4.13.34

Jnanesh Sathisha karmar - EE25BTECH11029

September 30,2025

Question

The equations to a pair of opposite sides of parallogram are
$x^2 - 5x + 6 = 0$ and $y^2 - 6y + 5 = 0$, the equations to its diagonals are

1. $x + 4y = 13, y = 4x - 7$
2. $4x + y = 13, y = 4x - 7$
3. $4x + y = 13, 4y = x - 7$
4. $y - 4x = 13, y + 4x = 7$

Equation

Given details
Equation 1:

$$x^2 - 5x + 6 = 0 \tag{1}$$

$$\text{This equation can be factored into:} \tag{2}$$

$$(x - 2)(x - 3) = 0 \tag{3}$$

$$\text{This gives us two vertical lines:} \tag{4}$$

$$x = 2 \tag{5}$$

$$x = 3 \tag{6}$$

Equation 2:

$$y^2 - 6y + 5 = 0 \tag{7}$$

$$\text{This equation can be factored into:} \tag{8}$$

$$(y - 1)(y - 5) = 0 \tag{9}$$

$$\text{This gives us two horizontal lines:} \tag{10}$$

$$y = 1 \tag{11}$$

$$y = 5 \tag{12}$$

Theoretical Solution

Through the intersection of these 4 lines we can find the 4 vertices of the parallelogram

Intersection of $x = 2$ and $y = 1$ is the point $(2, 1)$. Let vector $\mathbf{A} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$

$$(13)$$

Intersection of $x = 3$ and $y = 1$ is the point $(3, 1)$. Let vector $\mathbf{B} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$

$$(14)$$

Intersection of $x = 3$ and $y = 5$ is the point $(3, 5)$. Let vector $\mathbf{C} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}$

$$(15)$$

Intersection of $x = 2$ and $y = 5$ is the point $(2, 5)$. Let vector $\mathbf{D} = \begin{pmatrix} 2 \\ 5 \end{pmatrix}$

$$(16)$$

(17)

Theoretical Solution

The equations of the diagonals can be found using the matrix method.
The equation of a line through (x_1, y_1) and (x_2, y_2) is given by setting the
determinant of the matrix of coordinates to zero, as three collinear points
form a triangle with zero area.

$$\det \begin{pmatrix} x & y & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{pmatrix} = 0$$

Theoretical Solution

The equation of the diagonal **AC**, passing through **A** $(2, 1)$ and **C** $(3, 5)$, is:

$$\det \begin{pmatrix} x & y & 1 \\ 2 & 1 & 1 \\ 3 & 5 & 1 \end{pmatrix} = 0 \tag{18}$$

$$x(1 \cdot 1 - 5 \cdot 1) - y(2 \cdot 1 - 3 \cdot 1) + 1(2 \cdot 5 - 3 \cdot 1) = 0 \tag{19}$$

$$x(1 - 5) - y(2 - 3) + 1(10 - 3) = 0 \tag{20}$$

$$-4x - y(-1) + 7 = 0 \tag{21}$$

$$-4x + y + 7 = 0 \tag{22}$$

$$y = 4x - 7 \tag{23}$$

Theoretical Solution

The equation of the diagonal **BD**, passing through **B**$(3,1)$ and **D**$(2,5)$, is:

$$\det \begin{pmatrix} x & y & 1 \\ 3 & 1 & 1 \\ 2 & 5 & 1 \end{pmatrix} = 0 \qquad (24)$$

$$x(1 \cdot 1 - 5 \cdot 1) - y(3 \cdot 1 - 2 \cdot 1) + 1(3 \cdot 5 - 2 \cdot 1) = 0 \qquad (25)$$

$$x(1-5) - y(3-2) + 1(15-2) = 0 \qquad (26)$$

$$-4x - y(1) + 13 = 0 \qquad (27)$$

$$-4x - y + 13 = 0 \qquad (28)$$

$$4x + y = 13 \qquad (29)$$

Therefore the equations of both the diagonals are:

$$y = 4x - 7 \tag{30}$$

$$4x + y = 13 \tag{31}$$

Hence the answer is option 2

C Code (1) - Function to store the points

```c
#include <stddef.h>

void find_diagonals(double x1, double x2, double y1, double y2,
    double* diag1_coeffs, double* diag2_coeffs) {
    if (diag1_coeffs == NULL || diag2_coeffs == NULL) {
        return;
    }

    diag1_coeffs[0] = y2 - y1;
    diag1_coeffs[1] = x1 - x2;
    diag1_coeffs[2] = (x2 * y1) - (x1 * y2);

    diag2_coeffs[0] = y1 - y2;
    diag2_coeffs[1] = x1 - x2;
    diag2_coeffs[2] = (x2 * y2) - (x1 * y1);
}
```

```python
import ctypes
import numpy as np
import matplotlib.pyplot as plt
def solve_quadratic(a, b, c):
    discriminant = b**2 - 4*a*c
    if discriminant < 0:
        return None, None
    r1 = (-b + np.sqrt(discriminant)) / (2*a)
    r2 = (-b - np.sqrt(discriminant)) / (2*a)
    return r1, r2
def main():
    lib_path = "./diagonals.so"
    try:
        diag_lib = ctypes.CDLL(lib_path)
    except OSError as e:
        print(f"Error loading shared library: {e}")
        print("Please ensure 'diag_calculator.so' exists. You may
            need to compile the C code first using 'sh compile.sh
            '")
```

```python
diag_lib.find_diagonals.argtypes = [
    ctypes.c_double, ctypes.c_double,
    ctypes.c_double, ctypes.c_double,
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double)
]
diag_lib.find_diagonals.restype = None

x1, x2 = solve_quadratic(1, -5, 6)
y1, y2 = solve_quadratic(1, -6, 5)

if x1 is None or y1 is None:
    print("Error: Could not solve quadratic equations. Check
        coefficients.")
    return
```

Python Code - Using Shared Object

```python
        print(f"Parallelogram lines: x={x1}, x={x2}, y={y1}, y={y2
            }")

    Diag1Coeffs = (ctypes.c_double * 3)()
    Diag2Coeffs = (ctypes.c_double * 3)()

    diag_lib.find_diagonals(x1, x2, y1, y2, Diag1Coeffs,
        Diag2Coeffs)

    d1 = [c for c in Diag1Coeffs]
    d2 = [c for c in Diag2Coeffs]
    print(f"Diagonal 1 (Ax+By+C=0): A={d1[0]:.1f}, B={d1[1]:.1f},
        C={d1[2]:.1f}")
    print(f"Diagonal 2 (Ax+By+C=0): A={d2[0]:.1f}, B={d2[1]:.1f},
        C={d2[2]:.1f}")
```

Python Code - Using Shared Object

```python
fig, ax = plt.subplots(figsize=(8, 8))

    parallelogram_x = [x1, x2, x2, x1, x1]
    parallelogram_y = [y1, y1, y2, y2, y1]
    ax.plot(parallelogram_x, parallelogram_y, 'b-', label='
        Parallelogram Sides', linewidth=2)


    plot_x_range = np.linspace(min(x1, x2) - 1, max(x1, x2) + 1,
        100)


    def plot_line(coeffs, style, label):
        A, B, C = coeffs
        if abs(B) > 0:
            y_vals = (-A * plot_x_range - C) / B
            ax.plot(plot_x_range, y_vals, style, label=label)
        else:
            x_val = -C / A
            ax.axvline(x=x_val, linestyle=style.strip('-'), color=
                style[0], label=label)
```

```python
plot_line(d1, 'r--', 'Diagonal 1')
plot_line(d2, 'g--', 'Diagonal 2')


ax.set_title('Parallelogram and its Diagonals')
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.grid(True, linestyle=':', alpha=0.6)
ax.legend()


padding = 2
ax.set_xlim(min(x1, x2) - padding, max(x1, x2) + padding)
ax.set_ylim(min(y1, y2) - padding, max(y1, y2) + padding)
ax.set_aspect('equal', adjustable='box')
```
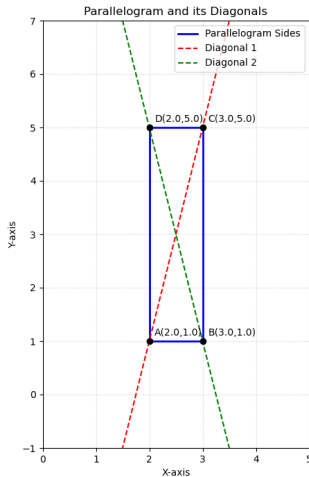
```python
    x_min, x_max = min(x1, x2), max(x1, x2)
    y_min, y_max = min(y1, y2), max(y1, y2)
    vertices = {'A':(x_min, y_min), 'B':(x_max, y_min), 'C':(
        x_max, y_max), 'D':(x_min, y_max)}
    for name, (px, py) in vertices.items():
        ax.plot(px, py, 'ko')
        ax.text(px + 0.1, py + 0.1, f'{name}({px:.1f},{py:.1f})')

    plt.savefig("./figs/diagonals.png")
    subprocess.run(shlex.split('termux-open ../figs/diagonals.png
        '))
    plt.show()

if __name__ == "__main__":
    main()
```

Plot-Using Both C and Python

Python Code

```python
import numpy as np
import matplotlib.pyplot as plt

def solve_quadratic(a, b, c):
    discriminant = b**2 - 4*a*c
    if discriminant < 0:
        return None, None
    r1 = (-b + np.sqrt(discriminant)) / (2*a)
    r2 = (-b - np.sqrt(discriminant)) / (2*a)
    return r1, r2

def find_diagonals_python(x1, x2, y1, y2):
    A1 = y2 - y1
    B1 = x1 - x2
    C1 = (x2 * y1) - (x1 * y2)
    diag1_coeffs = [A1, B1, C1]
```

Python Code

```python
    A2 = y1 - y2
    B2 = x1 - x2
    C2 = (x2 * y2) - (x1 * y1)
    diag2_coeffs = [A2, B2, C2]

    return diag1_coeffs, diag2_coeffs

def main():
    x1, x2 = solve_quadratic(1, -5, 6)
    y1, y2 = solve_quadratic(1, -6, 5)

    if x1 is None or y1 is None:
        print("Error: Could not solve quadratic equations. Check
            coefficients.")
        return
```

Python Code

```python
print(f"Parallelogram lines: x={x1}, x={x2}, y={y1}, y={y2}")

d1, d2 = find_diagonals_python(x1, x2, y1, y2)

print(f"Diagonal 1 (Ax+By+C=0): A={d1[0]:.1f}, B={d1[1]:.1f},
    C={d1[2]:.1f}")
print(f"Diagonal 2 (Ax+By+C=0): A={d2[0]:.1f}, B={d2[1]:.1f},
    C={d2[2]:.1f}")

fig, ax = plt.subplots(figsize=(8, 8))

parallelogram_x = [x1, x2, x2, x1, x1]
parallelogram_y = [y1, y1, y2, y2, y1]
ax.plot(parallelogram_x, parallelogram_y, 'b-', label='
    Parallelogram Sides', linewidth=2)
```

Python Code

```python
plot_x_range = np.linspace(min(x1, x2) - 1, max(x1, x2) + 1,
    100)

def plot_line(coeffs, style, label):
    A, B, C = coeffs
    if abs(B) > 0:
        y_vals = (-A * plot_x_range - C) / B
        ax.plot(plot_x_range, y_vals, style, label=label)
    else:
        x_val = -C / A
        ax.axvline(x=x_val, linestyle=style.strip('-'), color=
            style[0], label=label)

plot_line(d1, 'r--', 'Diagonal 1')
plot_line(d2, 'g--', 'Diagonal 2')
```

Python Code

```python
ax.set_title('Parallelogram and its Diagonals')
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.grid(True, linestyle=':', alpha=0.6)
ax.legend()

padding = 2
ax.set_xlim(min(x1, x2) - padding, max(x1, x2) + padding)
ax.set_ylim(min(y1, y2) - padding, max(y1, y2) + padding)
ax.set_aspect('equal', adjustable='box')
```

Python Code

```python
    x_min, x_max = min(x1, x2), max(x1, x2)
    y_min, y_max = min(y1, y2), max(y1, y2)
    vertices = {'A':(x_min, y_min), 'B':(x_max, y_min), 'C':(
        x_max, y_max), 'D':(x_min, y_max)}
    for name, (px, py) in vertices.items():
        ax.plot(px, py, 'ko')
        ax.text(px + 0.1, py + 0.1, f'{name}({px:.1f},{py:.1f})')
    plt.savefig("./figs/diagonals2.png")
    plt.show()

if __name__ == "__main__":
    main()
```

Plot-Using only Python