

Решение однородного уравнения Лапласа

Данько Д. И.

РУДН им. Патриса Лумумбы

Постановка задачи

Нужно написать программу графического представления решения уравнения теплопроводности. А также подсчитать разницу в значении относительно аналитического решения.

Код программы

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

N = 20
Tau = 10**(-6)

lenX = lenY = N
delta = 1

def f(r):
    return np.sin((5*r*np.pi/2)/N)

r = np.linspace(0, N, N)

Ttop = f(r)
Tbottom = 0
Tleft = 0
Tright = 0

Tguess = 10

colorinterpolation = 50
colourMap = plt.cm.jet

X, Y = np.meshgrid(np.arange(0, lenX), np.arange(0, lenY))

T = np.empty((lenX, lenY))
T.fill(Tguess)

print(np.shape(Ttop))
T[(lenY-1):, :] = Ttop
T[:, 0] = Tbottom
T[:, (lenX-1):] = Tright
T[:, :1] = Tleft

S = np.empty((N, N))
```

```
Error = 1
while Error > Tau:
    Error = 0
    for i in range(1, lenX-1, delta):
        for j in range(1, lenY-1, delta):
            T[i, j] = 0.25 * (T[i+1][j] + T[i-1][j] + T[i][j+1] + T[i][j-1])
            e = abs(T[i, j]-S[i,j])
            S[i,j] = T[i, j]
            if Error < e: Error = e
    print (Error)

plt.title("Contour of Temperature")
plt.contourf(X, Y, T, colorinterpolation, cmap=colourMap)
plt.colorbar()
plt.show()

print("")

ax = plt.axes(projection='3d')
ax.plot_surface(X, Y, T, cmap='viridis', edgecolor='green')
ax.set_title('Surface plot geeks for geeks')
plt.show()
```

Описание программы

- 1) Необходимо подключить нужные библиотеки и ввести постоянные параметры (количество разбиений и величину ошибки)

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

N = 20
Tau = 10**(-6)
```

- 2) Далее задаём граничные условия

```
def f(r):
    return np.sin((5*r*np.pi/2)/N)

r = np.linspace(0, N, N)

Ttop = f(r)
Tbottom = 0
Tleft = 0
Tright = 0
```

- 3) Далее создаём нашу матрицу значений и заполняем её Н.Д., которые, вообще говоря, неважны (пусть 10)

```
Tguess = 10

colorinterpolation = 50
colourMap = plt.cm.jet

X, Y = np.meshgrid(np.arange(0, lenX), np.arange(0, lenY))

T = np.empty((lenX, lenY))
T.fill(Tguess)
```

- 4) Задаём пустой массив для сравнения ошибки и саму ошибку

```
S = np.empty((N, N))

Error = 1
```

- 5) По циклу, пока ошибка не станет меньше заданной, высчитываем коэффициенты

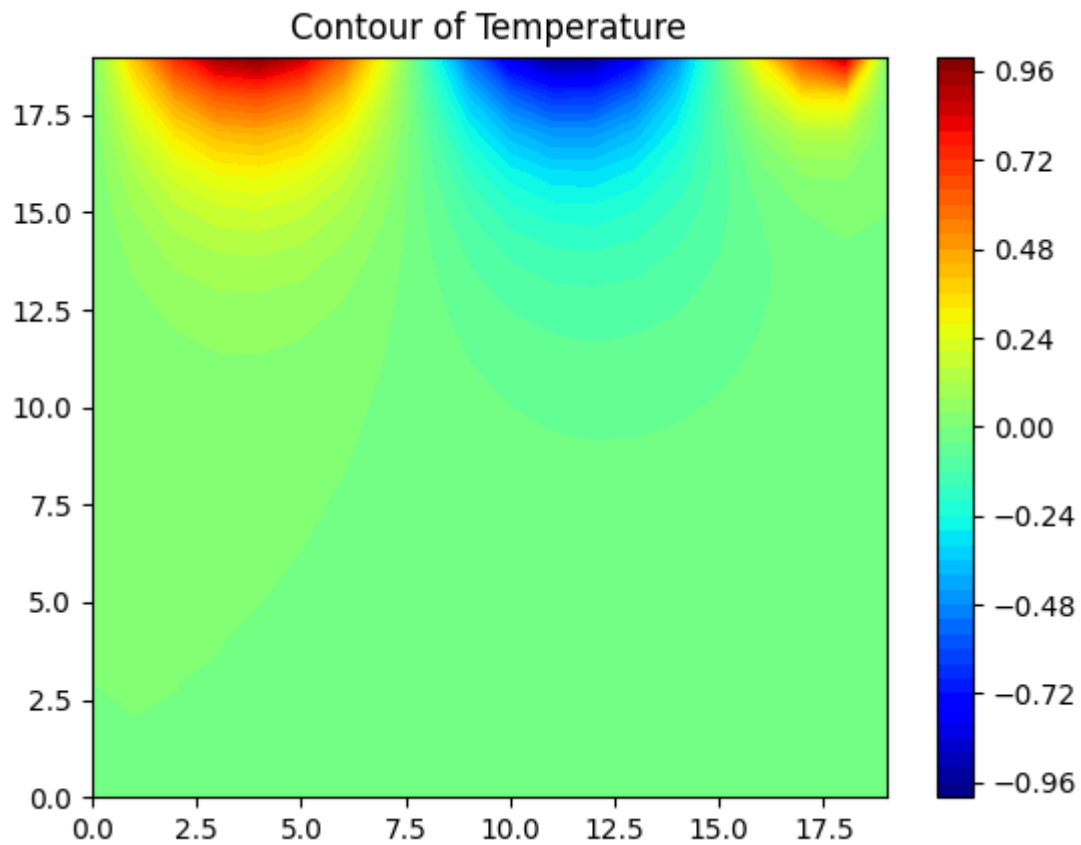
```
while Error > Tau:
    Error = 0
    for i in range(1, lenX-1, delta):
        for j in range(1, lenY-1, delta):
            T[i, j] = 0.25 * (T[i+1][j] + T[i-1][j] + T[i][j+1] + T[i][j-1])
            e = abs(T[i, j]-S[i,j])
            S[i,j] = T[i, j]
            if Error < e: Error = e
    print (Error)
```

- 6) Далее выводим наши результаты с помощью двух графиков

```
plt.title("Contour of Temperature")
plt.contourf(X, Y, T, colorinterpolation, cmap=colourMap)
plt.colorbar()
plt.show()

print("")
```

```
ax = plt.axes(projection='3d')  
ax.plot_surface(X, Y, T, cmap='viridis', edgecolor='green')  
ax.set_title('Surface plot geeks for geeks')  
plt.show()
```



Contour of Temperature

