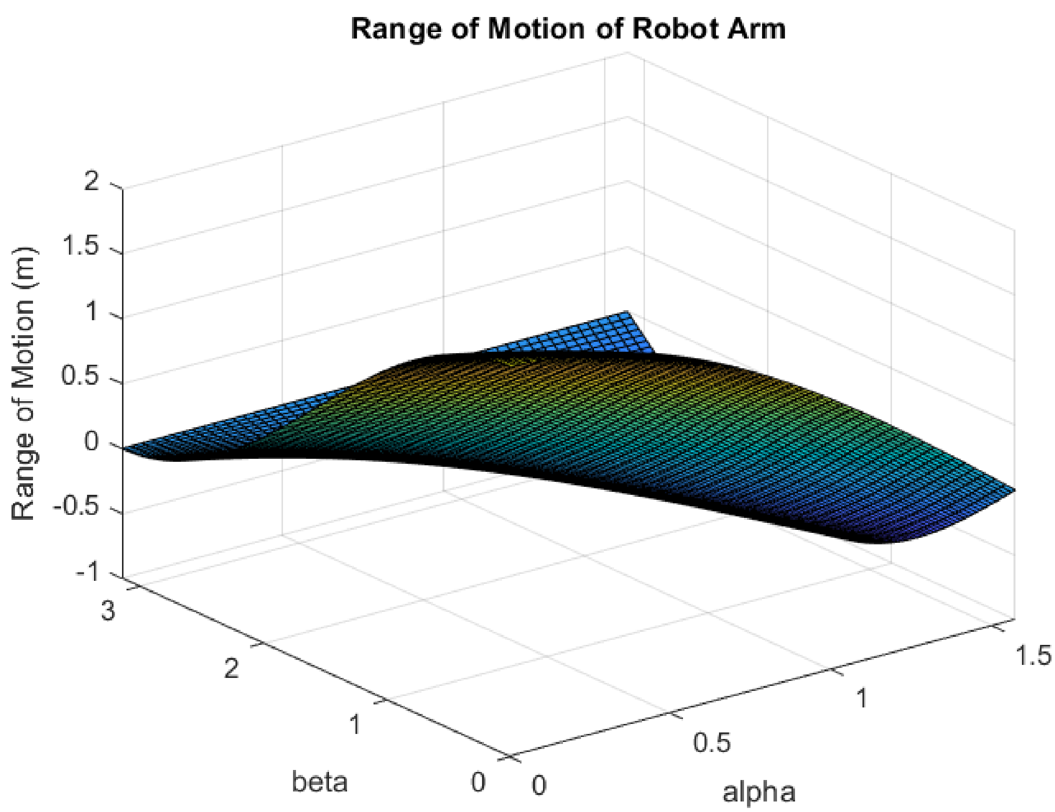
Name: Jingsi Zhou
Date: 11/16/2020
Section: 17460
Assignment: Lab 8

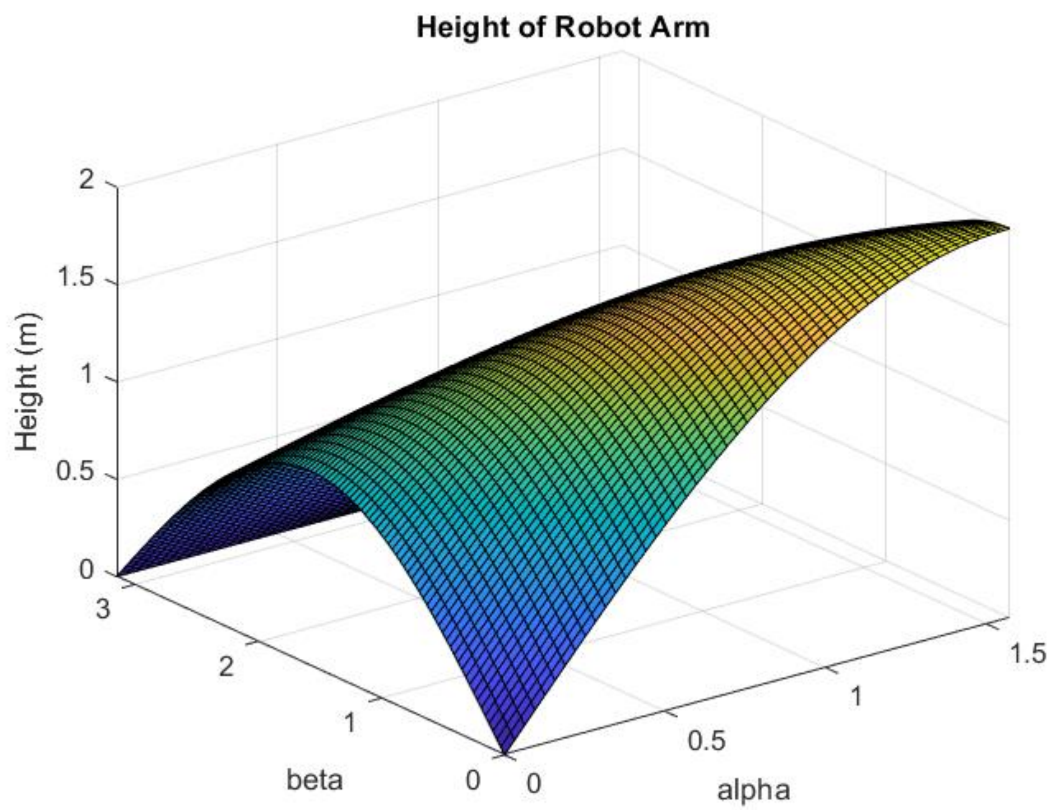
Problem 8.1 Using d_1 = 1.0, d_2 = 1.0, α = 0: π /100: π /2 and β = 0: π /100: π :

1. Plot the range of motion (R) with respect to both angles using **mesh** or **surf**. Include a well labeled plot and lines of code used to generate the plot in your solution. Make sure you comment your code well.




```
alph = 0:(pi/100):pi/2;  %given angle ranges
beta = 0:pi/100:pi;
[x1g, x2g] = meshgrid(alph, beta);  %create grid
y = 1.*cos(x1g) + 1.*cos(x1g+x2g);  % range equation
surf(x1g, x2g, y)                  %plot surface
xlabel('alpha')
ylabel('beta')
zlabel('Range of Motion (m)')
title('Range of Motion of Robot Arm')
```

2. Add to your m-file the commands necessary to similarly plot the range of heights (H) in a new figure window, Figure No. 2. To make sure you've plotted it correctly, ask yourself the following questions: What angles allow the robot arm to reach the highest (the greatest height)? Can the robot arm reach down, at all (negative height)? Does this make sense? (If you are not sure, pretend your arm is the robot's arm and try it out). Include a copy of your figure along with a small write-up about the analysis of the plot.




```
figure(2);
alph = 0:(pi/100):pi/2;  %given angle ranges
beta = 0:pi/100:pi;
[x1g, x2g] = meshgrid(alph, beta);  %create grid
h = 1.*sin(x1g) + 1.*sin(x1g+x2g);  % range equation
surf(x1g, x2g, h)                   %plot surface
xlabel('alpha')
ylabel('beta')
zlabel('Height (m)')
title('Height of Robot Arm')
```

The greatest height that may be reached is when sin(x) is maxed. Given the alpha and beta limits of this problem, the max argument of the first sin is pi/2 while the max argument (x) for the second sin is also pi/2. Thus, the max allowable height is 2 m which is shown by the graph. A negative height is not possible. This is because d2's allowable range of rotation is 180 degrees, which only allows for a net gain in height because it cannot rotate such that d2 goes to a negative height value. Also, because d1's angle of rotation is limited to 90 degrees, only a positive height can be achieved with this angle limit. This also agrees with the plot. Also, there are less height values near 0 and the max height because of the two variables and their angle constraints.

Problem 8.2 Using $d_1 = 1.0$, $d_2 = 1.0$, $R_{des} = 1.0$ and $H_{des} = 1.1$:

1. Define Eq. 8.3 and Eq. 8.4 in a function. This function should take a vector of joint angles (α and β) as input and should return a **column vector** containing the two functions (f_1 and f_2)

evaluated at those angles. The function should contain the link lengths and values of R_{des} and H_{des} for calculation. Include your code in your solution.

```
function out = f1f2(ang)
alph = ang(1);
beta = ang(2);
val1 = 1.*cos(alph) + 1.*cos(alph+beta) - 1;
val2 = 1.*sin(alph) + 1.*sin(alph+beta) - 1.1;
out = [val1; val2];
```

2. Create a separate function that accepts input of a single **row vector** containing the pair of angles α and β. The output of the function should be the Jacobian (a 2 by 2 matrix). First, calculate the derivatives of Eqs. 8.3-8.4 by hand, then put them into your function. Include your function code in your solution.

```
function out = Jacobian(RV)
alph = RV(1);
beta = RV(2);
df1_alph = -1 .* sin(alph) - 1 .* sin(alph + beta);
df1_beta = - 1 .* sin(alph + beta);
df2_alph = 1 .* cos(alph) + 1 .* cos(alph +beta);
df2_beta = 1 .* cos(alph +beta);
out = [df1_alph df1_beta; df2_alph df2_beta];
```

Problem 8.3 Write the code for the above algorithm and include it in your solution.

```
function sol = Newton(G)
count = 0;
x_new = [1; 1];
x_old = G;
while count < 100
    val_old = f1f2(x_old);
    J_old = Jacobian(x_old);
    x_new = x_old - inv(J_old) * val_old;
    x_oldold = x_old;
    x_old = x_new;
    if sqrt((x_old(1)-x_oldold(1))^2 + ((x_old(2)-x_oldold(2))^2)) <=
0.01
        break
    end
    count = count + 1
end
sol = x_new;
```

Problem 8.4 Run your code with accuracy as 0.01, maximum number of steps as 100 and initial guesses of your choice. Use $d_1 = 1.0$, $d_2 = 1.0$, $R_{des} = 1.0$ and $H_{des} = 1.1$.

1. What are the angles that Newton's method returns? Include your guess and the result in the Results section. Describe if it makes sense when seen within the range of motion surface plot. (Just saying “Yes” or “No” will not be given any credit.)

```
>> Newton([pi/4; pi/4])

ans =

    0.1002
    1.4656
```

The angles that the Newton’s method returns is where the desired range is 1.0 and the desired height is 1.1. When these new values are input into the f1 and f2 equations, the value of f1 and f2 turn out to be about 1.0 and 1.1, respectively. These values make sense with the range of motion surface plot because there are possible alpha and beta points that match this answer and that also yield the 1.0 desired range.

2. Find a value for the initial guess x0 that causes Newton's method to fail. What is your “bad” guess? How does this make sense? Write down a thorough reasoning.

```
Newton([0;0])
```

An initial guess that caused the Newton’s method to fail was [0;0] This makes sense because, looking at the range of motion graph, there seems to be no possible alpha and beta value near enough to the initial guess that can yield a range of 1.0. Using the [0;0] guess made the function go through the max number of iterations. MATLAB also gave a warning that some of the intermediate matrices may have been singular or difficult to round.
