ME30400

Assignment 2

ME21BTECH11001

Abhishek Ghosh

Question 1

A triangular face has the homogeneous coordinates P1 (3, 1.5, 2, 1), P2 (2.5, 2, 2, 1) and P3 (3, 2, 1.5, 1). Develop the auxiliary view to show the true shape of the triangle by rotating the vector passing through the centroid of the triangle and directed along the outward normal to the triangular face and projecting the triangle on the x – y plane. Find the final position vectors of the points P1, P2 and P3. Write a code to solve this problem.

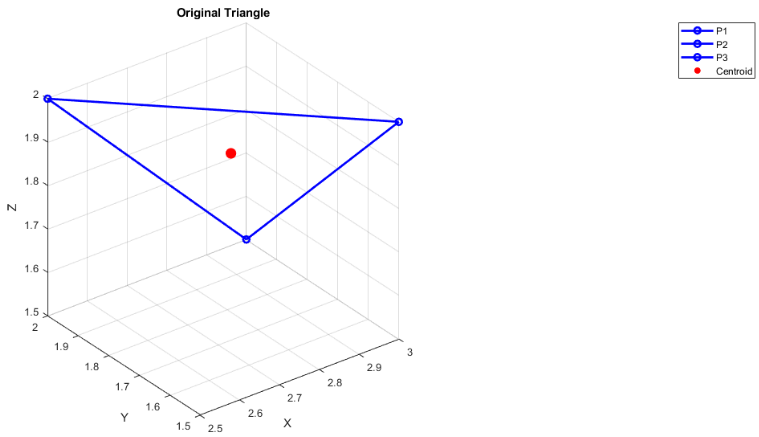
Ans:

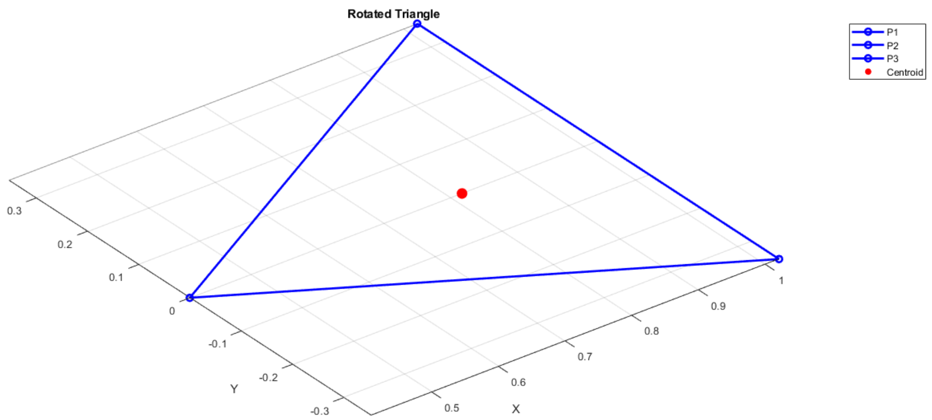
```
% % ME21BTECH11001
% % Abhishek Ghosh

% Define the points
P1 = [3, 1.5, 2, 1];
P2 = [2.5, 2, 2, 1];
P3 = [3, 2, 1.5, 1];

% Calculate the centroid
C = (P1 + P2 + P3) / 3;

% Calculate vectors along the edges of the triangle
v1 = P2(1:3) - P1(1:3);
v2 = P3(1:3) - P1(1:3);

% Calculate the normal vector of the triangle
normal = cross(v1, v2);
normal = normal / norm(normal);
normal = [normal, 1]; % Append 1 to make it homogeneous

% Calculate the angle alpha for rotation around the x-axis
alpha = atan2(normal(1), normal(2));
Rx = [
    1, 0, 0, 0;
    0, cos(alpha), sin(alpha), 0;
    0, -sin(alpha), cos(alpha), 0;
    0, 0, 0, 1
];

% Rotate the normal vector around the x-axis
normal_Rx = normal * Rx;

% Calculate the angle beta for rotation around the y-axis
beta = atan2(normal_Rx(1), normal_Rx(3));
Ry = [
    cos(beta), 0, sin(beta), 0;
    0, 1, 0, 0;
    -sin(beta), 0, cos(beta), 0;
    0, 0, 0, 1
];

% Rotate the normal vector around the y-axis
normal_z = normal_Rx * Ry;

% Apply the combined rotation matrix to the points
```

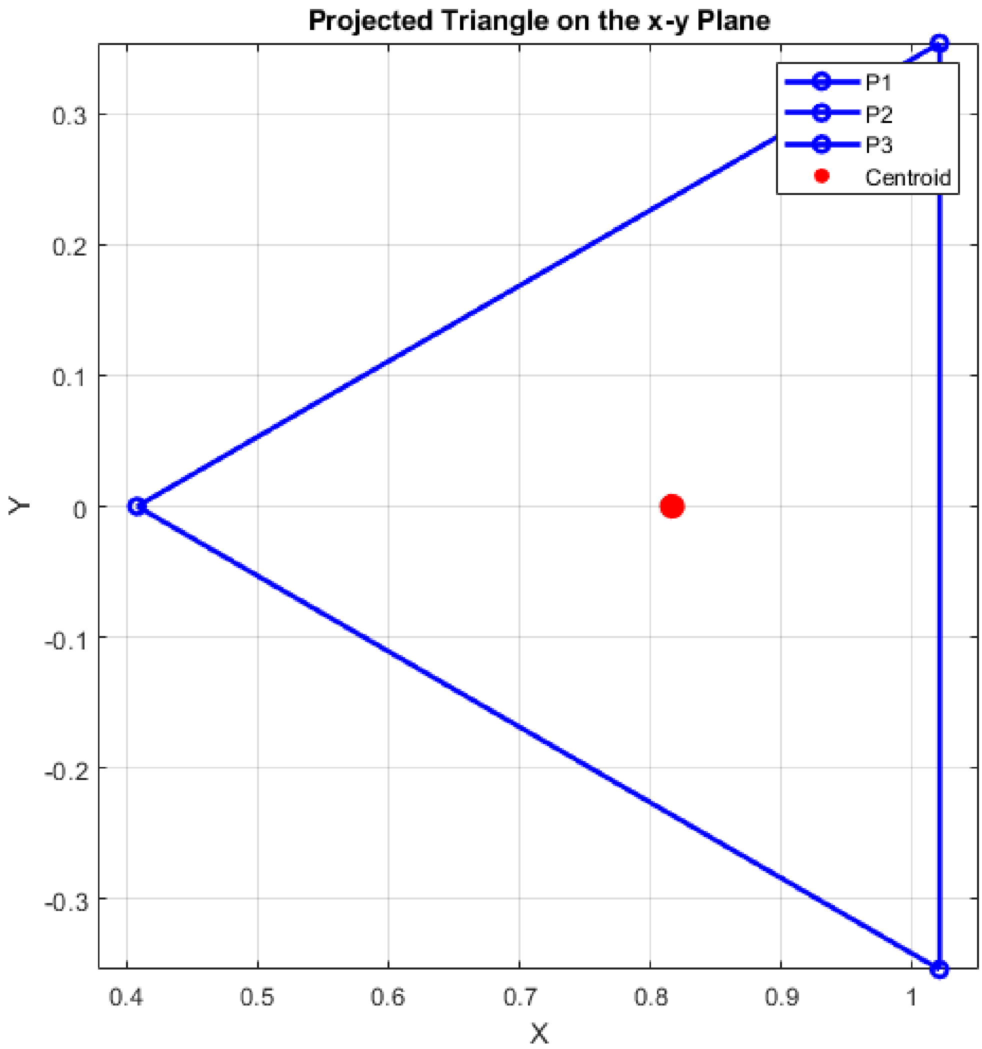
```
transform = Rx * Ry;
P1_rot = P1 * transform;
P2_rot = P2 * transform;
P3_rot = P3 * transform;
C_rot = (P1_rot + P2_rot + P3_rot) / 3;

% Projection matrix for x-y plane
xy_proj = [
    1, 0, 0, 0;
    0, 1, 0, 0;
    0, 0, 0, 0;
    0, 0, 0, 1
];

% Project the rotated points onto the x-y plane
P1_proj = P1_rot * xy_proj;
P2_proj = P2_rot * xy_proj;
P3_proj = P3_rot * xy_proj;
C_proj = (P1_proj + P2_proj + P3_proj) / 3;

% Display the final position vectors
disp('Final position vectors:');
disp(['P1: ', mat2str(P1_proj)]);
disp(['P2: ', mat2str(P2_proj)]);
disp(['P3: ', mat2str(P3_proj)]);

% Define edge color
edgeColor = 'b'; % Blue for all edges

% Plot the original triangle
figure('Position', [100, 100, 1500, 600]);
% subplot(1, 2, 1);
plot3([P1(1), P2(1)], [P1(2), P2(2)], [P1(3), P2(3)], [edgeColor, 'o-'],
'LineWidth', 2);
hold on;
plot3([P2(1), P3(1)], [P2(2), P3(2)], [P2(3), P3(3)], [edgeColor, 'o-'],
'LineWidth', 2);
plot3([P3(1), P1(1)], [P3(2), P1(2)], [P3(3), P1(3)], [edgeColor, 'o-'],
'LineWidth', 2);
scatter3(C(1), C(2), C(3), 100, 'r', 'filled'); % Red centroid
title('Original Triangle');
xlabel('X');
ylabel('Y');
zlabel('Z');
legend('P1','P2','P3', 'Centroid');
grid on;
axis equal;

% Plot the rotated triangle
figure('Position', [100, 100, 1500, 600]);
% subplot(1, 2, 2);
plot3([P1_rot(1), P2_rot(1)], [P1_rot(2), P2_rot(2)], [P1_rot(3), P2_rot(3)],
[edgeColor, 'o-'], 'LineWidth', 2);
hold on;
plot3([P2_rot(1), P3_rot(1)], [P2_rot(2), P3_rot(2)], [P2_rot(3), P3_rot(3)],
[edgeColor, 'o-'], 'LineWidth', 2);
plot3([P3_rot(1), P1_rot(1)], [P3_rot(2), P1_rot(2)], [P3_rot(3), P1_rot(3)],
[edgeColor, 'o-'], 'LineWidth', 2);
scatter3(C_rot(1), C_rot(2), C_rot(3), 100, 'r', 'filled'); % Red centroid
title('Rotated Triangle');
xlabel('X');
ylabel('Y');
zlabel('Z');
legend('P1','P2','P3', 'Centroid');
grid on;
axis equal;

% Plot the projected triangle on the x-y plane
```

```
figure('Position', [700, 100, 600, 600]);
plot([P1_proj(1), P2_proj(1)], [P1_proj(2), P2_proj(2)], [edgeColor, 'o-'],
'LineWidth', 2);
hold on;
plot([P2_proj(1), P3_proj(1)], [P2_proj(2), P3_proj(2)], [edgeColor, 'o-'],
'LineWidth', 2);
plot([P3_proj(1), P1_proj(1)], [P3_proj(2), P1_proj(2)], [edgeColor, 'o-'],
'LineWidth', 2);
scatter(C_proj(1), C_proj(2), 100, 'r', 'filled'); % Red centroid
title('Projected Triangle on the x-y Plane');
xlabel('X');
ylabel('Y');
grid on;
axis equal;
legend('P1','P2','P3', 'Centroid');
```

```
Final position vectors:

P1: [1.02062072615966 0.353553390593274 0 1]

P2: [0.408248290463863 2.22044604925031e-16 0 1]

P3: [1.02062072615966 -0.353553390593274 0 1]
```







Published with MATLAB® R2023a

Question 2

A dimetric projection is constructed by a rotation about the y axis by an angle ϕ followed by a rotation about the x axis through an angle θ and projection onto the z plane. Derive the foreshortening ratio, fx, fy and fz in terms of θ and ϕ. If $f_z = 1/\sqrt{2}$, calculate the possible values of θ and ϕ.

Ans:

Questn 2

Given, dimetric projectn

→ Rotatn about y axis by ϕ $= R_y$

→ Rotatn about x axis by θ $= R_x$

→ Projectn onto z-plane $= P_z$

The concatenatn matrix

$$[T] = [R_y][R_x][P_z]$$

$$= \begin{bmatrix} \cos\phi & 0 & -\sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\phi & 0 & -\sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & 0 & 0 \\ 0 & -\sin\theta & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow [T] = \begin{bmatrix} \cos\phi & \sin\phi\sin\theta & 0 & 0 \\ 0 & \cos\theta & 0 & 0 \\ \sin\phi & -\cos\phi\sin\theta & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The unit vector on the x, y and z principal axes transform to

$$[U][T] = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & \sin\phi\sin\theta & 0 & 0 \\ 0 & \cos\theta & 0 & 0 \\ \sin\phi & -\cos\phi\sin\theta & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\phi & \sin\phi\sin\theta & 0 & 1 \\ 0 & \cos\theta & 0 & 1 \\ \sin\phi & -\cos\phi\sin\theta & 0 & 1 \end{bmatrix} = \begin{bmatrix} x^*_x & y^*_x & 0 & 1 \\ x^*_y & y^*_y & 0 & 1 \\ x^*_z & y^*_z & 0 & 1 \end{bmatrix}$$

2. forshortening ratios

$$b_x = \sqrt{x_x^{*2} + y_x^{*2}} = \sqrt{\cos^2\phi + \sin^2\phi \sin^2\theta}$$

$$b_y = \sqrt{x_y^{*2} + y_y^{*2}} = \sqrt{\cos^2\theta} = \cos\theta$$

$$b_z = \sqrt{x_z^{*2} + y_z^{*2}} = \sqrt{\sin^2\phi + \cos^2\phi \sin^2\theta}$$

for dimetric projectn

$$b_x = b_y$$

$$\Rightarrow \cos^2\phi + \sin^2\phi \sin^2\theta = \cos^2\theta$$

$$\Rightarrow 1 - \sin^2\phi + \sin^2\phi \sin^2\theta = \cos^2\theta$$

$$\Rightarrow \sin^2\phi = \frac{\sin^2\theta}{1-\sin^2\theta} = \tan^2\theta$$

$$b_z^2 = \frac{\sin^2\theta}{1-\sin^2\theta} + \left(1 - \frac{\sin^2\theta}{1-\sin^2\theta}\right)\sin^2\theta$$

$$= \frac{2\sin^2\theta - 2\sin^4\theta}{1-\sin^2\theta}$$

$$\Rightarrow 2\sin^4\theta - (2+b_z^2)\sin^2\theta + b_z^2 = 0$$

By solving quadratic eqtn we get,

$$b_z^2 = 2\sin^2\theta \Rightarrow \sin\theta = \pm\frac{b_z}{\sqrt{2}}$$

$$\Rightarrow \boxed{\theta = \sin^{-1}\left(\frac{\pm b_z}{\sqrt{2}}\right)}$$

$$\boxed{\phi = \sin^{-1}\left(\frac{\pm b_z}{\sqrt{2-b_z^2}}\right)}$$

Given, $b_2 = \frac{1}{\sqrt{2}}$

$$\theta = \sin^{-1}\left(\pm \frac{1}{2}\right) \Rightarrow \theta = \pm \frac{\pi}{6}$$

$$\phi = \sin^{-1}\left(\pm \frac{b_2}{\sqrt{2 - b_2^2}}\right)$$

$$= \sin^{-1}\left(\pm \frac{1/\sqrt{2}}{\sqrt{2 - 1/2}}\right)$$

$$= \sin^{-1}\left(\pm 1/\sqrt{3}\right)$$

$\therefore$ possible values of θ are $\pm \pi/6$

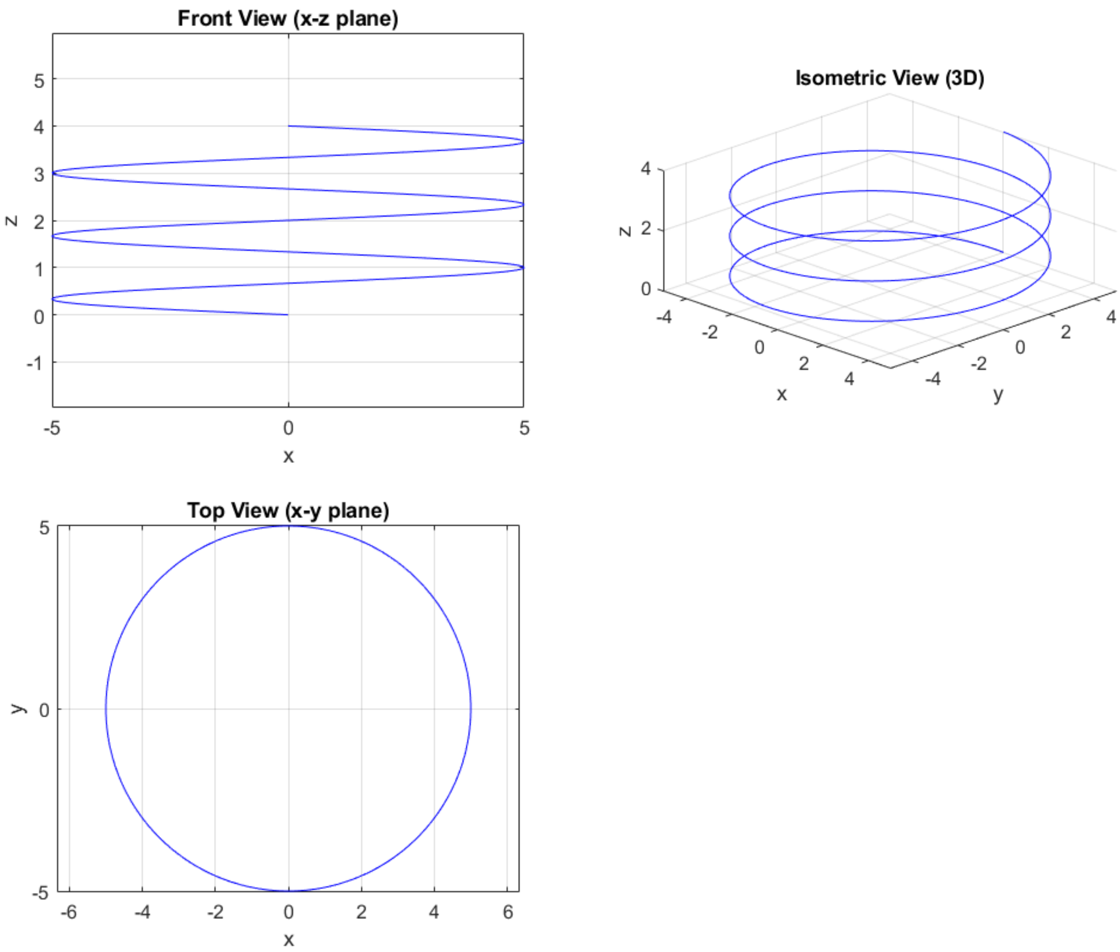
& ϕ are $\sin^{-1}(\pm 1/\sqrt{3})$

Question 3

Using a parametric curve representation approach, write a Matlab code to plot the following curve based on the different views of the curve given.

Ans:

```
% ME21BTECH11001
% Abhishek Ghosh


t = linspace(0, 6*pi, 1000);  % parameter t
r = 5;  % radius
c = (4/3)/(2*pi); % pitch = 4/3

x = -r * sin(t);
y = r * cos(t);
z = c * t;

% Front view (x-z plane)
figure;
plot(x, z, 'b');
title('Front View (x-z plane)');
xlabel('x');
ylabel('z');
xlim([-5, 5]);
ylim([0, max(z)]);
axis equal;
grid on;
set(gca, 'FontSize', 12);
set(gcf, 'Color', 'w');
set(gcf, 'Position', [100, 100, 500, 400]);

% Isometric view (3D plot)
figure;
plot3(x, y, z, 'b');
title('Isometric View (3D)');
xlabel('x');
ylabel('y');
zlabel('z');
xlim([-5, 5]);
ylim([-5, 5]);
zlim([0, max(z)]);
axis equal;
grid on;
set(gca, 'FontSize', 12);
set(gcf, 'Color', 'w');
set(gcf, 'Position', [650, 100, 500, 400]);
view([45, 20]);

% Top view (x-y plane)
figure;
plot(x, y, 'b');
title('Top View (x-y plane)');
xlabel('x');
ylabel('y');
xlim([-5, 5]);
ylim([-5, 5]);
axis equal;
grid on;
set(gca, 'FontSize', 12);
set(gcf, 'Color', 'w');
set(gcf, 'Position', [1200, 100, 500, 400]);
```


Front View (x-z plane)
z
x
Isometric View (3D)
z
x
y
Top View (x-y plane)
y
x