3.2.3

Vivek K Kumar - EE25BTECH11062

September 14, 2025

Question

Draw a parallelogram ABCD in which $BC = 5cm, AB = 3cm$ and $\angle ABC = 60^\circ$, divide it into triangles ACB and ABD by the diagonal BD

Solution

Let **A**, **B**, **C** and **D** represent position vectors of the vertices of parallelogram.
Given information,

$$\|\mathbf{A}-\mathbf{B}\| = 3 \tag{1}$$

$$\|\mathbf{C}-\mathbf{B}\| = 5 \tag{2}$$

$$\angle B = \frac{\pi}{3} \tag{3}$$

The coordinates of $\mathbf{A}, \mathbf{B}, \mathbf{C}$ can be expressed as

$$\mathbf{B} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{4}$$

$$\mathbf{C} = \|\mathbf{C}-\mathbf{B}\| \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{5}$$

$$\mathbf{A} = \|\mathbf{A}-\mathbf{B}\| \begin{pmatrix} cosB \\ sinB \end{pmatrix} \tag{6}$$

Solution

Since $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ form vertices of a parallelogram,

$$\frac{\mathbf{A}+\mathbf{C}}{2} = \frac{\mathbf{B}+\mathbf{D}}{2} \tag{7}$$

$$\mathbf{D} = \mathbf{A}+\mathbf{C}-\mathbf{B} \tag{8}$$

$$\mathbf{D} = \|\mathbf{A}-\mathbf{B}\| \begin{pmatrix} cosB \\ sinB \end{pmatrix} + \|\mathbf{C}-\mathbf{B}\| \begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{9}$$

Substituting values,

$$\mathbf{A} = \begin{pmatrix} 3/2 \\ 3\sqrt{3}/2 \end{pmatrix} \tag{10}$$

$$\mathbf{B} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{11}$$

$$\mathbf{C} = \begin{pmatrix} 5 \\ 0 \end{pmatrix} \tag{12}$$

Variables used

$$\mathbf{D} = \begin{pmatrix} 13/2 \\ 3\sqrt{3}/2 \end{pmatrix} \tag{13}$$

Name	Point
A	$\begin{pmatrix} 3/2 \\ 3\sqrt{3}/2 \end{pmatrix}$
B	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$
C	$\begin{pmatrix} 5 \\ 0 \end{pmatrix}$
D	$\begin{pmatrix} 13/2 \\ 3\sqrt{3}/2 \end{pmatrix}$

Table: Coordinates of vertices of parallelogram

Python - Importing libraries and checking system

```
import sys
import numpy as np
import numpy.linalg as LA
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import math

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

import subprocess
import shlex

print('Using termux?(y/n)')
y = input()
```

Python - Defining vertices of parallelogram

```
A = np.array([3/2, 3*math.sqrt(3)/2]).reshape(-1, 1)
B = np.array([0, 0]).reshape(-1, 1)
C = np.array([5, 0]).reshape(-1, 1)
D = np.array([13/2, 3*math.sqrt(3)/2]).reshape(-1,1)
```

Python - Generating points and plotting

```
p_A = line_gen(A, B)
p_B = line_gen(B, C)
p_C = line_gen(C, D)
p_D = line_gen(D, A)

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(p_A[0, :], p_A[1, :], label = 'Line AB')
ax.plot(p_B[0, :], p_B[1, :], label = 'Line BC')
ax.plot(p_C[0, :], p_C[1, :], label = 'Line CD')
ax.plot(p_D[0, :], p_D[1, :], label = 'Line DA')
```

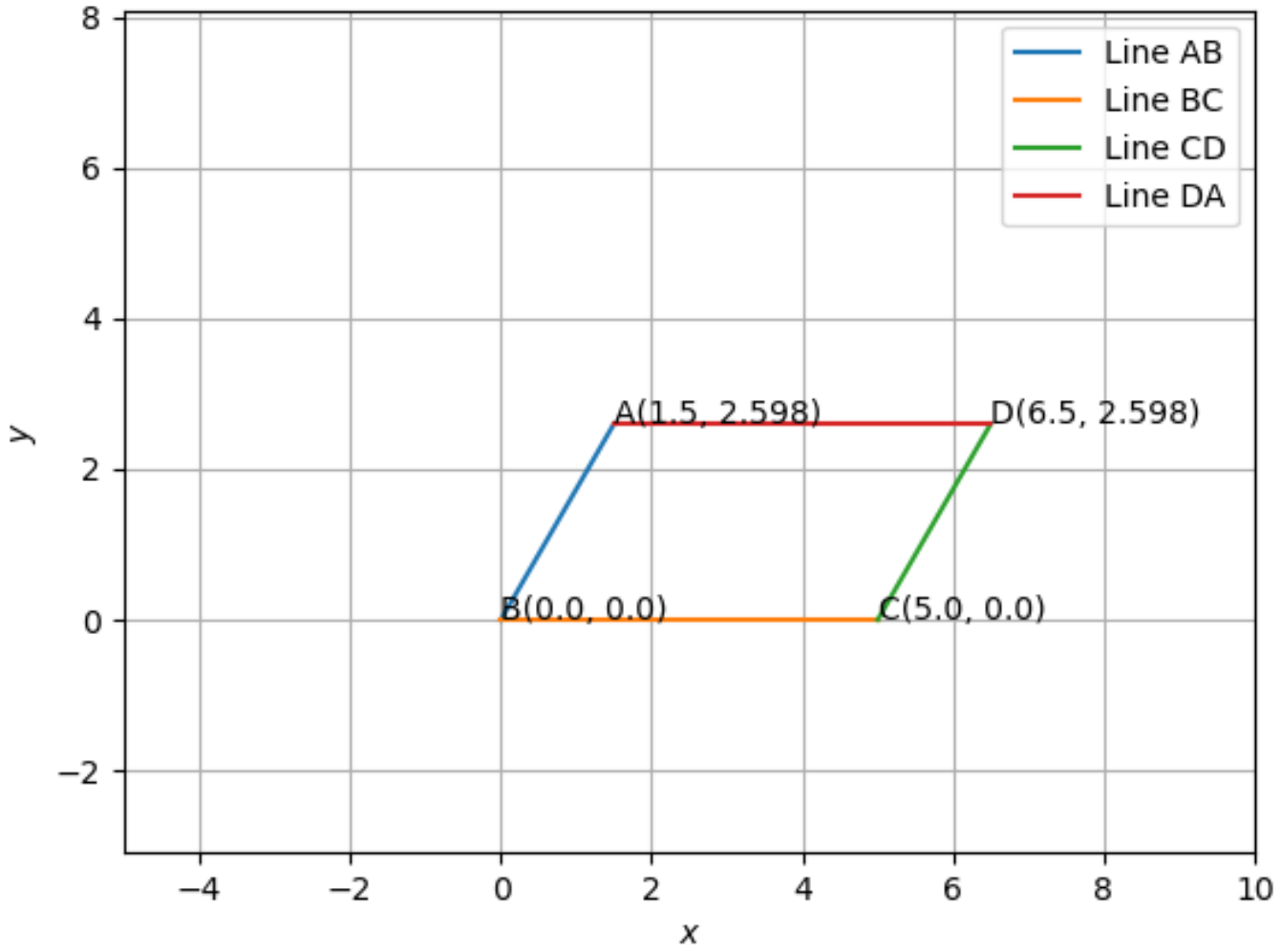
Python - Labelling points

```
pts = np.block([A, B, C, D])
names = ['A', 'B', 'C', 'D']
for i in range(4):
    X = pts[:, i]
    ax.text(X[0], X[1], s=f'{names[i]}({round(X[0], 3)}, {round(X
        [1],3)})')

ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
ax.legend(loc='best')
ax.grid(True)
ax.axis('equal')
ax.set_xlim([-5, 10])
ax.set_ylim([-5, 10])
```

Python - Saving figure and opening it

```
fig.savefig('../figs/fig.png')
print('Saved figure to ../figs/fig.png')

if(y == 'y'):
    subprocess.run(shlex.split('termux-open ../figs/fig.png'))
else:
    subprocess.run(["open", "../figs/fig.png"])
```

Plot-Using only Python



C Code (0) - Importing libraries

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include "libs/matfun.h"
#include "libs/geofun.h"
```

C Code (1) - Function to Generate Points on a Line

```c
void point_gen(FILE *p_file, double **A, double **B, int rows,
    int cols, int npts){
    for(int i = 0; i <= npts; i++){
     double **output = Matadd(A, Matscale(Matsub(B, A, rows, cols
         ), rows, cols, (double)i/npts), rows, cols);
     fprintf(p_file, "%lf, %lf\n", output[0][0], output[1][0]);
     freeMat(output, rows);
    }
}
```

C Code (2) - Function to write points b/w given point and origin to a file

```
void write_points(double x1, double y1, double x2, double y2,
    double x3, double y3, double x4, double y4, int npts){
    int m = 2;
    int n = 1;

    double **A = createMat(m, n);
    double **B = createMat(m, n);
    double **C = createMat(m, n);
    double **D = createMat(m, n);

    B[0][0] = x2;
    B[1][0] = y2;
```

C Code (2) - Function to write points b/w given 2 points to a file

```
    A[0][0] = x1;
    A[1][0] = y1;

    C[0][0] = x3;
    C[1][0] = y3;

    D[0][0] = x4;
    D[1][0] = y4;

    FILE *p_file;
    p_file = fopen("plot.dat", "w");
    if(p_file == NULL)
        printf("Error opening one of the data files\n");
```

C Code (2) - Function to write points b/w given 2 points to a file

```
    point_gen(p_file, A, B, m, n, npts);
    point_gen(p_file, B, C, m, n, npts);
    point_gen(p_file, C, D, m, n, npts);
    point_gen(p_file, D, A, m, n, npts);

    freeMat(A, m);
    freeMat(B, m);
    freeMat(C, m);
    freeMat(D, m);

    fclose(p_file);
}
```

Python Code (0) - Importing libraries and checking system

```
import numpy as np
import matplotlib.pyplot as plt
import ctypes
import os
import sys
import subprocess
import math

print('Using termux? (y/n)')
termux = input()
```

Python Code (1) - Using Shared Object

```
lib_path = os.path.join(os.path.dirname(__file__), 'plot.so')
my_lib = ctypes.CDLL(lib_path)

my_lib.write_points.argtypes = [ctypes.c_double, ctypes.c_double,
     ctypes.c_double, ctypes.c_double, ctypes.c_double, ctypes.
    c_double, ctypes.c_double, ctypes.c_double, ctypes.c_int]
my_lib.write_points.restype = None
A = np.array([3/2, 3*math.sqrt(3)/2]).reshape(-1, 1)
B = np.array([0, 0]).reshape(-1, 1)
C = np.array([5, 0]).reshape(-1, 1)
D = np.array([13/2, 3*math.sqrt(3)/2]).reshape(-1, 1)
npts = 20000
```

Python Code (2) - Loading points and plotting them

```
my_lib.write_points(A[0][0], A[1][0], B[0][0], B[1][0], C[0][0],
    C[1][0], D[0][0], D[1][0], npts)

fig = plt.figure()
ax = fig.add_subplot(111)
labels = ['AB', 'BC', 'CD', 'DA']
pts = np.block([A, B, C, D])
vertices = ['A', 'B', 'C', 'D']
for i,label in enumerate(labels):
    points = np.loadtxt('plot.dat', delimiter = ',', usecols
        =(0,1))[i*(npts+1):(i+1)*(npts+1)]
    ax.plot(points[:, 0], points[:, 1], label = f'Line {label}')
    ax.text(pts[:, i][0], pts[:, i][1], s=f'{vertices[i]}({round(
        pts[:, i][0],3)}, {round(pts[:, i][1],3)})')
```

Python Code (3) - Labelling plot

```
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
ax.legend(loc='best')
ax.grid()
ax.axis('equal')
ax.set_xlim([-5, 10])
ax.set_ylim([-5, 10])
```

Python Code (4) - Saving and displaying plot

```
fig.savefig('../figs/fig2.png')
print('Saved figure to ../figs/fig2.png')

if(termux == 'y'):
    subprocess.run(shlex.split('termux-open ../figs/fig2.png'))
else:
    subprocess.run(["open", "../figs/fig2.png"])
```

Plot-Using Both C and Python