ECE661 Fall 2024: Homework 1

By – Aayush Bhadani

abhadani@purdue.edu

Problem 1:

Origin which is $(0,0)$ in the physical space $\mathbb{R}^2$ can be represented as $(0,0,1) \in \mathbb{R}^3$ in homogeneous coordinates. Multiplying this 3D representation by any $k \in \mathbb{R}\, and\, k \neq 0$ will give $k \cdot (0,0,1)$ that represents the same physical point in $\mathbb{R}^2$ as that corresponding to $(0,0,1)$. Thus, origin $(0,0) \in \mathbb{R}^2$ can be represented by $k \cdot (0,0,1) \in \mathbb{R}^3\ where\ \ k \in \mathbb{R}\, and\, k \neq 0$.

Problem 2:

No, not all points at infinity in the physical plane $\mathbb{R}^2$ are the same. These points at infinity in $\mathbb{R}^2$ can be represented as $(u, v, 0) \in \mathbb{R}^3$ in homogeneous coordinates (HC). If we consider a general point in HC i.e $(u, v, w) \in \mathbb{R}^3$ where $w \to 0$, we find that the corresponding point in physical space $\mathbb{R}^2$ which is $(x, y) = \left(\frac{u}{w}, \frac{v}{w}\right)$ will approach infinity. This approach to infinity is controlled by $u, v \in \mathbb{R}$ and will be along a specific direction based on their signs.

Problem 3:

A degenerate conic can be represented as $C = lm^T + ml^T$

where $l = \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix}$ and $m = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix}$

If we consider the first term in the degenerate conic equation i.e $lm^T = \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix} \cdot (m_1 m_2 m_3)$ we find that this matrix has only one linearly independent column vector as other columns are just scaled versions of the first column. So, the rank of this matrix will be 1.

Similarly, for the second term of the degenerate conic equation i.e $ml^T = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} \cdot (l_1 l_2 l_3)$ we will get rank 1. Since both the terms of the degenerate conic equation have rank 1, the degenerate conic can at most have rank 2.

Problem 4:

A line in $\mathbb{R}^2$ has the general form $ax + by + c = 0$ and can be defined by two points.

A conic is given by the equation: $C = ax^2 + bxy + cy^2 + dx + ey + f = 0$ and will require 5 points in general position to be defined.

We can rewrite this conic equation as $C = x^2 + \left(\frac{b}{a}\right)xy + \left(\frac{c}{a}\right)y^2 + \left(\frac{d}{a}\right)x + \left(\frac{e}{a}\right)y + \left(\frac{f}{a}\right) = 0$

If $B' = \left(\frac{b}{a}\right), C' = \left(\frac{c}{a}\right), D' = \left(\frac{d}{a}\right), E' = \left(\frac{e}{a}\right), F' = \left(\frac{f}{a}\right)$

Conic equation $C = x^2 + B'xy + C'y^2 + D'x + E'y + F' = 0$

Here, we have 5 unknowns: B', C', D', E', F' and would thus need 5 points (resulting 5 equations) to get these conic coefficients. We can also think of counting the coefficients of $x, y\ terms$ in the original conic equation to arrive at the same result (i.e 5 – with a, b, c, d, e as coefficients of $x, y\ terms$).

When 3 points are collinear, we get a degenerate conic (reducible as it contains a line).

Problem 5:

Step 1: Line $l_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} -4 \\ 3 \\ 0 \end{bmatrix}$ (HC representation)

Step 2: Line $l_2 = \begin{bmatrix} -1 \\ 4 \\ 1 \end{bmatrix} \times \begin{bmatrix} 3/2 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 5/2 \\ -5 \end{bmatrix}$ (HC representation)

Step 3: Intersection point of $l_1 \& l_2 = l_1 \times l_2$

$$= \begin{bmatrix} -4 \\ 3 \\ 0 \end{bmatrix} \times \begin{bmatrix} 5 \\ 5/2 \\ -5 \end{bmatrix} = \begin{bmatrix} -15 \\ -20 \\ -25 \end{bmatrix}$$ (HC representation)

In the physical space, this point will be $(x, y) = (\frac{3}{5}, \frac{4}{5})$.

Now, if we consider a line l_1' that passes through $(-1,2)\ and\ (1,-2)$:

$$l_1' = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ 0 \end{bmatrix} \to \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}$$ (HC representation)

Comparing l_1' with l_2 where $l_2 = \begin{bmatrix} 5 \\ 5/2 \\ -5 \end{bmatrix} \to \begin{bmatrix} 2 \\ 1 \\ -2 \end{bmatrix}$, we find that l_1' and l_2 are parallel and thus intersect at infinity. This can be verified by finding the intersection point of l_1' and l_2 which is given by $x_1' = l_1' \times \, l_2 = \begin{bmatrix}4\\2\\0\end{bmatrix} \times \begin{bmatrix}5\\5/2\\-5\end{bmatrix} = \begin{bmatrix}-10\\20\\0\end{bmatrix}$. This point x_1' corresponds to infinity in the physical space.

Thus, in this case, it will take us only 2 steps. First is finding $l_1'$and second is comparing l_1' with l_2.

Problem 6:

It is given that the conic is an ellipse centered at $(1{,}4)$ with semi-minor axis as 1 and semi-major axis as 2. With minor axis parallel to x-axis and major axis parallel to y-axis we have a vertically oriented ellipse. Using the given information we can write the equation of this ellipse as:

$$\frac{(x-1)^2}{1^2} + \frac{(y-4)^2}{2^2} = 1$$

We can re-write this equation as:

$$4x^2 + y^2 - 8x - 8y + 16 = 0$$

Using this we get the conic matrix $C = \begin{bmatrix}4 & 0 & -4\\0 & 1 & -4\\-4 & -4 & 16\end{bmatrix}$

Point p which is origin is represented in HC as: $\begin{bmatrix}0\\0\\1\end{bmatrix}$

The polar line is $l = Cp = \begin{bmatrix}4 & 0 & -4\\0 & 1 & -4\\-4 & -4 & 16\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix} = \begin{bmatrix}-4\\-4\\16\end{bmatrix} \rightarrow \begin{bmatrix}-1\\-1\\4\end{bmatrix}$

Before we find the x-y intercept of the polar line, we need to get the axis lines. We get the axis line using origin and another point on the axis as given below:

x-axis line = $\begin{bmatrix}0\\0\\1\end{bmatrix} \times \begin{bmatrix}1\\0\\1\end{bmatrix} = \begin{bmatrix}0\\1\\0\end{bmatrix}$

y-axis line = $\begin{bmatrix}0\\0\\1\end{bmatrix} \times \begin{bmatrix}0\\1\\1\end{bmatrix} = \begin{bmatrix}-1\\0\\0\end{bmatrix}$

Intersection points of the polar line with x-axis and y-axis are given as:

x-intercept = $\begin{bmatrix}-1\\-1\\4\end{bmatrix} \times \begin{bmatrix}0\\1\\0\end{bmatrix} = \begin{bmatrix}-4\\0\\-1\end{bmatrix}$

y-intercept = $\begin{bmatrix}-1\\-1\\4\end{bmatrix} \times \begin{bmatrix}-1\\0\\0\end{bmatrix} = \begin{bmatrix}0\\-4\\-1\end{bmatrix}$

In the physical space we represent x-intercept as $(4,0)$ and y-intercept as $(4,0)$.

Problem 7:

Coordinates of the enemy triangle vertices are given in physical space as:

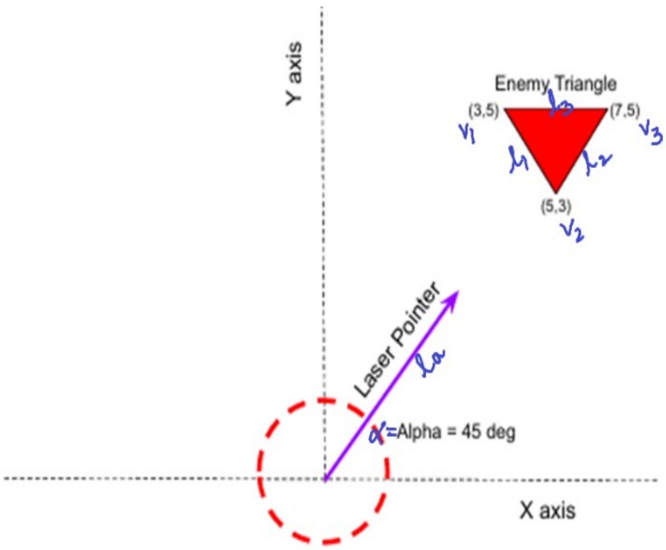
$$v_1 = (3,5), v_2 = (5,3), v_3 = (7,5)$$

Let the sides of the triangle be part of lines l_1, l_2, l_3 which are represented in HC as:

$$l_1 = \begin{bmatrix}3\\5\\1\end{bmatrix} \times \begin{bmatrix}5\\3\\1\end{bmatrix} = \begin{bmatrix}2\\2\\-16\end{bmatrix}$$

$$l_2 = \begin{bmatrix}5\\3\\1\end{bmatrix} \times \begin{bmatrix}7\\5\\1\end{bmatrix} = \begin{bmatrix}-2\\2\\4\end{bmatrix}$$

$$l_3 = \begin{bmatrix}7\\5\\1\end{bmatrix} \times \begin{bmatrix}3\\5\\1\end{bmatrix} = \begin{bmatrix}0\\-4\\20\end{bmatrix}$$



Lasor pointer line l_a is given as $y = \tan(\alpha)\ x$

In HC this pointer line is given as $l_a = \begin{bmatrix}\tan(\alpha)\\-1\\0\end{bmatrix} = \begin{bmatrix}1\\-1\\0\end{bmatrix}$ $(\alpha = 45\text{deg})$

Now we find the points of intersection on lines l_1, l_2, l_3 with pointer line l_a below:

$$x_1 = l_a \times l_1 = \begin{bmatrix}1\\-1\\0\end{bmatrix} \times \begin{bmatrix}2\\2\\-16\end{bmatrix} = \begin{bmatrix}16\\16\\4\end{bmatrix} \to \begin{bmatrix}4\\4\\1\end{bmatrix}$$

$$x_2 = l_a \times l_2 = \begin{bmatrix}1\\-1\\0\end{bmatrix} \times \begin{bmatrix}-2\\2\\4\end{bmatrix} = \begin{bmatrix}-4\\4\\0\end{bmatrix}$$

$$x_3 = l_a \times l_3 = \begin{bmatrix}1\\-1\\0\end{bmatrix} \times \begin{bmatrix}0\\-4\\20\end{bmatrix} = \begin{bmatrix}-20\\-20\\-4\end{bmatrix} \to \begin{bmatrix}5\\5\\1\end{bmatrix}$$

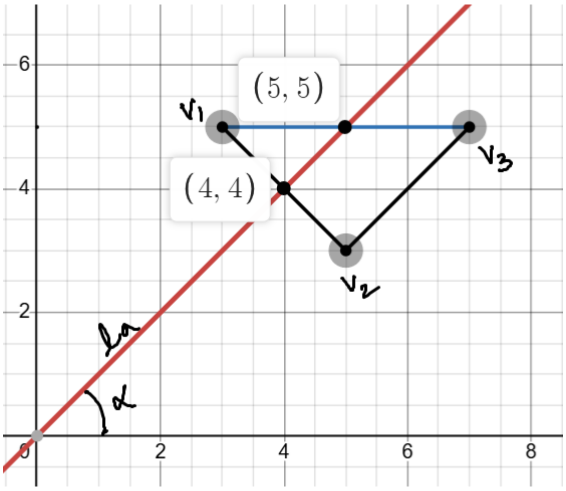
When we represent these intersection points in physical space, we get x_1 as $(4,4)$, x_2 at infinity (no intersection of $l_a\ with\ l_2$) and x_3 as $(5,5)$.

Finally, to check if the pointer aim hits the enemy triangle correctly we need to make sure that the intersection point lie on the side of the triangle. If the intersection point lies between the vertices defining the line-segment (side) of the triangle then it will be a successful hit.

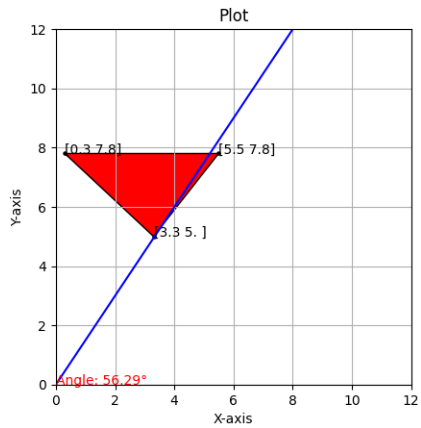
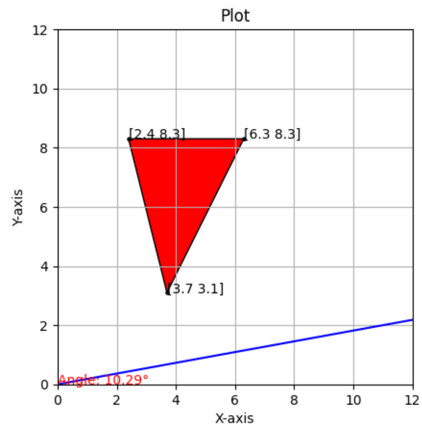
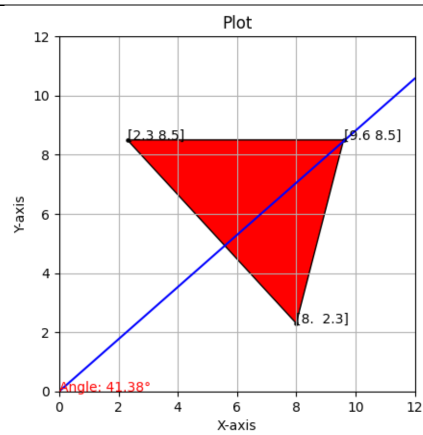
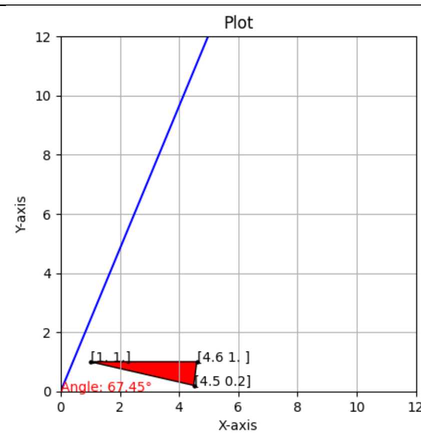
Point $x_1 = (4,4)$ lie on l_1 and is between vertices $v_1 = (3,5)$ and $v_2 = (5,3)$

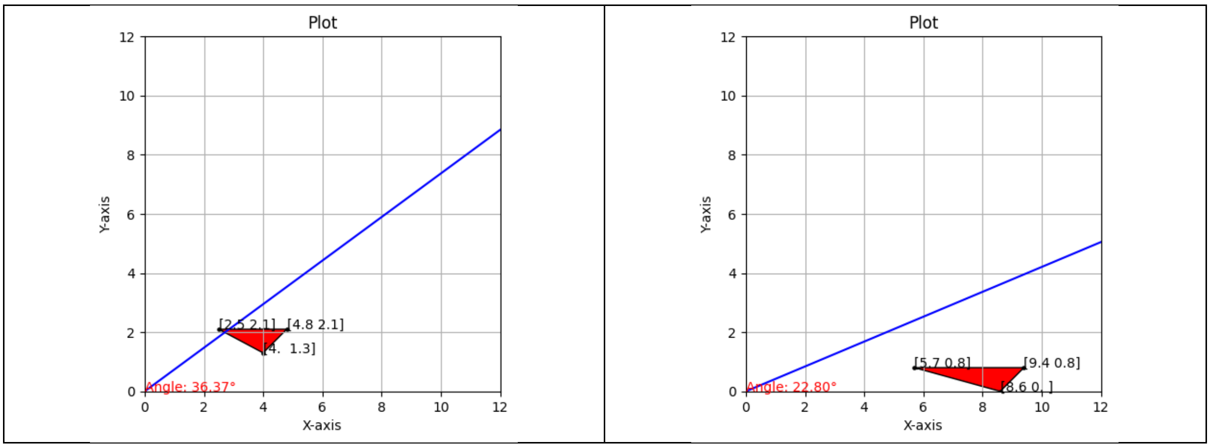
Point $x_3 = (5,5)$ lie on l_3 and is between vertices $v_1 = (3,5)$ and $v_3 = (7,5)$

As we have an intersection of pointer line and line-segment of the triangle lying on enemy triangle, we have a successful hit and thus user's aim is correct.



Extra:

<table>
<tr><th>HIT</th><th>MISS</th></tr>
<tr>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>
</td>
<td>
</td>
</tr>
</table>



Code:

```
import numpy as np
import matplotlib.pyplot as plt
import random
```

```
x_limit = 10
y_limit = 10

# x1,y1 = 3,5
# x2,y2 = 5,3
# x3,y3 = 7,5

x1,y1 = round(random.uniform(0,x_limit),1),round(random.uniform(0,y_limit),1)
x2,y2 = round(random.uniform(0,x_limit),1),y1#random.uniform(0,y_limit)
x3,y3 = round(random.uniform(min(x1,x2),max(x1,x2)),1),round(random.uniform(0,min(y1,y2)),1)

v1 = (x1,y1)
v2 = (x2,y2)
v3 = (x3,y3)

vertices = [v1,v2,v3]

#alpha = np.pi/4 #45deg
alpha = random.uniform(0,np.pi/2)
print('angle_rad:',alpha)
#print('angle_deg:',np.degrees(alpha))

pointerHC = np.array([np.tan(alpha),-1,0])

verticesHC = []
for v in vertices:
    verticesHC.append(np.array(v+(1,)))
N_pts = len(verticesHC)
```

```
linesHC = []
for i in range(N_pts):
  pts = [verticesHC[i],verticesHC[(i+1)%N_pts]]
  linesHC.append(np.cross(pts[0],pts[1]))

intersect_ptsHC = []
for l in linesHC:
  intersect_ptsHC.append(np.cross(pointerHC,l))
```

```
hit_check = False
for i in range(N_pts):
  end_pts = [verticesHC[i],verticesHC[(i+1)%N_pts]]
  intersect_pt = intersect_ptsHC[i]
  # print(end_pts)
  # print(intersect_pt)
  x_low,x_high = min(end_pts[0][0],end_pts[1][0]), max(end_pts[0][0],end_pts[1][0])
  y_low,y_high = min(end_pts[0][1],end_pts[1][1]), max(end_pts[0][1],end_pts[1][1])
  if x_low<=intersect_pt[0]/intersect_pt[2]<=x_high and y_low<=intersect_pt[1]/intersect_pt[2]<=y_high:
    hit_check = True
    break

if hit_check == True:
  print('TARGET-','SUCESSFUL HIT')
else:
  print('TARGET-','MISS')
```

```
vertices_arr = np.array(vertices)

fig, ax = plt.subplots()

# Plot the triangle
triangle = plt.Polygon(vertices_arr, closed=True, edgecolor='black', facecolor='red')
ax.add_patch(triangle)
plt.scatter(vertices_arr[:,0],vertices_arr[:,1],s=5,c='black')
for i in range(len(vertices_arr)):
    plt.annotate(vertices_arr[i], vertices_arr[i])

#Plot pointer line
x_pointer = np.linspace(0, x_limit+2, 100)
y_pointer = (-pointerHC[0] * x_pointer - pointerHC[2])/pointerHC[1]
alpha_deg = np.degrees(alpha)
plt.plot(x_pointer, y_pointer, color='blue')
plt.text(0, 0, f'Angle: {alpha_deg:.2f}°', color='red')

ax.set_xlim(0, x_limit+2)
ax.set_ylim(0, x_limit+2)
ax.set_aspect('equal')

plt.grid(True)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Plot')

plt.show()
```

For this exercise, a random triangle and random aiming angle is selected in a plane constraint between 0-10 along the axes. Triangle is pointed downwards as asked in the question.