

Решение неоднородного уравнения Лапласа

Лабораторная работ №4 и №5

Данько Д. И.

РУДН им. Патриса Лумумбы

Постановка задачи

Нужно написать программу графического представления решения одномерного уравнения теплопроводности. А также подсчитать разницу в значении относительно аналитического и численных решений. Работы отличаются точностью интерпретации граничных условий

$$\frac{d^2 u}{dx^2} = \sin x \quad \left[\frac{\pi}{2}, \frac{3\pi}{2} \right]$$

$$\frac{du}{dx} \Big|_{x=\frac{\pi}{2}} = 0$$

$$u \Big|_{x=\frac{3\pi}{2}} = 1$$

1) $O(h)$ $\frac{u_{i+1} - u_{i-1}}{h} = 0$ ($= 0$)

2) $O(h^2)$ $\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = 0$

$\frac{u_2 - u_0}{2h} = 0$

$\frac{1}{h^2}(u_2 - 2u_1 + u_0) = \sin(x_1)$

$u_0 = d_1 u_1 + p_1$

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 = 100

X_0 = np.pi/2
X_N = 3*np.pi/2
U_N = 1
h = (X_N - X_0)/N
x = np.zeros(N+1)
for i in range(N+1):
    x[i] = np.pi/2 + h*i

def f(x): return np.sin(x)

u = np.zeros(N+1)
```

2) Далее задаём 3-х диагональную матрицу и массивы решений

```
A, B, C = 1, -2, 1

a, b = np.zeros(N+1), np.zeros(N+1)
a[0] = 1
b[0] = 0

for i in range(0, N):
    a[i+1] = ((-1)*C)/(A*a[i]+B)
    b[i+1] = (f(x[i])*h*h - A*b[i])/(A*a[i]+B)
```

3) Далее определяем функции меры и решаем уравнение

```
def Measure(A, B):
    N = len(B)
    s = 0
    for i in range(N):
        s = s + (abs(A[i] - B[i]))**2

def M(A, B):
    N = len(B)
    s = 0
    for i in range(N):
        if s <= abs(A[i] - B[i]): s = abs(A[i] - B[i])
    return s

u[N] = 1
for i in range(N-1, -1, -1):
    u[i] = a[i]*u[i+1]+b[i]

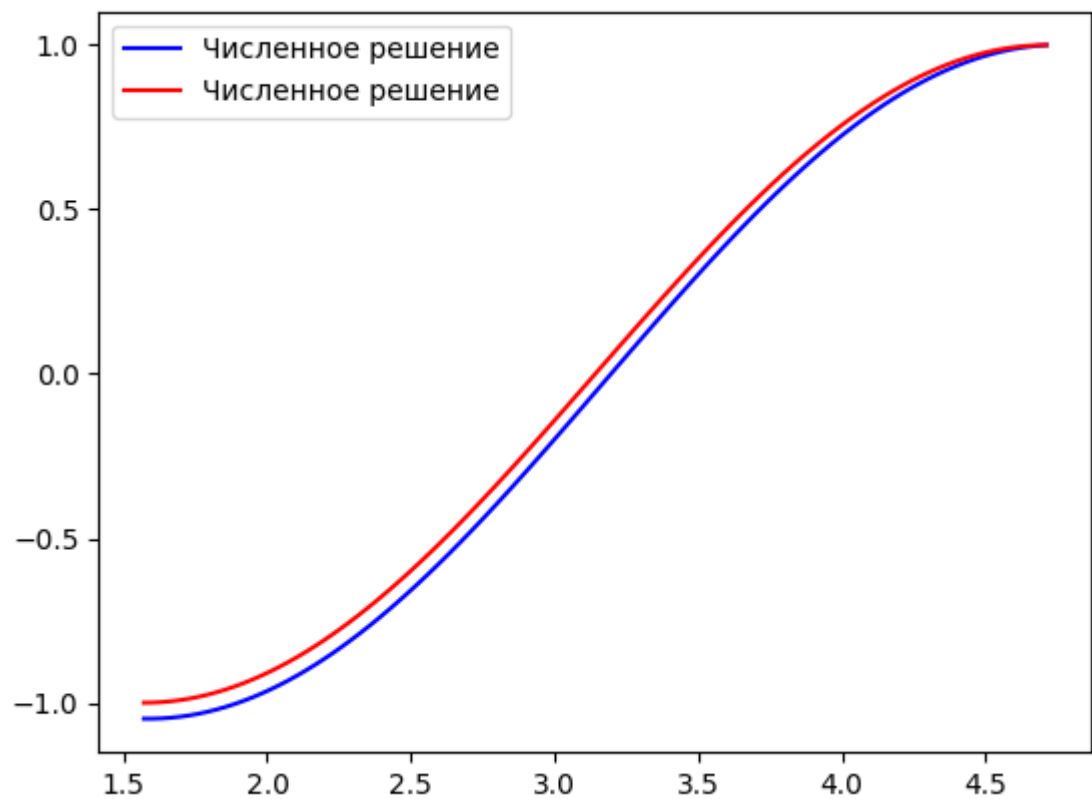
analit = [-f(x[i]) for i in range(N+1)]

print (M(analit, u))

t = np.linspace(X_0, X_N, N+1)
plt.plot(t, u, color = 'blue', label = 'Численное решение')
plt.plot(t, -f(t), color = 'r', label = 'Численное решение')
plt.legend()
plt.show()
```

На выдачу имеем абсолютную абсолютную ошибку: 0.05948807575885373

И соответственно график:



4) Аналогично для 5 лабораторной

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

```
Error = 1
```

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

```
def Measure(A, B):
    N = len(B)
    s = 0
    for i in range(N):
        s = s + (abs(A[i] - B[i]))**2

    return np.sqrt(s)
```

```
N = 100
```

```
def M(A, B):
    N = len(B)
```

```

s = 0
for i in range(N):
    if s <= abs(A[i] - B[i]): s= abs(A[i] - B[i])
return s
X_0 = np.pi/2
X_N = 3*np.pi/2
U_N = 1
h = (X_N - X_0)/N
x = np.zeros(N+1)
for i in range(N+1):
    x[i] = np.pi/2 + h*i

def f(x): return np.sin(x)

u = np.zeros(N+1)
A, B, C = 1, -2, 1

a, b = np.zeros(N+1), np.zeros(N+1)
a[0] = 1
b[0] = f(x[0])*h*h/2

for i in range(0, N):
    a[i+1] = ((-1)*C)/(A*a[i]+B)
    b[i+1] = (f(x[i])*h*h - A*b[i])/(A*a[i]+B)

u[N] = 1
for i in range(N-1, -1, -1):
    u[i] = a[i]*u[i+1]+b[i]

# print (u)

analit = [-f(x[i]) for i in range(N+1)]

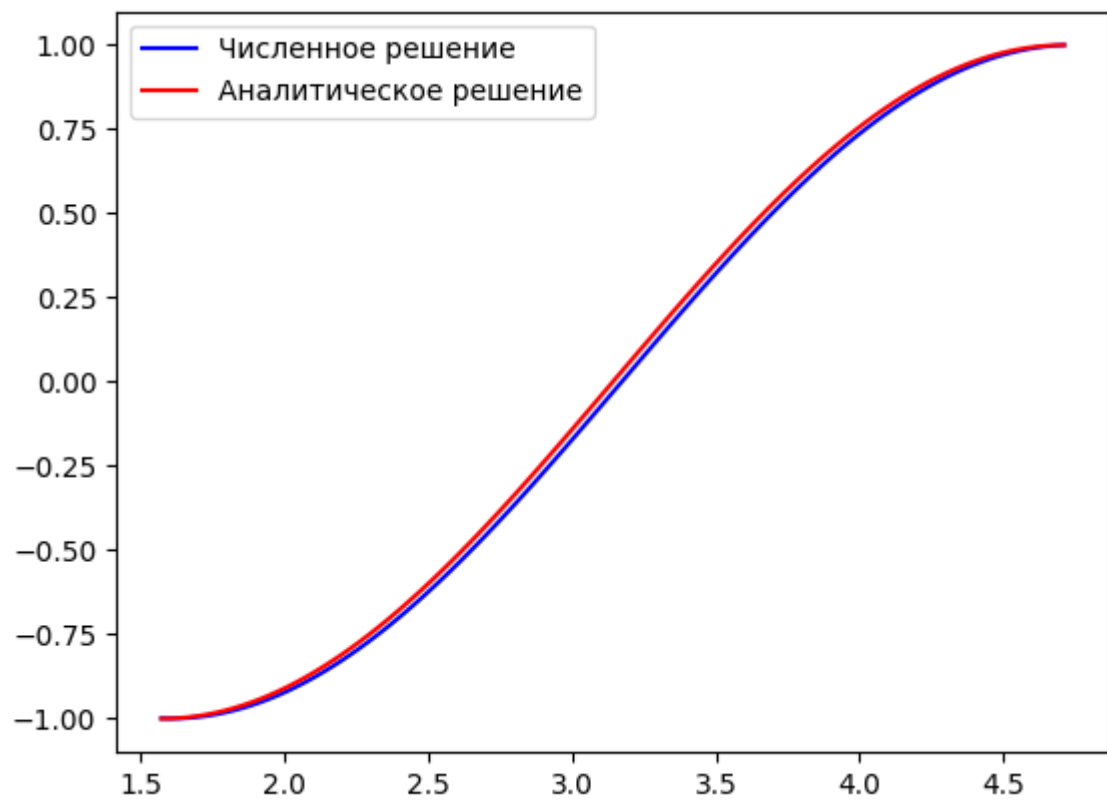
print (M(analit, u))

t = np.linspace(X_0, X_N, N+1)
plt.plot(t, u, color = 'blue', label = 'Численное решение')
plt.plot(t, -f(t), color = 'r', label = 'Аналитическое решение')
plt.legend()
plt.show()

```

На выдачу имеем абсолютную абсолютную ошибку: 0.031002113179163283

И соответственно график:



Вывод

При втором решении количество итераций не изменилось, но ошибка меньше.