Лабораторная работа №2

Назаров Рустам M3232

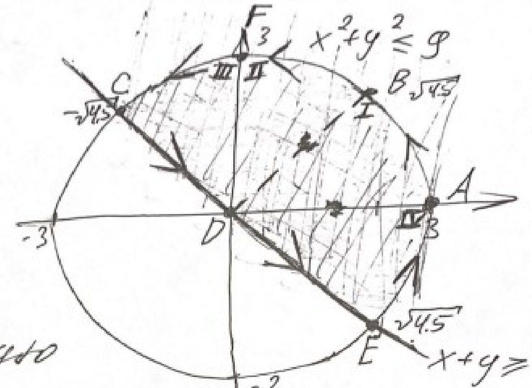
Вариант 18

Аналитический метод

Назаров М3232
Рустам

Лабораторная работа №2
Вариант 18
Аналитическая часть

$$\oint\limits_L x^2y\,dx+y^2x\,dy$$



$L: D: x^2+y^2\le 9$
$x+y\ge 0$

1) Криволинейный напрямую

Разделю замкнутую кривую на 4 сектора, равных (I, II, III, IV)

Решу каждый ~~напрямую~~ Напрямую

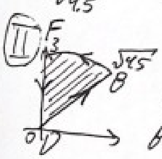
I. $$\oint\limits_{ABD} x^2y\,dx+y^2x\,dy=\int\limits_{AB}x^2y\,dx+y^2x\,dy+\int\limits_{BD}x^2y\,dx+y^2x\,dy+\int\limits_{DA}x^2y\,dx+y^2x\,dy =$$

AB: $y=\sqrt{9-x^2}$, $dy=\frac{-x}{\sqrt{9-x^2}}dx$, $x:[3;\sqrt{4.5}]$

BD: $y=x$, $dy=dx$, $x:[\sqrt{4.5};0]$

DA: $y=0$, $dy=0$, $x:[0;3]$

$$= -\int\limits_{\sqrt{4.5}}^{3} \underbrace{x^2\sqrt{9-x^2}\,dx+(9-x^2)x\cdot\frac{-x}{\sqrt{9-x^2}}dx}_{0} - \int\limits_0^{\sqrt{4.5}} x^3dx+x^2dx+\int\limits_0^3 0 = -\int\limits_0^{\sqrt{4.5}}2x^3dx=\boxed{-\frac{81}{8}}$$

II. 

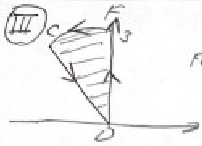
$$\oint\limits_{BFD} x^2y\,dx+y^2x\,dy=\int\limits_{BF}f+\int\limits_{FD}f+\int\limits_{DB}f=-\int\limits_0^{\sqrt{4.5}}\underbrace{x^2\sqrt{9-x^2}\,dx+x^2\sqrt{9-x^2}\,dx}_{0}-\int\limits_0^3 0+\int\limits_0^{\sqrt{4.5}}2x^3=$$

BF: $y=\sqrt{9-x^2}$, $dy=\frac{-x}{\sqrt{9-x^2}}dx$, $x:[\sqrt{4.5};0]$

FD: $x=0$, $dx=0$, $y[3;0]$

DB: $y=x$, $dy=dx$, $x:[0;\sqrt{4.5}]$

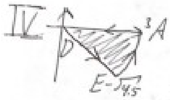
$$=\boxed{\frac{81}{8}}$$

III 

$$\oint\limits_{FCD} x^2y\,dx+y^2x\,dy=\int\limits_{FC} f+\int\limits_{CD} f+\int\limits_{DF} f =$$

$FC: y=\sqrt{9-x^2}$, $dy=\frac{-x}{\sqrt{9-x^2}}dx$, $x:[0;-\sqrt{4.5}]$; $CD: y=-x$, $dy=-dx$, $x:[-\sqrt{4.5};0]$; $DF: x=0$, $dx=0$, $y:[0;3]$

$$= -\int_{-\sqrt{4.5}}^{0}(x^2\sqrt{9-x^2}-x^2\sqrt{9-x^2})dx+\int_{-\sqrt{4.5}}^{0}-2x^3dx+\int_0^3 0=\boxed{-\frac{81}{8}}$$

IV 

$$\oint\limits_{ADE} x^2y\,dx+y^2x\,dy=\int\limits_{AD} f+\int\limits_{DE} f+\int\limits_{EA} f =$$

$AD: y=0$, $dy=0$, $x:[3;0]$; $DE: y=-x$, $dy=-dx$, $x:[0;\sqrt{4.5}]$; $EA: y=-\sqrt{9-x^2}$, $dy=\frac{x}{\sqrt{9-x^2}}dx$, $x:[-\sqrt{4.5};3]$

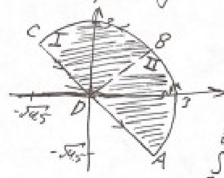
$$= -\int_0^3 0+\int_0^{\sqrt{4.5}}-2x^3dx+\int_{-\sqrt{4.5}}^{3}\underbrace{-x^2\sqrt{9-x^2}+(9-x^2)x\cdot\frac{x}{\sqrt{9-x^2}}}_{0}dx=\boxed{-\frac{81}{8}}$$

$$\oint\limits_{L} x^2y\,dx+y^2x\,dy=\oint\limits_{ABD} f+\oint\limits_{BFD} f+\oint\limits_{FCD} f+\oint\limits_{ADE} f=\boxed{\frac{81}{2}}$$

Ответ: $\frac{81}{2}$

2) Решение криволинейного интеграла по Грину

$$\oint\limits_{L} \underbrace{x^2y}_{P}dx+\underbrace{y^2x}_{Q}dy=\iint\limits_{L}(\underbrace{y^2}_{\frac{\partial Q}{\partial x}}-\underbrace{x^2}_{\frac{\partial P}{\partial y}})dxdy=$$



Разобью на 2 сектора

$$=\iint\limits_{BCO} f+\iint\limits_{ABD} f=\boxed{\frac{81}{2}}$$

$\int\limits_{|x|}^{\sqrt{9-x^2}}$ 

$\int\limits_{|y|}^{\sqrt{9-y^2}}$ 

$$\iint\limits_{BCO} f=\int_{-\sqrt{4.5}}^{\sqrt{4.5}}dx\int_{|x|}^{\sqrt{9-x^2}}(y^2-x^2)dy=\int_{-\sqrt{4.5}}^{\sqrt{4.5}}\left(\frac{2x^2|x|}{3}-\frac{\sqrt{9-x^2}(4x^2-9)}{3}\right)dx=\frac{81}{4}$$

$$\iint\limits_{ABD}=\int_{-\sqrt{4.5}}^{\sqrt{4.5}}dy\int_{|y|}^{\sqrt{9-y^2}}(y^2-x^2)dx=\int_{-\sqrt{4.5}}^{\sqrt{4.5}}\left(\frac{\sqrt{9-y^2}(4y^2-9)}{3}-\frac{2y^2|y|}{3}\right)dy=-\frac{81}{4}$$

Ответ: $\frac{81}{2}$

Ответы сошлись! Это доказывает правильность нашего решения и правильность способов решения т.к. площадь не может поменяться

$$\boxed{\frac{81}{2}=\frac{81}{2}}$$

Численный метод

язык: Python 3.8

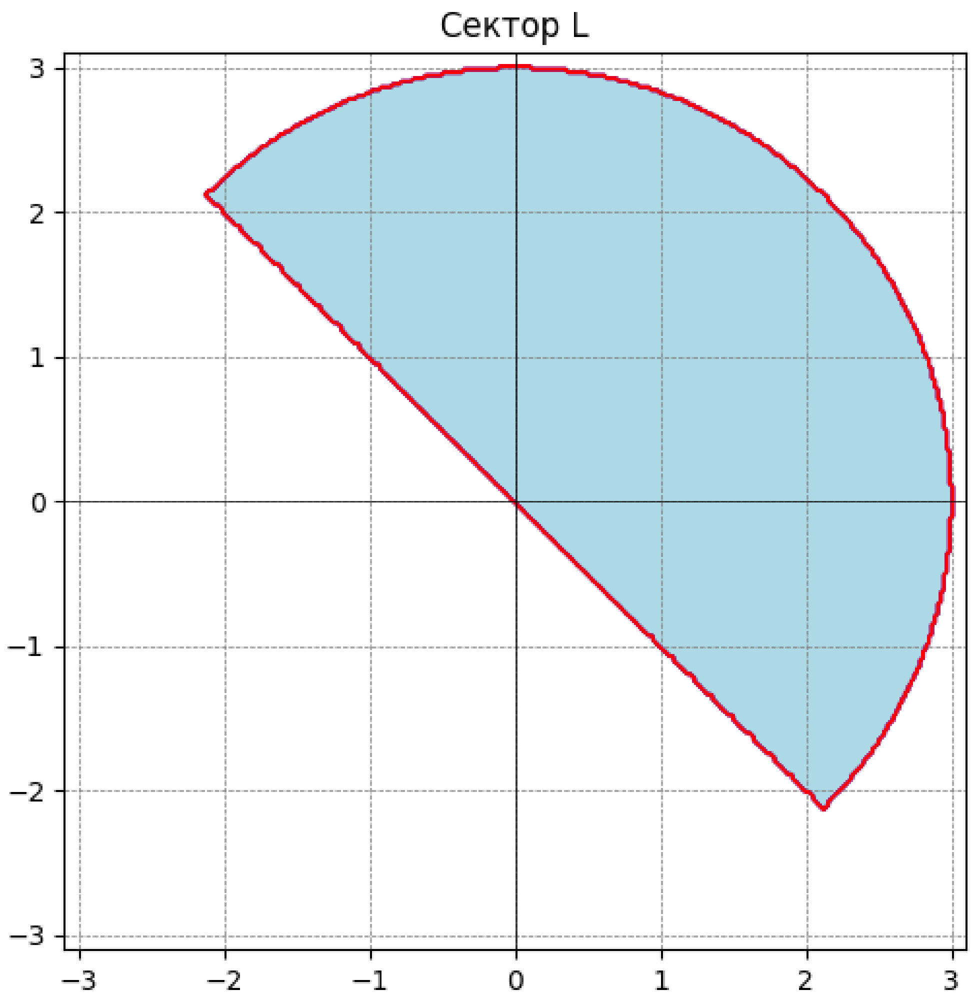
```
#Импорты
import sympy as sp
import numpy as np
import matplotlib.pyplot as plt
```

```python
import time
import matplotlib.lines as mlines
from scipy.integrate import quad
import pandas as pd

ANSWER = 81/2 # Ответ из аналитики с которым хотим совпасть
print(ANSWER)
```

```
40.5
```

Функция рисования графика

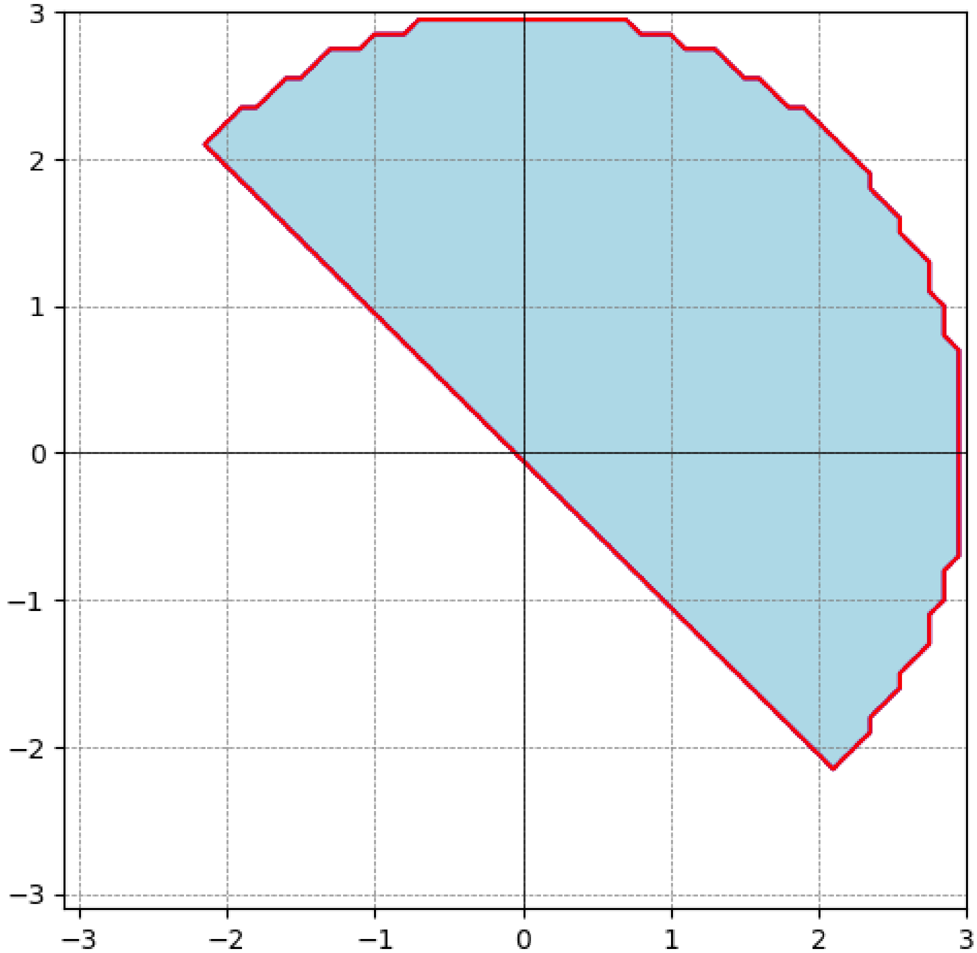
```python
def f(x, y, str):
    x, y = np.meshgrid(x, y)

    # Определяем условия
    condition1 = x**2 + y**2 <= 9
    condition2 = x + y >= 0

    # Объединяем условия
    condition = np.logical_and(condition1, condition2)

    # Рисуем область, удовлетворяющую условиям
    plt.figure(figsize=(6,6))
    plt.contourf(x, y, condition, colors='lightblue', levels=[0.5, 1])

    # Рисуем границы областей только внутри области, удовлетворяющей
обоим условиям
    plt.contour(x, y, np.logical_and(condition1, condition),
colors='blue', levels=[0.5])
    plt.contour(x, y, np.logical_and(condition2, condition),
colors='red', levels=[0.5])

    # Рисуем оси координат
    plt.axhline(0, color='black',linewidth=0.5)
    plt.axvline(0, color='black',linewidth=0.5)
    plt.grid(color = 'gray', linestyle = '--', linewidth = 0.5)
    plt.title(str)

    plt.show()

# Создаем сетку точек
x = np.linspace(-3.1, 3.1, 305)
y = np.linspace(-3.1, 3.1, 305)
f(x, y, "Сектор L")
```

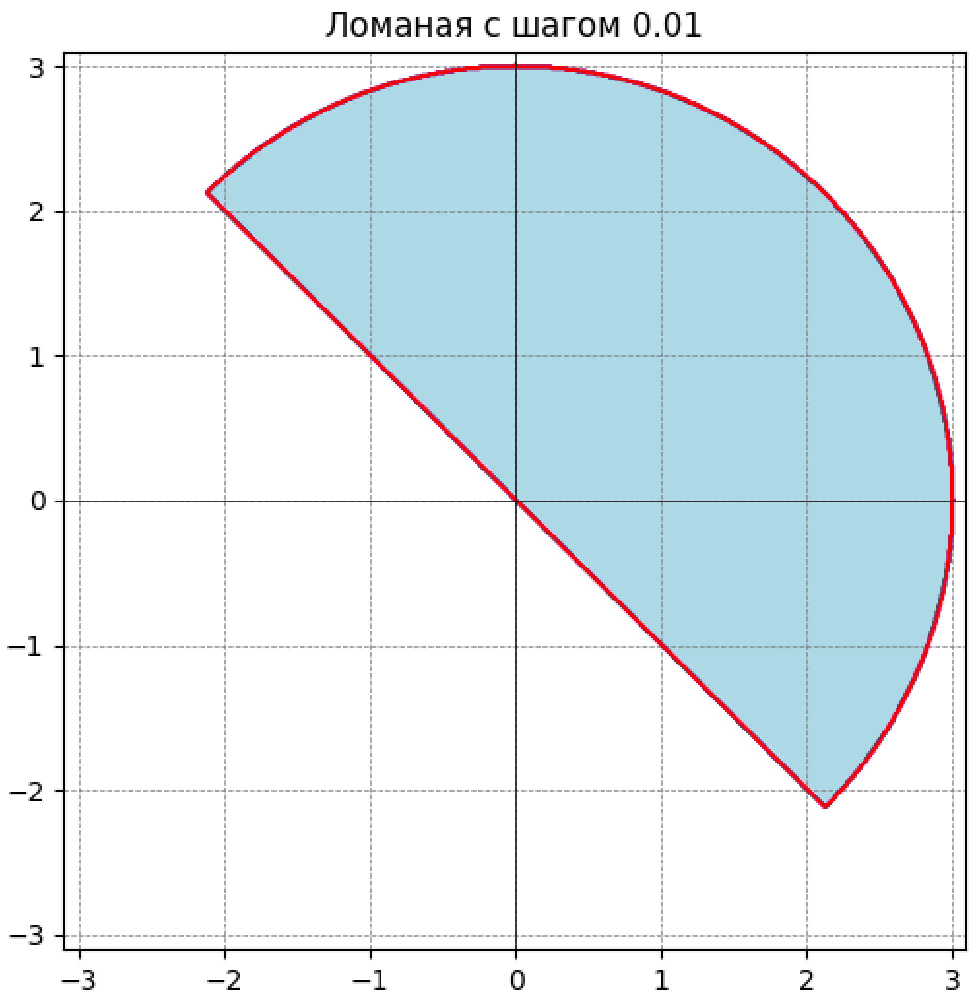


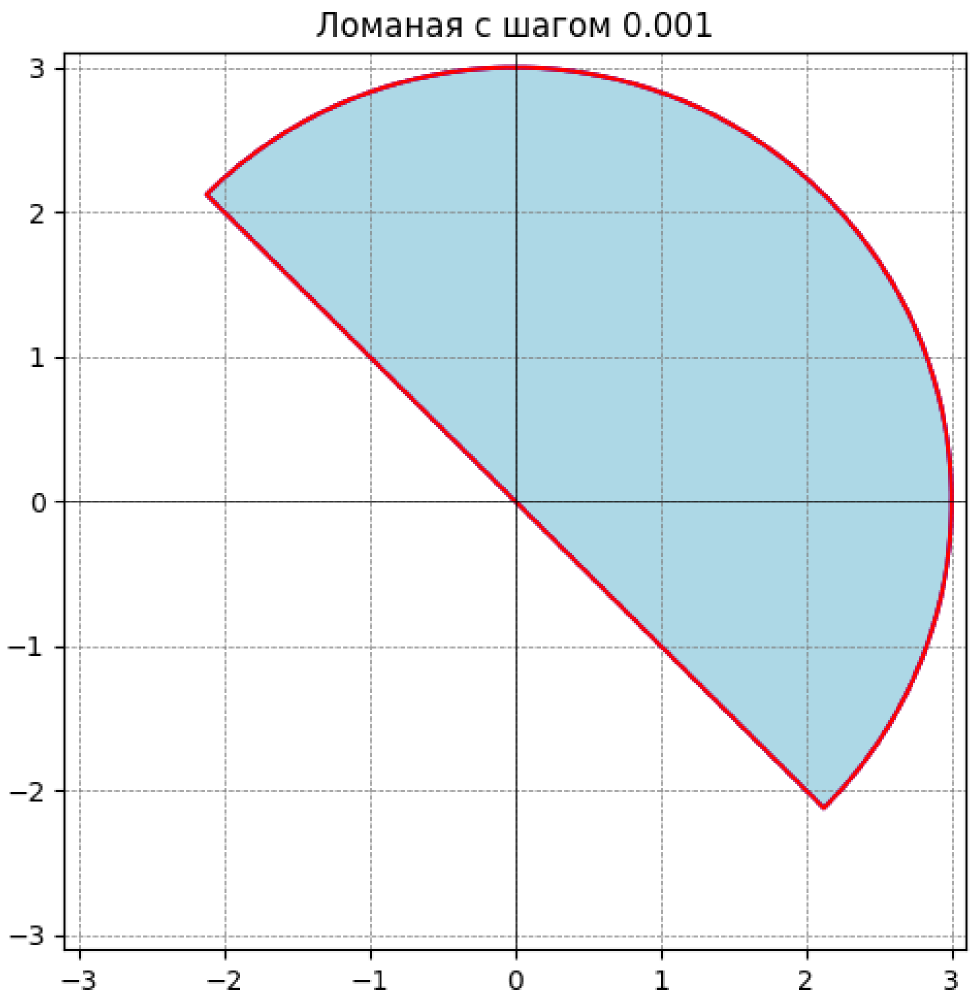
2.1

```
# Создаем сетку точек
x = np.arange(-3.1, 3.1, 0.1)
y = np.arange(-3.1, 3.1, 0.1)
f(x, y, "Ломаная с шагом 0.1")
x = np.arange(-3.1, 3.1, 0.01)
y = np.arange(-3.1, 3.1, 0.01)
f(x, y, "Ломаная с шагом 0.01")
x = np.arange(-3.1, 3.1, 0.001)
y = np.arange(-3.1, 3.1, 0.001)
f(x, y, "Ломаная с шагом 0.001")
```

Ломаная с шагом 0.1




Ломаная с шагом 0.01
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3




```
deltas = [0.1, 0.01, 0.001, 0.0001, 0.00001]

import numpy as np

# функции, ограничивающие область интегрирования
def curve1(t):
    return t, np.sqrt(9 - t**2)  # верхняя часть окружности

def curve2(t):
    return t, -np.sqrt(9 - t**2)  # нижняя часть окружности

def curve3(t):
    return t, -t  # линия y = -x

def curve4(t):
    return t, t  # линия y = x

# функции для дифференциальной формы
def f1(x, y):
```

```
    return x**2 * y

def f2(x, y):
    return y**2 * x

def count(a, a1, a2, a3, b, b1, b2, b3, c, c1, c2, c3):
    # Вычисляю интегральные суммы для каждого значения дельта и по
заданным функциям и границам
    total_integral_sum = np.array([])
    times = []
    for delta in deltas:
        sta = time.time()
        t1 = np.linspace(a1, a2, int((a2-a1)/delta))
        t2 = np.linspace(b1, b2, int((b2-b1)/delta))
        t3 = np.linspace(c1, c2, int((c2-c1)/delta))
        x1, y1 = a(t1)
        x2, y2 = b(t2)
        x3, y3 = c(t3)
        integral_sum1 = a3 * (np.sum(f1(x1[:-1], y1[:-1]) *
np.diff(x1)) + np.sum(f2(x1[:-1], y1[:-1]) * np.diff(y1)))
        integral_sum2 = b3 * (np.sum(f1(x2[:-1], y2[:-1]) *
np.diff(x2)) + np.sum(f2(x2[:-1], y2[:-1]) * np.diff(y2)))
        integral_sum3 = c3 * (np.sum(f1(x3[:-1], y3[:-1]) *
np.diff(x3)) + np.sum(f2(x3[:-1], y3[:-1]) * np.diff(y3)))
        total_integral_sum = np.append(total_integral_sum,
np.abs(integral_sum1 + integral_sum2 + integral_sum3))
        fn = time.time()
        ti = fn-sta
        times.append(ti)
    return total_integral_sum, times

a, t1 = count(curve1, np.sqrt(4.5), 3, -1, curve2, 0, 0, 0, curve4, 0,
np.sqrt(4.5), -1) # 1 сектор из аналитики
b, t2 = count(curve1, 0, 0, 0, curve2, np.sqrt(4.5), 3, 1, curve3, 0,
np.sqrt(4.5), 1) # 2 сектор из аналитики
c, t3 = count(curve1, 0, np.sqrt(4.5), -1, curve2, 0, 0, 0, curve4, 0,
np.sqrt(4.5), 1) # 3 сектор из аналитики
d, t4 = count(curve1, -3, -np.sqrt(4.5), -1, curve2, 0, 0, 0, curve3,
-np.sqrt(4.5), 0, -1) # 4 сектор из аналитики
e = a + b + c + d
data = {
    "Дельта": deltas,
    "Подсчет": [e[0], e[1], e[2], e[3], e[4]],
    "Отклонение": [ANSWER-e[0], ANSWER-e[1], ANSWER-e[2], ANSWER-e[3],
ANSWER-e[4]],
    "Время": [t1[0] + t2[0] + t3[0] + t4[0],
              t1[1] + t2[1] + t3[1] + t4[1],
              t1[2] + t2[2] + t3[2] + t4[2],
              t1[3] + t2[3] + t3[3] + t4[3],
              t1[4] + t2[4] + t3[4] + t4[4]]
```

```
}

df = pd.DataFrame(data)

print(df)
```

```
    Дельта    Подсчет  Отклонение     Время
0  0.10000  36.271550    4.228450  0.000000
1  0.01000  40.210369    0.289631  0.000996
2  0.00100  40.474006    0.025994  0.000986
3  0.00010  40.497490    0.002510  0.003371
4  0.00001  40.499752    0.000248  0.045066
```

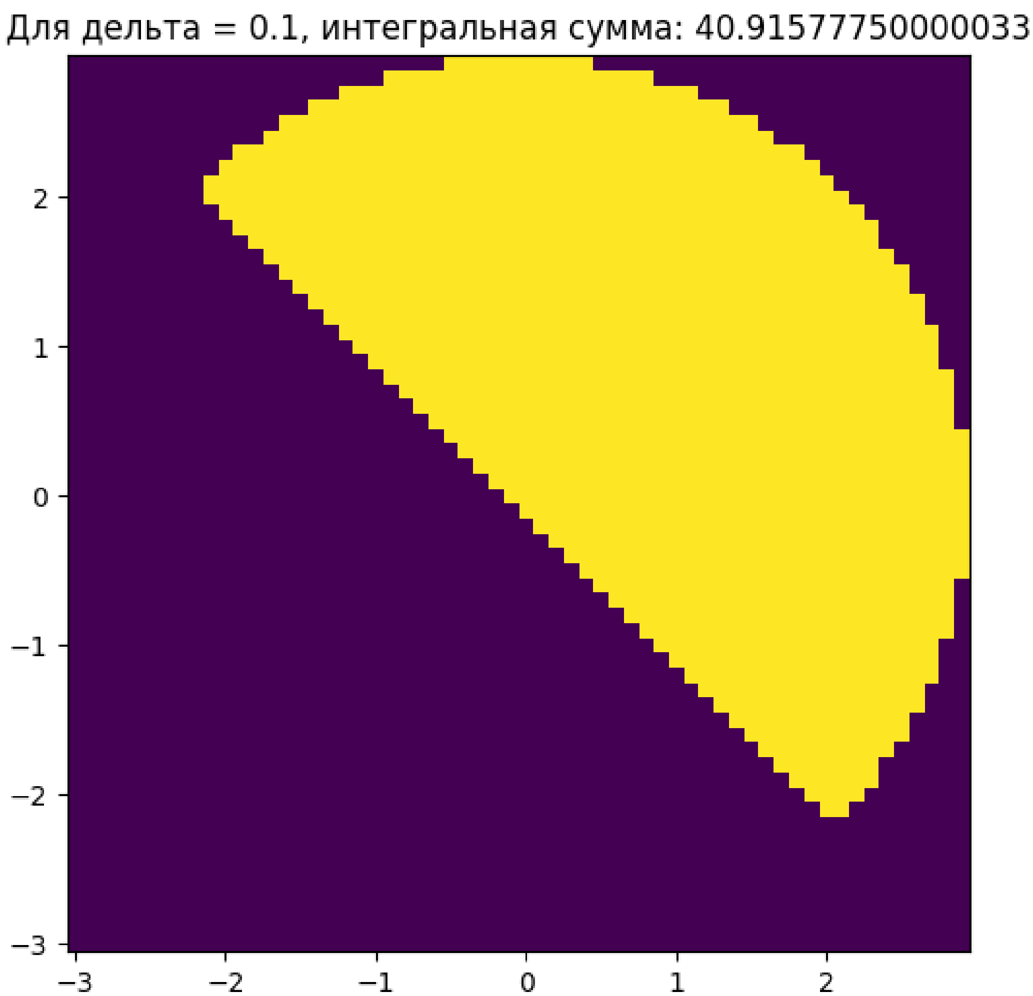
^ОТВЕТ НА 2.1^

2.2

```
deltas = [0.1, 0.01, 0.001]

# функции для дифференциальной формы
def f(x, y):
    return x**2 * y + y**2 * x

# функции, ограничивающие область интегрирования
def inside_curve(x, y):
    return np.logical_and(x**2 + y**2 <= 9, y >= -x)

sums = []
times = []
for delta in deltas:
    sta = time.time()
    x = np.arange(-3, 3, delta)
    y = np.arange(-3, 3, delta)
    X, Y = np.meshgrid(x, y)
    Z = np.abs(f(X + delta/2, Y + delta/2)) # =значение функции в 
центрах ячеек
    mask = inside_curve(X + delta/2, Y + delta/2)  # =какие центры 
ячеек находятся внутри замкнутой кривой
    integral_sum = np.sum(Z[mask]) * delta**2  # интегральная сумма
    integral_sum -= 1
    fn = time.time()

    sums.append(integral_sum)
    times.append(fn-sta)
    # график
    print(f'Для дельта = {delta}, интегральная сумма: {integral_sum}')
    plt.figure(figsize=(6, 6))
    plt.pcolormesh(X, Y, mask, cmap='viridis')
    plt.title(f'Для дельта = {delta}, интегральная сумма: 
```

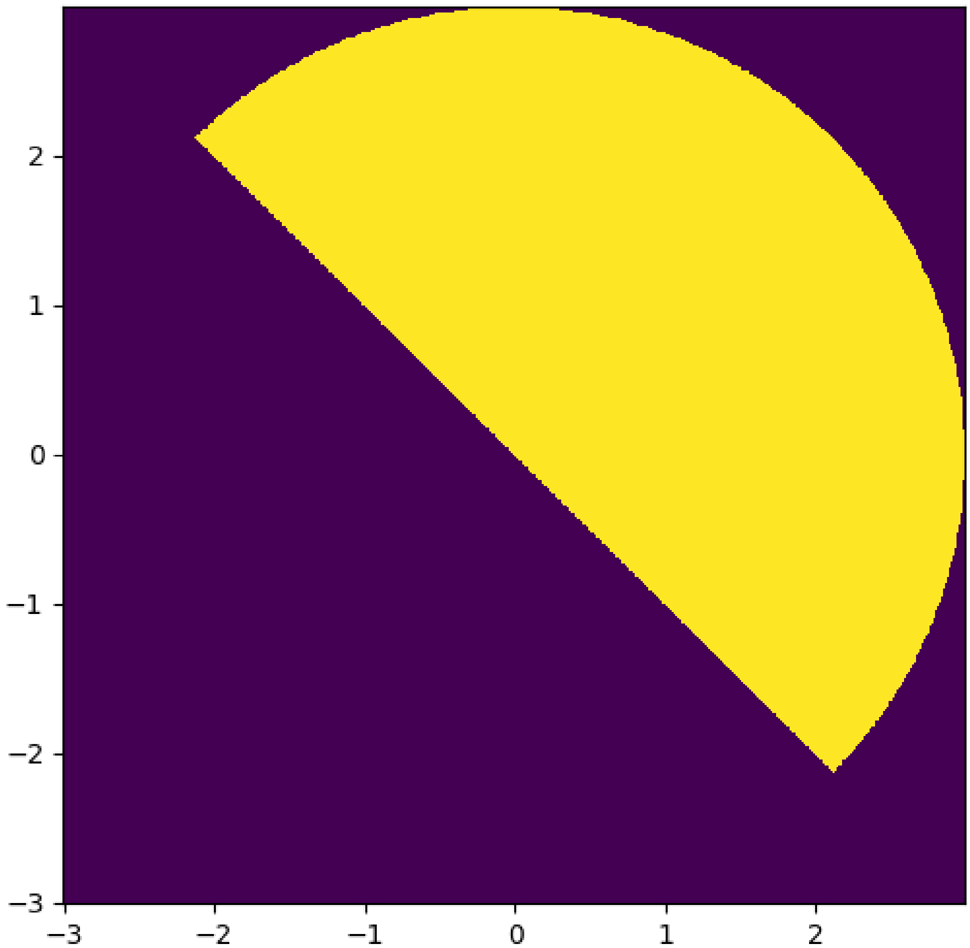
```
{integral_sum}')
    plt.show()
data = {
    "Дельта": deltas,
    "Подсчет": [sums[0], sums[1], sums[2]],
    "Отклонение": [ANSWER-sums[0], ANSWER-sums[1], ANSWER-sums[2]],
    "Время": [times[0], times[1], times[2]]
}

# Создайте DataFrame
df = pd.DataFrame(data)

# Выведите DataFrame
print(df)
```

```
Для дельта = 0.1, интегральная сумма: 40.91577750000033
```

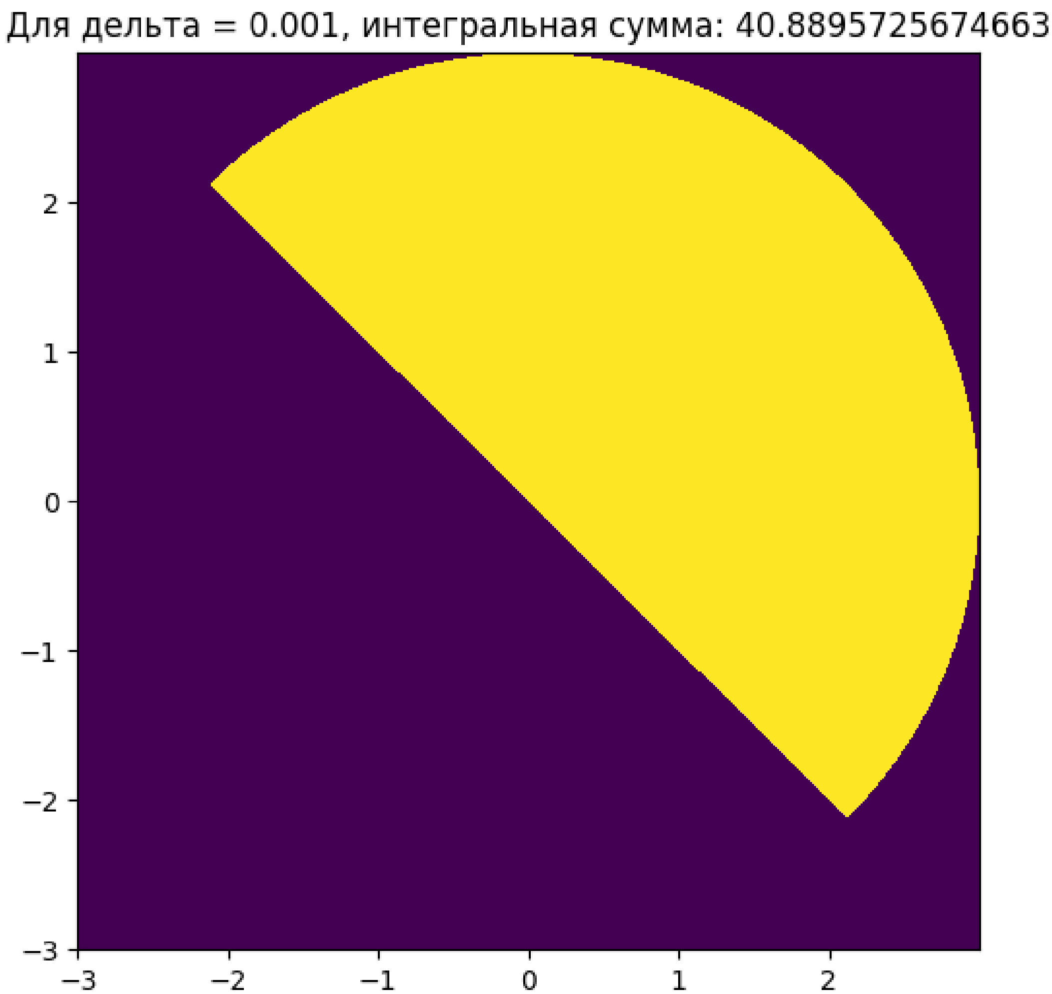



```
Для дельта = 0.01, интегральная сумма: 40.90684685974215
```

Для дельта = 0.01, интегральная сумма: 40.90684685974215




```
Для дельта = 0.001, интегральная сумма: 40.8895725674663
```



	Дельта	Подсчет	Отклонение	Время
0	0.100	40.915778	-0.415778	0.071840
1	0.010	40.906847	-0.406847	0.023034
2	0.001	40.889573	-0.389573	2.479321

^ОТВЕТ НА 2.2^

Вывод:

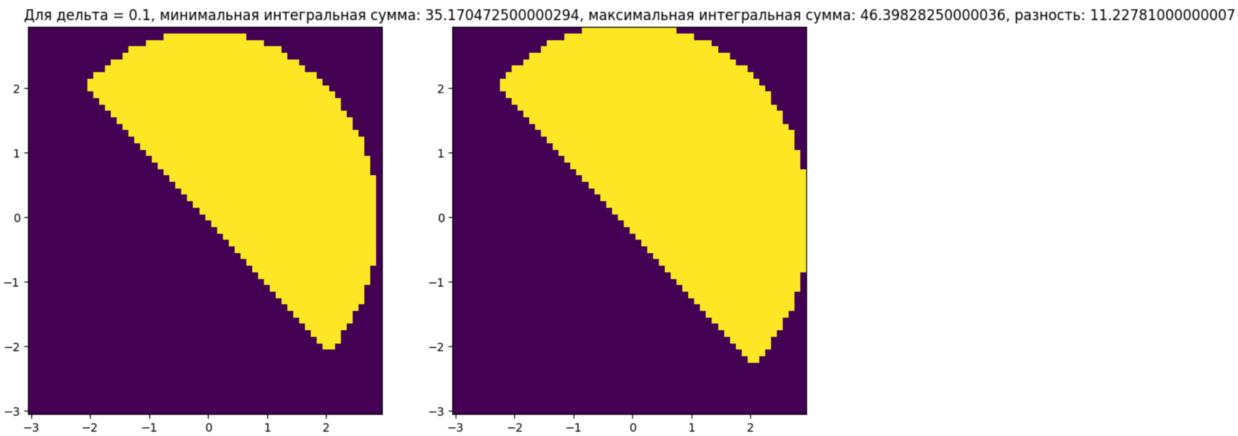
Итак, 2.1 пункт стремится к 40.5 снизу, так как соей ломаной мы очевидно уменьшаем площадь сектора 2.2 пункт стремится к 40.5 сверху, так как, проверяя центры квадратов, куски торчат и увеличивают площадь сектора

Но, как и ожидалось, обе функции стремятся к ответу из Аналитики. Это доказывает правильность вычислений и примененных методов.

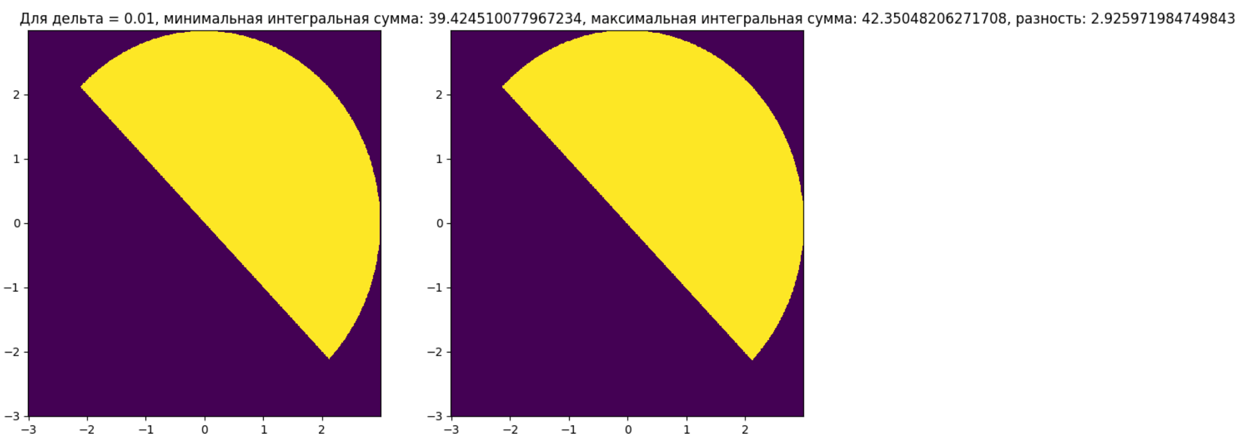
```
# функции для дифференциальной формы
def f(x, y):
```

```
    return x**2 * y + y**2 * x

# функции, ограничивающие область интегрирования
def inside_curve(x, y):
    return np.logical_and(x**2 + y**2 <= 9, y >= -x)

def on_border(x, y, delta):
    return inside_curve(x, y) | inside_curve(x + delta, y + delta) |
inside_curve(x + delta, y) | inside_curve(x, y + delta)

def in_border(x, y, delta):
    return inside_curve(x, y) & inside_curve(x + delta, y + delta) &
inside_curve(x + delta, y) & inside_curve(x, y + delta)

# интегральные суммы для каждого значения дельта
deltas = [0.1, 0.01, 0.001]
for delta in deltas:
    x = np.arange(-3, 3, delta)
    y = np.arange(-3, 3, delta)
    X, Y = np.meshgrid(x, y)
    Z = np.abs(f(X + delta/2, Y + delta/2))  # значение функции в
центрах ячеек
    mask_inside = in_border(X, Y, delta)  # какие центры ячеек
находятся внутри замкнутой кривой
    mask_on_border = on_border(X, Y, delta)  # какие ячейки
пересекаются с D
    integral_sum_min = np.sum(Z[mask_inside]) * delta**2  #
минимальную интегральную сумму
    integral_sum_min -= 2
    integral_sum_max = np.sum(Z[mask_on_border]) * delta**2  #
максимальную интегральную сумму
    print(f'Для дельта = {delta}, минимальная интегральная сумма:
{integral_sum_min}, максимальная интегральная сумма:
{integral_sum_max}, разность: {integral_sum_max - integral_sum_min}')

    # Постройте график
    plt.figure(figsize=(12, 6))
    plt.subplot(1, 2, 1)
    plt.pcolormesh(X, Y, mask_inside, cmap='viridis')
    plt.subplot(1, 2, 2)
    plt.pcolormesh(X, Y, mask_on_border, cmap='viridis')
    plt.title(f'Для дельта = {delta}, минимальная интегральная сумма:
{integral_sum_min}, максимальная интегральная сумма:
{integral_sum_max}, разность: {integral_sum_max - integral_sum_min}')
    plt.show()
```

```
Для дельта = 0.1, минимальная интегральная сумма: 35.170472500000294,
максимальная интегральная сумма: 46.39828250000036, разность:
11.22781000000007
```




```
Для дельта = 0.01, минимальная интегральная сумма: 39.424510077967234,
максимальная интегральная сумма: 42.35048206271708, разность:
2.925971984749843
```




```
Для дельта = 0.001, минимальная интегральная сумма:
39.843414757638016, максимальная интегральная сумма: 41.9359790838814,
разность: 2.0925643262433837
```