Python with Linear Algebra: 2D

G V V Sharma*

Contents

Abstract—**This manual introduces matrix computations using python and the properties of a triangle.**

1 Line

1.1 Let

$$\mathbf{A} = \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \mathbf{C} = \begin{pmatrix} 4 \\ -1 \end{pmatrix}. \tag{1}$$

Draw ΔABC.
Solution: The following code yields the desired plot in Fig. 1.1

```
#Code by GVV Sharma
#January 28, 2019
#released under GNU GPL
import numpy as np
import matplotlib.pyplot as plt
#if using termux
import subprocess
import shlex
#end if

A = np.array([-2,-2])
B = np.array([1,3])
C = np.array([4,-1])

len =10
lam_1 = np.linspace(0,1,len)

x_AB = np.zeros((2,len))
x_BC = np.zeros((2,len))
x_CA = np.zeros((2,len))
for i in range(len):
  temp1 = A + lam_1[i]*(B-A)
  x_AB[:,i]= temp1.T
  temp2 = B + lam_1[i]*(C-B)
  x_BC[:,i]= temp2.T
  temp3 = C + lam_1[i]*(A-C)
  x_CA[:,i]= temp3.T

#print(x_AB[0,:],x_AB[1,:])
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')
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()
```

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

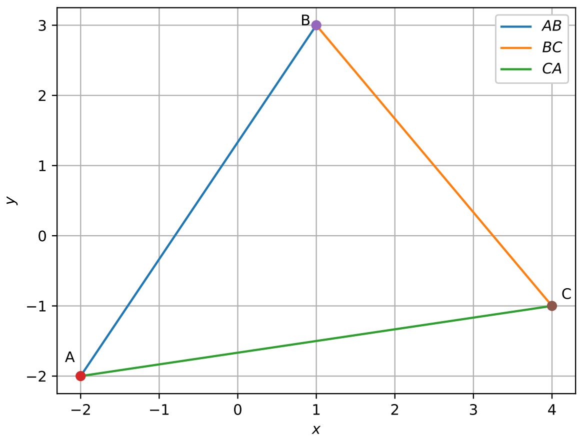

Fig. 1.1

1.2 Find the equation of AB.
Solution: The desired equation is obtained as

$$AB: \quad \mathbf{x} = \mathbf{A} + \lambda_1 (\mathbf{B} - \mathbf{A}) \tag{2}$$

$$= -\begin{pmatrix} 2 \\ 2 \end{pmatrix} + \lambda_1 \begin{pmatrix} 3 \\ 5 \end{pmatrix} \tag{3}$$

1.3 Find the direction vector and the normal vector for AB
Solution: Let

$$T_{AB} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ -2 & 3 \end{pmatrix} \tag{4}$$

The direction vector of AB is

$$\mathbf{m} = \mathbf{B} - \mathbf{A} = T_{AB} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix} \tag{5}$$

The normal vector $\mathbf{n}$ is defined as

$$\mathbf{n}^T \mathbf{m} = 0 \tag{6}$$

$$\implies \mathbf{n} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \mathbf{m} = \begin{pmatrix} 5 \\ -3 \end{pmatrix} \tag{7}$$

1.4 Write a python code for computing the direction and normal vectors. **Solution:** Save the following code as **coeffs.py** and execute.

```
import numpy as np


def dir_vec(AB):
  return np.matmul(AB,dvec)

def norm_vec(AB):
  return np.matmul(omat,np.matmul(AB,dvec
      ))

A = np.array([−2,−2])
B = np.array([1,3])
dvec = np.array([−1,1])
omat = np.array([[0,1],[−1,0]])
AB =np.vstack((A,B)).T

#print (dir_vec(AB))
#print (norm_vec(AB))
```

1.5 Find the equation of the line in terms of the normal vector.
Solution: The desired equation is

$$\mathbf{n}^T (\mathbf{x} - \mathbf{A}) = \mathbf{n}^T (\mathbf{x} - \mathbf{B}) = 0 \tag{8}$$

$$\implies \begin{pmatrix} 5 & -3 \end{pmatrix} \mathbf{x} = -\begin{pmatrix} 5 & -3 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \end{pmatrix} = -4 \tag{9}$$

1.6 Find the equations of BC and CA.

2 Altitudes of a Triangle

2.1 In ΔABC, Let $\mathbf{P}$ be a point on BC such that $AP \perp BC$. Then AP is defined to be an *altitude* of ΔABC.

2.2 Find the equation of AP.
Solution: The normal vector of AP is $\mathbf{B} - \mathbf{C}$. From (8), the equation of AP is

$$(\mathbf{B} - \mathbf{C})^T (\mathbf{x} - \mathbf{A}) = 0 \tag{10}$$

$$\implies \begin{pmatrix} -3 & 4 \end{pmatrix} \mathbf{x} = -\begin{pmatrix} -3 & 4 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \end{pmatrix} = -2 \tag{11}$$

2.3 Find the equation of the altitude BQ.
Solution: The desired equation is

$$(\mathbf{C} - \mathbf{A})^T (\mathbf{x} - \mathbf{B}) = 0 \tag{12}$$

$$\implies \begin{pmatrix} 6 & 1 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 6 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = 9 \tag{13}$$

2.4 Find the equation of the altitude CR.

2.5 Find the point of intersection of AP and BQ.
Solution: (10) and (12) can be stacked together into the matrix equation

$$\begin{pmatrix} -3 & 4 \\ 6 & 1 \end{pmatrix} \mathbf{x} = \begin{pmatrix} -2 \\ 9 \end{pmatrix} \tag{14}$$

The following code computes the point of intersection.

```
#This program calculates the orthocentre of
    Triangle ABC
```

```
import numpy as np
from coeffs import *

A = np.array([−2,−2])
B = np.array([1,3])
C = np.array([4,−1])

AB =np.vstack((A,B)).T
BC =np.vstack((B,C)).T
CA =np.vstack((C,A)).T

p = np.zeros(2)

#AD
n1 = dir_vec(BC)
p[0] = np.matmul(n1,A)
#BQ
n2 = dir_vec(CA)
p[1] = np.matmul(n2,B)

#Intersection
N=np.vstack((n1,n2))
H=np.matmul(np.linalg.inv(N),p)
print(H)
```

2.6 Find the point of intersection of and BQ and CR. Comment.
2.7 Find $\mathbf{P}, \mathbf{Q}, \mathbf{R}$.
2.8 Draw AP, BQ and CR and verify that they meet at a point $\mathbf{H}$.

3 Medians of a Triangle

3.1 Find the coordinates of D, E and F of the mid points of AB, BC and CA respectively for ΔABC.
Solution: The coordinates of the mid points are given by

$$D = \frac{B+C}{2}, E = \frac{C+A}{2}, F = \frac{A+B}{2} \tag{15}$$

The following code computes the values resulting in

$$D = \begin{pmatrix} 2.5 \\ 1 \end{pmatrix}, E = \begin{pmatrix} 1 \\ -1.5 \end{pmatrix}, F = \begin{pmatrix} -0.5 \\ 0.5 \end{pmatrix}, \tag{16}$$

```
#This program calculates the mid point
     between
#any two coordinates
import numpy as np
import matplotlib.pyplot as plt

def mid_pt(B,C):
                    D = (B+C)/2
                    return D

A = np. matrix('−2;−2')
B = np. matrix('1;3')
C = np. matrix('4;−1')

print(mid_pt(B,C))
print(mid_pt(C,A))
print(mid_pt(A,B))
```

3.2 Find the equations of AD, BE and CF. These lines are the *medians* of ΔABC
Solution: Use the code in Problem 1.4.
3.3 Find the point of intersection of AD and CF.
Solution: Let the respective equations be

$$\mathbf{n}_1^T \mathbf{x} = p_1 \text{ and} \tag{17}$$

$$\mathbf{n}_2^T \mathbf{x} = p_2 \tag{18}$$

This can be written as the matrix equation

$$\begin{pmatrix} \mathbf{n}_1^T \\ \mathbf{n}_2^T \end{pmatrix} \mathbf{x} = \mathbf{p} \tag{19}$$

$$\implies N^T \mathbf{x} = \mathbf{p} \tag{20}$$

where

$$N = \begin{pmatrix} \mathbf{n}_1 & \mathbf{n}_2 \end{pmatrix}, \tag{21}$$

The point of intersection is then obtained as

$$\mathbf{x} = \left(N^T\right)^{-1} \mathbf{p} \tag{22}$$

$$= N^{-T} \mathbf{p} \tag{23}$$

The following code yields the point of intersection

$$\mathbf{G} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{24}$$

```
#This program calculates the
#intersection of AD and CF
import numpy as np

def mid_pt(B,C):
  D = (B+C)/2
  return D

def dir_vec(AB):
  return np.matmul(AB,dvec)
def norm_vec(AB):
  return np.matmul(omat,dir_vec(AB))
```

```
def line_intersect(AD,CF):
  n1=norm_vec(AD)
  n2=norm_vec(CF)
  N =np.vstack((n1,n2))
  p = np.zeros(2)
  p[0] = np.matmul(n1,AD[:,0])
  p[1] = np.matmul(n2,CF[:,0])
  return np.matmul(np.linalg.inv(N),p)

A = np.array([−2,−2])
B = np.array([1,3])
C = np.array([4,−1])

D = mid_pt(B,C)
F = mid_pt(A,B)

AD =np.vstack((A,D)).T
CF =np.vstack((C,F)).T

dvec = np.array([−1,1])
omat = np.array([[0,1],[−1,0]])



#print(line_intersect(AD,CF))
```

3.4 Using the code in Problem 3.3, verify that $\mathbf{G}$ is the point of intersection of BE, CF as well as AD, BE. $\mathbf{G}$ is known as the *centroid* of ΔABC.

3.5 Graphically show that the medians of ΔABC meet at the centroid.

3.6 Verify that

$$G = \frac{A + B + C}{3} \tag{25}$$

4 Angle Bisectors of a Triangle

4.1 In ΔABC, let U be a point on BC such that $\angle BAU = \angle CAU$. Then AU is known as the *angle bisector*.

4.2 Find the length of AB, BC and CA
Solution: The length of CA is given by

$$CA = \|\mathbf{C} - \mathbf{A}\| \tag{26}$$

The following code calculates the respective values as

$$AB = 5.83, BC = 5, CA = 6.08 \tag{27}$$

```
#This program calculates the distance
    between
#two points
import numpy as np
import matplotlib.pyplot as plt


A = np.array([−2,−2])
B = np.array([1,3])
C = np.array([4,−1])

print (np.linalg.norm(A−B))
```

4.3 If AU, BV and CW are the angle bisectors, find the coordinates of $\mathbf{U}, \mathbf{V}$ and $\mathbf{W}$.
Solution: Using the section formula,

$$\mathbf{W} = \frac{AW.\mathbf{B} + WB.\mathbf{A}}{AW + WB} = \frac{\frac{AW}{WB}.\mathbf{B} + \mathbf{A}}{\frac{AW}{WB} + 1} \tag{28}$$

$$= \frac{\frac{CA}{BC}.\mathbf{B} + \mathbf{A}}{\frac{CA}{BC} + 1} \tag{29}$$

$$= \frac{CA \times \mathbf{B} + BC \times \mathbf{A}}{BC + CA} \tag{30}$$

$$= \frac{a \times \mathbf{A} + b \times \mathbf{B}}{a + b} \tag{31}$$

where $a = BC, b = CA$, since the angle bisector has the property that

$$\frac{AW}{WB} = \frac{CA}{AB} \tag{32}$$

4.4 Write a program to find $\mathbf{U}, \mathbf{V}, \mathbf{W}$.

4.5 Find the intersection of AU and BV.
Solution: Using the code in Problem 3.3, the desired point of intersection is

$$\mathbf{I} = \begin{pmatrix} 1.15 \\ 0.14 \end{pmatrix} \tag{33}$$

It is easy to verify that even BV and CW meet at the same point. $\mathbf{I}$ is known as the *incentre* of ΔABC.

4.6 Draw AU, BV and CW and verify that they meet at a point $\mathbf{I}$.

4.7 Verify that

$$\mathbf{I} = \frac{BC.\mathbf{A} + CA.\mathbf{B} + AB.\mathbf{C}}{AB + BC + CA} \tag{34}$$

4.8 Let the perpendicular from $\mathbf{I}$ to AB be IX. If the equation of AB is

$$\mathbf{n}^T(\mathbf{x}-\mathbf{A}) = 0 \tag{35}$$

show that

$$IX = \frac{\left|\mathbf{n}^T(\mathbf{I}-\mathbf{A})\right|}{\|\mathbf{n}\|} \tag{36}$$

Verify through a Python script.

4.9 If $IY \perp BC$ and $IZ \perp CA$, verify that

$$IX = IY = IZ = r \tag{37}$$

r is known as the *inradius* of ΔABC.

4.10 Draw the incircle of $\triangle ABC$

4.11 Draw the circumcircle of $\triangle ABC$