

1.9.21

Kartik Lahoti - EE25BTECH11032

August 26,2025

Question

Given vertices of a parallelogram

A $(-2, 1)$, **B** $(a, 0)$, **C** $(4, b)$, and **D** $(1, 2)$. Find the values of a and b .
Hence, find the lengths of its sides.

Theoretical Solution

Given,
A parallelogram $ABCD$ with ,

$$\mathbf{A} = \begin{pmatrix} -2 \\ 1 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} a \\ 0 \end{pmatrix}, \mathbf{C} = \begin{pmatrix} 4 \\ b \end{pmatrix}, \mathbf{D} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad (1)$$

In a Parallelogram $PQRS$ the opposite side are parallel and equal , i.e.
 $PQ \parallel RS$ and $PQ = RS$ and similarly $QR \parallel PS$ and $QR = PS$
 \therefore Here we can say,

$$\mathbf{AB = DC} \quad (2)$$

Theoretical Solution

Calculating **AB** ,

$$\mathbf{AB} = \mathbf{B} - \mathbf{A} \quad (3)$$

$$\mathbf{AB} = \begin{pmatrix} a \\ 0 \end{pmatrix} - \begin{pmatrix} -2 \\ 1 \end{pmatrix} = \begin{pmatrix} a+2 \\ -1 \end{pmatrix} \quad (4)$$

Similarly,

$$\mathbf{DC} = \begin{pmatrix} 3 \\ b-2 \end{pmatrix} \quad (5)$$

Theoretical Solution

From Eqn (2) we get :

$$\begin{pmatrix} a+2 \\ -1 \end{pmatrix} = \begin{pmatrix} 3 \\ b-2 \end{pmatrix} \quad (6)$$

(7)

$$\implies a = 1 \text{ and } b = 1 \quad (8)$$

$\therefore a = 1$ and $b = 1$

Theoretical Solution

Calculating the side lengths ,

$$\therefore \mathbf{A} - \mathbf{B} = \begin{pmatrix} -3 \\ 1 \end{pmatrix}, \quad (9)$$

$$(\mathbf{A} - \mathbf{B})^T (\mathbf{A} - \mathbf{B}) = 10 \quad (10)$$

Thus, the desired length \mathbf{AB} is

$$d_1 = \|\mathbf{A} - \mathbf{B}\| = \sqrt{10} \quad (11)$$

Theoretical Solution

Similarly,

$$\therefore \mathbf{B} - \mathbf{C} = \begin{pmatrix} -3 \\ -1 \end{pmatrix}, \quad (12)$$

$$(\mathbf{B} - \mathbf{C})^T (\mathbf{B} - \mathbf{C}) = 10 \quad (13)$$

Thus, the desired length \mathbf{BC} is

$$d_2 = \|\mathbf{B} - \mathbf{C}\| = \sqrt{10} \quad (14)$$

Hence: The length of the sides of the parallelogram is $\sqrt{10}$

C Code (1) - Function to Magnitude of Vector AB

```
#include <math.h>
double length(double *A , double *B , int m )
{
    double sum = 0.0;
    for ( int i = 0 ; i < m ; i++ )
    {
        sum += pow(A[i]-B[i] , 2 );
    }
    return sqrt(sum) ;
}
```

C Code (2) - Function to Generate Points on Line

```
void linegen(double *X, double *Y , double *A , double *B , int n
, int m )
{
    double temp[m] ;
    for (int i = 0 ; i < m ; i++)
    {
        temp [ i ] = (B[i]- A[i]) /(double) n ;
    }
    for (int i = 0 ; i <= n ; i++ )
    {
        X[i] = A[0] + temp[0] * i ;
        Y[i] = A[1] + temp[1] * i ;
    }
}
```

Python Code - Using Shared Object

```
import ctypes
import numpy as np
import matplotlib.pyplot as plt
def length_func (P: np.ndarray , Q: np.ndarray, m ) -> float:
    handc1 = ctypes.CDLL("./length.so")

    handc1.length.argtypes = [
        ctypes.POINTER(ctypes.c_double),
        ctypes.POINTER(ctypes.c_double),
        ctypes.c_int ]
    handc1.length.restype = ctypes.c_double

    len = handc1.length (
        P.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
        Q.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
        m )
    return len
```

Python Code - Using Shared Object

```
A = np.array([[ -2],[1]], dtype=np.float64)
B = np.array([[1],[0]], dtype=np.float64)
C = np.array([[4],[1]], dtype=np.float64)
D = np.array([[1],[2]], dtype=np.float64)

d1 = length_func(A,B,2)
d2 = length_func(B,C,2)

if d1 == d2 :
    print("Length of Sides =",d1)
else:
    print("Length of Side AB and CD = ",d1)
    print("Length of Side BC and AD = ",d2)
```

Python Code - Using Shared Object

```
def line_cre(P: np.ndarray , Q: np.ndarray, str):  
    handc2 = ctypes.CDLL("./line_gen.so")  
  
    handc2.linegen.argtypes = [  
        ctypes.POINTER(ctypes.c_double),  
        ctypes.POINTER(ctypes.c_double),  
        ctypes.POINTER(ctypes.c_double),  
        ctypes.POINTER(ctypes.c_double),  
        ctypes.c_int , ctypes.c_int  
    ]  
  
    handc2.linegen.restype = None
```

Python Code - Using Shared Object

```
n = 200
X_1 = np.zeros(n,dtype=np.float64)
Y_1 = np.zeros(n,dtype=np.float64)

handc2.linegen (
    X_1.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    Y_1.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    P.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    Q.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    n,2
)
plt.plot([X_1[0],X_1[-1]], [Y_1[0],Y_1[-1]],str)
```

Python Code - Using Shared Object

```
plt.figure()

line_cre(A,B,"g-")
line_cre(B,C,"r-")
line_cre(C,D,"b-")
line_cre(D,A,"y-")

coords = np.block([[A,B,C,D]])
plt.scatter(coords[0,:],coords[1,:])
vert_labels = ['A','B','C','D']
# for i , txt in enumerate(vert_labels):
#     plt.annotate(txt,(coords[0,i],coords[1,i]),textcoords="offset
#         points", xytext=(0,10),ha='center')
```

Python Code - Using Shared Object

```
for i, txt in enumerate(vert_labels):
    plt.annotate(f'{{txt}}\n({coords[0,i]:.0f}, {coords[1,i]:.0f})'
                ,
                (coords[0,i], coords[1,i]),
                textcoords="offset points",
                xytext=(20,0),
                ha='center')

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


Python Code - Using Shared Object

```
plt.title("Fig:1.9.21")
plt.axis('equal')

plt.savefig("../figs/p_gram1.png")
plt.show()

#plt.savefig('figs/triangle/ang-bisect.pdf')
#subprocess.run(shlex.split("termux-open figs/triangle/ang-bisect
.pdf"))
```

Python Code

```
import math
import sys
sys.path.insert(0, '/home/kartik-lahoti/matgeo/codes/CoordGeo')
import numpy as np
import numpy.linalg as LA
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

from line.funcs import *
#from triangle.funcs import *
#from conics.funcs import circ_gen

#if using termux
#import subprocess
#import shlex
```

Python Code

```
def length(P,Q) :  
    return LA.norm(P-Q)  
  
A = np.array([-2,1]).reshape(-1,1)  
B = np.array([1,0]).reshape(-1,1)  
C = np.array([4,1]).reshape(-1,1)  
D = np.array([1,2]).reshape(-1,1)  
  
d1 = length(A,B)  
d2 = length(B,C)  
if d1 != d2 :  
    print("Length of AB and CD = ",d1)  
    print("Length of BC and AD = ",d2)  
else :  
    print("Length of all sides = ",d1)
```

Python Code

```
def plot_it(P,Q,str):  
    x_l = line_gen_num(P,Q,20)  
    plt.plot(x_l[0,:],x_l[1,:] , str )  
  
plt.figure()  
  
plot_it(A,B,"g-")  
plot_it(B,C,"r-")  
plot_it(C,D,"b-")  
plot_it(D,A,"y-")
```

```
coords = np.block([[A,B,C,D]])
plt.scatter(coords[0,:],coords[1,:])
vert_labels = ['A','B','C','D']
#for i , txt in enumerate(vert_labels):
#    plt.annotate(txt,(coords[0,i],coords[1,i]),textcoords="offset
#        points", xytext=(0,10),ha='center')
for i, txt in enumerate(vert_labels):
    plt.annotate(f'{txt}\n({coords[0,i]:.0f}, {coords[1,i]:.0f})'
        ,
                (coords[0,i], coords[1,i]),
                textcoords="offset points",
                xytext=(20,0),
                ha='center')
```

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

plt.title("Fig:1.9.21")
plt.axis('equal')

plt.savefig("../figs/p_gram2.png")
plt.show()

#plt.savefig('figs/triangle/ang-bisect.pdf')
#subprocess.run(shlex.split("termux-open figs/triangle/ang-bisect
.pdf"))
```

