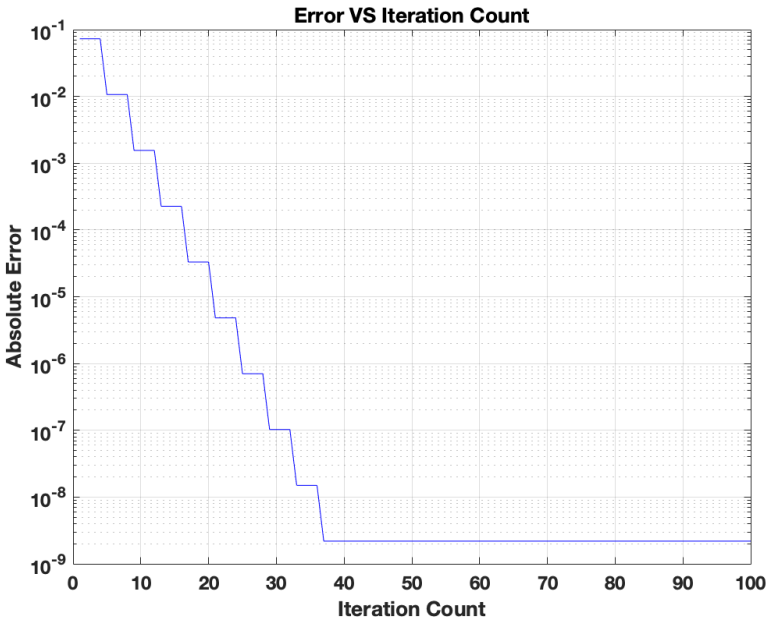
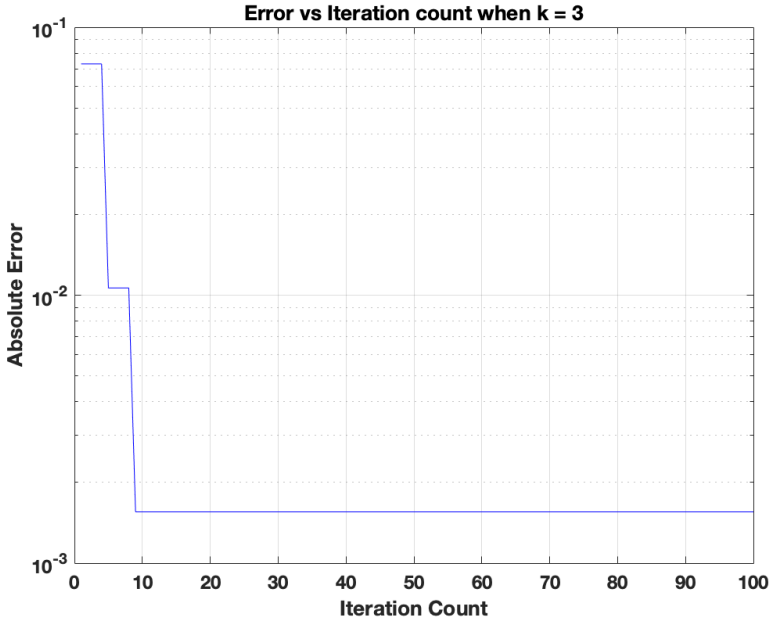
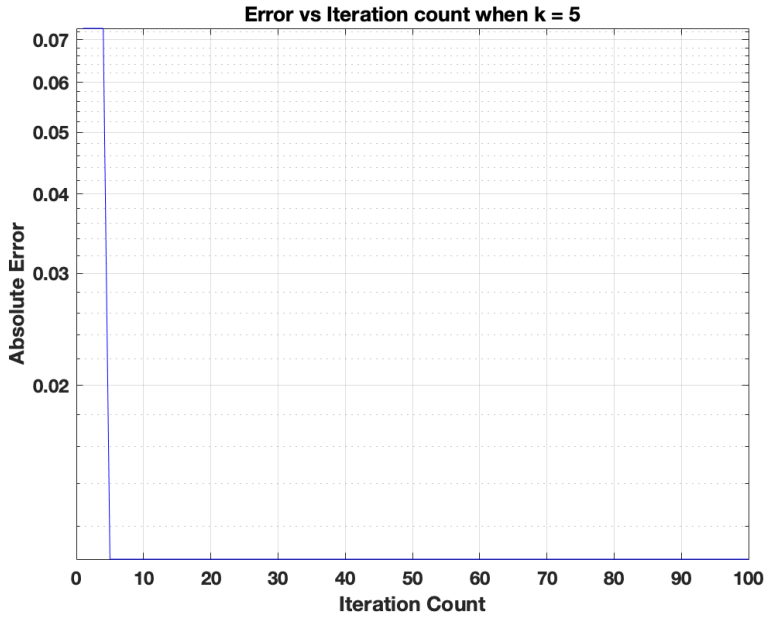
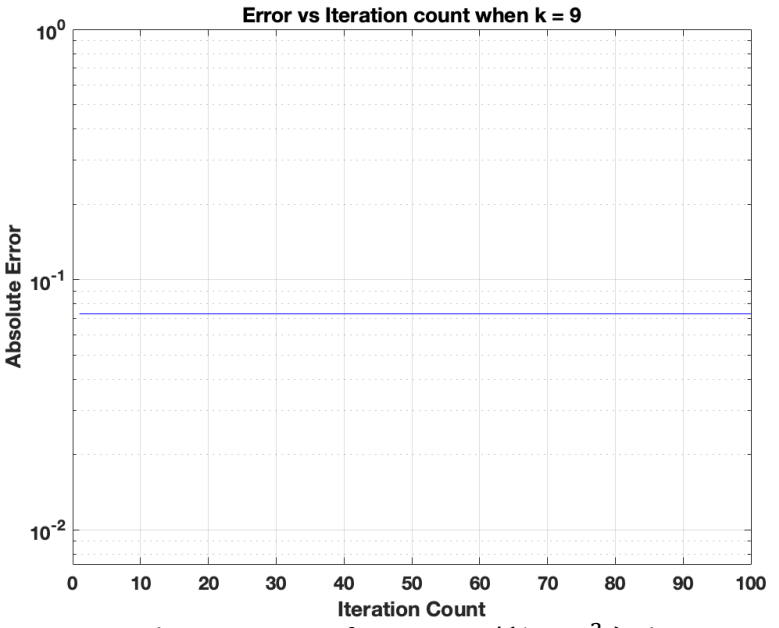
MACM316 COMPUTING ASSIGNMENT 4 – Ngawang Kyirong – 301312227









* The top left figure shows the absolute error of the approximation to the minimum of $y = cosx/(1 + x^2)$ changing with respect to the amount iterations of the algorithm used. This algorithm appears to increase in accuracy as the number of iterations increase until around the 38th iteration where it flattens out to just under 10^{-9} due to round off error. I would consider this accurate and especially after the 38th iteration. In terms of efficiency, I would not consider this algorithm efficient as it takes multiple iterations (4 to 5) per error decrease. For robustness, I would also not consider this robust due to the round off error causing it flatten before 10^{-9} as this is not machine epsilon.

* For the second question (y = $cosx^k/(1 + x^{2k})$) , as k increases, the error is also increasing which can be seen in the 3 graphs above. This result shows that the error is changing with respect to the parameters in a bad way. This is not accurate. For efficiency, this converges much quicker than the graph shown in the top left. It is efficient as every iteration is contributing to error reduction until it flattens out. However, the amount of flops is increasing as k increases as well. In terms of robustness, it is also not robust because of the fact that increasing values of k lead to higher error, thus trending away from machine epsilon due to round off error (flattening out).

* Lastly, x should only get to within 10^{-8} of the true minimum. If x* represents the minimum then $f'(x^*)$ = 0. Thus, the second iteration of taylors expansion is: $f(x) = f(x^*) + \frac{1}{2}f''(x^*)(x - x^*)^2$. If x is approaching x* from the algorithm, the (x-x*) term in taylors expansion will be small. Furthermore, this means $(x - x^*)^2$ is even smaller. Moreover, we will reach a point where $f(x) \approx f(x^*)$ and the computer will see this as $f(x) = f(x^*)$ because of machine epsilon. Therefore, solving for the inequality: $\frac{1}{2}f''(x^*)(x - x^*)^2 < \varepsilon|f(x^*)|$ (this means finding a point x closest to x*) will show us that 10^{-8} is the closest we can get.

A4.m

```
% part 1
%f = @(x) -cos(x)/(1 + power(x,2));

% part 2
%k = 5;
%k = 9;
k = 3;
f = @(x) -cos(power(x,k))/(1 + power(x,2*k));

% params
minimum = 0;
N = 100;
an = -1;
bn = 1/2;
cn = 1;
error=1:N;

for i = 1:N
    [an,bn,cn] = minima(f, an, bn, cn);
    error(i) = abs(minimum-bn);
end

semilogy(1:N,error, 'b')  % Plot the error versus iteration number
%title('Error VS Iteration Count')
title('Error vs Iteration count when k = 3')
xlabel('Iteration Count')
ylabel('Absolute Error')
grid on;
```

minima.m

```
% function to compute algorithm
function [a,b,c] = minima(f, an, bn, cn)
gamma = (3-sqrt(5))/2;


if ( (cn-bn) > (bn-an) )
    x = bn + gamma * ( cn - bn );
elseif ( (cn-bn) < (bn-an) )
    x = bn + gamma * ( an - bn );
end

if ( x<bn && f(x)<f(bn) )
    a = an;
    b = x;
    c = bn;
elseif ( x<bn && f(x)>f(bn) )
    a = x;
    b = bn;
    c = cn;
elseif ( x>bn && f(x)<f(bn) )
    a = bn;
    b = x;
    c = cn;
else
    a = an;
    b = bn;
    c = x;

end
```