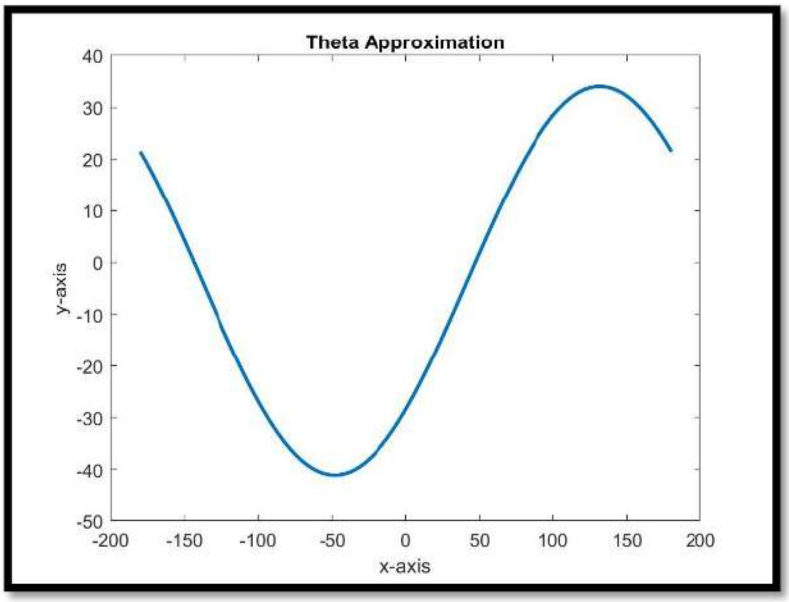
COMPUTING ASSIGNMENT 2

NAVJOT KAUR (301404765)

In the MATLAB code provided below, a graph is plotted in degrees by taking θ on x-axis and $f(\theta)$ on y-axis. The plot shown below has two roots at approximately -143.5899 and 47.1131. The graph is continuous and differentiable on the given interval i.e., [-180°, 180°] and have a wave-like behavior. To find the approximate roots to the given anonymous function $f(\theta) = wSin(\theta) - hcos(\theta) - b$, three different types of root finding algorithms are used namely Bisection method, Fixed point method and Newton's method. All the calculations for three methods are first done in the form of radian and then displayed in the form of degrees.

Plot:



Comparing the cost of three methods:

Bisection method converges to 46.406250 with 6 iterations with an initial guess of [0°, 90°] and 1° of tolerance. Here no. of iterations = cost because only $f(\theta)$ is used. So, cost = 6.

Fixed point method uses the initial guess of [-180°, 180°] and it converges to root 46.635966 with 48 iterations. The function $gfunc$ used here is $\theta = \frac{\arcsin(hcos\theta + b)}{w}$. Here no. of iterations = cost because only $f(\theta)$ is used.
So, cost = 48.

Newton's method converges to 47.110405 with 2 iterations where the initial guess is $\theta = 0°$. Here cost = no. of iterations*2. So, cost = 4.

The differences in the cost is that Newton's method converges very fast in just 2 iterations with cost = 4 whereas Bisection method and Fixed-point method took 6 and 48 iterations, respectively.

It can be concluded from comparing the costs that the Newton's method is the closest approximation to the exact root that is calculated in part (c). The exact root is 47.110456 and the Newton's approximation is 47.110405, which is true up to 6 significant digits as compared to Fixed point and Bisection which are true up to 1 significant digit after round off.

Using the most accurate approximation that is of Newton's method, $d1$ and $d2$ are calculated where

$d1 = 10.041931$ and $d2 = 17.060968$, that are obtained by putting the values of h=25, w=28, b=3.5, a, c and α. After that, $d1$ and $d2$ are calculated by the exact root where $d1 = 10.041916$ and $d2 = 17.060954$.

After computing the relative error, the relative error in $d1 = 0.000001$ and in $d2 = 0.000001$. It can be concluded from these values that $d1$ and $d2$ are accurate within a tolerance of 0.1 inches.

NAVJOT KAUR (301404765)

MATLAB CODE

```
function f_root = RootFinding()
%part(A)
w = 28; h = 25; b = 3.5;
angle = rad2deg(pi);
theta = linspace(-angle,angle);
f=@(theta) (w*sind(theta)) - (h* cosd(theta))- b;
plot(theta,f(theta),"LineWidth",2);
xlabel("x-axis"); ylabel("y-axis");
title("Theta Approximation");
%part b
func = '28*sin(theta)-25*cos(theta)-3.5'; %bisect method
[b_root, nitr, rlist] = bisect2(func, [0,pi/2], pi/180);
b_root=rad2deg(b_root);
fprintf("Bisection Method\n");
fprintf("Roots :  %f\n", b_root);
fprintf("Iterations :  %i\n", nitr);
gfunc = 'asin(((25*cos(theta))+3.5)/28)'; %fixed point
[xfinal, niter, xlist] = fixedpt(gfunc, [-pi,pi], pi/180);
xfinal=rad2deg(xfinal);
fprintf("Fixed Point Method\n");
fprintf("Roots :  %f\n", xfinal);
fprintf("Iterations :  %i\n", niter);
func_1 = '28*sin(theta)-25*cos(theta)-3.5';
f_derivative = '28*cos(theta)+25*sin(theta)'; %newton method
[n_root, iter, xlist] = newton(func_1, f_derivative, 0, pi/180);
fprintf("Newton method\n");
n_root = rad2deg(n_root);
fprintf("Roots :  %f\n", n_root);
fprintf("Iterations :  %i\n", iter);
%part c
theta_star = acos(((-b*h)+sqrt((b^2*h^2)+(h^2+w^2)*(w^2-b^2)))/(h^2+w^2));
theta_star=rad2deg(theta_star);
fprintf("Exact root :  %f\n", theta_star);
fprintf("Most approximate root is calculated using Newton's method. \n");
%part d
d2_approximate=(h)/(2*sind(n_root));
alpha_approximate=(90-n_root);
a_approximate=b/tand(alpha_approximate);
c_approximate=b/tand(n_root);
d1_approximate = d2_approximate - a_approximate - c_approximate;
fprintf("Approximate value of d2 when calculated with newton root is :  %f\n",
d2_approximate);
fprintf("Approximate value of d1 when calculated with newton root is :  %f\n",
d1_approximate);
d2_star=(h)/(2*sind(theta_star));
alpha_star=(90-theta_star);
a_star=b/tand(alpha_star);
c_star=b/tand(theta_star);
d1_star=d2_star - a_star - c_star;
d1_relative = abs((d1_star - d1_approximate)/d1_star);
d2_realtive = abs((d2_star - d2_approximate)/d2_star);
fprintf("Value of d2 when calculated with exact root is :  %f\n", d2_star);
fprintf("Value of d1 when calculated with exact root is :  %f\n", d1_star);
fprintf("Relative value of d1 is :  %f\n", d1_relative);
fprintf("Relative value of d2 is :  %f\n", d2_realtive);
end
```