Geometric Constructions through Python

G V V Sharma*

Contents

***Abstract*—This manual shows how to construct geometric figures using Python. The problems are based on NCERT math textbooks of Class 9 and 10.**

1 Triangle

1.1 Consider $\triangle ABC$ with $BC = a, CA = b$ and $AB = c$. Let

$$\mathbf{A} = \begin{pmatrix} p \\ q \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \mathbf{C} = \begin{pmatrix} a \\ 0 \end{pmatrix} \tag{1}$$

Find the p and q.

Solution: Since

$$\mathbf{p^2 + q^2} = c^2 \tag{2}$$

$$(\mathbf{p - a})^2 + \mathbf{q^2} = b^2, \tag{3}$$

we obtain

$$p = \frac{a^2 + c^2 - b^2}{2a}, q = \sqrt{c^2 - p^2} \tag{4}$$

1.2 Write a program to compute p and q when $a = 8, b = 11$ and $c = 13$.

1.3 Plot $\triangle ABC$ for $a = 8, b = 11$ and $c = 13$.

Solution: The following program plots $\triangle ABC$ in Fig. 1.3

```
#Code by GVV Sharma
#March 26, 2019
#released under GNU GPL
import numpy as np
import matplotlib.pyplot as plt

#if using termux
import subprocess
import shlex
#end if

#Generate line points
def line_gen(A,B):
  len =10
  x_AB = np.zeros((2,len))
  lam_1 = np.linspace(0,1,len)
  for i in range(len):
    temp1 = A + lam_1[i]*(B-A)
    x_AB[:,i]= temp1.T
  return x_AB


#Triangle sides
a = 8
b = 11
c = 13
p = (a**2 + c**2-b**2 )/(2*a)
q = np.sqrt(c**2-p**2)

#Triangle vertices
A = np.array([p,q])
B = np.array([0,0])
C = np.array([a,0])


#Generating all lines
x_AB = line_gen(A,B)
x_BC = line_gen(B,C)
x_CA = line_gen(C,A)

#Plotting all lines
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='$CA$')

plt.plot(A[0], A[1], 'o')
```

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

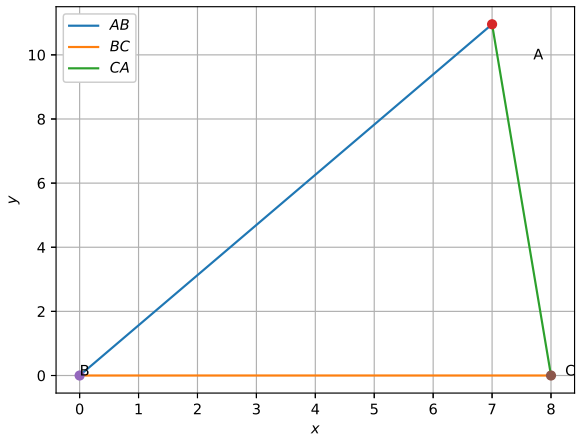

Fig. 1.3

```
plt.text(A[0] * (1 + 0.1), A[1] * (1 - 0.1) , '
    A')
plt.plot(B[0], B[1], 'o')
plt.text(B[0] * (1 - 0.2), B[1] * (1) , 'B')
plt.plot(C[0], C[1], 'o')
plt.text(C[0] * (1 + 0.03), C[1] * (1 - 0.1) ,
    'C')

plt.xlabel('$x$')
plt.ylabel('$y$')
plt.legend(loc='best')
plt.grid() # minor

#if using termux
plt.savefig('../figs/triangle.pdf')
plt.savefig('../figs/triangle.eps')
subprocess.run(shlex.split("termux-open ../
    figs/triangle.pdf"))
#else
#plt.show()
```

1.4 In $\triangle ABC$, $a = 7, \angle B = 75^\circ$ and $b + c = 13$. If

$$a^2 + c^2 - b^2 = 2ac\cos B \tag{5}$$

find b and c and sketch $\triangle ABC$.

1.5 In $\triangle ABC$, $a = 8, \angle B = 45^\circ$ and $c - b = 3.5$. Sketch $\triangle ABC$.

1.6 In $\triangle ABC$, $a = 6, \angle B = 60^\circ$ and $b - c = 2$. Sketch $\triangle ABC$.

1.7 $\triangle ABC$ is right angled at **B**. If $a = 12$ and $b + c = 18$, find a, b, c and draw the triangle.

1.8 In $\triangle ABC$, given that $a + b + c = 11, \angle B = 45^\circ$ and $\angle C = 45^\circ$, find a, b, c.

Solution: We have

$$a = b\cos C + c\cos B \tag{6}$$

$$b\sin C = c\sin B \tag{7}$$

$$a + b + c = 11 \tag{8}$$

Solving the equivalent matrix equation gives the desired answer.

1.9 Draw $\triangle ABC$, given that $a + b + c = 11, \angle B = 30^\circ$ and $\angle C = 90^\circ$, find a, b, c.

2 Circle

2.1 Draw a circle with centre **B** and radius 6. If **C** be a point 10 units away from its centre, construct the pair of tangents AC and CD to the circle.

Solution: From the given information, $AC \perp BC, a = 10$ and $c = 6$.

$$b = \sqrt{a^2 - c^2} \tag{9}$$

The following code draws the circle and tangents in Fig. 2.1

```
#Code by GVV Sharma
#March 26, 2019
#released under GNU GPL
import numpy as np
import matplotlib.pyplot as plt

#if using termux
import subprocess
import shlex
#end if

#Generate line points
def line_gen(A,B):
  len =10
  x_AB = np.zeros((2,len))
  lam_1 = np.linspace(0,1,len)
  for i in range(len):
    temp1 = A + lam_1[i]*(B-A)
    x_AB[:,i]= temp1.T
  return x_AB


#Triangle sides
a = 10
c = 6
b = np.sqrt(a**2-c**2)
```

```
p = (a**2 + c**2-b**2 )/(2*a)
q = np.sqrt(c**2-p**2)

#Triangle vertices
A = np.array([p,q])
B = np.array([0,0])
C = np.array([a,0])
D = np.array([p,-q])


#Generating all lines
x_AB = line_gen(A,B)
x_BC = line_gen(B,C)
x_CA = line_gen(C,A)
x_CD = line_gen(C,D)

#Plotting all lines
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='$CA$')
plt.plot(x_CD[0,:],x_CD[1,:],label='$CD$')

plt.plot(A[0], A[1], 'o')
plt.text(A[0] * (1 + 0.1), A[1] * (1 - 0.1) , '
    A')
plt.plot(B[0], B[1], 'o')
plt.text(B[0] * (1 - 0.2), B[1] * (1) , 'B')
plt.plot(C[0], C[1], 'o')
plt.text(C[0] * (1 + 0.03), C[1] * (1 - 0.1) ,
    'C')
plt.plot(D[0], D[1], 'o')
plt.text(D[0] * (1 - 0.2), D[1] * (1) , 'D')

#Plotting the circle

theta = np.linspace(0,2*np.pi,50)
x = c*np.cos(theta)
y = c*np.sin(theta)

plt.plot(x,y)

plt.xlabel('$x$')
plt.ylabel('$y$')
plt.legend(loc='best')
plt.grid() # minor
plt.axis('equal')
#if using termux
plt.savefig('../figs/circle.pdf')
plt.savefig('../figs/circle.eps')
subprocess.run(shlex.split("termux-open ../
    figs/circle.pdf"))
#else
#plt.show()
```

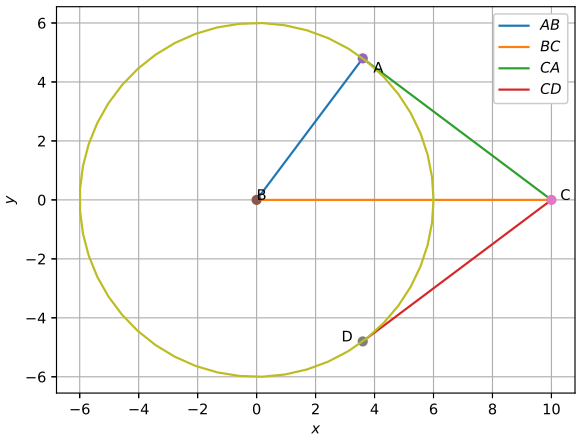

Fig. 2.1

2.2 Construct a tangent to a circle of radius 4 units from a point on the concentric circle of radius 6 units.

2.3 Draw a circle of radius 3 units. Take two points **P** and **Q** on one of its extended diameter each at a distance of 7 units from its centre. Draw tangents to the circle from these two points **P** and **Q**.

2.4 Draw a pair of tangents to a circle of radius 5 units which are inclined to each other at an angle of 60°.

2.5 Draw a line segment AB of length 8 units. Taking **A** as centre, draw a circle of radius 4 units and taking **B** as centre, draw another circle of radius 3 units. Construct tangents to each circle from the centre of the other circle.

2.6 Let ABC be a right triangle in which $a = 8, c = 6$ and $\angle B = 90^{\circ}$. BD is the perpendicular from **B** on AC. The circle through **B, C, D** is drawn. Construct the tangents from **A** to this circle.