Finding Area of a Bound Region Using Matrices

S A Aravind Eswar - EE24BTECH11053

November 5, 2024

1. Problem

2. Solution
   - Finding equation of conic
   - Finding points of intersection
   - Plotting the curve
   - Integration
   - Code

Problem Statement

Find the area of region bounded by the curve

$$\begin{align} x^2 &= 4y \tag{2.1}\\ y &= 2 \tag{2.2}\\ y &= 4 \tag{2.3}\\ & \tag{2.4} \end{align}$$

and the y-axis in the first quadrant.

Finding equation of conic

$$x^2 = 4y \tag{3.1}$$

$$\mathbf{V} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{3.2}$$

$$\mathbf{u} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \tag{3.3}$$

$$f = 0 \tag{3.4}$$

$$g(\mathbf{x}) = x^\top \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \mathbf{x} + 2 \begin{pmatrix} 0 \\ 2 \end{pmatrix}^\top \mathbf{x} = 0 \tag{3.5}$$

Finding points of intersection

Using,

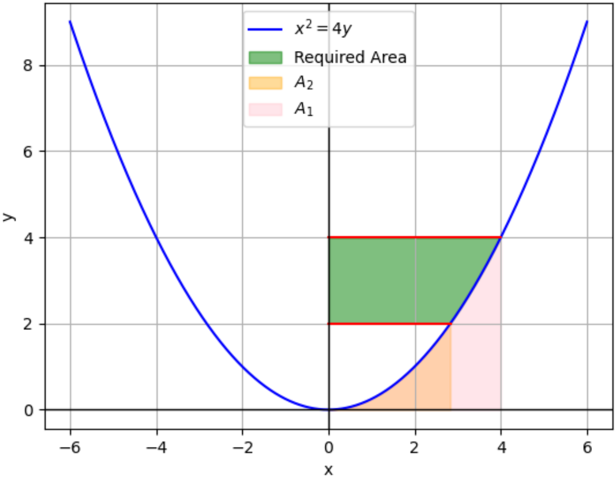
$$\kappa_i = \frac{1}{\mathbf{m}^\top \mathbf{V}\mathbf{m}} \left(-\mathbf{m}^\top \left(\mathbf{V}\mathbf{h} + \mathbf{u} \right) \pm \sqrt{\left[\mathbf{m}^\top \left(\mathbf{V}\mathbf{h} + \mathbf{u} \right) - g(\mathbf{h})(\mathbf{m}^\top \mathbf{V}\mathbf{m}) \right]} \right) \tag{3.6}$$

and substituing in the line equation $(\mathbf{x} = \mathbf{h} - \kappa\mathbf{m})$
we get points of intersection,

$$a_1 = \begin{pmatrix} 2\sqrt{2} \\ 2 \end{pmatrix} \tag{3.7}$$

$$a_2 = \begin{pmatrix} 4 \\ 4 \end{pmatrix} \tag{3.8}$$

Plotting the curve



Integration

Calculating A_1 and A_2,

$$A_1 = \int_0^4 \frac{x^2}{4} dx = \frac{16}{3} \tag{3.9}$$

$$A_2 = \int_0^{2\sqrt{2}} \frac{x^2}{4} dx = \frac{4\sqrt{2}}{3} \tag{3.10}$$

Final area,

$$A = \left(\int_0^4 4dx - A_1 \right) - \left(\int_0^{2\sqrt{2}} 2dx - A_2 \right) = \frac{32 - 8\sqrt{2}}{3} \tag{3.11}$$

Code I

Code for generating the points using C:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "libs/matfun.h"
#include "libs/geofun.h"



double conicForm(double** V, double **u, double f, double** x){
    return **Matadd(Matadd(Matmul(Matmul(transposeMat(x,2,1),V
        ,1,2,2),x,1,2,1),Matscale(Matmul(transposeMat(u,2,1),x
        ,1,2,1),1,1,2),1,1),Matscale(Mateye(1),1,1,f),1,1);
}

double* kapG(double** V, double** u, double f, double** h, double
    ** m){
```

Code II

```
double a = (**(Matscale(Matmul(transposeMat(m,2,1),Matadd(
    Matmul(V,h,2,2,1),u,2,1),1,2,1),1,1,-1))+sqrt(**Matadd(
    Matmul(transposeMat(m,2,1),Matadd(Matmul(V,h,2,2,1),u
    ,2,1),1,2,1),Matscale(Matmul(Matmul(transposeMat(m, 2, 1)
    ,V,1,2,2),m,1,2,1),1,1,-conicForm(V, u, f, h)),1,1)))/
    **(Matmul(Matmul(transposeMat(m, 2, 1),V,1,2,2),m,1,2,1))
    ;
double b = (**(Matscale(Matmul(transposeMat(m,2,1),Matadd(
    Matmul(V,h,2,2,1),u,2,1),1,2,1),1,1,-1))-sqrt(**Matadd(
    Matmul(transposeMat(m,2,1),Matadd(Matmul(V,h,2,2,1),u
    ,2,1),1,2,1),Matscale(Matmul(Matmul(transposeMat(m, 2, 1)
    ,V,1,2,2),m,1,2,1),1,1,-conicForm(V, u, f, h)),1,1)))/
    **(Matmul(Matmul(transposeMat(m, 2, 1),V,1,2,2),m,1,2,1))
    ;
double* x; x = malloc(2*sizeof(double));
x[0] = a;
x[1] = b;
return x;
```

Code III

```
}

double*** inter(double** m, double** h, double* kap){
    double** x1 = Matsub(h,Matscale(m,2,1,kap[0]),2,1);
    double** x2 = Matsub(h,Matscale(m,2,1,kap[1]),2,1);

    double*** x; malloc(2*sizeof(*x));
    x[0] = x1;
    x[1] = x2;
    return x;
}

void point_gen(FILE *fptr, double **A, double **B, int num_points
    ) {
    for (int i = 0; i <= num_points; i++) {
        double temp = (double)i/(double)num_points;
        double temp1 = 1-temp;
```

Code IV

```
        double **output = Matadd(Matscale(A,2,1,temp1),Matscale(B
            ,2,1,temp),2,1);
        //printf("%lf,%lf\n",output[0][0],output[1][0]);
        fprintf(fptr, "%lf,%lf\n", output[0][0], output[1][0]);
        freeMat(output,2);
    }}


void yparabola_gen(FILE *fptr, double a, double num_points,
    double **vertex){
    for(int i=num_points;i>=0;i--){
        double t = 3*i/num_points;
        double **output=createMat(2,1);
        output[1][0]=vertex[0][0]+a*t*t;
        output[0][0]=vertex[1][0]+2*a*t;
        fprintf(fptr,"%lf,%lf\n",output[0][0],output[1][0]);
        freeMat(output,2);
    }
```

Code V

```
    for(int i=0;i<=num_points;i++){
        double t = -3*i/num_points;
        double **output=createMat(2,1);
        output[1][0]=a*t*t;
        output[0][0]=2*a*t;
        fprintf(fptr,"%lf,%lf\n",output[0][0],output[1][0]);
        freeMat(output,2);
    }

}

int main() {
    double x1, y1,x2,y2;
// scanf("%lf %lf %lf %lf %lf %lf", &x1, &y1, &x2, &y2, &x3, &y3)
    ;
    x1 = 0; y1 = 0; x2=8;y2=15;
    int m = 2, n = 1;
    double **vertex = createMat(m, n);
```

Code VI

```
    double **a1= createMat(m,n);
    double **a2=createMat(m,n);
    double **a01 = createMat(m,n);
    double **a02 = createMat(m,n);
    vertex[0][0] = x1;
    double** V;
    V[0][0] = 1; V[0][1] = 0;
    V[1][0] = 0; V[1][1] = 0;
    double** u;
    u[0][0] = 0;
    u[1][0] = 2;
    double f = 0;
    double** h1 = createMat(2,1); double** h2 = createMat(2,1);
    h1[0][0] = 0; h1[1][0] = 2;
    h2[0][0] = 0; h2[1][0] = 4;
    double** m1 = createMat(2,1);
    m1[0][0] = 1; m1[1][0] = 0;
    double *kap1; double* kap2;
```

Code VII

```
    kap1 = kapG(V, u, f, h1, m1);
    kap2 = kapG(V, u, f, h2, m1);
    double*** in1 = inter(m1, h1, kap1);
    double*** in2 = inter(m1, h2, kap2);

    for(int i=0;i<2;i++){
        if(**in1[i]>0) a1 = in1[i];
        if(**in2[i]>0) a2 = in2[i];
    }
    vertex[1][0] = y1;
    // a1[0][0]=2*sqrt(2);a1[1][0]=2;
    // a2[0][0]=4;a2[1][0]=4;
    a01[0][0] = 0; a01[1][0] = 2;
    a02[0][0] = 0; a02[1][0] = 4;

    double radius = 4;
    FILE *fptr;
    fptr = fopen("line_points.txt", "w");
```

Code VIII

```
    if (fptr == NULL) {
        printf("Error␣opening␣file!\n");
        return 1;
    }
    double a = 1;
    yparabola_gen(fptr, a, 1000,vertex);
    point_gen(fptr,a01,a1,11);
    point_gen(fptr,a02,a2,11);
    fclose(fptr);
    return 0;
}
```

Code for plotting the graph of the curve and area using Python:

Code IX

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

# Load the points from the text file
points = np.loadtxt("line_points.txt", delimiter=',', max_rows=
    len(list(open("./line_points.txt"))))
# Extract the x and y coordinates
centre=np.array([points[0][0],points[0][1]])
x1 = points[:-24, 0]
y1 = points[:-24, 1]
x2 = points[-24:-12,0]
y2 = points[-24:-12,1]
x3 = points[-12:,0]
y3 = points[-12:,1]
plt.figure()
```

Code X

```
x5 = [0 for i in range(12)]
plt.plot(x1, y1, label=’$x^2=4y$’, color=’blue’)
plt.plot(x2, y2, label=’’, color=’red’)
plt.plot(x3, y3, label=’’, color=’red’)
#plt.plot(x5, y3, label=’’, color=’black’)
where_l = []
for i in range(2002):
    if y1[i]<=4 and y1[i]>=2 and x1[i]>=0:
        where_l.append(True)
    else: where_l.append(False)
where1 = []
where2 = []
for i in range(2002):
    if x1[i]>=0 and x1[i]<=2*math.sqrt(2):
        where1.append(True)
    else: where1.append(False)

for i in range(2002):
```

Code XI

```
    if x1[i]>=0 and x1[i]<=4:
        where2.append(True)
    else: where2.append(False)
plt.fill_betweenx(y1, x1, 0, where=where_l, color='green', alpha
    =0.5, label='Required␣Area')
plt.fill_between(x1, y1, 0, where=(where1), color='orange', alpha
    =0.4, label="$A_2$")
plt.fill_between(x1, y1, 0, where=(where2), color='pink', alpha
    =0.4, label="$A_1$")

#plt.fill_between(y1, x1, x2, where=(x2*x2 >= x1*x1 and y1<=8),
    color='orange', alpha=0.5)
#plt.fill_between(x1, y1, y2, where=(y2 >= y1), color='lightblue
    ', alpha=0.5)
plt.gca().set_aspect('equal', adjustable='box')
#vector_max = max(np.abs(directionvector).max(), np.abs(
    normalvector).max())
#plt.xlim(-vector_max * 3, vector_max * 3)
```

Code XII

```
#plt.ylim(-vector_max * 3, vector_max * 3)
#plt.xlim(-1,0.5)
plt.axhline(0, color=’black’,linewidth=1) # x-axis
plt.axvline(0, color=’black’,linewidth=1) # x-axis
#plt.ylim(-15,15)
plt.xlabel("x")
plt.ylabel("y")
plt.title("␣")
plt.grid(True)
plt.legend()
#plt.legend()
plt.savefig("../figs/fig1.png")
plt.show()
```

The codes in:

Code XIII

```
https://github.com/me-coder-1204/EE1030/blob/main/Assignment5/codes/plot.py
https://github.com/me-coder-1204/EE1030/blob/main/Assignment5/codes/points.c
```