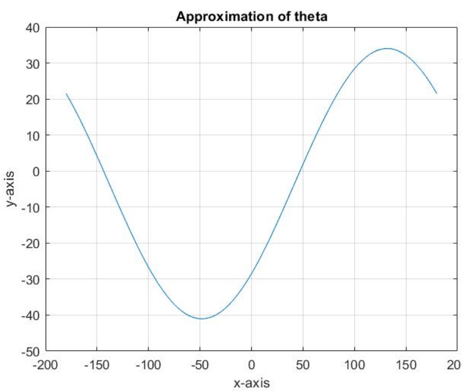
RITIKA GOYAL (301401516)

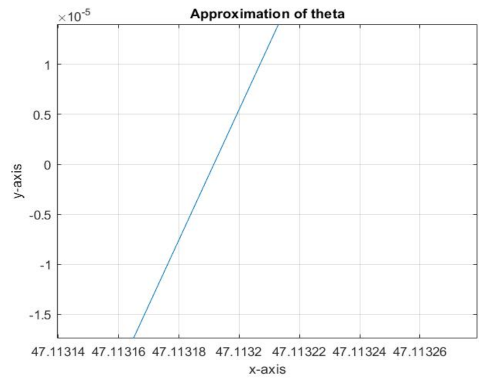
MACM316 CA2

(a): The given values of h = 25, w = 28 and b = 3.5 is used to calculate roots of an anonymous function f = @(theta) w*sind(theta) – h*cosd(theta) -b by plotting it on MATLAB on the interval [-180,180]. From the given graph, it is seen that there are two roots of this equation and its behavior is wave like(sinusoidal). The graph is continuous and differential in the given interval. After zooming in, the approximate location of roots are, 47.11319 and -143.589963.

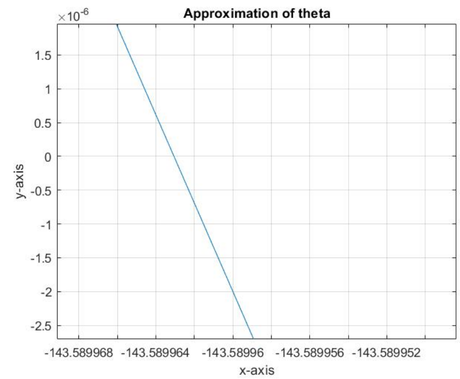
Plot without zoomed



First root when zoomed



Second root when zoomed



(b): Computed positive root by three root-finding algorithms:

METHOD	ROOTS	NITTER	TOL	COST	EXPLAINATION OF COST
Bisection	46.406250	6	1 degree	6	The number of iterations = Cost, because we are just computing value of f(x)
Fixed	46.635966	48	1 degree	48	The number of iterations = Cost, because we are just computing value of g(x)
Newton	47.110405	2	1 degree	2*2=4	The number of iteration *2 = Cost, because we are computing value of both f(x) and f'(x) which doubles the work.

Cost of Newton< Bisection< Fixed and Newton gave the closest root to the exact value which is calculated in part c.

(c): Calculated the exact value of theta by putting h, w, b in the equation:
Theta = acos ((-b*h + sqrt(b^2*h^2) + (h^2+w^2) * (w^2-b^2)))/(h^2 + w^2)). The exact root is 47.110456 which matches up to 7 significant digits with Newton's method while Fixed method matches up to 2 significant digits (after rounding) and Bisection matches just one significant digit (after rounding).

The absolute and relative errors in the calculation of positive roots with different methods are:
Bisection method: absolute error = 0.704206 , relative error = 0.0000015
Fixed method: absolute error = 0.47449 , relative error = 0.0100719
Newton method: absolute error = 0.000051 , relative error = 0.0000010

(d): The most accurate approximation from part b is Newtons method whose root is used to find the measurement of d1 and d2.
Newtons method: d1 = 10.041931, d2 = 17.060968
Exact method: d1 = 10.041916, d2 = 17.060954
The relative error in d1 and d2 (abs((exact – calculated)/exact)) is 0.000001 which is less that 0.1 tolerance.

RITIKA GOYAL (301401516)

MATLAB CODE

```
function  root_geo_design()
w=28;h=25;b=3.5;
% part a:
fprintf("part a :\n");
left_lim = rad2deg(-pi);
right_lim = rad2deg(pi);
theta=linspace(left_lim,right_lim);
f = @(theta)  w*sind(theta) - h*cosd(theta)-b;
plot(theta,f(theta));
grid on;
xlabel("x-axis");
ylabel("y-axis");
title("Approximation of theta");
disp("approximate roots are (found by zooming in the graph) = -143.5899,
47.11319.");
fprintf("Number of roots = 2\n\n");
%part b)
fprintf("part b:\n");
[bisect_root, niter_bisect] = bisect2( '28*sin(theta) - 25*cos(theta)-3.5',
[0,pi/2],0.017453);
bisect_root = rad2deg(bisect_root);
fprintf('root by BISECTION method = %f, Number of iterations in bisect method =
%d\n', bisect_root, niter_bisect);
[fixed_root, niter_fixed] = fixedpt( 'asin((25*cos(theta)+3.5)/28)' , [-pi,pi] ,
0.017453);
fixed_root = rad2deg(fixed_root);
fprintf('root by FIXED method = %f,',  fixed_root);
fprintf('Number of iterations in fixed method = %d\n',niter_fixed);
[newton_root, niter_newton] = newton( '28*sin(theta) - 25*cos(theta)-3.5',
'28*cos(theta)+25*sin(theta)', 0 , 0.017453);
newton_root = rad2deg(newton_root);
fprintf('root by NEWTON method = %f, Number of iterations in newton method =
%d\n\n', newton_root, niter_newton);
%part c)
fprintf("part c: \n");
thetaStar = acos((-b*h + sqrt((b^2*h^2) + (h^2+w^2)*(w^2-b^2) ))/(h^2 + w^2));
thetaStar = rad2deg(thetaStar);
fprintf("EXACT root : %f \n\n",thetaStar);
fprintf("part d: \n");
%part d)
%values of exact d1 and d2
exact_d2 =  h/(2*sind(thetaStar));
exact_alpha = rad2deg(pi/2) - thetaStar;
exact_d1 = exact_d2 - (b/tand(exact_alpha)) - (b/tand(thetaStar));
fprintf("Using exact value of theta: d1 = %f, d2 = %f\n\n ",exact_d1,exact_d2);
%values of d1 and d2 by newton method
d2 = h/(2*sind(newton_root));
%b2 = w - 2*d2*cosd(newton_root);
alpha = rad2deg(pi/2) - newton_root;
d1 = d2 - (b/tand(alpha)) - (b/tand(newton_root));
fprintf("Using APPROXIMATED value of theta: d1 = %f, d2 = %f\n\n ",d1,d2);
%claculating relative error to compute accuracy
re_d1 = abs((exact_d1 - d1)/exact_d1);
re_d2 = abs((exact_d2 - d2)/exact_d1);
fprintf("relative error in d1: %f\nrelative error in d2: %f\n",re_d1,re_d2);
end
```