Points on Parabola and Area Calculation

Patnam Shariq Faraz Muhammed
EE24BTECH11049

Question

Using integration, find the area of the region enclosed by the curve $y = x^2$, the x-axis, and the ordinates $x = -2$ and $x = 1$.

Given	formula
$y = x^2$	$\mathbf{x^T V x} + \mathbf{2u^T x} + f = 0$
$x = -2$	$\begin{pmatrix} -2 \\ 4 \end{pmatrix}$
$x = 1$	$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$

Parameters of Conic - Parabola

Substituting the given values, we have:

$$\mathbf{V} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{0.1}$$

$$\mathbf{u} = \begin{pmatrix} \frac{-1}{2} \\ 0 \end{pmatrix} \tag{0.2}$$

$$f = 0 \tag{0.3}$$

Equation of Conic and Line in Matrix Form

We get the equation of the curve as

$$\mathbf{y} = \mathbf{x}^{\mathbf{T}}\mathbf{V}\mathbf{x} \tag{0.4}$$

Line equation of the form $\mathbf{x} = \mathbf{h} + k\mathbf{m}$

Intersection of Line and Conic

If a line intersects the conic, the k value of the intersecting point is given by:

$$k_i = \frac{-\mathbf{m}^\top \left(\mathbf{V}\mathbf{h} + \mathbf{u}\right) \pm \sqrt{\left[\mathbf{m}^\top \left(\mathbf{V}\mathbf{h} + \mathbf{u}\right)\right]^2 - g(h)\left(\mathbf{m}^\top \mathbf{V}\mathbf{m}\right)}}{\mathbf{m}^\top \mathbf{V}\mathbf{m}} \tag{0.5}$$

Points of Intersection

Substituting the values, we get the point of intersection as:

$$\kappa_i = -\begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} \frac{-1}{2} & 0 \end{pmatrix} \pm \sqrt{\left[\begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{-1}{2} \\ 0 \end{pmatrix}\right]^2 + 1 \cdot (1)} \tag{0.6}$$

$$\kappa_i = 1 \tag{0.7}$$

Hence, the point of intersection is $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Similarly, the other point is given by $\begin{pmatrix} -2 \\ 4 \end{pmatrix}$.

Area Calculation by Integration

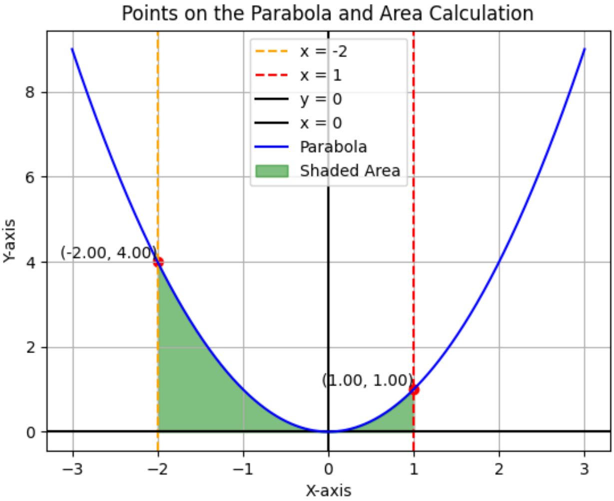
The area bounded by the curve and the line is:

$$\int_{-2}^{1} \left(x^2\right) dx = \frac{1}{3}\left(1 - (-8)\right) \tag{0.8}$$

$$= 3 \tag{0.9}$$

Hence the required area is 3.

A Plot of the Given Question



C Code: Area and Points on the Curve I

```c
#include <stdio.h>
#include <stdlib.h>

// Define the struct for the Parabola with coefficients a, b, c, and variable x
typedef struct {
    double a;
    double b;
    double c;
    double x;
} Parabola;

// Define the function for the parabola y = ax^2 + bx + c using the Parabola struct
double function(Parabola *p) {
    return p->a * p->x * p->x + p->b * p->x + p->c;
}

// Calculate the area using Riemann sum from lower_limit to upper_limit
double area(Parabola *p, double lower_limit, double upper_limit) {
    double sum = 0.0;
    double delta_x = 1e-7; // Width of each rectangle
    int num_points = (int)((upper_limit - lower_limit) / delta_x);

    for (int i = 0; i < num_points; i++) {
        p->x = lower_limit + i * delta_x; // Set x value from lower limit to upper limit
        sum += function(p) * delta_x; // Height * Width
    }

    return sum;
```

C Code: Area and Points on the Curve II

```c
}

// Generate points in the range from x_start to x_end and store them in the arrays
void generate_points(Parabola *p, double x_start, double x_end, int num_points, double *points_x, double *
      points_y)
{
    double delta_x = (x_end - x_start) / (num_points - 1);

    for (int i = 0; i < num_points; i++) {
        p->x = x_start + i * delta_x;
        points_x[i] = p->x;
        points_y[i] = function(p);
    }
}
```

Python: To plot the points I

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

# Load the shared C library
lib = ctypes.CDLL('./libarea_calculator.so')

# Define the Parabola struct equivalent in Python
class Parabola(ctypes.Structure):
    _fields_ = [("a", ctypes.c_double),
                ("b", ctypes.c_double),
                ("c", ctypes.c_double),
                ("x", ctypes.c_double)]

# Define the C function argument and return types
lib.area.argtypes = [ctypes.POINTER(Parabola), ctypes.c_double, ctypes.c_double]
lib.area.restype = ctypes.c_double

lib.generate_points.argtypes = [
    ctypes.POINTER(Parabola), ctypes.c_double, ctypes.c_double,
    ctypes.c_int, ctypes.POINTER(ctypes.c_double), ctypes.POINTER(ctypes.c_double)
]

# Function to calculate the value of the parabola at a given x
def function(a, b, c, x):
    return a * x**2 + b * x + c

def main(a, b, c, x_start, x_end, num_points, area_start, area_end):
```

Python: To plot the points II

```
# Create a Parabola struct and set the values of a, b, c
p = Parabola(a=a, b=b, c=c, x=0.0)

# Calculate the area using the C function from area_start to area_end
area = lib.area(ctypes.byref(p), area_start, area_end) # Calculate area using C function

# Corrected print statement for area calculation
print(f"Calculated␣Area␣(using␣Riemann␣sum)␣from␣({area_start},"
      f"{function(a,␣b,␣c,␣area_start):.2f})␣to␣({area_end},␣"
      f"{function(a,␣b,␣c,␣area_end):.2f}):␣{area:.2f}")

# Allocate memory for points
points_x = (ctypes.c_double * num_points)()
points_y = (ctypes.c_double * num_points)()

# Generate points using the C function for the range from x_start to x_end
lib.generate_points(ctypes.byref(p), x_start, x_end, num_points, points_x, points_y)

# Prepare data for plotting
plot_x_vals = np.array(points_x)
plot_y_vals = np.array(points_y)

# Plot the lines for x_start and x_end
plt.axvline(x=area_start, label=f'x␣=␣{area_start}', color='orange', linestyle='--')
plt.axvline(x=area_end, label=f'x␣=␣{area_end}', color='red', linestyle='--')

plt.axhline(y=0, label='y␣=␣0', color='black', linestyle='-') # Horizontal line for y=0
plt.axvline(x=0, label='x␣=␣0', color='black', linestyle='-') # Vertical line for x=0
```

Python: To plot the points III

```
    # Plot the parabola
    plt.plot(plot_x_vals, plot_y_vals, label='Parabola', color='blue')

    # Shade the area under the curve between area_start and area_end
    plt.fill_between(plot_x_vals, plot_y_vals, 0, where=((plot_x_vals >= area_start) & (plot_x_vals <=
         area_end)), color='green', alpha=0.5, label='Shaded Area')

    # Get points at the limits area_start and area_end
    limit_points = [(area_start, function(a, b, c, area_start)), (area_end, function(a, b, c, area_end))]

    # Plot the limit points
    for x, y in limit_points:
        plt.scatter(x, y, color='red', marker='o')
        plt.text(x, y, f'({x:.2f}, {y:.2f})', fontsize=10, verticalalignment='bottom', horizontalalignment='
             right')

    plt.title('Points on the Parabola and Area Calculation')
    plt.xlabel('X-axis')
    plt.ylabel('Y-axis')
    plt.grid()
    plt.legend()
    plt.savefig('../figs/fig.png')
    plt.show()

if __name__ == "__main__":
    # Example usage with x_start = -3, x_end = 3 for generating points
    # and area_start = -2, area_end = 1 for area calculation
```

Python: To plot the points IV

```
main(a=1, b=0, c=0, x_start=-3, x_end=3, num_points=1000, area_start=-2, area_end=1)
```