Санкт-Петербургский политехнический университет Петра Великого
Физико-механический институт
Высшая школа теоретической механики и математической физики

Р А Б О Т А № 4

Нахождение функций форм

по дисциплине «Вычислительная механика»

Выполнил
студент гр. 5030103/10301 <*подпись*> А.Г. Фёдоров

Руководитель
Доцент, к.ф.-м.н. <*подпись*> Е.Ю. Витохин

«___» __________ 2023 г.

Санкт-Петербург

2023

Постановка задачи

В данной работе необходимо найти функции форм для квадратичного тетраэдра (согласно моему варианту №9).

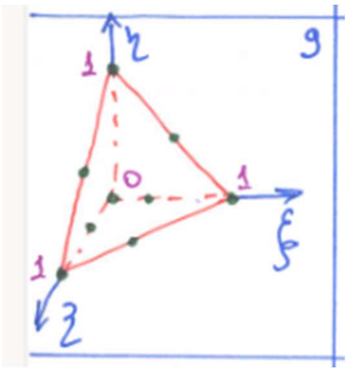

Рис.1. Квадратичный тетраэдр

Интерполяционный полином квадратичного тетраэдра выглядит так:

$$\boldsymbol{u = A + Bx + Cy + Dz + Ex^2 + Fy^2 + Gz^2 + Hxy + Ixz + Jyz}$$

Рассматриваемый элемент имеет узлы в следующих координатах:

$(\boldsymbol{x_1, y_1, z_1}) = (0, 0, 0),$

$(\boldsymbol{x_2, y_2, z_2}) = (1, 0, 0),$

$(\boldsymbol{x_3, y_3, z_3}) =(0, 1, 0),$

$(\boldsymbol{x_4, y_4, z_4}) = (0, 0, 1)$

$(\boldsymbol{x_5, y_5, z_5}) = (0.5, 0, 0)$

$(\boldsymbol{x_6, y_6, z_6}) = (0, 0.5, 0)$

$(\boldsymbol{x_7, y_7, z_7}) = (0, 0, 0.5)$

$(\boldsymbol{x_8, y_8, z_8}) = (0.5, 0.5, 0)$

$(\boldsymbol{x_9, y_9, z_9}) = (0.5, 0, 0.5)$

$(\boldsymbol{x_{10}, y_{10}, z_{10}}) = (0, 0.5, 0.5)$

Описание метода решения

Функции формы для квадратичного тетраэдра представляют собой плоскости, область определения которых ограничена:

$$N_1 = A_1 + B_1x + C_1y + D_1z + E_1x^2 + F_1y^2 + G_1z^2 + H_1xy + I_1xz + J_1yz$$

$$N_2 = A_2 + B_2x + C_2y + D_2z + E_2x^2 + F_2y^2 + G_2z^2 + H_2xy + I_2xz + J_2yz$$

$$N_3 = A_3 + B_3x + C_3y + D_3z + E_3x^2 + F_3y^2 + G_3z^2 + H_3xy + I_3xz + J_3yz$$

$$N_4 = A_4 + B_4x + C_4y + D_4z + E_4x^2 + F_4y^2 + G_4z^2 + H_4xy + I_4xz + J_4yz$$

$$N_5 = A_5 + B_5x + C_5y + D_5z + E_5x^2 + F_5y^2 + G_5z^2 + H_5xy + I_5xz + J_5yz$$

$$N_6 = A_6 + B_6x + C_6y + D_6z + E_6x^2 + F_6y^2 + G_6z^2 + H_6xy + I_6xz + J_6yz$$

$$N_7 = A_7 + B_7x + C_7y + D_7z + E_7x^2 + F_7y^2 + G_7z^2 + H_7xy + I_7xz + J_7yz$$

$$N_8 = A_8 + B_8x + C_8y + D_8z + E_8x^2 + F_8y^2 + G_8z^2 + H_8xy + I_8xz + J_8yz$$

$$N_9 = A_9 + B_9x + C_9y + D_9z + E_9x^2 + F_9y^2 + G_9z^2 + H_9xy + I_9xz + J_9yz$$

$$N_{10} = A_{10} + B_{10}x + C_{10}y + D_{10}z + E_{10}x^2 + F_{10}y^2 + G_{10}z^2 + H_{10}xy + I_{10}xz + J_{10}yz$$

То есть необходимо найти 100 коэффициентов. Для их определения используем основные свойства функций форм. В своем узле функция формы равна 1, а в остальных узлах равна 0.

$$N_1(\mathrm{x}_1, \mathrm{y}_1, \mathrm{z}_1) = 1$$

$$N_2(\mathrm{x}_1, \mathrm{y}_1, \mathrm{z}_1) = 0$$

$$N_3(\mathrm{x}_1, \mathrm{y}_1, \mathrm{z}_1) = 0$$

$$N_4(\mathrm{x}_1, \mathrm{y}_1, \mathrm{z}_1) = 0$$

$$N_5(\mathrm{x}_1, \mathrm{y}_1, \mathrm{z}_1) = 0$$

$$N_6(\mathrm{x}_1, \mathrm{y}_1, \mathrm{z}_1) = 0$$

$$N_7(\mathrm{x}_1, \mathrm{y}_1, \mathrm{z}_1) = 0$$

$$N_8(\mathrm{x}_1, \mathrm{y}_1, \mathrm{z}_1) = 0$$

$$N_9(\mathrm{x}_1, \mathrm{y}_1, \mathrm{z}_1) = 0$$

$$N_{10}(\mathrm{x}_1, \mathrm{y}_1, \mathrm{z}_1) = 0$$

Можно записать аналогичные уравнения для N_2 - N_{10} и получится всего 100 уравнений.

Если записать уравнения в матричном виде:

$$\begin{bmatrix} 1 & x_1 & y_1 & \dots & z_1y_1 \\ 1 & x_2 & y_2 & \dots & y_2z_2 \\ & \dots & & & \\ 1 & x_{10} & y_{10} & \dots & y_{10}z_{10} \end{bmatrix} \begin{bmatrix} A_1 & A_2 & \dots & A_{10} \\ B_1 & B_2 & \dots & B_{10} \\ C_1 & C_2 & \dots & C_{10} \\ D_1 & D_2 & \dots & D_{10} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ & & \dots & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Либо:

$$[X] \cdot [A] = [E]$$

$[X]$ - матрица с координатами

$[A]$ – матрица с коэффициентами функций формы

Последнее соотношение можно представить как:

$$[A] = [X]^{-1}$$

Представим интерполяционный полином в виде вектора-столбца:

$$\{P\}^T = \{1,\ x,\ y,\ z,\ x^2,\ y^2,\ z^2,\ xy,\ xz,\ yz\ \}$$

Теперь матрицу с функциями форм вида

$$[\widetilde{N}] = [N_1,\ N_2,\ N_3,\ N_4,\ N_5,\ N_6,\ N_7,\ N_8,\ N_9,\ N_{10}\ ]$$

Можно получить таким образом:

$$[\widetilde{N}] = \{P\}^T [A]$$

Результаты

Были получены следующие функции форм:

$$N_1(\mathrm{x,y,z}) = 1 - 3\mathrm{x} - 3\mathrm{y} - 3\mathrm{z} + 2\mathrm{x}^2 + 2\mathrm{y}^2 + 2\mathrm{z}^2 + 4\mathrm{xy} + 4\mathrm{xz} + 4\mathrm{yz}$$

$$N_2(\mathrm{x,y,z}) = -\mathrm{x} + 2\mathrm{x}^2$$

$$N_3(\mathrm{x,y,z}) = -\mathrm{y} + 2\mathrm{y}^2$$

$$N_4(\mathrm{x,y,z}) = -\mathrm{z} + 2\mathrm{z}^2$$

$$N_5(\mathrm{x,y,z}) = 4\mathrm{z} - 4\mathrm{z}^2 - 4\mathrm{xz} - 4\mathrm{yz}$$

$$N_6(\mathrm{x,y,z}) = 4\mathrm{x} - 4\mathrm{x}^2 - 4\mathrm{xy} - 4\mathrm{xz}$$

$$N_7(\mathrm{x,y,z}) = 4\mathrm{y} - 4\mathrm{y}^2 - 4\mathrm{xy} - 4\mathrm{yz}$$

$$N_8(\mathrm{x,y,z}) = 4\mathrm{yz}$$

$$N_9(\mathrm{x,y,z}) = 4\mathrm{xz}$$

$$N_{10}(\mathrm{x,y,z}) = 4\mathrm{xy}$$

Были получены графики распределения функций форм для каждого узла в проекциях на плоскости. На рисунках 2-11 представлено распределение функций форм для десяти узлов.

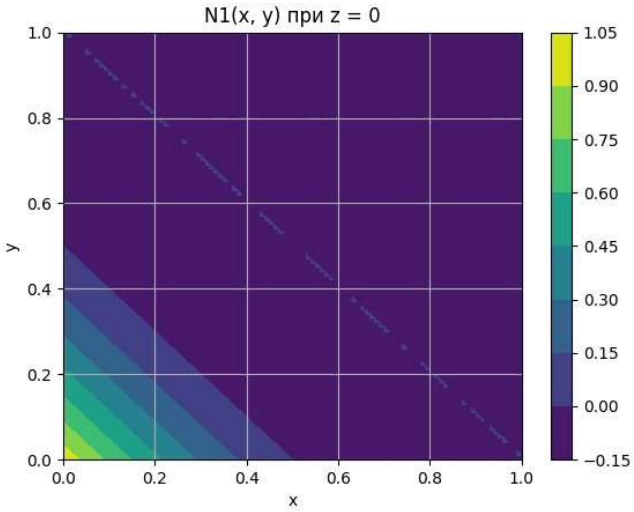

Рис.2. Проекция N_1 на OXY

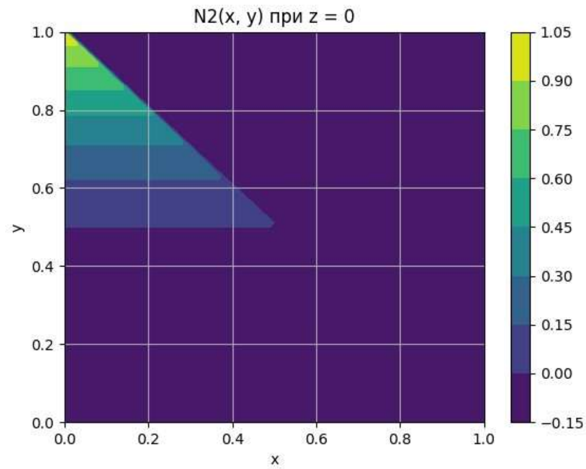

Рис.3. Проекция N_2 на OXY

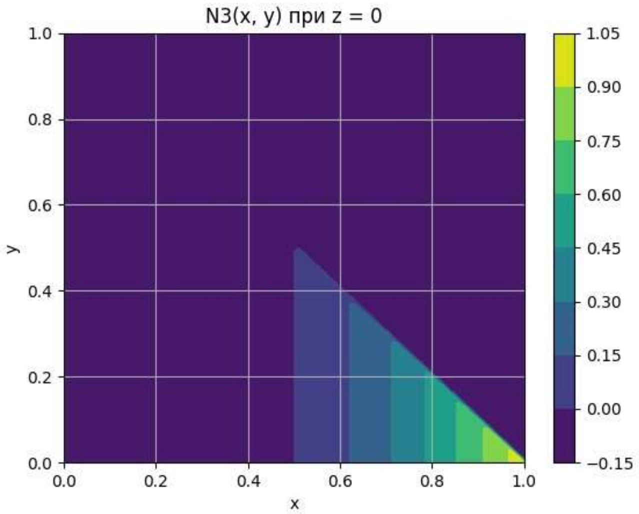

Рис.4. Проекция N_3 на OYZ

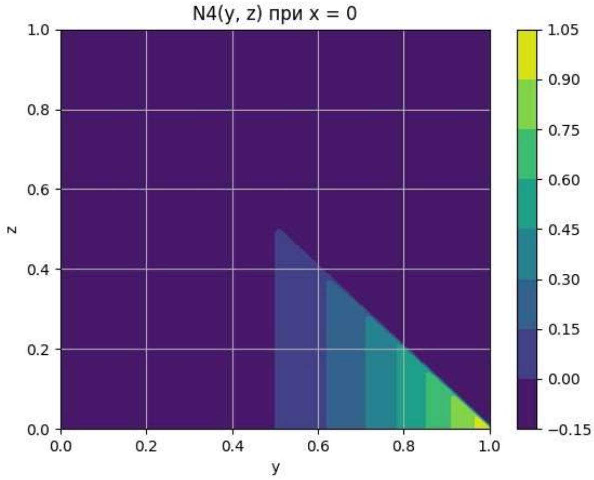

Рис.5. Проекция N_4 на OXY

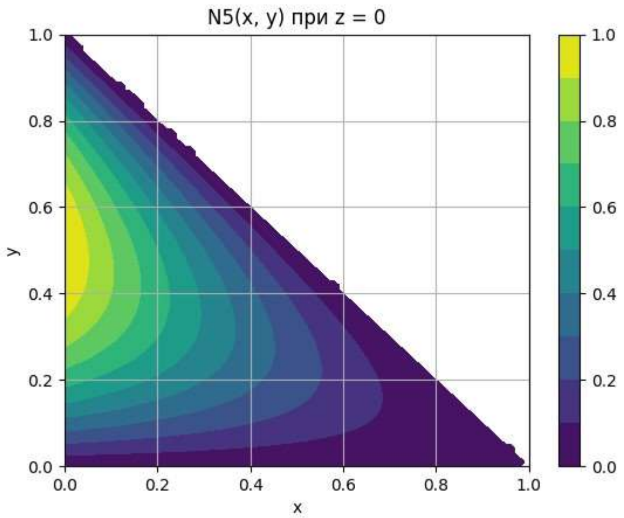

Рис.6. Проекция N_5 на OXY

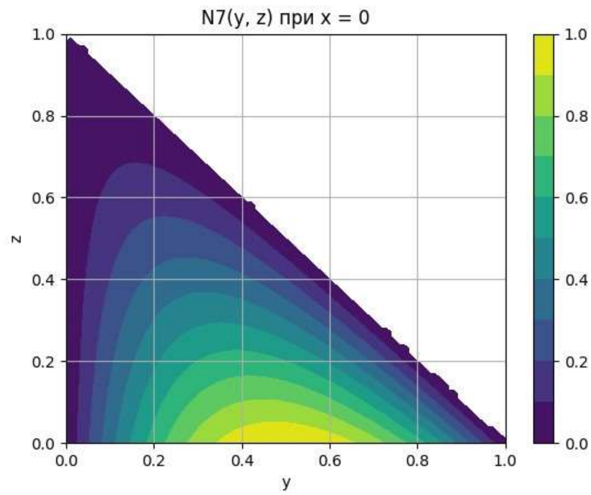

Рис.7. Проекция N_6 на OXY

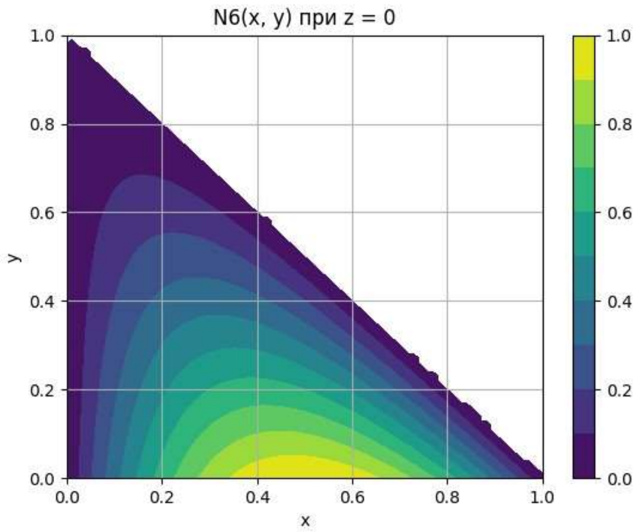

Рис.8. Проекция N_7 на OYZ

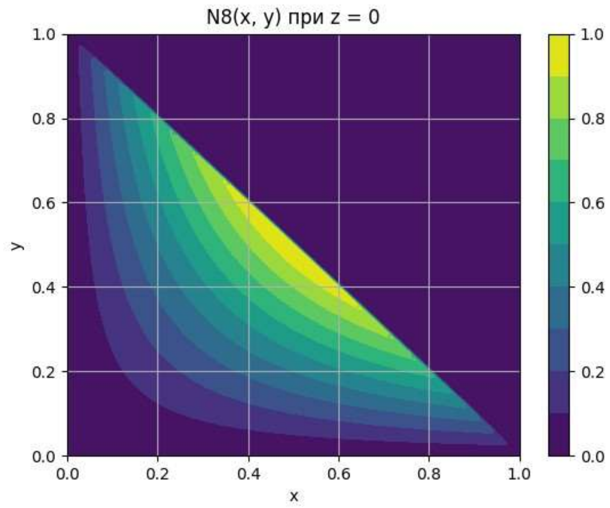

Рис.9. Проекция N_8 на OXY

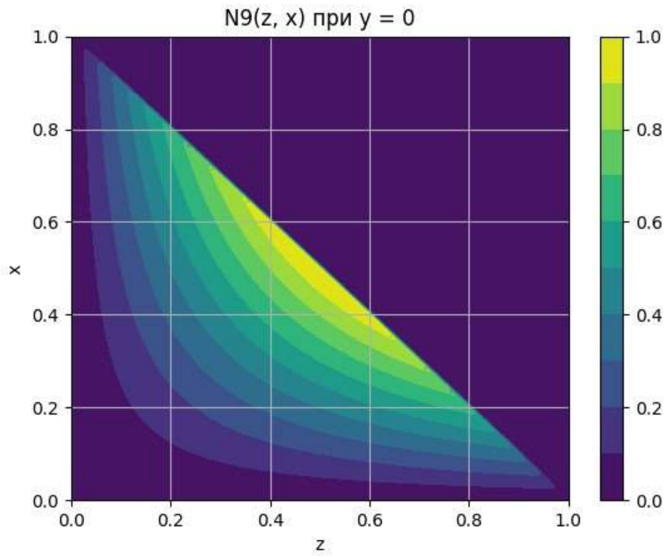

Рис.10. Проекция N_9 на OZX

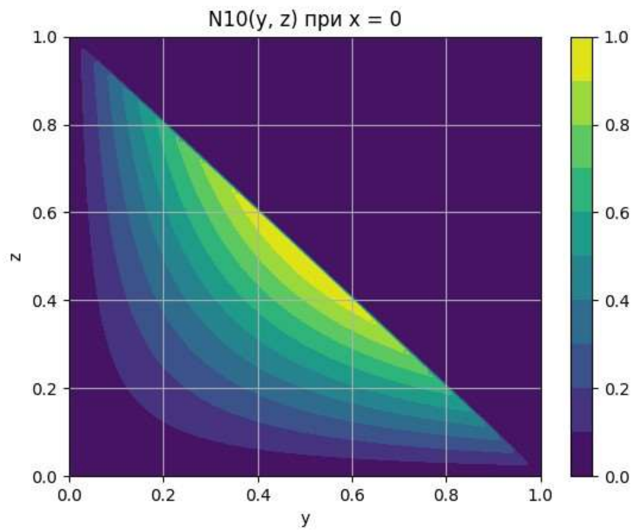

Рис.11. Проекция N_{10} на OYZ

Код программы

Файл interpolation_polynom.py

```
from ctypes import sizeof
import numpy as np
import matplotlib.pyplot as plt
import element as f
import sympy as sp


def get_x_matrix(variables):
    dimension = len(variables)
    X = np.zeros( (dimension, dimension) )
```

```
    for i in range(0, dimension):
        current_set = variables[i, :]
        X[i, : ] = f.fill_form_vector(current_set[0], current_set[1],
current_set[2])
    X = np.linalg.inv(X)
    X = X.T
    return X

def graph(F, step, title, xlabel, ylabel, projection):
    xl = np.linspace(0, 1, step)
    xgrid, ygrid = np.meshgrid(xl, xl)
    fig, ax = plt.subplots()
    cs = ax.contourf(xgrid, ygrid, F, levels = 10)
    cbar = plt.colorbar(cs)
    ax.set_xlabel("{}".format(xlabel))
    ax.set_ylabel("{}".format(ylabel))
    ax.set_title("{0}({1}, {2}) при {3} = 0" .format(title, xlabel, ylabel,
projection))
    ax.grid()
    plt.savefig("fileout/Функция формы({}).png".format(title))


def main():
    varis = f.getFigure()
    X = get_x_matrix(varis)
    print(X)
    step = 101
    step1 = 101
    N1 = np.zeros((step, step))
    N2 = np.zeros((step, step))
    N3 = np.zeros((step, step))
    N4 = np.zeros((step, step))
    N5 = np.zeros((step, step))
    N6 = np.zeros((step, step))
    N7 = np.zeros((step, step))
    N8 = np.zeros((step, step))
    N9 = np.zeros((step, step))
    N10 = np.zeros((step, step))
    xcoord = np.linspace(0, 1, step)
    ycoord = np.linspace(0, 1, step)
    zcoord = np.linspace(0, 1, step)
    for i in range(step):
        for j in range(step1):
            N1[i][j] = 1 - 3 * xcoord[i] - 3 * ycoord[j] + 2 * xcoord[i] ** 2 + 2
* ycoord[j] ** 2 + 4 * xcoord[i] * ycoord[j]
            N2[i][j] = -xcoord[i] + 2 * xcoord[i] ** 2
            N3[i][j] = -ycoord[j] + 2 * ycoord[j] ** 2
            N4[i][j] = -zcoord[j] + 2 * zcoord[j] ** 2
            N5[i][j] = 4 * xcoord[i] - 4 * xcoord[i] ** 2 - 4 * xcoord[i] *
ycoord[j]
            N6[i][j] = 4 * ycoord[j] - 4 * ycoord[j] ** 2 - 4 * xcoord[i] *
ycoord[j]
            N7[i][j] = 4 * zcoord[j] - 4 * zcoord[j] ** 2 - 4 * ycoord[i] *
zcoord[j]
            N8[i][j] = 4 * xcoord[i] * ycoord[j]
            N9[i][j] = 4 * zcoord[i] * xcoord[j]
            N10[i][j] = 4 * ycoord[i] * zcoord[j]

        step1 = step1 - 1
    graph(N1, step,'N1', 'x', 'y', 'z')
    graph(N2, step,'N2', 'x', 'y', 'z')
    graph(N3, step,'N3', 'x', 'y', 'z')
    graph(N4, step,'N4', 'y', 'z', 'x')
    graph(N5, step,'N5', 'x', 'y', 'z')
    graph(N6, step,'N6', 'x', 'y', 'z')
    graph(N7, step,'N7', 'y', 'z', 'x')
```

```
    graph(N8, step,'N8', 'x', 'y', 'z')
    graph(N9, step,'N9', 'z', 'x', 'y')
    graph(N10, step,'N10', 'y', 'z', 'x')


if __name__ == "__main__":
    main()
```

Файл element.py

```
import numpy as np

def getFigure():
    X= [[0, 0, 0],
   [1, 0, 0],
   [0, 1, 0],
   [0, 0, 1],
   [0.5, 0, 0],
   [0, 0.5, 0],
   [0, 0, 0.5],
   [0.5, 0.5, 0],
   [0.5, 0, 0.5],
   [0, 0.5, 0.5]]

    return np.array( X )



def fill_form_vector(x, y, z):
   return [1+0*x, x, y, z, x ** 2, y ** 2, z ** 2, x * y, x * z, y * z]
```