

# Mat-Geo Problem Solution Series

---

Aditya Tripathy  
Dept. of Electrical Engg.,  
IIT Hyderabad.  
November 5, 2024

# Outline

---

# Table Of Contents

Problem

Solution

Matrix Representation

Calculation

Graph

C - Code

Helper Functions

C - Code

Triangle Generation

# Problem

---

## Problem Statement

Construct an equilateral triangle  $ABC$  with each side 5cm.

## Solution

---

# Matrix Representation

Let  $\mathbf{A} = \mathbf{0}$ , and  $\mathbf{B} = \begin{pmatrix} 5 \\ 0 \end{pmatrix}$ .  $\mathbf{C}$  is the required vertex.

Since the triangle is equilateral,

$$\mathbf{C} = R(\mathbf{B} - \mathbf{A}) \quad (3.1)$$

$$R = \begin{pmatrix} \cos \frac{\pi}{3} & -\sin \frac{\pi}{3} \\ \sin \frac{\pi}{3} & \cos \frac{\pi}{3} \end{pmatrix} \quad (3.2)$$

where  $R$  is the rotation matrix which rotates the vector  $\mathbf{B} - \mathbf{A}$  by angle  $\frac{\pi}{3}$ .

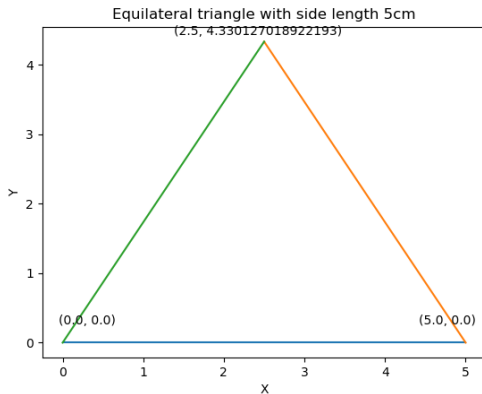
On calculation,

$$\mathbf{C} = \begin{pmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 5 \\ 0 \end{pmatrix} \quad (3.3)$$

$$\mathbf{C} = \begin{pmatrix} \frac{5}{2} \\ \frac{5\sqrt{3}}{2} \end{pmatrix} \quad (3.4)$$



# Graph



**Figure 1:** equilateral triangle of side 5cm

# Required Structures

## Listing 1: C example

```
struct point {  
    double x;  
    double y;  
};  
  
struct triangle {  
    double length;  
    struct point vertex1;  
    struct point vertex2;  
    struct point vertex3;  
    int num_points;  
    struct point **sides;  
};
```

## Listing 2: C example

```
struct point *get_vertex(struct point point1, struct point point2)
{
    struct point *vertex = malloc(sizeof(struct point));
    double vec_x = point2.x - point1.x;
    double vec_y = point2.y - point1.y;

    vertex->x = point1.x + (vec_x / 2) - (sqrt(3) * vec_y / 2);
    vertex->y = point1.y + (sqrt(3) * vec_x / 2) + (vec_y / 2);

    return vertex;
}
```

# Helper Functions

## Listing 3: C example

```
struct point *generate_line_points(struct point point1, struct
    point point2, int num_points) {
    double dx = point2.x - point1.x;
    double dy = point2.y - point1.y;
    struct point *sides = (struct point*) malloc(num_points *
        sizeof(struct point));
    for (int i = 0; i < num_points; i++) {
        double t = (double)i / (num_points - 1);
        sides[i].x = point1.x + t * dx;
        sides[i].y = point1.y + t * dy;
    }
    return sides;
}
```

# Triangle Generation

## Listing 4: C example

```
void generate_triangle(struct triangle *to_draw) {  
    to_draw->sides = (struct point**) malloc(3 * sizeof(struct  
        point*));  
    to_draw->sides[0] = generate_line_points(to_draw->vertex1,  
        to_draw->vertex2, to_draw->num_points);  
    to_draw->vertex3 = *get_vertex(to_draw->vertex1, to_draw  
        ->vertex2);  
    to_draw->sides[1] = generate_line_points(to_draw->vertex2,  
        to_draw->vertex3, to_draw->num_points);  
    to_draw->sides[2] = generate_line_points(to_draw->vertex1,  
        to_draw->vertex3, to_draw->num_points);  
}
```