3.2.30

AI25BTECH11014 - Gooty Suhas

September 30, 2025

Question

Construct a triangle $\triangle ABC$ given:

$$\angle B = 105^\circ, \quad \angle C = 90^\circ, \quad AB + BC + CA = 10 \text{ cm}$$

Matrix Formulation

Let the side lengths be:

$$\mathbf{x} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ -1 & \cos C & \cos B \\ 0 & \sin C & -\sin B \end{bmatrix} \mathbf{x} = \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix}$$

Numerical Substitution

Substitute:

$$\cos C = 0, \quad \sin C = 1$$

$$\cos B \approx -0.2588, \quad \sin B \approx 0.9659$$

$$\begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & -0.2588 \\ 0 & 1 & -0.9659 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix}$$

Matrix Solution

Solving the system:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} -1.52 \\ 5.66 \\ 5.86 \end{bmatrix}$$

Conclusion

Since side $a \approx -1.52$ is negative, the triangle is not physically constructible.

Construction is not possible.

Python Code (Part 1)

```
import numpy as np

A = np.array([
    [1, 1, 1],
    [−1, 0, −0.2588],
    [0, 1, −0.9659]
], dtype=np.float32)

b = np.array([10, 0, 0], dtype=np.float32)

x = np.linalg.solve(A, b)
print("Solution␣[a␣b␣c]:", x)
```

C Code for .so File

```
#include <stdio.h>

void solve_triangle(float* A,
                    float* b,
                    float* x) {

  float invA[9] = {
    -0.152, 0.332, 0.820,
     0.566, 0.566, -0.132,
     0.586, 0.102, -0.688
  };
```

C Code Continued

```
    for (int i = 0; i < 3; i++) {
        x[i] = 0;
        for (int j = 0; j < 3; j++) {
            x[i] += invA[3*i + j] * b[j];
        }
    }
}
```

Python Code Using .so File

```
import ctypes
import numpy as np

lib = ctypes.CDLL('./libtriangle.so')
lib.solve_triangle.argtypes = [
    ctypes.POINTER(ctypes.c_float),
    ctypes.POINTER(ctypes.c_float),
    ctypes.POINTER(ctypes.c_float)
]
```

Python Code Continued

```
A = np.array([
    [1, 1, 1],
    [−1, 0, −0.2588],
    [0, 1, −0.9659]
], dtype=np.float32)

b = np.array([10, 0, 0], dtype=np.float32)
x = np.zeros(3, dtype=np.float32)

lib.solve_triangle(
    A.ctypes.data_as(ctypes.POINTER(ctypes.c_float)),
    b.ctypes.data_as(ctypes.POINTER(ctypes.c_float)),
    x.ctypes.data_as(ctypes.POINTER(ctypes.c_float))
)

print("Solution␣[a␣b␣c]:", x)
```