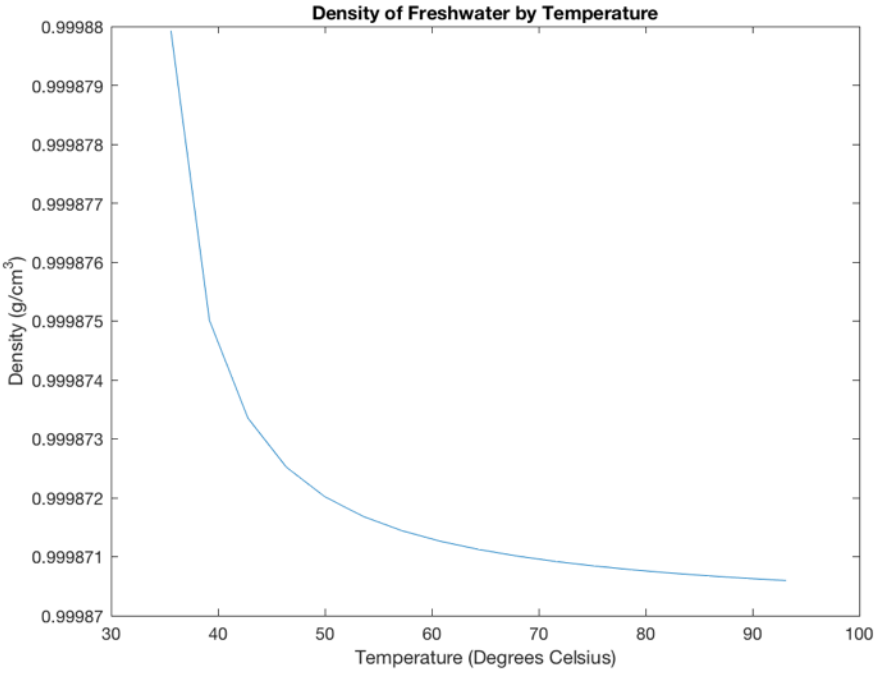
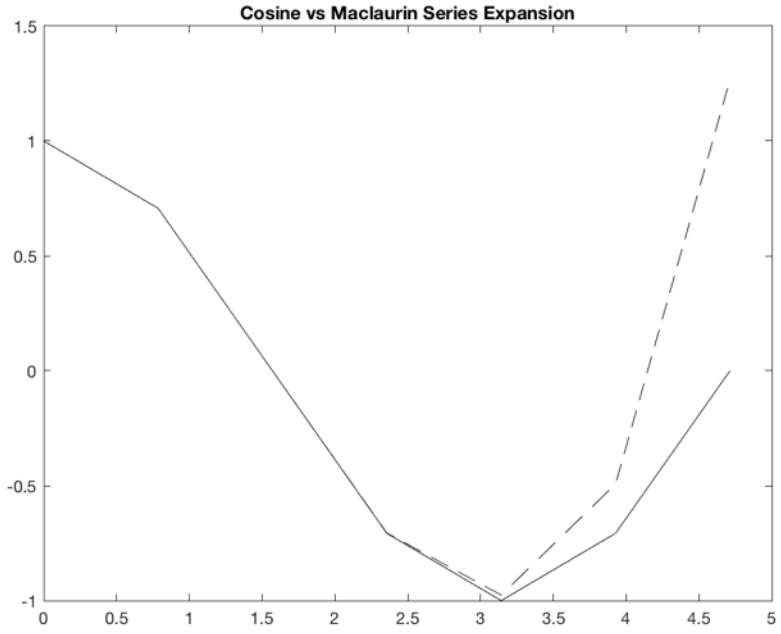
Problem 1

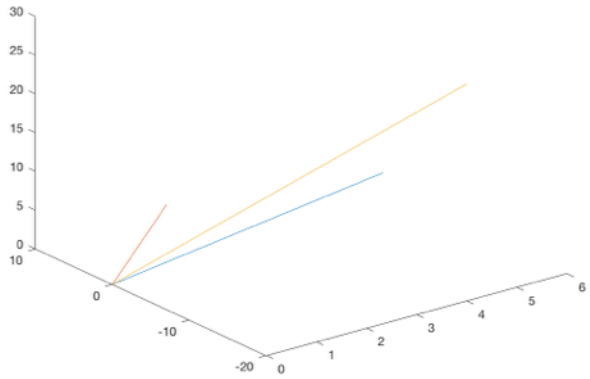


Problem 2



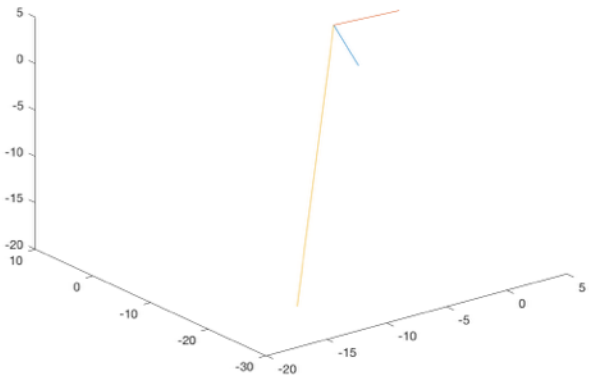
Problem 3

x	y	θ (radians)	r
2	0	2.0000	0
2	1	2.2361	0.4636
0	3	3.0000	1.5708
-3	1	3.1623	2.8198
-2	0	2.0000	3.1416
-1	-2	2.2361	-2.0344
0	0	0	0
0	-2	2.0000	-1.5708
2	2	2.8284	0.7854



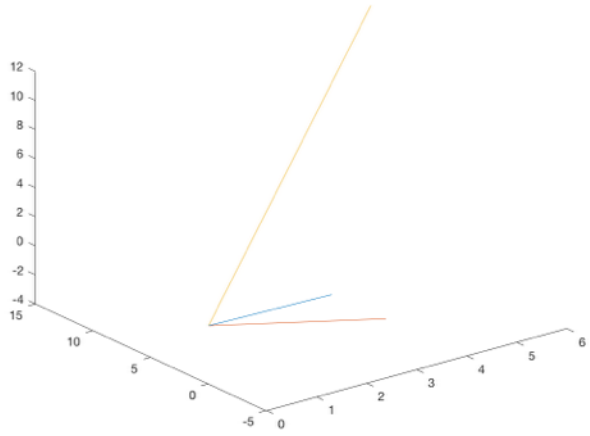
4a)

θ = 0.6669 radians
c = <4, -20, 28>
lcl = 34.6410



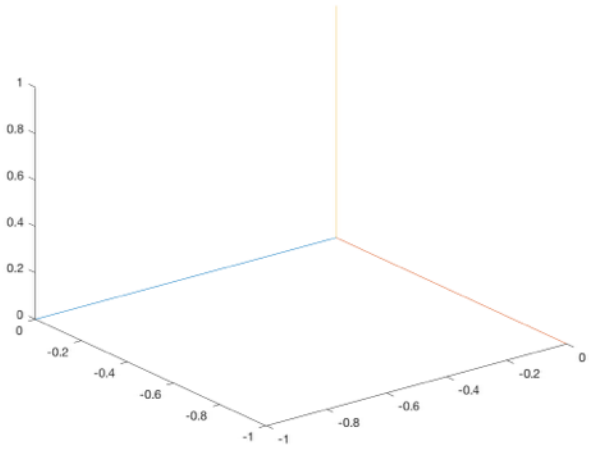
4b)

θ = 1.5708 radians
c = <-16, -27, -17>
lcl = 35.6931



4c)

θ = 1.5708 radians
c = <6, 12, 12>
|c| = 18



4d)

θ = 1.5708 radians
c = <0, 0, 1>
|c| = 1

```
% Problem 1

y1 = [];
for t = 32:3.6:93.2
    y1 = [y1 waterdensity(t)];
end

plot(32:3.6:93.2, y1);
title('Density of Freshwater by Temperature');
xlabel('Temperature (Degrees Celsius)');
ylabel('Density (g/cm^3)');

% Problem 2

x2 = 0:pi/4:(3 * pi)/2;
y2_1 = [];
for i = 0:pi/4:(3 * pi)/2
    y2_1 = [y2_1 maclaurin(i, 4)];
end
y2_2 = [];
for i = 0:pi/4:(3 * pi)/2
    y2_2 = [y2_2 cos(i)];
end

plot(x2, y2_1, '--k', x2, y2_2, '-k');
title('Cosine vs Maclaurin Series Expansion');

% Problem 3

x3 = [2 2 0 -3 -2 -1 0 0 2];
y3 = [0 1 3 1 0 -2 0 -2 2];
A = [];

for i = 1:length(x3)
    [r, t] = cartesianToPolar(x3(i), y3(i));
    A = [A ; r t];
end

% Problem 4

A1 = [6, 4, 2];
B1 = [2, 6, 4];
plotCrossProduct(A1, B1);

A2 = [3, 2, -6];
B2 = [4, -3, 1];
plotCrossProduct(A2, B2);

A3 = [2, -2, 1];
```

```
B3 = [4, 2, -4];
plotCrossProduct(A3, B3);

A4 = [-1, 0, 0];
B4 = [0, -1, 0];
plotCrossProduct(A4, B4);

function m = maclaurin(x, i)
% maclaurin:
% m = expansion of the maclaurin series up to the term x^8/8!
% input:
% x = value of x
% i = number of terms
% output:
% m = maclaurin series expansion

m = 1;
for j = 1:i
   term = (x^(2 * j))/factorial(2 * j);
   if mod(j,2) == 0
      m = m + term;
   else
      m = m - term;
   end
end

function d = waterdensity(f)
% waterDensity: Density of freshwater computed as a function of temperature
% d = waterDensity computes the density of freshwater as a function of
% temperature
% input:
% f = temperature in Fahrenheit
% output:
% d = density (g/cm^3)

t = fahrenheit_celsius(f)
d = (5.5289 * 10^(-8) * t^3) - (8.5016 * 10^(-6) * t^2) + ...
   (6.5622 * 10^(-5) * t) + 0.99987;

function c = fahrenheit_celsius(f)
% fahrenheit_celsius: Converts degrees fahrenheit to degrees celsius
% c = celsius conversion
% input:
% f = temperature in Fahrenheit
% output:
% c = temperature in Celsius

c = 5/(9 * (f - 32));
```

```
function [r, t] = cartesianToPolar(x, y)
% cartestianToPolar: converts cartesian to polar coordinates
% input:
% x = value of x in cartesian coordinates
% y = value of y in cartesian coordinates
% output:
% r = radius in polar coordinates
% t = theta (radians) in polar coordinates
r = getRadius(x, y);
t = getTheta(x, y);

function r = getRadius(x, y)
% getRadius: get value of radius from x and y
% input:
% x = value of x in cartesian coordinates
% y = value of y in cartesian coordinates
% output:
% r = radius in polar coordinates
r = sqrt(x^2 + y^2);

function t = getTheta(x, y)
% getTheta: get value of theta (radians) from x and y
% input:
% x = value of x in cartesian coordinates
% y = value of y in cartesian coordinates
% output:
% t = theta in polar coordinates
if (x > 0)
   t = atan(y / x);
elseif (x < 0 && y > 0)
   t = atan(y / x) + pi;
elseif (x < 0 && y < 0)
   t = atan(y / x) - pi;
elseif (x < 0)
   t = pi;
elseif (x == 0 && y > 0)
   t = pi / 2;
elseif (x == 0 && y < 0)
   t = - pi / 2;
else
   t = 0;
end

function [t, c, m] = plotCrossProduct(a, b)
% plotCrossProduct: plot cross product of vectors a and b going through the origin.
% input:
% a = vector a
% b = vector b
% output:
```

```
% t = value of theta from the dot product (a * b = |a||b| cos(theta))
% c = vector retrieved from cross product of a and b
% m = magnitude of vector c
t = getTheta(a, b);
[c, m] = getCrossProduct(a, b);
a = [0 0 0; a];
b = [0 0 0; b];
c = [0 0 0; c];
plot3(a(:, 1), a(:, 2), a(:, 3), b(:, 1), b(:, 2), b(:, 3), ...
    c(:, 1), c(:, 2), c(:, 3));
function t = getTheta(a, b)
t = acos(dot(a, b) / (norm(a) * norm (b)));

function [c, m] = getCrossProduct(a, b)
c = cross(a, b);
m = norm(c);
```