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. $y - 4x = 13, y + 4x = 7$

Theoretical Solution

We can solve this problem by treating a pair of parallel lines as a degenerate conic section and finding where a line (the diagonal) intersects it. The general equation for a conic is given by $g(\mathbf{x}) = \mathbf{x}^\intercal \left(V \right) \mathbf{x} + 2\mathbf{u}^\intercal \mathbf{x} + f = 0$, and a parametric line is given by $\mathbf{x} = \mathbf{h} + \kappa \mathbf{m}$. The intersection points are found using the formula:

$$\kappa_{1,2} = \frac{-\mathbf{m}^\top \left(\left(V \right) \mathbf{h} + \mathbf{u} \right) \pm \sqrt{\left(\mathbf{m}^\top \left(\left(V \right) \mathbf{h} + \mathbf{u} \right) \right)^2 - \left(\mathbf{m}^\top \left(V \right) \mathbf{m} \right) g(\mathbf{h})}}{\mathbf{m}^\top \left(V \right) \mathbf{m}} \tag{1}$$

Theoretical Solution

Let's represent the pair of vertical lines $x^2 - (x_1 + x_2)x + x_1x_2 = 0$ as our conic section.

$$\mathbf{V} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} -\frac{x_1+x_2}{2} \\ 0 \end{pmatrix} \; f = x_1x_2 \tag{2}$$

The diagonal is a line starting from the parallelogram's center **h** with a direction vector **m**.

$$\mathbf{h} = \begin{pmatrix} \frac{x_1+x_2}{2} \\ \frac{y_1+y_2}{2} \end{pmatrix}, \; \mathbf{m} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix} \tag{3}$$

Theoretical Solution

First, we evaluate the term $\mathbf{Vh} + \mathbf{u}$:

$$\mathbf{Vh} + \mathbf{u} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{x_1+x_2}{2} \\ \frac{y_1+y_2}{2} \end{pmatrix} + \begin{pmatrix} -\frac{x_1+x_2}{2} \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{x_1+x_2}{2} \\ 0 \end{pmatrix} - \begin{pmatrix} \frac{x_1+x_2}{2} \\ 0 \end{pmatrix} = \mathbf{0} \tag{4}$$

Since $\mathbf{Vh} + \mathbf{u} = \mathbf{0}$, the formula for κ simplifies dramatically:

$$\kappa_{1,2} = \frac{\pm\sqrt{-(\mathbf{m}^\top \left(V\right) \mathbf{m}) g(\mathbf{h})}}{\mathbf{m}^\top \left(V\right) \mathbf{m}} \tag{5}$$

Next, we evaluate the remaining terms in the general case:

$$\mathbf{m}^\top \left(V\right) \mathbf{m} = (x_2 - x_1)^2 \tag{6}$$

$$g(\mathbf{h}) = \left(\frac{x_1 + x_2}{2}\right)^2 - (x_1 + x_2)\left(\frac{x_1 + x_2}{2}\right) + x_1 x_2 = -\frac{(x_2 - x_1)^2}{4} \tag{7}$$

Substituting these back into the simplified formula for κ:

$$\kappa_{1,2} = \frac{\pm\sqrt{-(x_2 - x_1)^2\left(-\frac{(x_2-x_1)^2}{4}\right)}}{(x_2 - x_1)^2} = \frac{\pm\sqrt{\frac{(x_2-x_1)^4}{4}}}{(x_2 - x_1)^2} = \frac{\pm\frac{(x_2-x_1)^2}{2}}{(x_2 - x_1)^2} = \pm\frac{1}{2} \tag{8}$$

This shows that the vertices are located at $\kappa = \pm 1/2$ from the center along the direction vector $\mathbf{m}$.

Applying to the specific problem:
From $x^2 - 5x + 6 = 0$, we have $x_1 = 2, x_2 = 3$.
From $y^2 - 6y + 5 = 0$, we have $y_1 = 1, y_2 = 5$.
The center point $\mathbf{h}$ and diagonal direction vectors $\mathbf{m}_1, \mathbf{m}_2$ are:

$$\mathbf{h} = \begin{pmatrix} 2.5 \\ 3 \end{pmatrix}, \ \mathbf{m}_1 = \begin{pmatrix} 3-2 \\ 5-1 \end{pmatrix} = \begin{pmatrix} 1 \\ 4 \end{pmatrix}, \ \mathbf{m}_2 = \begin{pmatrix} 2-3 \\ 5-1 \end{pmatrix} = \begin{pmatrix} -1 \\ 4 \end{pmatrix} \tag{9}$$

The vertices are $\mathbf{v} = \mathbf{h} \pm \frac{1}{2}\mathbf{m}$.
Diagonal 1 (using $\mathbf{m}_1$):

$$\mathbf{v}_C = \begin{pmatrix} 2.5 \\ 3 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 1 \\ 4 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix} \text{ and } \mathbf{v}_A = \begin{pmatrix} 2.5 \\ 3 \end{pmatrix} - \frac{1}{2}\begin{pmatrix} 1 \\ 4 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

The line passing through $\mathbf{A}(2,1)$ and $\mathbf{C}(3,5)$ is
$y - 1 = 4(x-2) \implies \boldsymbol{y = 4x - 7}$.

Diagonal 2 (using $\mathbf{m}_2$):

$$\mathbf{v}_D = \begin{pmatrix} 2.5 \\ 3 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} -1 \\ 4 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \end{pmatrix} \text{ and } \mathbf{v}_B = \begin{pmatrix} 2.5 \\ 3 \end{pmatrix} - \frac{1}{2}\begin{pmatrix} -1 \\ 4 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

The line passing through $\mathbf{B}\,(3, 1)$ and $\mathbf{D}\,(2, 5)$ is
$y - 1 = -4\,(x - 3) \implies \boldsymbol{4x + y = 13}$.

C Code (1) - Function to store the points

```
#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 Code - Using Shared Object

```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 Code - Using Shared Object

```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

```
        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

```
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 Code - Using Shared Object

```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')
```

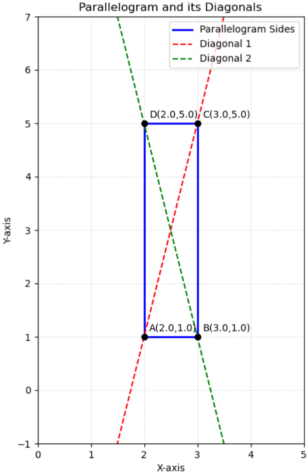
Python Code - Using Shared Object

```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

```
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

```
    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

```
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

```
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

```
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

```
    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