

Візуалізація даних

Лабораторна робота №3

Засоби побудови полів у Matplotlib



Візуалізація скалярних полів

Скалярне поле (скалярна функція) на деякому кінцевому просторі V – функція, що ставить у відповідність кожній точці з деякої області цього простору (область визначення) скаляр, тобто дійсне або комплексне число. При фіксованому базисі простору скалярне поле можна задати як функцію кількох змінних від координат точки.

Різниця між числовою функцією декількох змінних і скалярним полем полягає в тому, що в іншому базисі скалярне поле як функція координат змінюється так, якщо новий набір аргументів представляє ту ж точку простору в новому базисі, то значення скалярної функції не змінюється.

Наприклад, якщо в деякому ортонормованому базисі двовимірного векторного простору скалярна функція має вигляд $f(v) = x^2 + 2y^2$, то в іншому базисі, поверненому на 45 градусів до цього, ця ж функція в нових координатах матиме вигляд $f(v) = 3x'^2 + 3y'^2 - 2x'y'$.

Найчастіше розглядаються скалярні функції, що є безперервними або диференційованими (гладкими) достатню кількість разів. На практиці знайшли широко зустрічається:

- Функція трьох змінних: $u = u(\mathbf{r}) = u(x, y, z)$ (скалярне поле на (в) тривимірному просторі).
- Функція двох змінних: $u = u(\mathbf{r}) = u(x, y)$ (скалярне поле на (в) двовимірному просторі).

Прикладами скалярних полів на тривимірному просторі є:

- температура (якщо вона різна в різних точках простору);
- електростатичний потенціал;
- потенціал у ньютонівській теорії тяжіння;
- поле тиску в рідкому середовищі.

Приклади плоских (двовимірних) скалярних полів:

- глибина моря, відзначена будь-яким чином на плоскій карті;
- густина заряду на плоскій поверхні провідника.

Однією з важливих характеристик скалярного поля є його *градієнт*.

Градiєнтом скалярного поля є векторне поле з компонентами:

$$\text{grad}(u(x, y)) = \left(\frac{\partial u(x, y)}{\partial x}, \frac{\partial u(x, y)}{\partial y} \right) \quad (1)$$

Приклад. Побувати візуалізацію скалярного поля $u(x, y) = \sin(x^2 + y^2)$, $x \in [-3; 3]$; $y \in [-3; 3]$. Реалізація представлена у лістингу 3.1.

```
import numpy as np
from matplotlib import pyplot as plt

n = 256
x = np.linspace(-3., 3., n) # діапазон по X
y = np.linspace(-3., 3., n) # діапазон по Y
X, Y = np.meshgrid(x, y)

Z = np.sin(X**2+Y**2) # Формула скалярного поля

plt.pcolormesh(X, Y, Z)
plt.show()
```

Лістинг 3.1 – Приклад коду для візуалізації скалярного поля

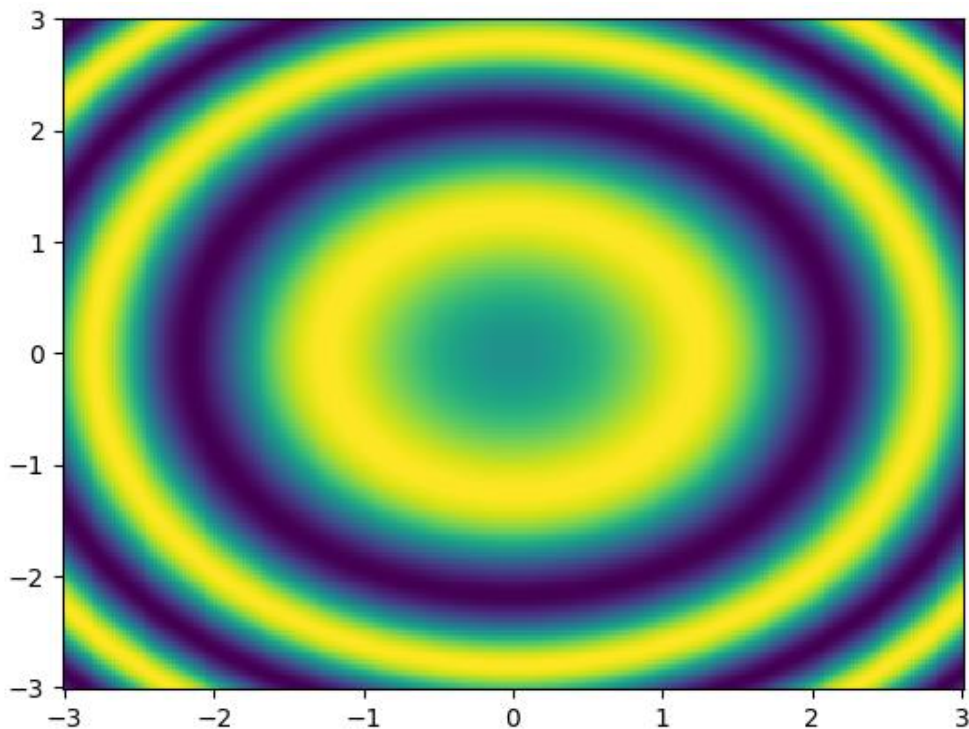


Рисунок 3.1 – Візуалізація скалярного поля

Візуалізація векторних полів

Векторне поле — векторнозначна функція – відображення, яке кожній точці даного простору ставить у відповідність вектор. У випадку евклідового (скінченновимірний векторний простір зі скалярним добутком) простору, поняття векторного поля стає наочним, і векторне поле інтерпретується як спосіб завдання рухів деякої динамічної системи: вектор у даній точці описує напрям і швидкість руху точки по фазовій кривій. Для декартової системи координат поле може бути представлене у вигляді $\mathbf{F}(\mathbf{r}) = \{u(x, y, z), v(x, y, z), w(x, y, z)\}$.

Приклад. Побудувати візуалізацію плоского векторного поля

$$F(r) = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -y / \sqrt{x^2 + y^2} \\ x / \sqrt{x^2 + y^2} \end{pmatrix}, x, y \in [-5; 5]; \quad (2)$$

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

def u(x, y):
    return -y / np.sqrt(x ** 2 + y ** 2)

def v(x, y):
    return x / np.sqrt(x ** 2 + y ** 2)

xx, yy = np.meshgrid(np.linspace(-5, 5, 10),
                     np.linspace(-5, 5, 10))
u_val = u(xx, yy)
v_val = v(xx, yy)
plt.quiver(xx, yy, u_val, v_val)
plt.show()

plt.streamplot(xx, yy, u_val, v_val)
plt.show()
```

Лістинг 3.2 – Приклад коду для візуалізації векторного поля

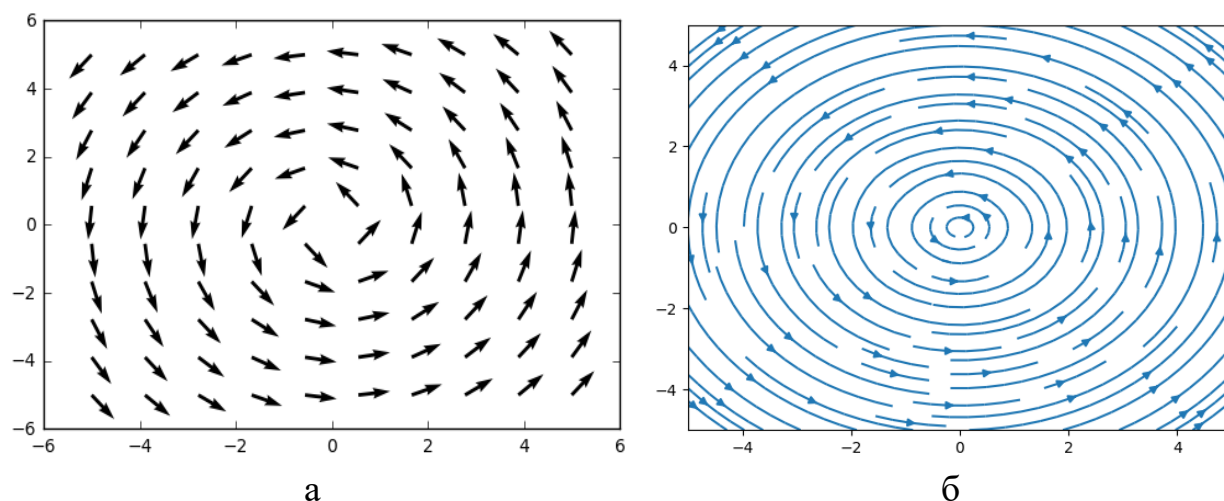


Рисунок 3.2 – Візуалізація векторного поля: а – векторами; б – лініями потоку

Також відносно просто можна власноруч будувати лінії току знаходячи траєкторію безмасової точки виконуючі інтегрування за наступний принципом (див. лістинг 3.3):

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} u(t_i) \\ v(t_i) \end{pmatrix} \Delta t \quad (3)$$

Результат побудови ліній току можна побачити на рис. 3.3.

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

def u(x, y):
    return -y / np.sqrt(x ** 2 + y ** 2)

def v(x, y):
    return x / np.sqrt(x ** 2 + y ** 2)

def create_stream_line(x0, y0, u, v, t0=0, t1=10, dt=0.001):
    t = np.arange(t0, t1, dt)
    xx_new = np.zeros_like(t)
    yy_new = np.zeros_like(t)
    xx_new[0] = x0
    yy_new[0] = y0

    for i in range(1, t.size):
        xx_new[i] = x0 + u(x0, y0) * dt
        yy_new[i] = y0 + v(x0, y0) * dt
```

```

        x0, y0 = xx_new[i], yy_new[i]

    return xx_new, yy_new

for i in range(1, 10):
    x1, y1 = create_stream_line(i, 0, u, v)
    plt.plot(x1, y1)

plt.show()

```

Лістинг 3.3 – Приклад коду для розрахунку та візуалізації ліній току

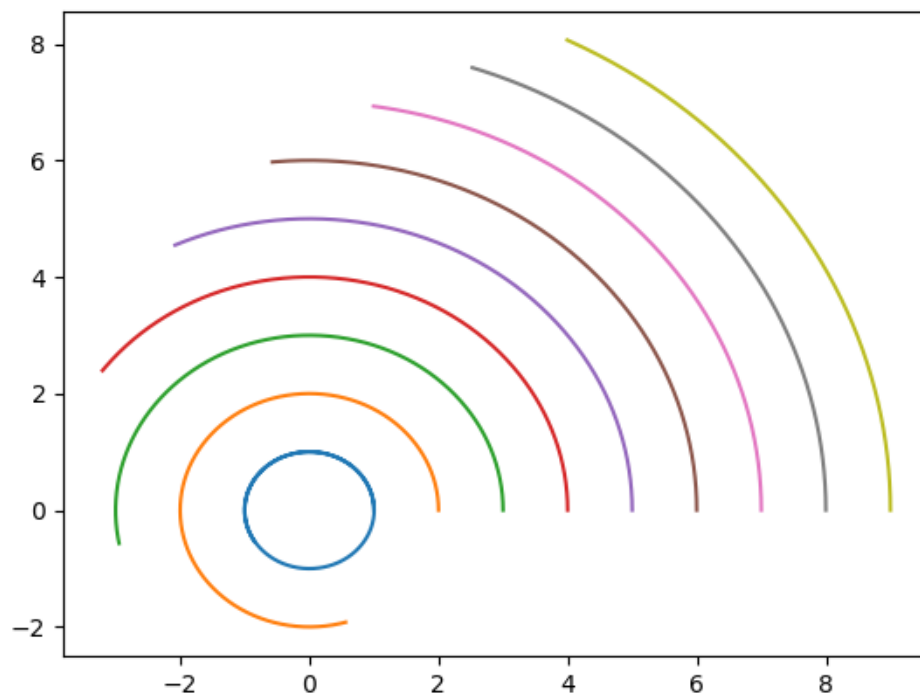


Рисунок 3.3 – Візуалізація ліній току

Приклад. Побудувати візуалізацію векторного поля

$$F(r) = \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} \sin(\pi x) \cos(\pi y) \cos(\pi z) \\ -\cos(\pi x) \sin(\pi y) \cos(\pi z) \\ \sqrt{2}/3 \cos(\pi x) \cos(\pi y) \sin(\pi z) \end{pmatrix}, x, y, z \in [-0.8; 0.8]; \quad (4)$$

```

from mpl_toolkits.mplot3d import axes3d
import matplotlib.pyplot as plt
import numpy as np

ax = plt.figure().add_subplot(projection='3d')

x, y, z = np.meshgrid(np.arange(-0.8, 1, 0.2),
                      np.arange(-0.8, 1, 0.2),
                      np.arange(-0.8, 1, 0.8))

```

```

u = np.sin(np.pi * x) * np.cos(np.pi * y) * np.cos(np.pi * z)
v = -np.cos(np.pi * x) * np.sin(np.pi * y) * np.cos(np.pi * z)
w = (np.sqrt(2.0 / 3.0) * np.cos(np.pi * x) * np.cos(np.pi * y)
* np.sin(np.pi * z))

ax.quiver(x, y, z, u, v, w, length=0.2, color = 'black')

plt.show()

```

Лістинг 3.4 – Візуалізація тривимірного векторного поля

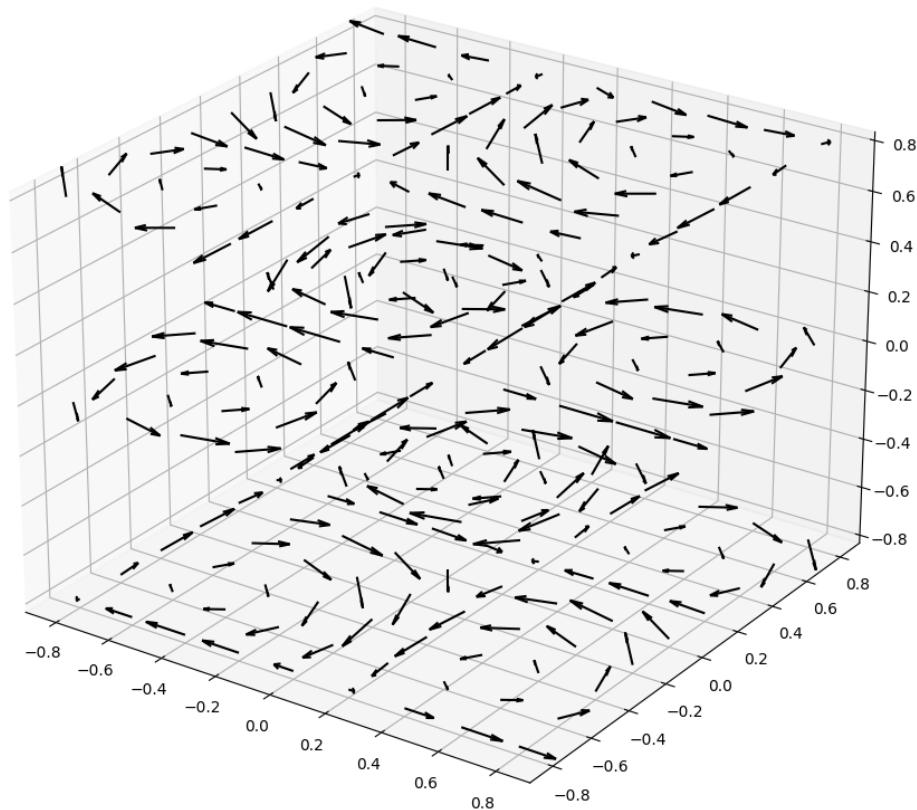


Рисунок 3.4 – Візуалізація тривимірного векторного поля

Візуалізація тензорних полів

Під тензорним полем, по аналогії з векторним полем, розуміють відображення, яке кожній точці простору ставить у відповідність тензор. Найпростішим тензором є тензор рангу 2 і він є «поєднанням» двох векторів в один об'єкт. Так вектор у 3-вимірному просторі має 3 компоненти, то тензор 2-го порядку містить в собі 3×3 елементів і представляється за

допомогою відповідної матриці. Якщо тензор симетричний, то з 9-ти його компонентів незалежними є тільки 6. Одним з широко поширених тензорів – є тензор механічних напружень (рис. 3.5):

$$\sigma = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{pmatrix} \quad (5)$$

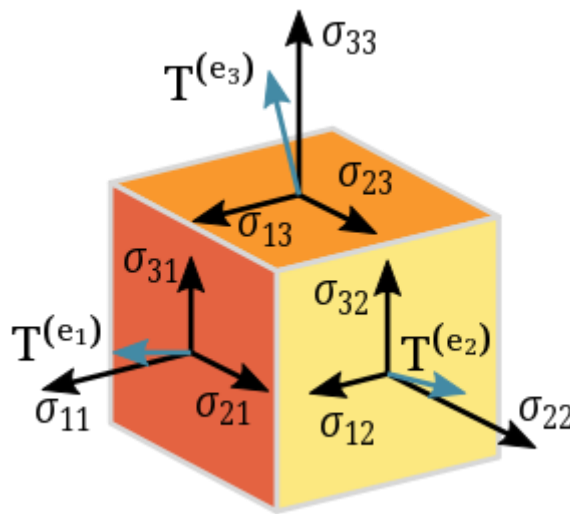
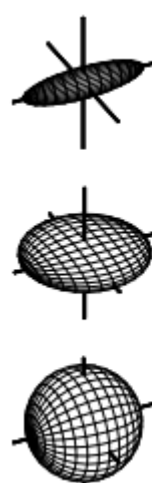


Рисунок 3.5 – Компоненти тензора напружень σ
(Рисунок взято з <https://uk.wikipedia.org/wiki/Тензор>)

Для побудови зображень (гліфів) вважається, що тензор може бути представлений у вигляді:

$$\sigma = R\Lambda R^{-1}, \quad (6)$$

де R – матриця повороту координат до базису, що співпадає з головними напрямками; $\Lambda = (\lambda_1, \lambda_2, \lambda_3)$ – власні значення тензору. Більш детально про побудову гліфів можна дізнатись зі статті Г. Кінделманна (Kindlmann G. Superquadric tensor glyphs // Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization. – 2004. – С. 147-154. <https://people.cs.uchicago.edu/~glk/pubs/pdf/Kindlmann-Superquad1-VisSym-2004.pdf>).



$$c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3}$$

$$c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3}$$

$$c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$$
(7)

Сума трьох метрик c_l , c_p , c_s становить одиницю і визначає барицентричну параметризацію трикутної області з крайніми значеннями лінійних, площинних і сферичних форм у трьох кутах.

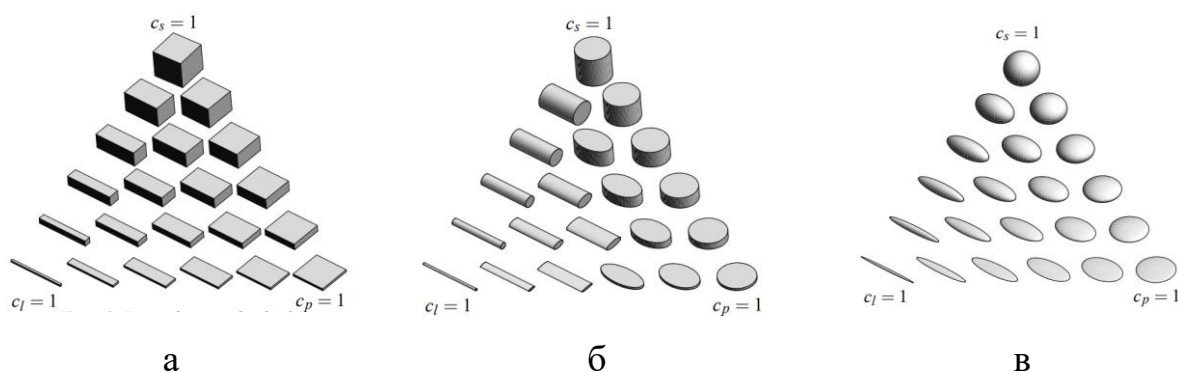


Рисунок 3.6 – Візуалізація тензора напружень при різних значеннях c_l , c_p , c_s :
а – кубами, б – циліндрами, в – еліпсоїдами

Для реалізації цієї ідеї на кафедрі динаміки та міцності машин НТУ «ХП» розроблена бібліотека `glyph_visualization_lib` (автори Руслан Бабуджан, Олексій Водка). Ця бібліотека підтримує два варіанти для візуалізації:

- За допомогою `matplotlib` (працює повільно, але не потребує додаткових залежностей);
- За допомогою `mayavi` (Швидший варіант).

Для встановлення `mayavi` необхідно встановити пакети `mayavi` та `pyqt5`.

```
pip install mayavi pyqt5
```


Приклад. Побудувати візуалізацію тензорного поля:

$$\sigma = \begin{pmatrix} -\sin x & 5\sin(x+y) & 10\sin(xz) \\ & -\sin y & 3\sin(y+z) \\ \text{symm} & & \cos(z) \end{pmatrix} \quad (8)$$

Код, що виконує візуалізацію наведено у лістингу 3.5. Для того, щоб код запрацював необхідно завантажити файл *glyph_visualization_lib.py* та додати його в каталог проекту. Результат візуалізації представлений на рис. 3.7.

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

from mayavi import mlab
import glyph_visualization_lib as gvl

def main():
    x = np.linspace(0, np.pi, 8, dtype=float, endpoint=True)
    y = np.linspace(-np.pi, -1, 8, dtype=float, endpoint=True)
    z = np.linspace(0, 2 * np.pi, 8, dtype=float, endpoint=True)

    X, Y, Z = np.meshgrid(x, y, z)

    stress_tensor = np.array([
        [-np.sin(X), 5 * np.sin(X + Y), 10 * np.sin(X * Z)],
        [5 * np.sin(X + Y), -np.sin(Y), 3 * np.sin(Y + Z)],
        [10 * np.sin(X * Z), 3 * np.sin(Y + Z), np.cos(Z)]
    ])

    print(stress_tensor.shape)

    vm_stress = gvl.get_von_Mises_stress(stress_tensor)

    glyph_radius = 0.25
    limits = [np.min(vm_stress), np.max(vm_stress)]
    colormap = plt.get_cmap('rainbow', 120)
    fig = mlab.figure(bgcolor=(1, 1, 1))

    fig2 = plt.figure()
    ax = fig2.add_subplot(111, projection='3d')

    for i in range(x.size):
        for j in range(y.size):
```

```

        for k in range(z.size):
            center = [x[i], y[j], z[k]]
            data = stress_tensor[:, :, i, j, k]
            color =
colormap(gvl.get_colormap_ratio_on_stress(vm_stress[i, j, k],
limits))[:3]

            """
            glyph_type = {0: 'cuboid', 1: 'cylinder', 2:
'ellipsoid', 3: 'superquadric'}

            if glyph_type == 3 (superquadric)
            there are glyph shape type
            0 - superquadrics,
            1 - Kindlmann_glyph,
            2 - Kindlmann_modified_glyph

            """

            x_g, y_g, z_g = gvl.get_glyph_data(center, data,
limits, glyph_points=12, glyph_radius=glyph_radius,
                                                    glyph_type=3,
superquadrics_option=2)

            # surf = ax.plot_surface(x_g, y_g, z_g,
linewidth=0, antialiased=True, color=color)
            # surf = ax.plot_wireframe(x_g, y_g, z_g,
linewidth=1, antialiased=True, color=color)

            mlab.mesh(x_g, y_g, z_g, color=color)

            mlab.move(forward=1.8)
            mlab.savefig("superquadric-Kindlmann_modified-viz.png",
size=(100, 100))
            mlab.show()

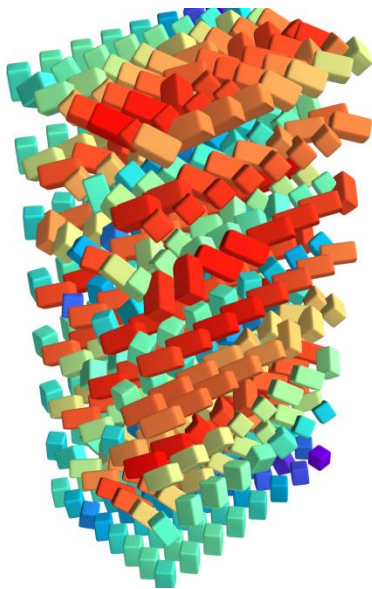
            # plt.show()

        pass

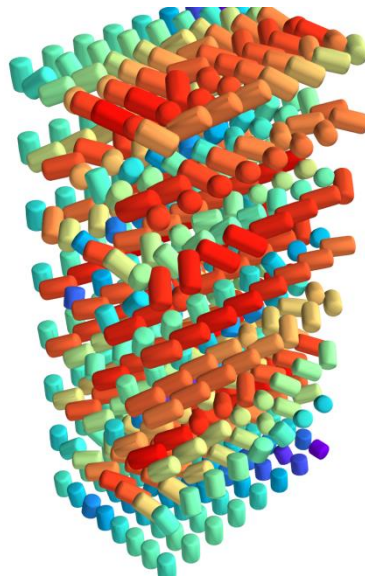
if __name__ == '__main__':
    main()

```

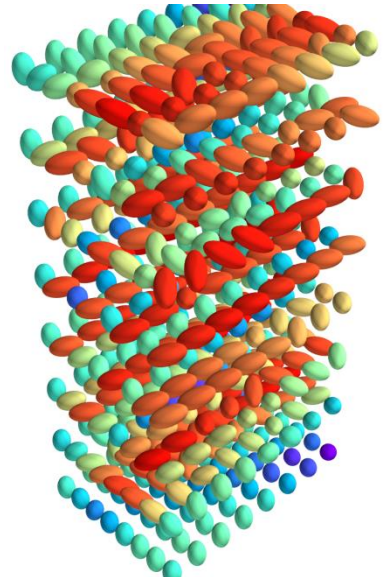
Лістинг 3.5 – Візуалізація тензорного поля



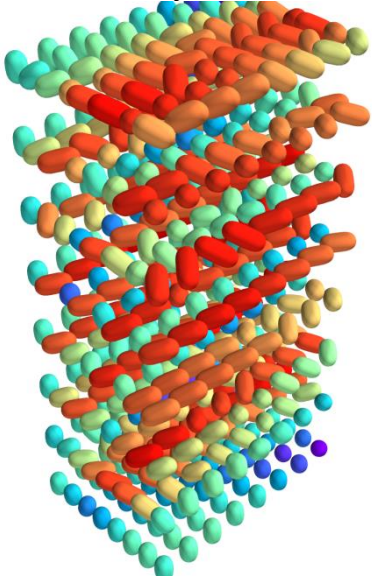
а. кубоїд



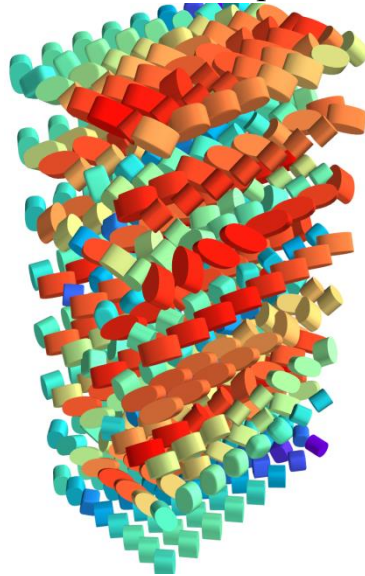
б. циліндр



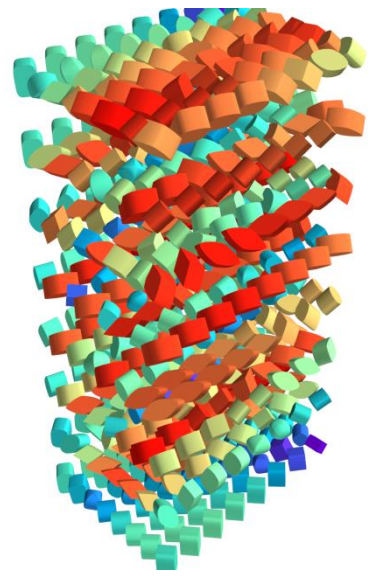
в. еліпсоїд



г. суперквадр



д. суперквадр
кінделмана



е. модифікований
суперквадр Кінделмана

Рисунок 3.7 – Візуалізація тензорного поля

Завдання для виконання. Згідно за варіантом побудуйте:

1. Візуалізацію скалярного поля. Знайдіть його градієнт та візуалізуйте його як плоске векторне поле;
2. Побудуйте візуалізацію плоского векторного поля як за допомогою векторів та ліній току з бібліотеки `matplotlib` та за допомогою коду з лістингу;
3. Побудуйте тривимірну візуалізацію векторного поля; За додатковий бал (не обов'язково) модернізуйте алгоритм побудови ліній току на випадок 3-вимірного поля.
4. Побудуйте візуалізацію тензорного поля за допомогою еліпсоїдів, кубоїдів, циліндрів та будь-якого суперквядру.

Варіант 1.

1. $u(x, y) = 4 \ln(x^2 + y^2) - 8xy; x, y \in [-5; 3]$
2. $F = (x^2 - 2y; y^2 - 2x); x, y \in [-5; 3]$
3. $F = \left(\frac{y}{z} + 2x; \frac{x}{z}; \frac{xy}{z^2} \right); x, y, z \in [-5; 3]$
4. $T = \begin{pmatrix} x & \ln(xy) & -\ln(xz) \\ & -y & \ln(yz) \\ & & z \end{pmatrix}, x, y, z \in [1..10]$

Варіант 2.

1. $u(x, y) = x\sqrt{y} + y\sqrt{x}; x, y \in [0; 5]$
2. $F = (x^2 + 2y; y^2 + 2x); x, y \in [-4; 4]$
3. $F = \left(\frac{yz}{x^2 + y^2 + z^2}; \frac{xz}{x^2 + y^2 + z^2}; \frac{xy}{x^2 + y^2 + z^2} \right); x, y, z \in [-3; 4]$
4. $T = \begin{pmatrix} \sin(x) & x + y & x + z \\ & \cos(y) & y + z \\ & & \cos(z) \end{pmatrix}, x, y, z \in [-2\pi..2\pi]$

Варіант 3.

1. $u(x, y) = -2 \ln(x^2 + 5) - 4xy; x, y \in [-3; 6]$
2. $F = (x + 2y^2; y + 2x^2); x, y \in [-3; 6]$
3. $F = \left(y - \frac{z}{x^2}; x + \frac{1}{z}; \frac{1}{x} - \frac{y}{z^2} \right); x, y, z \in [-3; 6]$
4. $T = \begin{pmatrix} \exp(-x^2) & \sin(x+y) & \sin(x+z) \\ & \exp(-y^2) & \sin(y+z) \\ & & \exp(-z^2) \end{pmatrix}, x, y, z \in [-2..2]$

Варіант 4.

1. $u(x, y) = xy^2 - \sqrt{x^3 y}; x, y \in [0; 8]$
2. $F = (x^3; -y^3); x, y \in [-8; 8]$
3. $F = \left(\frac{x^2}{x^3 + y^3 + z^3}; \frac{y^2}{x^3 + y^3 + z^3}; \frac{z^2}{x^3 + y^3 + z^3} \right); x, y, z \in [-7; 7]$
4. $T = \begin{pmatrix} \frac{\sin(x+y)}{x^2 + y^2 + z^2} & x & y \\ & \frac{\cos(y+z)}{x^2 + y^2 + z^2} & z \\ & & \frac{\cos(x+z)}{x^2 + y^2 + z^2} \end{pmatrix}, x, y, z \in [-2..2]$

Варіант 5.

1. $u(x, y) = x\sqrt{y} - yx^2; x, y \in [0; 9]$
2. $F = (x + y; x - y); x, y \in [-9; 9]$
3. $F = \left(\frac{1}{y} - \frac{z}{x^2}; \frac{1}{z} - \frac{x}{y^2}; \frac{1}{x} - \frac{y}{z^2} \right); x, y, z \in [-9; 9]$
4. $T = \begin{pmatrix} \frac{\ln(x)}{x^2 + y^2 + z^2} & \sqrt{x} & \sqrt{y} \\ & \frac{\ln(y)}{x^2 + y^2 + z^2} & \sqrt{z} \\ & & \frac{\ln(z)}{x^2 + y^2 + z^2} \end{pmatrix}, x, y, z \in [1..5]$

Варіант 6.

$$1. u(x, y) = 7 \ln(x^2 + \frac{1}{13}) - 4 \sin(xy); x, y \in [-10; 10]$$

$$2. F = (x^2 y; -y); x, y \in [-10; 10]$$

$$3. F = \left(\frac{x+z}{x^2}; \frac{1}{y}; \frac{1}{z} \right); x, y, z \in [-10; 10]$$

$$4. T = \begin{pmatrix} \frac{\ln(x)}{\sin(x)} & \sqrt{x}/y & \sqrt{y}/z \\ & \frac{\ln(y)}{\sin(y)} & \sqrt{z}/x \\ & & \frac{\ln(z)}{\sin(z)} \end{pmatrix}, x, y, z \in [1..3]$$

Варіант 7.

$$1. u(x, y) = \arcsin\left(\frac{2y}{x^2 + y^2}\right); x, y \in [-\pi/2; \pi/2]$$

$$2. F = (2xy - y; x^2 + x); x, y \in [-11; 11]$$

$$3. F = \left(\frac{2x}{y} + 1; \frac{x^2}{y^2}; -6z^2 \right); x, y, z \in [-11; 11]$$

$$4. T = \begin{pmatrix} 1 & \sin(x)/x & \sin(y)/y \\ & 1 & \sin(z)/z \\ & & 1 \end{pmatrix}, x, y, z \in [-3..3]$$

Варіант 8.

$$1. u(x, y) = \arcsin\left(\frac{2(x+y)}{x^2 + y^2}\right); x, y \in [-\pi/2; \pi/2]$$

$$2. F = (\sin(x); \sin(y); \sin(z)); x, y, z \in [-2\pi; 2\pi]$$

$$3. F = \left(\frac{x+z}{x^2}; \frac{y+x}{y^2}; \frac{z+x}{z^2} \right); x, y, z \in [-11; 11]$$

$$4. T = \begin{pmatrix} \frac{x}{x^2 + y^2 + z^2} & \frac{x+y}{z} & \frac{x+z}{y} \\ & \frac{y}{x^2 + y^2 + z^2} & \frac{y+z}{x} \\ & & \frac{z}{x^2 + y^2 + z^2} \end{pmatrix}, x, y, z \in [1..5]$$

Варіант 9.

$$1. u(x, y) = \frac{1}{4} x^2 y - \sqrt{x^2 + 5y^2}; x, y \in [-2; 7]$$

$$2. F = (x + y; 2x); x, y \in [-2; 7]$$

$$3. F = \left(\frac{x}{(x^2 + y^2 + z^2)^3}; \frac{y}{(x^2 + y^2 + z^2)^3}; \frac{z}{(x^2 + y^2 + z^2)^3} \right); x, y, z \in [-2; 7]$$

$$4. T = \begin{pmatrix} \frac{x}{x^2 + y^2 + z^2} & x & y \\ & \frac{y}{x^2 + y^2 + z^2} & z \\ & & \frac{z}{x^2 + y^2 + z^2} \end{pmatrix}, x, y, z \in [-2..2]$$