

Практическое задание по ОММ

Задача #2

Выполнил студент 335 группы
Будамян Я. С.
Преподаватели: Пикун В. М.
Домбровская Ж. О.

1 Постановка задачи

Задача 2. Используя метод переменных направлений, решите краевую задачу:

$$\left\{ \begin{array}{l} \frac{\partial u}{\partial t} = \Delta u \quad 0 < x < 1, \quad 0 < y < 2, \quad t > 0 \\ u|_{x=0} = u|_{x=1} = 0, \\ u|_{y=0} = u|_{y=2} = 0, \\ u|_{t=0} = \sin 2\pi x \sin \pi y \end{array} \right. \quad (1)$$

2 Метод решения

2.1 Метод переменных направлений

Для численного решения задачи, введем двумерную пространственную и временную равномерные сетки:

$$\begin{aligned} \bar{\omega}_{h_1} &= \{x_i = ih_1; \ i = \overline{0, N_1}; \ h_1 N_1 = 1\} \\ \bar{\omega}_{h_2} &= \{y_j = jh_2; \ j = \overline{0, N_2}; \ h_2 N_2 = 2\} \\ \bar{\omega}_\tau &= \{t_k = k\tau; \ k = \overline{0, S}; \ \tau S = T\} \\ \bar{\omega}_{h_1 h_2 \tau} &= \bar{\omega}_{h_1} \times \bar{\omega}_{h_2} \times \bar{\omega}_\tau = \{(x_i, y_j, t_k) \in \bar{D}\} \\ \bar{D} &= \{0 < x < 1; \ 0 < y < 2; \ 0 \leq t \leq T\} \end{aligned}$$

Введем сеточную функцию:

$$\omega_{h_1 h_2}^k \stackrel{\text{def}}{=} u(x_i, y_j, t_k)$$

Запишем разностную аппроксимацию оператора Лапласа:

$$L\omega \rightarrow \Lambda\omega = \Lambda_1\omega + \Lambda_2\omega,$$

где

$$\Lambda_1\omega = \frac{\omega_{i-1,j} - 2\omega_{ij} + \omega_{i+1,j}}{h_1^2}$$

$$\Lambda_2 \omega = \frac{\omega_{i,j-1} - 2\omega_{ij} + \omega_{i,j+1}}{h_2^2}$$

При решении данной задачи используется схема переменных направлений, являющаяся экономичной разностной схемой. Такая схема сочетает в себе ряд преимуществ, таких как сложность $O(N_1 N_2)$ и безусловная устойчивость. В этой схеме переход с одного временного слоя на другой происходит в 2 шага, привлекая промежуточный (дробный) слой. Разностная аппроксимация примет вид:

$$\begin{cases} \frac{\omega^{k+\frac{1}{2}} - \omega^k}{0.5\tau} = \Lambda_1 \omega^{k+\frac{1}{2}} + \Lambda_2 \omega^k \\ \frac{\omega^{k+1} - \omega^{k+\frac{1}{2}}}{0.5\tau} = \Lambda_1 \omega^{k+\frac{1}{2}} + \Lambda_2 \omega^{k+1} \end{cases} \quad (2)$$

Рассмотрим переход $k \rightarrow k + \frac{1}{2}$. Используя явный вид операторов Λ_1 и Λ_2 , имеем:

$$\begin{cases} \frac{\gamma_1}{2} \omega_{i-1,j}^{k+\frac{1}{2}} - (1 + \gamma_1) \omega_{ij}^{k+\frac{1}{2}} + \frac{\gamma_1}{2} \omega_{i+1,j}^{k+\frac{1}{2}} = -\frac{\gamma_2}{2} \omega_{i,j-1}^k - (1 - \gamma_2) \omega_{ij}^k - \frac{\gamma_2}{2} \omega_{i,j+1}^k \\ \omega_{0j}^{k+\frac{1}{2}} = \omega_{N_1,j}^{k+\frac{1}{2}} = 0 \quad j = \overline{1, N_2 - 1}, \end{cases} \quad (3)$$

где $\gamma_1 = \frac{\tau}{h_1^2}$, $\gamma_2 = \frac{\tau}{h_2^2}$. Введя обозначения:

$$A^{(1)} = B^{(1)} = \frac{\gamma_1}{2}$$

$$C^{(1)} = 1 + \gamma_1$$

$$F_{ij}^k = \frac{\gamma_2}{2} \omega_{i,j-1}^k + (1 - \gamma_2) \omega_{ij}^k + \frac{\gamma_2}{2} \omega_{i,j+1}^k,$$

получим систему:

$$\begin{cases} A^{(1)} \omega_{i-1,j}^{k+\frac{1}{2}} - C^{(1)} \omega_{ij}^{k+\frac{1}{2}} + B^{(1)} \omega_{i+1,j}^{k+\frac{1}{2}} = -F_{ij}^k \\ \omega_{0j}^{k+\frac{1}{2}} = \omega_{N_1,j}^{k+\frac{1}{2}} = 0 \quad j = \overline{1, N_2 - 1} \end{cases} \quad (4)$$

Аналогично, вводя обозначения

$$A^{(2)} = B^{(2)} = \frac{\gamma_2}{2}$$

$$C^{(2)} = 1 + \gamma_2$$

$$F_{ij}^{k+\frac{1}{2}} = \frac{\gamma_1}{2} \omega_{i-1,j}^{k+\frac{1}{2}} + (1 - \gamma_2) \omega_{ij}^{k+\frac{1}{2}} + \frac{\gamma_2}{2} \omega_{i+1,j}^{k+\frac{1}{2}},$$

получим систему для перехода $k + \frac{1}{2} \rightarrow k + 1$:

$$\begin{cases} A^{(2)} \omega_{i,j-1}^{k+1} - C^{(2)} \omega_{ij}^{k+1} + B^{(2)} \omega_{i,j+1}^{k+1} = -F_{ij}^{k+\frac{1}{2}} \\ \omega_{i0}^{k+1} = \omega_{i,N_2}^{k+1} = 0 \quad i = \overline{1, N_1 - 1} \end{cases} \quad (5)$$

Полученные системы (4), (5) решаются методом прогонки.

2.2 Метод прогонки

Метод прогонки используется для решения систем линейных уравнений вида $Ax = F$, где F - трехдиагональная матрица. Пусть дана система уравнений

$$\begin{cases} A_n y_{n-1} - C_n y_n + B_n y_{n+1} = -F_n, & n = \overline{1, N-1} \\ A_n \neq 0, \quad B_n \neq 0, & n = \overline{1, N-1} \\ y_0 = k_1 y_1 + \mu_1, \quad y_N = k_2 y_{N-1} + \mu_2 \end{cases} \quad (6)$$

Ее решение будем искать в виде:

$$y_n = \alpha_{n+1} y_{n+1} + \beta_{n+1}, \quad n = \overline{1, N-1}$$

Подставляя и исключая y_{n-1} и y_n , получаем:

$$[(A_n \alpha_n - C_n) \alpha_{n+1} + B_n] y_{n+1} + [(A_n \alpha_n - C_n) \beta_{n+1} + (A_n \beta_n + F_n)]$$

Получившееся уравнение будет удовлетворено вместе с граничными условиями, если

$$\begin{aligned} \alpha_1 &= k_1, \quad \beta_1 = \mu_1 \\ \alpha_{n+1} &= \frac{B_n}{C_n - \alpha_n A_n}, \quad \beta_{n+1} = \frac{A_n \beta_n + F_n}{C_n - \alpha_n A_n}, \quad n = \overline{1, N-1} \\ y_N &= \frac{\mu_2 + \beta_N k_2}{1 - \alpha_N k_2} \end{aligned}$$

Используя эти формулы, сначала вычисляются все коэффициенты α_n, β_n , а потом все неизвестные $y_{N-1}, y_{N-2}, \dots, y_0$.

Достаточные условия устойчивости метода прогонки:

$$|C_n| \geq |A_n| + |B_n|, \quad n = \overline{1, N-1}$$

$$|k_\alpha| \leq 1, \quad \alpha = 1, 2, \quad |k_1| + |k_2| < 2$$

Проверим эти условия для системы (4):

$$1 + \gamma_1 = |C^{(1)}| \geq |A^{(1)}| + |B^{(1)}| = \gamma_1$$

$$k_1 = k_2 \equiv 0$$

Аналогично, условия выполняются и для системы (5).

3 Порядок аппроксимации

4 Устойчивость схемы

Пусть $\delta\omega$ - погрешность решения. Она удовлетворяет системе:

$$\begin{cases} \frac{\delta\omega^{k+\frac{1}{2}} - \delta\omega^k}{0.5\tau} = \Lambda_1 \delta\omega^{k+\frac{1}{2}} + \Lambda_2 \delta\omega^k \\ \frac{\delta\omega^{k+1} - \delta\omega^{k+\frac{1}{2}}}{0.5\tau} = \Lambda_1 \delta\omega^{k+\frac{1}{2}} + \Lambda_2 \delta\omega^{k+1} \end{cases} \quad (7)$$

Будем искать для этой системы частные решения с разделенными переменными в виде:

$$\delta\omega_{n,m}^k = e^{i(\alpha n + \beta m)}$$

$$\delta\omega_{n,m}^{k+\frac{1}{2}} = \rho_1 \delta\omega_{n,m}^k$$

$$\delta\omega_{n,m}^{k+1} = \rho_2 \delta\omega_{n,m}^{k+\frac{1}{2}}$$

Подставляя в систему и сокращая на $e^{i(\alpha n + \beta m)}$, получаем систему:

$$\begin{cases} 2(\rho_1 - 1) = \gamma_1 \rho_1 (e^{-i\alpha} + e^{i\alpha} - 2) + \gamma_2 (e^{-i\beta} + e^{i\beta} - 2) \\ 2(\rho_2 - 1) = \gamma_1 (e^{-i\alpha} + e^{i\alpha} - 2) + \gamma_2 \rho_2 (e^{-i\beta} + e^{i\beta} - 2) \end{cases} \quad (8)$$

Решая, находим:

$$\rho_1 = \frac{1 - \gamma_2 \sin^2\left(\frac{\beta}{2}\right)}{1 + \gamma_1 \sin^2\left(\frac{\alpha}{2}\right)}$$

$$\rho_2 = \frac{1 - \gamma_1 \sin^2\left(\frac{\alpha}{2}\right)}{1 + \gamma_2 \sin^2\left(\frac{\beta}{2}\right)}$$

$$|\rho_1| |\rho_2| \leq 1$$

Следовательно, при переходе $k \rightarrow k + 1$ ошибки не накапливаются и схема безусловно устойчива.

5 Результаты

Графики, получившиеся в результате численного моделирования по описанному алгоритму приведены в конце работы. Вид получившихся графиков позволяет предположить, что можно искать аналитическое решение поставленной задачи в виде

$$u(t, x, y) = T(t) \sin(2\pi x) \sin(\pi y)$$

Подставляя в уравнение и условия, получаем задачу для функции T :

$$\begin{cases} \frac{dT}{dt} = -5\pi^2 T \\ T(0) = 1 \end{cases} \quad (9)$$

Ее решением является функция $T(t) = e^{-5\pi^2 t}$. Следовательно, искомое аналитическое решение есть

$$u(t, x, y) = e^{-5\pi^2 t} \sin(2\pi x) \sin(\pi y)$$

6 Листинг программы

```
1  #-*- coding: utf-8 -*-
2  """
3  Created on Fri Jun 17 14:43:10 2016
4
5  @author: Yan
6  """
7
8  import numpy as np
9  import matplotlib.pyplot as plt
10 from mpl_toolkits.mplot3d import Axes3D
11 from math import sin
12
13 #функция граничных условий
14 def u_t0(x,y):
15     return sin(2*np.pi*x) * sin(np.pi*y)
16
17 #TMA - tridiagonal matrix algorithm - метод прогонки
18 def TMA(A, B, C, F, k1, k2, mu1, mu2):
19     N = A.size
20     w = np.zeros(N+1)
21     alpha = np.zeros(N+1)
22     beta = np.zeros(N+1)
23
24     #прямой ход прогонки
25     alpha[1] = k1
26     beta[1] = mu1
27
28     for n in range(1, N):
29         alpha[n+1] = B[n] / (C[n] - alpha[n]*A[n])
30         beta[n+1] = (A[n]*beta[n] + F[n]) / (C[n] - alpha[n]*A[n])
31
32     #обратный ход
33     w[N] = (mu2 + beta[N]*k2) / (1 - alpha[N]*k2)
34
35     for n in range(N-1, -1, -1):
36         w[n] = alpha[n+1]*w[n+1] + beta[n+1]
37
38     return w
39
40
41 #шаги по времени и координатам
42 N1 = 20
43 N2 = 40
44 S = 100
45
46 #рассматриваемый промежуток времени
47 T = 0.07
48
49 h1 = 1 / N1
50 h2 = 2 / N2
51 tau = T / S
52
53 #коэффициенты гамма_1 и гамма_2
54 g1 = tau / h1**2
55 g2 = tau / h2**2
56
57 #массив для искомого решения
58 #индексы: время(k) - x(i) - y(j)
59 u = np.zeros((S+1, N1+1, N2+1))
60 #массив для хранения значений на промежуточном слое k+1/2
61 u12 = np.zeros((S+1, N1+1, N2+1))
62
63 #заполняем граничные условия
64 for i in range(0, N1):
65     for j in range(0, N2):
66         u[0][i][j] = u_t0(i*h1, j*h2)
67
68 #основной цикл расчета
69 for k in range(S):
70     #шаг k -> k + 1/2
71     A = np.zeros(N1)
72     B = np.zeros(N1)
73     C = np.zeros(N1)
74     F = np.zeros(N1)
```

```

75
76     k1 = 0
77     k2 = 0
78     mu1 = 0
79     mu2 = 0
80
81     for j in range(1, N2):
82         for i in range(1, N1):
83             A[i] = g1 / 2
84             B[i] = g1 / 2
85             C[i] = 1 + g1
86             F[i] = (g2 / 2) * (u[k][i][j-1] + u[k][i][j+1]) + (1 - g2) * u[k][i][j]
87
88     w = TMA(A, B, C, F, k1, k2, mu1, mu2)
89
90     for i in range(N1+1):
91         u12[k][i][j] = w[i]
92
93     # шаг k + 1/2 -> k + 1
94     A = np.zeros(N2)
95     B = np.zeros(N2)
96     C = np.zeros(N2)
97     F = np.zeros(N2)
98
99     # значения k1, k2, mu1, mu2 не менялись
100    for i in range(1, N1):
101        for j in range(1, N2):
102            A[j] = g2 / 2
103            B[j] = g2 / 2
104            C[j] = 1 + g2
105            F[j] = (g1 / 2) * (u12[k][i-1][j] + u12[k][i+1][j]) + (1 - g1) * u12[k][i][j]
106
107    w = TMA(A, B, C, F, k1, k2, mu1, mu2)
108
109    for j in range(N2+1):
110        u[k+1][i][j] = w[j]
111
112    #визуализация
113    ax = Axes3D(plt.figure())
114    X = np.arange(0, 1 + h1, h1)
115    Y = np.arange(0, 2 + h2, h2)
116    X, Y = np.meshgrid(X, Y)
117    Z = np.transpose(np.array(u[S]))
118
119    ax.plot_surface(X, Y, Z, rstride=1, cstride=1, color='1')
120    ax.set_zlim3d(-1, 1)
121    ax.set_xlabel('x')
122    ax.set_ylabel('y')
123    ax.set_zlabel('u(x, y, {time})'.format(time=str(T)))
124    ax.view_init(elev=30, azim=230)
125
126    plt.show()

```

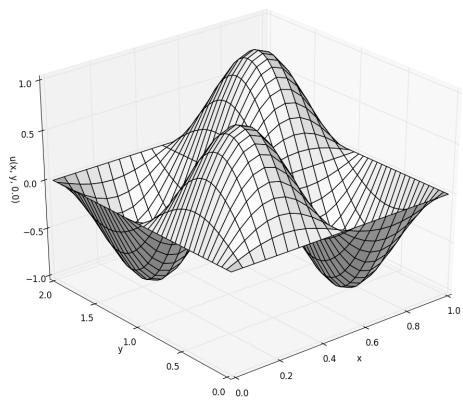



Рис. 1: $T = 0$

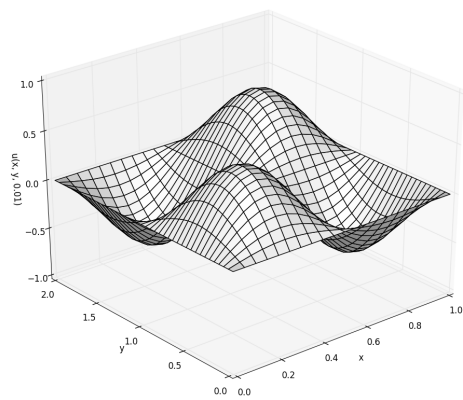


Рис. 2: $T = 0.01$

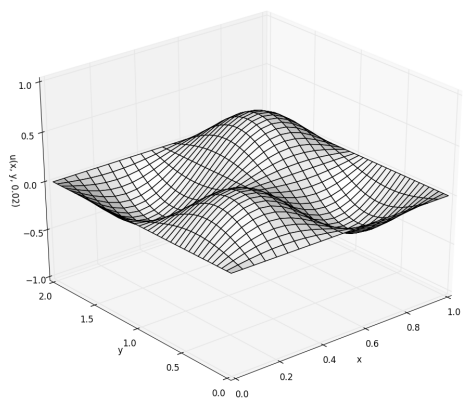


Рис. 3: $T = 0.02$

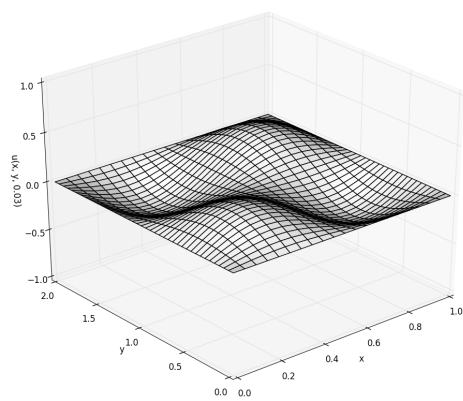


Рис. 4: $T = 0.03$

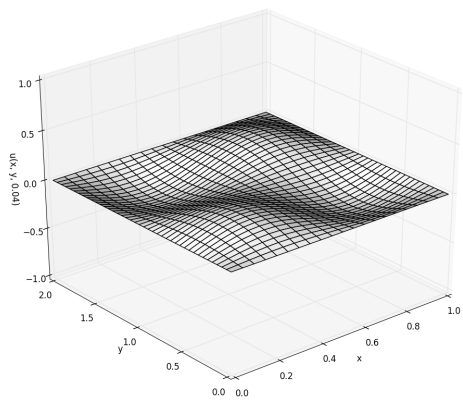


Рис. 5: $T = 0.04$

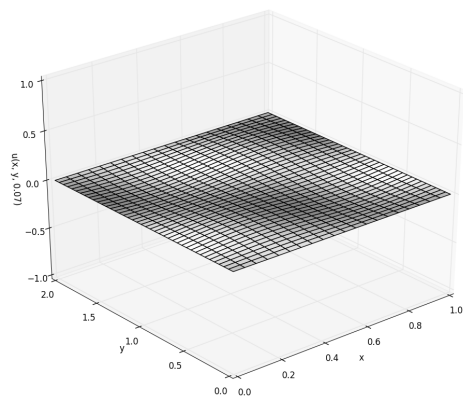


Рис. 6: $T = 0.07$