Matgeo Presentation

G. Abhimanyu Koushik
EE24BTECH11024

November 20, 2024

Problem Statement

Construct a Triangle ABC such that $\angle B = 60^\circ$, $\angle C = 45^\circ$ and $AB + BC + CA = 12cm$.

Input Parameters

Symbol	Description
a	length of side BC
b	length of side CA
c	length of side AB
$\angle A$	angle at vertex A
$\angle B$	angle at vertex B
$\angle C$	angle at vertex C
K	Perimeter of triangle

Linear Equation

From properties of triangle we get the following equations

$$a + b + c = K \tag{3.1}$$

$$b\cos(C) + c\cos(B) = a \tag{3.2}$$

$$\frac{b}{\sin(B)} = \frac{c}{\sin(C)} \tag{3.3}$$

Matrix Equation

The linear equations can be expxressed as

$$a + b + c = K \tag{3.4}$$

$$-a + b\cos(C) + c\cos(B) = 0 \tag{3.5}$$

$$b\sin(C) - c\sin(B) = 0 \tag{3.6}$$

resulting in the matrix equation

$$\frac{1}{K}\begin{pmatrix} 1 & 1 & 1 \\ -1 & \cos C & \cos B \\ 0 & \sin C & -\sin B \end{pmatrix} \times \mathbf{x} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{3.7}$$

where

$$\mathbf{x} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \tag{3.8}$$

Row Reduction

After substituting the values, (3.7) can be row reduced as follows

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & \frac{1}{\sqrt{2}} & \frac{1}{2} & 0 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{\sqrt{3}}{2} & 0 \end{pmatrix} \xleftrightarrow{R_2 \leftarrow R_1 + R_2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & \frac{1}{\sqrt{2}} + 1 & \frac{3}{2} & 1 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{\sqrt{3}}{2} & 0 \end{pmatrix} \tag{3.9}$$

$$\xleftrightarrow{R_3 \leftarrow R_2 - \left(\sqrt{2}+1\right) R_3} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & \frac{1}{\sqrt{2}} + 1 & \frac{3}{2} & 1 \\ 0 & 0 & \left(\sqrt{3} + \sqrt{2} + 1\right) \frac{\sqrt{3}}{2} & 1 \end{pmatrix} \tag{3.10}$$

$$\xleftrightarrow{R_3 \leftarrow \left(\frac{2}{\sqrt{3}\left(\sqrt{3}+\sqrt{2}+1\right)}\right) R_3} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & \frac{1}{\sqrt{2}} + 1 & \frac{3}{2} & 1 \\ 0 & 0 & 1 & \frac{2}{\sqrt{3}\left(\sqrt{3}+\sqrt{2}+1\right)} \end{pmatrix} \tag{3.11}$$

$$\xleftrightarrow{R_2 \leftarrow R_2 - \left(\frac{3}{2}\right) R_3} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & \frac{1}{\sqrt{2}} + 1 & 0 & \frac{\sqrt{2}+1}{\sqrt{3}+\sqrt{2}+1} \\ 0 & 0 & 1 & \frac{2}{\sqrt{3}\left(\sqrt{3}+\sqrt{2}+1\right)} \end{pmatrix} \tag{3.12}$$

$$\xleftrightarrow{R_2 \leftarrow \left(\frac{\sqrt{2}}{\sqrt{2}+1}\right) R_2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & \frac{\sqrt{2}}{\sqrt{3}+\sqrt{2}+1} \\ 0 & 0 & 1 & \frac{2}{\sqrt{3}\left(\sqrt{3}+\sqrt{2}+1\right)} \end{pmatrix} \tag{3.13}$$

$$\xleftrightarrow{R_1 \leftarrow R_1 - R_2} \begin{pmatrix} 1 & 0 & 1 & \frac{\sqrt{3}+1}{\sqrt{3}+\sqrt{2}+1} \\ 0 & 1 & 0 & \frac{\sqrt{2}}{\sqrt{3}+\sqrt{2}+1} \\ 0 & 0 & 1 & \frac{2}{\sqrt{3}\left(\sqrt{3}+\sqrt{2}+1\right)} \end{pmatrix} \tag{3.14}$$

$$\xleftrightarrow{R_1 \leftarrow R_1 - R_3} \begin{pmatrix} 1 & 0 & 0 & \frac{1+\sqrt{3}}{\sqrt{3}\left(\sqrt{3}+\sqrt{2}+1\right)} \\ 0 & 1 & 0 & \frac{1+\sqrt{3}}{\sqrt{3}\left(\sqrt{3}+\sqrt{2}+1\right)} \\ 0 & 0 & 1 & \frac{2}{\sqrt{3}\left(\sqrt{3}+\sqrt{2}+1\right)} \end{pmatrix} \tag{3.15}$$

Final Answer

Thus,

$$\left(\frac{1}{K}\right)\mathbf{x} = \begin{pmatrix} \frac{1+\sqrt{3}}{\sqrt{3}\left(\sqrt{3}+\sqrt{2}+1\right)} \\ \frac{1+\sqrt{3}}{\sqrt{3}\left(\sqrt{3}+\sqrt{2}+1\right)} \\ \frac{2}{\sqrt{3}\left(\sqrt{3}+\sqrt{2}+1\right)} \end{pmatrix} \tag{3.16}$$

$$\implies a = \frac{12+12\sqrt{3}}{\sqrt{3}\left(\sqrt{3}+\sqrt{2}+1\right)} \tag{3.17}$$

$$b = \frac{12\sqrt{2}}{\left(\sqrt{3}+\sqrt{2}+1\right)} \tag{3.18}$$

$$c = \frac{24}{\sqrt{3}\left(\sqrt{3}+\sqrt{2}+1\right)} \tag{3.19}$$

The codes below verifies (3.16).

Plot of the triangle

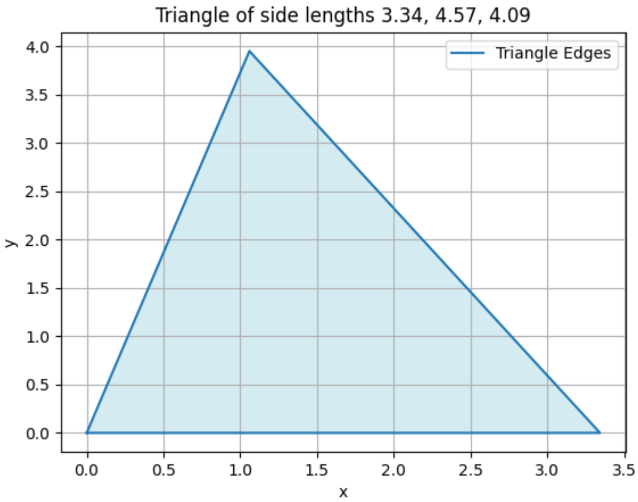

Figure: Triangle Satisfying given conditions

C Code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <unistd.h>
#include "libs/matfun.h"
#include "libs/geofun.h"

void point_gen(FILE *fptr, double **A, double **B, int no_rows, int
    no_cols, int num_points) {
    for (double i = 0; i <= num_points; i++) {
        double **output = Matadd(A, Matscale(Matsub(B,A,no_rows,
            no_cols),no_rows,no_cols,(double)i/num_points), no_rows,
            no_cols);
        fprintf(fptr, "%lf,%lf\n", output[0][0], output[1][0]);
        freeMat(output,no_rows);
```

```
        }
}

double** sidelength_vector_gen_2anglesperimeter(double angleB, double
    angleC, double perimeter) {

    // Solving the matrix equation in form of Ax=b with x=inv(A)b
    double **coeff = createMat(3, 3);
    double **b = createMat(3, 1);
    double **lengths = createMat(3, 1);
    double **sidematrix = createMat(3, 1);

    //Assigning the values
    coeff[0][0] = 1;
    coeff[1][0] = 1;
    coeff[2][0] = 1;
    coeff[0][1] = -1;
    coeff[1][1] = cos(angleC);
```

```
    coeff[2][1] = cos(angleB);
    coeff[0][2] = 0;
    coeff[1][2] = sin(angleC);
    coeff[2][2] = -sin(angleB);
    b[0][0] = 1;
    b[1][0] = 0;
    b[2][0] = 0;
    sidematrix = Matscale(b, 3, 1, perimeter);

    //Solving the equation and getting side lengths of the triangle
    lengths = Matmul(Matinv(coeff, 3), sidematrix, 3, 3, 1);

    // Free allocated memory
    freeMat(b, 3);
    freeMat(sidematrix, 3);

    return lengths;
}
```

```
void twoDtriangle_gen(double sideAB, double sideBC, double sideCA,
    char filename[]) {
    double xA, yA, xB, yB, xC, yC;

    // Correct formula for angle A
    double angleA = acos(((sideAB * sideAB) + (sideCA * sideCA) - (
        sideBC * sideBC)) / (2 * sideAB * sideCA));

    xA = 0; yA = 0;
    xB = sideAB; yB = 0;
    xC = cos(angleA) * sideCA; yC = sin(angleA) * sideCA;

    int m = 2, n = 1;

    // Open the file for writing
    FILE *fptr = fopen(filename, "w");
    if (fptr == NULL) {
        printf("Error opening file!\n");
```

```c
        return; // Return early if file cannot be opened
    }
    // Create matrices for vertices
    double **A = createMat(m, n);
    double **B = createMat(m, n);
    double **C = createMat(m, n);

    A[0][0] = xA;
    A[1][0] = yA;
    B[0][0] = xB;
    B[1][0] = yB;
    C[0][0] = xC;
    C[1][0] = yC;

    // Generate points along the triangle's edges
    point_gen(fptr, A, B, m, n, 20);
    point_gen(fptr, B, C, m, n, 20);
    point_gen(fptr, C, A, m, n, 20);
```

```
    freeMat(A, m);
    freeMat(B, m);
    freeMat(C, m);

    // Close the file
    fclose(fptr);
}

int main() {
    double sideAB, sideBC, sideCA;
    double **length;

    length = sidelength_vector_gen_2anglesperimeter(M_PI/3, M_PI/4,
        12);
    sideBC = length[0][0];
    sideCA = length[1][0];
    sideAB = length[2][0];
```

```
    twoDtriangle_gen(sideAB, sideBC, sideCA, "triangle.dat");

    return 0;
}
```

Python Code for Plotting

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

# Load the points from the text file
def plot_triangle(filename, file_no=""):
    points = np.loadtxt(filename, delimiter=',')

    # Extract the x and y coordinates
    x = points[:, 0]
    y = points[:, 1]

    # Plot the triangle
    plt.figure()
    plt.plot(x, y, label='Triangle Edges')
    plt.fill(x, y, 'lightblue', alpha=0.5)
    plt.xlabel("x")
```

```
    plt.ylabel("y")
    plt.title("Triangle of side lengths " + str(round(((x[20] - x[0])**2
        + (y[20] - y[0])**2)**0.5, 2)) + ", " +
            str(round(((x[41] - x[21])**2 + (y[41] - y[21])**2)**0.5,
                2)) + ", " +
            str(round(((x[62] - x[42])**2 + (y[62] - y[42])**2)**0.5,
                2)))
    plt.grid(True)
    plt.legend()

    # Save the plot to figs directory
    plt.savefig('../figs/fig' + str(file_no) + '.png')

plot_triangle('triangle.dat')
```