3.2.29

Gautham-AI25BTECH11013

September 30, 2025

Question

Construct a $\triangle ABC$ given

$$a = BC = 6 \text{ cm}, \qquad \angle B = 30^\circ, \qquad AC - AB = 4 \text{ cm}. \tag{1}$$

$$\tag{2}$$

Theoretical Solution

In the usual notation $a = BC,\ b = CA,\ c = AB$. From the cosine formula in $\triangle ABC$

$$b^2 = a^2 + c^2 - 2ac\cos B. \tag{3}$$

Put $b = c + k$ where $k = 4$.

$$(c+k)^2 = a^2 + c^2 - 2ac\cos B. \tag{4}$$

Canceling c^2 and collecting terms in c:

$$2kc + k^2 = a^2 - 2ac\cos B \quad \Longrightarrow \quad c\big(2k + 2a\cos B\big) = a^2 - k^2. \tag{5}$$

Hence the general expression for c when $b - c = k$ is

$$\boxed{c = \frac{a^2 - k^2}{2(k + a\cos B)}}. \tag{6}$$

Theoretical solution

Now substitute $a = 6,\ B = 30^\circ,\ k = 4$:

$$\cos 30^\circ = \frac{\sqrt{3}}{2}, \qquad c = \frac{6^2 - 4^2}{2\left(4 + 6\cos 30^\circ\right)} = \frac{36 - 16}{2\left(4 + 6 \cdot \frac{\sqrt{3}}{2}\right)} = \frac{20}{2\left(4 + 3\sqrt{3}\right)}. \tag{7}$$

Numerically,

$$c \approx 1.09 \text{ cm}, \qquad b = c + 4 \approx 5.09 \text{ cm}. \tag{8}$$

Place $\mathbf{B} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $\mathbf{C} = \begin{pmatrix} a \\ 0 \end{pmatrix} = \begin{pmatrix} 6 \\ 0 \end{pmatrix}$. Point A lies on this ray with $BA = c$, so

$$\mathbf{A} = c \begin{pmatrix} \cos B \\ \sin B \end{pmatrix} \approx (0.94, 0.54). \tag{9}$$

Python Code

```
 import sys
sys.path.insert(0, '/home/gauthamp/Documents/codes/CoordGeo')
import numpy as np
import numpy.linalg as LA
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

#local imports
from line.funcs import *
from triangle.funcs import *
from conics.funcs import circ_gen
theta = 30
#Given points
B = np.array(([0, 0])).reshape(-1,1)
C = np.array(([6, 0])).reshape(-1,1)
A= 1.09*np.array(([np.cos(np.deg2rad(theta)),np.sin(np.deg2rad(
    theta))])).reshape(-1,1)
```

Python Code

```
#Generating all lines
x_AB = line_gen(A,B)
x_BC = line_gen(B,C)
x_CA = line_gen(C,A)
plt.plot(x_AB[0,:],x_AB[1,:],label='$AB$')
plt.plot(x_BC[0,:],x_BC[1,:],label='$BC$')
plt.plot(x_CA[0,:],x_CA[1,:],label='$CA$')
colors = np.arange(1,4)
tri_coords = np.block([[A,B,C]])
plt.scatter(tri_coords[0,:], tri_coords[1,:], c=colors)
vert_labels = ['A','B','C']
for i, txt in enumerate(vert_labels):
    #plt.annotate(txt, # this is the text
    plt.annotate(f'{txt}\n({tri_coords[0,i]:.2f}, {tri_coords[1,i
        ]:.2f})',
                 (tri_coords[0,i], tri_coords[1,i]), textcoords="
                    offset points", xytext=(25,5), ha='center')
```

Python Code

```
ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['bottom'].set_position('zero')
'''
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.spines['bottom'].set_visible(False)
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.legend(loc='best')
plt.grid()
plt.axis('equal')
plt.savefig('/home/gauthamp/ee1030-2025/ai25btech11013/matgeo
    /3.2.29/figs/fig.png')
plt.show()
```

Figure