Лабораторная работа №1

Назаров Рустам

М3232 368563

Вариант: 18

Аналитический метод

Лабораторная работа №1

Назаров Рустам

М3232 368563

Вариант 18

Аналитическая часть

ООФ: $\begin{cases} x\neq 0 \\ y\neq 0 \end{cases}$ не могут одновременно

$z=f(x,y)$

$f(x,y)=x^2y^2\ln(x^2+y^2)$

1) Стационарные точки

$\frac{dz}{dx}=2xy^2\ln(x^2+y^2)+\frac{2y^2x^3}{x^2+y^2}$

$\frac{dz}{dy}=2yx^2\ln(x^2+y^2)+\frac{2x^2y^3}{x^2+y^2}$

$\begin{cases} 2xy^2\ln(x^2+y^2)+\frac{2y^2x^3}{x^2+y^2}=0 \ \Big| \cdot(x^2+y^2) \\ 2yx^2\ln(x^2+y^2)+\frac{2x^2y^3}{x^2+y^2}=0 \ \Big| \cdot(x^2+y^2) \end{cases}$

$\begin{cases} 2y^2x^3\ln(x^2+y^2)+2xy^4\ln(x^2+y^2)+2y^2x^3=0 \\ 2x^2y^3\ln(x^2+y^2)+2yx^4\ln(x^2+y^2)+2x^2y^3=0 \end{cases}$

$\begin{cases} 2y^2x(x^2\ln(x^2+y^2)+y^2\ln(x^2+y^2)+x^2)=0 \\ 2x^2y(x^2\ln(x^2+y^2)+y^2\ln(x^2+y^2)+y^2)=0 \end{cases}$ $\begin{matrix} \to \underline{x=0}, y\neq 0 \text{ (ООФ)} \\ \to \underline{y=0}, x\neq 0 \text{ (ООФ)} \end{matrix}$

$\begin{cases} (x^2+y^2)\ln(x^2+y^2)+x^2=0 \\ (x^2+y^2)\ln(x^2+y^2)+y^2=0 \end{cases}$ $-$

$(x^2+y^2)\ln(x^2+y^2)+x^2-(x^2+y^2)\ln(x^2+y^2)-y^2=0$

$x^2-y^2=0, \quad x^2=y^2 \quad \left[\begin{matrix} y=x \\ y=-x \end{matrix}\right.$

$\angle\ y=x \Rightarrow 2x^3\ln(2x^2)+x^3=0; \ x^3(2\ln(2x^2)+1)=0, \ \left[\begin{matrix} x=0\ (y\neq 0) \\ 2x^2=e^{-1/2}, \ \underline{x=\pm\sqrt{\frac{1}{2\sqrt{e}}}} \end{matrix}\right.$

Итог:

$A=\left(\sqrt{\frac{1}{2\sqrt{e}}};\sqrt{\frac{1}{2\sqrt{e}}}\right)$

$B=\left(\sqrt{\frac{1}{2\sqrt{e}}};-\sqrt{\frac{1}{2\sqrt{e}}}\right)$

$C=\left(-\sqrt{\frac{1}{2\sqrt{e}}};\sqrt{\frac{1}{2\sqrt{e}}}\right)$

$D=\left(-\sqrt{\frac{1}{2\sqrt{e}}};-\sqrt{\frac{1}{2\sqrt{e}}}\right)$

$E=(0,y)\ y\neq 0$

$F=(x,0)\ x=0$

2) Достаточное условие

$$f''_{xx}=2y^2\ln(x^2+y^2)+\frac{4y^2x^2}{x^2+y^2}+\frac{2y^2x^4+6x^2y^4}{(x^2+y^2)^2}=2y^2\ln(x^2+y^2)+\frac{6y^2x^4+10x^2y^4}{(x^2+y^2)^2}$$

$$f''_{yy}=2x^2\ln(x^2+y^2)+\frac{6x^2y^4+10y^2x^4}{(x^2+y^2)^2}$$

$$f''_{xy}=f''_{yx}=4yx\ln(x^2+y^2)+\frac{4y^3x}{x^2+y^2}+\frac{4yx^5\cancel{+4y^3x^3-4y^5x}}{(x^2+y^2)^2}=4yx\ln(x^2+y^2)+\frac{4xy(x^4+x^2y^2+y^4)}{(x^2+y^2)^2}=$$

$$=4yx\ln(x^2+y^2)+4xy-\frac{4x^3y^3}{(x^2+y^2)^2}=4xy\left(\ln(x^2+y^2)-\frac{x^2y^2}{(x^2+y^2)^2}+1\right)$$

$$A=\left(\sqrt{\frac{1}{2\sqrt{e}}};\sqrt{\frac{1}{2\sqrt{e}}}\right) \text{ и } D=\left(-\sqrt{\frac{1}{2\sqrt{e}}};-\sqrt{\frac{1}{2\sqrt{e}}}\right)$$

$$f''_{xx}(A)=2x^2\ln(2x^2)+4x^2=\frac{3}{2\sqrt{e}}$$

$$f''_{yy}(A)=\frac{3}{2\sqrt{e}}$$

$$f''_{xy}(A)=4x^2\left(\ln(2x^2)+\frac{3}{4}\right)=\frac{1}{2\sqrt{e}}$$

$$\begin{pmatrix}\frac{3}{2\sqrt{e}} & \frac{1}{2\sqrt{e}}\\ \frac{1}{2\sqrt{e}} & \frac{3}{2\sqrt{e}}\end{pmatrix}\begin{matrix}\Delta_1>0\\ \\ \Delta_2>0\end{matrix} \Rightarrow$$

A и D локальные min

$$B=\left(\sqrt{\frac{1}{2\sqrt{e}}};-\sqrt{\frac{1}{2\sqrt{e}}}\right) \text{ и } C=\left(-\sqrt{\frac{1}{2\sqrt{e}}};\sqrt{\frac{1}{2\sqrt{e}}}\right)$$

$$f''_{xx}(B)=\frac{3}{2\sqrt{e}} \quad f''_{yy}(B)=\frac{3}{2\sqrt{e}}$$

$$f''_{xy}(B)=-4x^2\left(\ln(2x^2)+\frac{3}{4}\right)=-\frac{1}{2\sqrt{e}}$$

$$\begin{pmatrix}\frac{3}{2\sqrt{e}} & -\frac{1}{2\sqrt{e}}\\ -\frac{1}{2\sqrt{e}} & \frac{3}{2\sqrt{e}}\end{pmatrix}\begin{matrix}\Delta_1>0\\ \\ \Delta_2>0\end{matrix}$$

B и C $\Rightarrow$ локальные min

$E=(0,y)\ y\neq 0$ $\quad f''_{xx}(E)=2y^2\ln y^2$

$f''_{yy}(E)=0$

$f''_{xy}(E)=0$

$$\begin{pmatrix}2y^2\ln y^2 & 0\\ 0 & 0\end{pmatrix}$$ вырожденная матрица

$$d^2f(E)=2y^2\ln y^2\,dx^2$$ – полуопределен!

Не выполнено достаточное условие для E

~~[illegible]~~

$F=(x,0)\ x\neq 0$ $\quad f''_{yy}(F)=2x^2\ln x^2$

$f''_{xx}(F)=0 \quad f''_{xy}(F)=0$

$$\begin{pmatrix}0 & 0\\ 0 & 2x^2\ln x^2\end{pmatrix}$$ вырожденная матрица

$$d^2f(F)=2x^2\ln x^2\,dy^2$$ – полу-определен.

Не выполнено достаточное условие для F

3) По определению

для E

$x_0 = E$

$\exists U(x_0)\ \forall x \in U(x_0) \cap E$

$f(x) \leq f(x_0)$

$(f(x) \geq f(x_0)$

$\exists\ x = \Delta x \qquad \Delta x; \Delta y \to 0$

$y = y + \Delta y \qquad y \to \infty$

$\Delta f = f(\Delta x, y + \Delta y) - f(0, y) =$

$\Delta f = \Delta x^2 (y + \Delta y)^2 \ln(\Delta x^2 + (y + \Delta y)^2)$

Так как $\ln(a) < 0,\ a < 1$

$\ln(a) > 0,\ a > 1$

и $\Delta x \to 0,\ \Delta y \to 0$, то

при $|y| < 1$ $\Delta f < 0$ $\Rightarrow$ E локальный max при $|y| < 1$

при $|y| > 1$ $\Delta f > 0$ $\Rightarrow$ E локальный min при $|y| > 1$

для F аналогично:

$(x, 0)\ x \neq 0$

$\exists\ x = x + \Delta x \qquad \Delta x, \Delta y \to 0$

$y = \Delta y \qquad x \to \infty$

$\Delta f = (\Delta x + x)^2 \Delta y^2 \ln((x + \Delta x)^2 + \Delta y^2)$

при $|x| < 1$ $\Delta f < 0$ $\Rightarrow$ F локальный max при $|x| < 1$

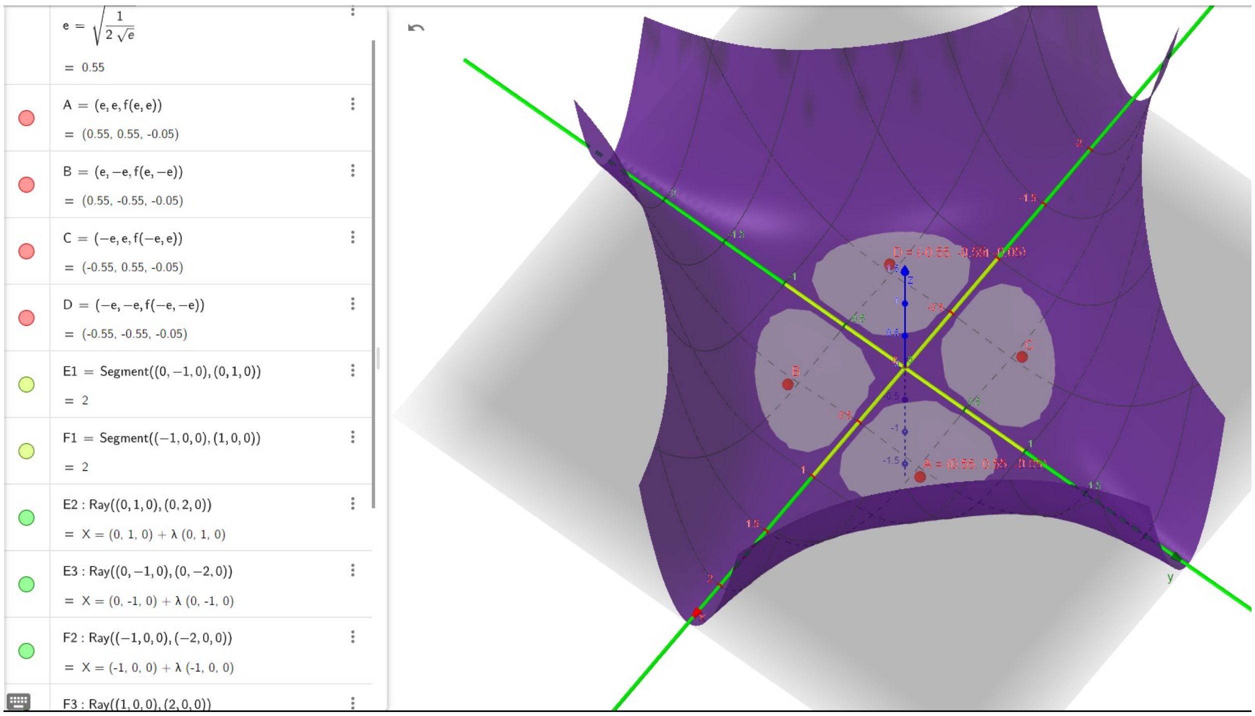
при $|x| > 1$ $\Delta f > 0$ $\Rightarrow$ F локальный min при $|x| > 1$

Численный метод

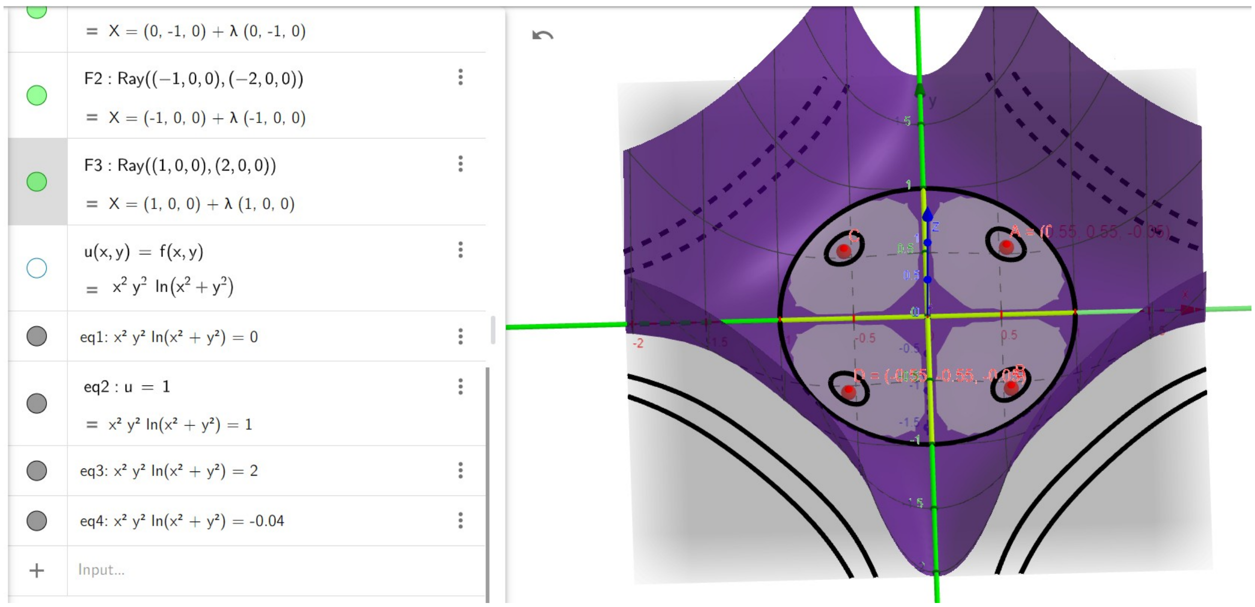
Отоборазим на функции наши Стационарные точки.

Красные: A, B, C, D

Желтые и Зеленые: E, F



Отобразим линии уровня. То есть приравняем f(x, y) = C. C={-0.4, 0, 1, 2}, так как f(x, y) не ниже ~-0.05



Видно как при уменьшении C круги приближаются к 4 точкам A, B, C, D. И как в C=0 это круг с радиусом 1, где наши E, F переходят из максимума в минимум

```
#Импорты
import sympy as sp
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import time
import matplotlib.lines as mlines

# Определяем переменные
x, y = sp.symbols('x y')

# Определяем функцию
f = x ** 2 * y ** 2 * sp.log(x ** 2 + y ** 2)
```

Условие останова: Если приращение функции меньше eps или приращение двух аргументов меньше beta, то останавливаем интерации. Так как, если разница между предыдущим значением и текущим очень мала, то мы подошли почти впритык к екстремуме и дальше итерироваться бессмысленно

```
epsilon = 0.000000001  # условие остановки на основе приращения функции
beta = 0.000000001  # условие остановки на основе приращения аргумента
```

Интерации по вычислению точек. Стремимся к точке А

```
# градиент
grad_f = sp.Matrix([f.diff(var) for var in (x, y)])

# Определяем начальную точку и скорость уменьшения шага
x_k, y_k = 0.1, 0.1
a_k = 0.3 # Постоянное

f_val_prev = f.subs({x: x_k, y: y_k}).evalf()  # Значение функции на начальном шаге

# Списки для хранения истории точек
x_points, y_points, z_points = [], [], []

start_time = time.time()

for i in range(200):  # Максимум 200 шагов
    # значение функции и градиент в текущей точке
    f_val = f.subs({x: x_k, y: y_k}).evalf()
    x_points.append(x_k)
    y_points.append(y_k)
    z_points.append(f_val)
    grad_val = grad_f.subs({x: x_k, y: y_k}).evalf()

    # приращение функции
    delta_f = f_val - f_val_prev

    # вывод каждые 12 раз
    if i % 12 == 0:
        print(f'{i + 1} Текущие точки: {x_k}, {y_k}')
        print(f'Значение функции: {f_val}')
        print(f'Приращение функции: {delta_f}')
        print(f'Градиент: {grad_val}')
        print()
```

```
    # следующая точка
    x_k_next, y_k_next = (sp.Matrix([x_k, y_k]) - a_k *
grad_val).evalf()

    # приращение аргумента
    delta_x = abs(x_k_next - x_k)
    delta_y = abs(y_k_next - y_k)

    # условия остановки
    if i > 0 and abs(delta_f) < epsilon or (delta_x < beta and delta_y
< beta):
        print(f'{i + 1} Текущие точки: {x_k}, {y_k}')
        print(f'Значение функции: {f_val}')
        print(f'Приращение функции: {delta_f}')
        print(f'Градиент: {grad_val}')
        print(f'\nИнтераций: {i + 1}')
        break

    # Обновляем текущие точки и значение функции
    x_k, y_k = x_k_next, y_k_next

    # Обновляем предыдущее значение функции
    f_val_prev = f_val

end_time = time.time()

print(f'Потраченное время: {(end_time - start_time):.4f} мс')
```

```
1 Текущие точки: 0.1, 0.1
Значение функции: -0.000391202300542815
Приращение функции: 0
Градиент: Matrix([[-0.00682404601085629], [-0.00682404601085629]])

13 Текущие точки: 0.133591335739991, 0.133591335739991
Значение функции: -0.00106150355416517
Приращение функции: -0.0000991524370942202
Градиент: Matrix([[-0.0135076452460145], [-0.0135076452460145]])

25 Текущие точки: 0.210361905276005, 0.210361905276005
Значение функции: -0.00474818041909765
Приращение функции: -0.000669211673340098
Градиент: Matrix([[-0.0358340057498928], [-0.0358340057498928]])

37 Текущие точки: 0.428488783453354, 0.428488783453354
Значение функции: -0.0337717611058428
Приращение функции: -0.00395166074399261
Градиент: Matrix([[-0.0789602785524611], [-0.0789602785524611]])

49 Текущие точки: 0.549186238205136, 0.549186238205136
Значение функции: -0.0459821802246783
```

```
Приращение функции: -0.000003947222237169465
Градиент: Matrix([[-0.00181808670990966], [-0.00181808670990966]])

59 Текущие точки: 0.550678777239436, 0.550678777239436
Значение функции: -0.0459849298146815
Приращение функции: -4.88060036829552E-10
Градиент: Matrix([[-2.00596941186404e-5], [-2.00596941186404e-5]])

Интераций: 59
Потраченное время: 0.3284 мс
```

```
e = sp.sqrt(1/(2*sp.sqrt(sp.E))) # 1/(2*sqrt(e))
z = f.subs({x: e, y: e}).evalf() # Экстремума A
```

Проверяем, что наш конечный X_k Y_k и правда стремился и дошел до наибольшей экстремумы A (Верно)

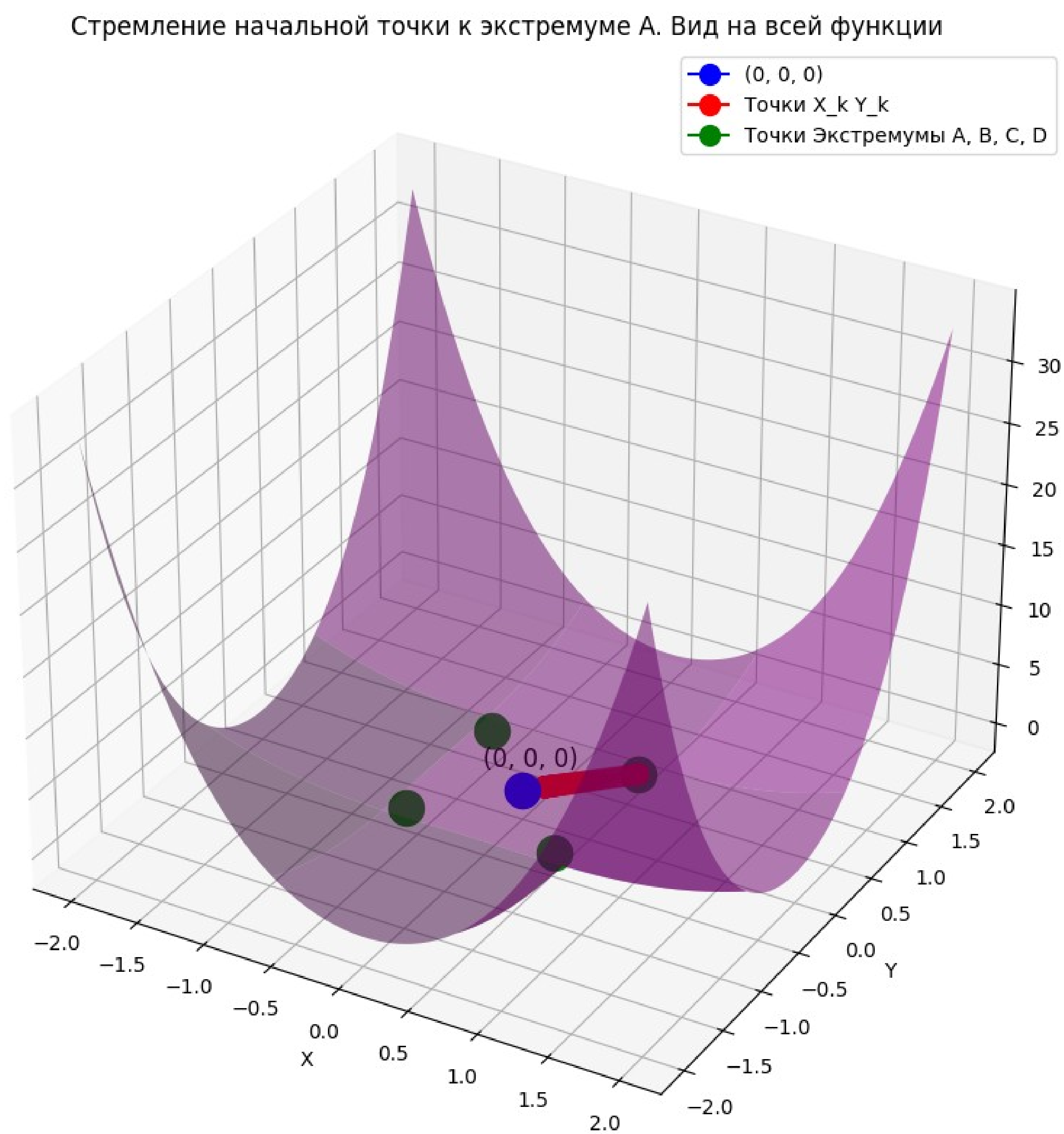
```
print(abs(f.subs({x: x_k, y: y_k}).evalf() - z) < epsilon)
```

```
True
```

Посмотрим на всей функции, как мы стремимся к экстремуме A

```
fig = plt.figure(figsize=(10, 10))
ax = fig.add_subplot(111, projection='3d')

# Создание сетки координат для построения поверхности
x_grid = np.linspace(-2, 2, 300)
y_grid = np.linspace(-2, 2, 300)
X, Y = np.meshgrid(x_grid, y_grid)

# Вычисление значений функции для каждой точки сетки
Z = np.array([[f.subs({x: x_val, y: y_val}).evalf() for x_val in
x_grid] for y_val in y_grid])

# Построение точек
ax.scatter(0, 0, 0, color='blue', s=300)
ax.text(-0.3, 0, 1, '(0, 0, 0)', fontsize=12)

ax.scatter(x_points, y_points, z_points, color='r', s=100)

ax.scatter(e, e, z, color='g', s=300)

ax.scatter(e, -e, z, color='g', s=300)

ax.scatter(-e, e, z, color='g', s=300)
```

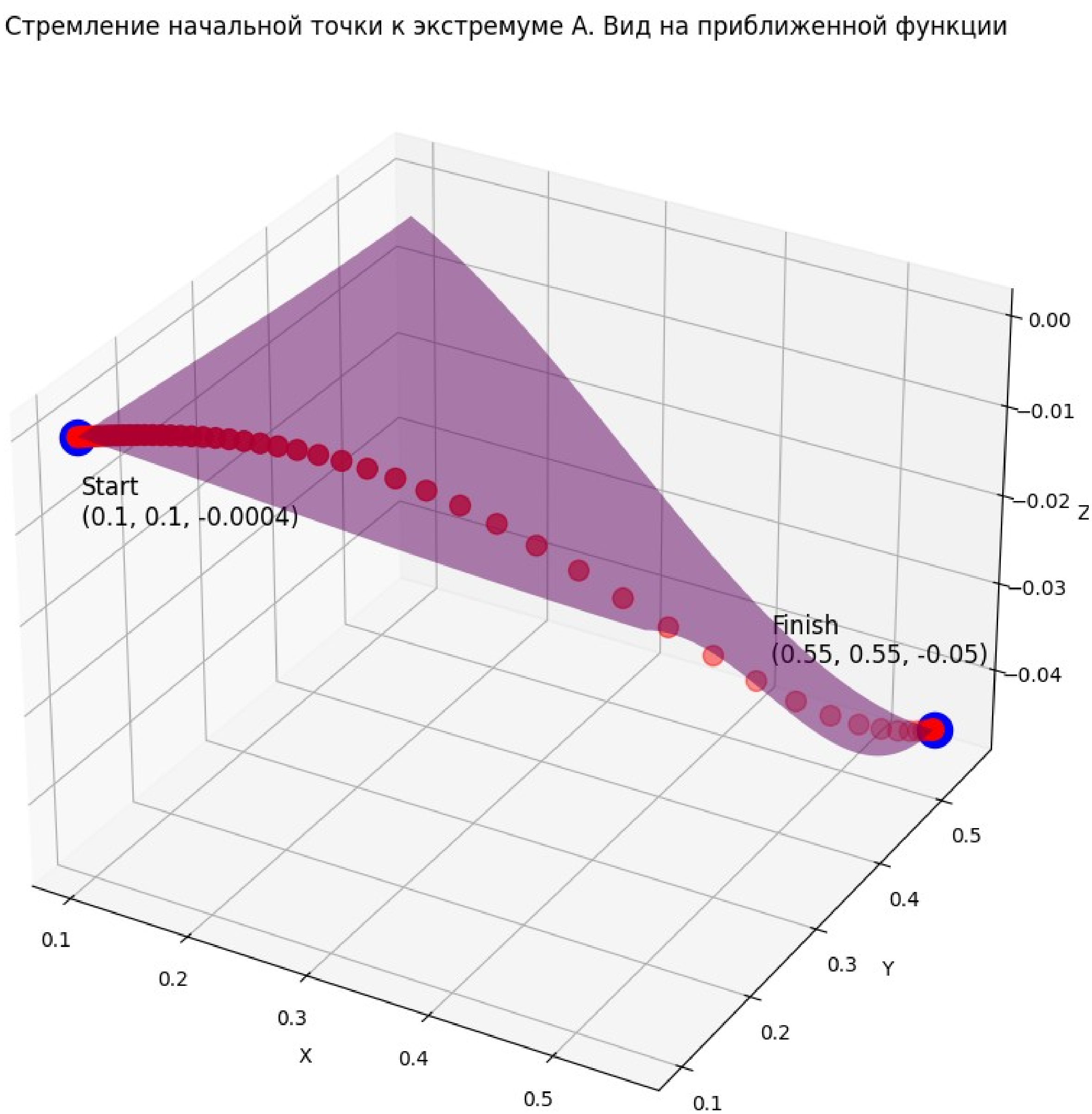
```
ax.scatter(-e, -e, z, color='g', s=300)

# Построение поверхности
ax.plot_surface(X, Y, Z, alpha=0.5, rstride=100, cstride=100,
color='purple')

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

# Добавление заголовка
ax.set_title('Стремление начальной точки к экстремуме A. Вид на всей
функции')

# Legend
start_line = mlines.Line2D([], [], color='blue', marker='o',
markersize=10, label='(0, 0, 0)')
points_line = mlines.Line2D([], [], color='red', marker='o',
markersize=10, label='Точки X_k Y_k')
finish_line = mlines.Line2D([], [], color='green', marker='o',
markersize=10, label='Точки Экстремумы A, B, C, D')
ax.legend(handles=[start_line, points_line, finish_line], loc='upper
right')

plt.show()
```



Приблизим, чтобы увидеть получше

```
fig = plt.figure(figsize=(10, 10))
ax = fig.add_subplot(111, projection='3d')

# Создание сетки координат для построения поверхности
x_grid = np.linspace(float(min(x_points)), float(max(x_points)), 100)
y_grid = np.linspace(float(min(y_points)), float(max(y_points)), 100)
X, Y = np.meshgrid(x_grid, y_grid)

# Вычисление значений функции для каждой точки сетки
```

```
Z = np.array([[f.subs({x: x_val, y: y_val}).evalf() for x_val in
x_grid] for y_val in y_grid])
# Построение поверхности
ax.plot_surface(X, Y, Z, alpha=0.5, rstride=100, cstride=100,
color='purple')

z = f.subs({x: 0.1, y: 0.1}).evalf() # старт

# Построение точек
ax.scatter(0.1, 0.1, z, color='blue', s=300)
ax.text(0.1, 0.1, z * 25, f'Start\n(0.1, 0.1, {z:.4f})', fontsize=12)

ax.scatter(x_points, y_points, z_points, color='r', s=100)

z = f.subs({x: e, y: e}).evalf()
ax.scatter(e, e, z, color='blue', s=300)
ax.text(e * 0.75, e, z * 0.95, f'Finish\n({e:.2f}, {e:.2f}, {z:.2f})',
fontsize=12)

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

# Добавление заголовка
ax.set_title('Стремление начальной точки к экстремуме A. Вид на
приближенной функции')

plt.show()
```



ССЫЛКА НА GEOGEBRA: https://www.geogebra.org/3d/jf5wnnjq

! МАТЕМАТИКА МАТЕМАТИКА МАТЕМАТИКА МАТЕМАТИКА !