f\_x = x\_guess/17298;

partial\_g\_y = 1/77841\*(2\*y\_guess - 1000);

partial\_g\_x = -1/172159\*(2\*x\_guess - 600);

out = [partial\_f\_x, partial\_f\_y; partial\_g\_x, partial\_g\_y];

function [out, count] = VectNewR(x\_guess, y\_guess)

x\_0 = x\_guess;

y\_0 = y\_guess;

count = 0;

while count < 10

count = count + 1;

A = [x\_0; y\_0];

J = Jacobian(x\_0, y\_0);

F = [f(x\_0, y\_0); g(x\_0, y\_0)];

var\_New = A - (inv(J))\*F;

x\_nm1 = x\_0;

y\_nm1 = y\_0;

x\_0 = var\_New(1);

y\_0 = var\_New(2);

end

out = [x\_0; y\_0];

count = count;

Command Window:

>> [out, count]= VectNewR(4,-4)

out =

**-8.4286499e+26**

**-2.6618638e+34**

count =

10

**Problem 3:**

**A picture containing diagram

Description automatically generated**