

Содержание

1	Задание	2
2	Теоретическая часть	2
2.1	Тестовые примеры	2
2.1.1	Плоскость	2
2.1.2	Линейно-квадратичная функция	2
2.1.3	Параболоид	3
2.1.4	Индивидуальная функция	3
2.2	Методы решения сеточных уравнений	3
2.2.1	$\omega - Jac$ (метод простой итерации с параметром)	3
2.2.2	Jac (метод простой итерации)	3
3	Практическая часть	4
3.1	Jac	4
3.1.1	Тестовая задача №1	4
3.1.2	Тестовая задача №2	4
3.1.3	Тестовая задача №3	5
3.1.4	Тестовая задача №4	5
3.2	$\omega - Jac$	5
3.2.1	Тестовая задача №1	5
3.2.2	Тестовая задача №2	6
3.2.3	Тестовая задача №3	6
3.2.4	Тестовая задача №4	7
4	Приложение	7

1 Задание

В единичном квадрате $\Omega = (0,1) \times (0,1)$ рассматривается краевая задача:

$$\begin{cases} -\Delta u = f, \\ u|_{\partial\Omega} = g. \end{cases} \quad (1)$$

Реализовать РС «крест». Для решения системы сеточных уравнений использовать два метода: *Jac*, $\omega - Jac$ с обходом по столбцам снизу вверх, справа налево.

Реализовать серию вычислительных экспериментов на стандартных тестах. Сравнить время работы методов, необходимое для достижения точности ε (задается пользователем). Составить (для каждого тестового расчета) таблицы зависимости времени расчета и количества итераций от точности и шага сетки h (по обоим переменным).

2 Теоретическая часть

2.1 Тестовые примеры

2.1.1 Плоскость

$$u(x,y) = 2x + 3y - 5$$

$$\begin{cases} -\Delta u = 0, \\ u|_{\partial\Omega} = \begin{cases} 2x - 5, y = 0 \\ 3y - 5, x = 0 \\ 3y - 3, x = 1 \\ 2x - 2, y = 1 \end{cases} \end{cases} \quad (2)$$

2.1.2 Линейно-квадратичная функция

$$u(x,y) = 2x^2 + y - 3$$

$$\begin{cases} -\Delta u = -4, \\ u|_{\partial\Omega} = \begin{cases} 2x^2 - 3, y = 0 \\ y - 3, x = 0 \\ 2x^2 - 2, y = 1 \\ y - 1, x = 1 \end{cases} \end{cases} \quad (3)$$

2.1.3 Параболоид

$$u(x, y) = 3x^2 - 2y^2 - 1$$

$$\begin{cases} -\Delta u = -2, \\ u|_{\partial\Omega} = \begin{cases} 3x^2 - 1, y = 0 \\ -2y^2 - 1, x = 0 \\ 2 - 2y^2, x = 1 \\ 3x^2 - 3, y = 1 \end{cases} \end{cases} \quad (4)$$

2.1.4 Индивидуальная функция

$$u(x, y) = e^{\sin^2 x + \cos^2 y}$$

$$\begin{cases} -\Delta u = -e^{\sin^2 x + \cos^2 y} (\sin^2 2x + 2 \cos 2x + \sin 2y - 2 \cos 2y) \\ u|_{\partial\Omega} = \begin{cases} e^{\sin^2 x + 1}, y = 0 \\ e^{\cos^2 y}, x = 0 \\ e^{\sin^2 x + \cos^2 1}, y = 1 \\ e^{\sin^2 1 + \cos^2 y}, x = 1 \end{cases} \end{cases} \quad (5)$$

2.2 Методы решения сеточных уравнений

2.2.1 $\omega - Jac$ (метод простой итерации с параметром)

$$U_{i,j}^{(m+1)} = (1 - \omega) U_{i,j}^{(m)} + \frac{\omega}{4} \left(U_{i-1,j}^{(m)} + U_{i+1,j}^{(m)} + U_{i,j-1}^{(m)} + U_{i,j+1}^{(m)} \right) + \frac{\omega h^2}{4} F_{i,j}$$

Индексы i и j изменяются в цикле от 1 до $N-1$ (каждый) в произвольном порядке. Параметр $\omega \in (0, 1)$.

2.2.2 Jac (метод простой итерации)

$$\frac{U_{i-1,j}^{(m)} - 2U_{i,j}^{(m+1)} + U_{i+1,j}^{(m)}}{h_1^2} + \frac{U_{i,j-1}^{(m)} - 2U_{i,j}^{(m+1)} + U_{i,j+1}^{(m)}}{h_2^2} = -F_{i,j}$$

Расчетные формулы для программирования:

$$U_{i,j}^{(m+1)} = C_0 F_{i,j} + C_1 \left(U_{i,j-1}^{(m)} + U_{i,j+1}^{(m)} \right) + C_2 \left(U_{i-1,j}^{(m)} + U_{i+1,j}^{(m)} \right)$$

где

$$C_0 = \frac{0.5h_1^2 h_2^2}{h_1^2 + h_2^2}, \quad C_1 = \frac{0.5h_1^2}{h_1^2 + h_2^2}, \quad C_2 = \frac{0.5h_2^2}{h_1^2 + h_2^2}$$

3 Практическая часть

3.1 Jac

$$h_1 = h_2 = h_0 = 0.1, \varepsilon_0 = 0.1$$

3.1.1 Тестовая задача №1

Время расчета(в секундах) от точности и шага сетки

	h_0	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
$\varepsilon_0/1$	0.0023	0.0086	0.036	0.1464	0.6671
$\varepsilon_0/10$	0.0127	0.0725	0.3163	1.334	5.482
$\varepsilon_0/100$	0.0182	0.217	2.123	12.9	51.89
$\varepsilon_0/1000$	0.0274	0.3697	4.541	50.1	433.4

Число итераций от точности и шага сетки

	h_0	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
$\varepsilon_0/1$	10	10	11	11	12
$\varepsilon_0/10$	50	94	95	101	105
$\varepsilon_0/100$	96	276	656	976	970
$\varepsilon_0/1000$	142	462	1400	3803	8123

3.1.2 Тестовая задача №2

Время расчета(в секундах) от точности и шага сетки

	h_0	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
$\varepsilon_0/1$	0.0013	0.004	0.0156	0.0663	0.257
$\varepsilon_0/10$	0.0059	0.0241	0.1098	0.4713	1.948
$\varepsilon_0/100$	0.0148	0.1489	1.066	4.881	18.3
$\varepsilon_0/1000$	0.0227	0.278	3.255	32.24	171.5

Число итераций от точности и шага сетки

	h_0	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
$\varepsilon_0/1$	4	4	4	4	4
$\varepsilon_0/10$	29	31	34	35	36
$\varepsilon_0/100$	74	187	323	319	338
$\varepsilon_0/1000$	120	372	1044	2390	3148

3.1.3 Тестовая задача №3

Время расчета(в секундах) от точности и шага сетки

	h_0	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
$\varepsilon_0/1$	0.0018	0.0068	0.0263	0.105	0.5103
$\varepsilon_0/10$	0.0069	0.0404	0.1966	0.8841	3.872
$\varepsilon_0/100$	0.0161	0.1614	1.304	6.852	32.81
$\varepsilon_0/1000$	0.026	0.3188	3.703	35.55	243.4

Число итераций от точности и шага сетки

	h_0	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
$\varepsilon_0/1$	6	7	7	7	8
$\varepsilon_0/10$	33	50	58	65	69
$\varepsilon_0/100$	78	202	398	513	604
$\varepsilon_0/1000$	124	387	1100	2625	4602

3.1.4 Тестовая задача №4

Время расчета(в секундах) от точности и шага сетки

	h_0	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
$\varepsilon_0/1$	0.0086	0.0423	0.1827	0.7716	3.067
$\varepsilon_0/10$	0.0364	0.2554	1.389	6.422	28.28
$\varepsilon_0/100$	0.0993	0.9446	8.053	47.99	234.6
$\varepsilon_0/1000$	0.123	1.663	20.41	213.6	1587

Число итераций от точности и шага сетки

	h_0	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
$\varepsilon_0/1$	8	9	10	10	10
$\varepsilon_0/10$	38	61	78	89	95
$\varepsilon_0/100$	84	223	463	661	821
$\varepsilon_0/1000$	130	408	1188	2961	5422

3.2 $\omega - Jac$

$$h_1 = h_2 = h_0 = 0.1, \varepsilon_0 = 0.1, \omega = 0.8$$

3.2.1 Тестовая задача №1

Время расчета(в секундах) от точности и шага сетки

	h_0	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
$\varepsilon_0/1$	0.0026	0.0103	0.0428	0.1942	0.8751
$\varepsilon_0/10$	0.0133	0.0943	0.3833	1.718	7.519
$\varepsilon_0/100$	0.0267	0.3062	2.96	16.51	71.39
$\varepsilon_0/1000$	0.0431	0.5913	7.153	76.67	629.9

Число итераций от точности и шага сетки

	h_0	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
$\varepsilon_0/1$	10	10	10	11	11
$\varepsilon_0/10$	57	98	96	102	105
$\varepsilon_0/100$	115	322	731	936	980
$\varepsilon_0/1000$	173	555	1659	4392	8798

3.2.2 Тестовая задача №2

Время расчета(в секундах) от точности и шага сетки

	h_0	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
$\varepsilon_0/1$	0.0011	0.0048	0.0217	0.1126	0.3279
$\varepsilon_0/10$	0.0073	0.0303	0.1383	0.5952	2.571
$\varepsilon_0/100$	0.0198	0.2023	1.296	5.304	22.96
$\varepsilon_0/1000$	0.0392	0.4402	5.254	44.9	205.1

Число итераций от точности и шага сетки

	h_0	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
$\varepsilon_0/1$	3	4	4	4	4
$\varepsilon_0/10$	31	32	34	35	36
$\varepsilon_0/100$	88	211	327	322	340
$\varepsilon_0/1000$	145	444	1214	2638	3074

3.2.3 Тестовая задача №3

Время расчета(в секундах) от точности и шага сетки

	h_0	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
$\varepsilon_0/1$	0.0018	0.0077	0.0308	0.1329	0.6415
$\varepsilon_0/10$	0.0082	0.0528	0.2559	1.115	4.743
$\varepsilon_0/100$	0.0203	0.2242	1.744	9.501	40.8
$\varepsilon_0/1000$	0.0355	0.4507	5.412	50.02	324.8

Число итераций от точности и шага сетки

	h_0	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
$\varepsilon_0/1$	6	7	7	7	8
$\varepsilon_0/10$	36	50	59	65	69
$\varepsilon_0/100$	92	229	429	530	615
$\varepsilon_0/1000$	149	461	1284	2942	4812

3.2.4 Тестовая задача №4

Время расчета(в секундах) от точности и шага сетки

	h_0	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
$\varepsilon_0/1$	0.0086	0.0399	0.1834	0.7685	3.13
$\varepsilon_0/10$	0.0404	0.2913	1.698	6.741	28.15
$\varepsilon_0/100$	0.0952	1.113	8.769	51.49	261.3
$\varepsilon_0/1000$	0.1581	2.115	25.95	253.9	1752

Число итераций от точности и шага сетки

	h_0	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
$\varepsilon_0/1$	8	9	9	10	10
$\varepsilon_0/10$	42	63	80	90	96
$\varepsilon_0/100$	99	256	502	690	840
$\varepsilon_0/1000$	156	488	1394	3348	5744

Вывод В ходе выполнения лабораторной работы были произведены вычислительные эксперименты, а именно использовались метод Якоби и Якоби с параметром для решения сеточных уравнений РС «крест». Я выяснил, что метод простой итерации с параметром совпадает с методом простой итерации, при выборе оптимального параметра. Также увидел зависимость числа итераций от шага: при уменьшении шага, увеличивается число итераций. Если смотреть на время вычислений обоих методов, то быстрее по времени получается метод Якоби.

4 Приложение

```
1 port time
2 port numpy as np
3 port matplotlib.pyplot as plt
4
5 = 1
```

```

6 silon = 0.1
7 1 = 0.1
8 2 = 0.1
9
10
11 f u_1(x, y):
12     return 2 * x + 3 * y - 5
13
14
15 f f_1(x, y):
16     return 0
17
18
19 f u_2(x, y):
20     return 2 * x ** 2 + y - 3
21
22
23 f f_2(x, y):
24     return -4
25
26
27 f u_3(x, y):
28     return 3 * x ** 2 - 2 * y ** 2 - 1
29
30
31 f f_3(x, y):
32     return -2
33
34
35 f u_4(x, y):
36     return np.exp((np.sin(x)) ** 2 + (np.cos(y)) ** 2)
37
38
39 f f_4(x, y):
40     return -1 * np.exp(
41         (np.sin(x)) ** 2 + (np.cos(y)) ** 2) * \
42         ((np.sin(2 * x)) ** 2 + 2 * np.cos(
43             2 * x) + np.sin(2 * y) -
44             2 * np.cos(2 * y))
45
46
47 f Cross_Jac(f, h_1, h_2, y0, ep):
48     tic = time.perf_counter()

```



```

49 C_0 = (0.5 * (h_1 ** 2) * (h_2 ** 2)) / (
50     h_1 ** 2 + h_2 ** 2)
51 C_1 = (0.5 * (h_1 ** 2)) / (h_1 ** 2 + h_2 ** 2)
52 C_2 = (0.5 * (h_2 ** 2)) / (h_1 ** 2 + h_2 ** 2)
53 Nx = int(L / h_1) + 1
54 Ny = int(L / h_2) + 1
55 U = np.zeros((Ny, Nx))
56 U_m = np.zeros((Ny, Nx))
57 x = np.linspace(0, L, Nx)
58 y = np.linspace(0, L, Ny)
59
60 # Заполняем начальное приближение с учетом нач.условий
61 for i in range(0, Ny):
62     for j in range(0, Nx):
63         U[i][j] = y0(x[j], 0)
64 for i in range(0, Nx): # y = 1
65     U[Ny - 1][i] = y0(x[i], L)
66 for i in range(0, Ny): # x = 1
67     U[i][Nx - 1] = y0(L, y[i])
68 for i in range(0, Ny): # x = 0
69     U[i][0] = y0(0, y[i])
70
71 norm = 1
72 k = 0
73 for i in range(0, Ny):
74     for j in range(0, Nx):
75         U_m[i][j] = y0(x[j], 0)
76 for i in range(0, Nx): # y = 1
77     U_m[Ny - 1][i] = y0(x[i], L)
78 for i in range(0, Ny): # x = 1
79     U_m[i][Nx - 1] = y0(L, y[i])
80 for i in range(0, Ny): # x = 0
81     U_m[i][0] = y0(0, y[i])
82
83 while norm > ep:
84
85     for i in range(Nx - 2, 0, -1):
86         for j in range(1, Ny - 1):
87             U_m[j][i] = C_0 * f(x[i], y[j]) + C_1 * (
88                 U[j - 1][i] + U[j + 1][
89                     i]) + C_2 * (U[j][i - 1] + U[j][
90                     i + 1])
91     norm = np.max(abs(U_m - U))

```

```

92     k += 1
93     for i in range(0, Ny):
94         for j in range(0, Nx):
95             U[i][j] = U_m[i][j]
96     print(norm)
97
98     toc = time.perf_counter()
99     tme = round(toc - tic, 4)
100     return tme, k
101
102
103 f Cross_Jac_with_param(f, h_1, h_2, y0, omega, ep):
104     tic = time.perf_counter()
105     Nx = int(L / h_1) + 1
106     Ny = int(L / h_2) + 1
107     U = np.zeros((Ny, Nx))
108     U_m = np.zeros((Ny, Nx))
109     x = np.linspace(0, L, Nx)
110     y = np.linspace(0, L, Ny)
111
112     # Заполняем начальное приближение с учетом нач.условий
113     for i in range(0, Ny):
114         for j in range(0, Nx):
115             U[i][j] = y0(x[j], 0)
116     for i in range(0, Nx): # y = 1
117         U[Ny - 1][i] = y0(x[i], L)
118     for i in range(0, Ny): # x = 1
119         U[i][Nx - 1] = y0(L, y[i])
120     for i in range(0, Ny): # x = 0
121         U[i][0] = y0(0, y[i])
122
123     for i in range(0, Ny):
124         for j in range(0, Nx):
125             U_m[i][j] = y0(x[j], 0)
126     for i in range(0, Nx): # y = 1
127         U_m[Ny - 1][i] = y0(x[i], L)
128     for i in range(0, Ny): # x = 1
129         U_m[i][Nx - 1] = y0(L, y[i])
130     for i in range(0, Ny): # x = 0
131         U_m[i][0] = y0(0, y[i])
132     norm = 1
133     k = 0
134     while norm > ep:

```

```

135     for i in range(Nx - 2, 0, -1):
136         for j in range(1, Ny - 1):
137             U_m[j][i] = (1 - omega) * U[j][i] + (
138                 omega / 4) \
139                 * (U[j - 1][i] + U[j + 1][i] +
140                    U[j][i - 1] + U[j][i + 1]) \
141                 + (omega * (h_1 ** 2) / 4) * f(
142                     x[i], y[j])
143     norm = np.max(abs(U_m - U))
144     k += 1
145     for i in range(0, Ny):
146         for j in range(0, Nx):
147             U[i][j] = U_m[i][j]
148     toc = time.perf_counter()
149     tme = round(toc - tic, 4)
150     return tme, k
151
152
153 f Table(name, func, h_1, h_2, y0, ep):
154     with open(name, 'w') as f:
155         f.write('& ' + 'h_{0}' & ' + 'h_{0}/2 & '
156                + 'h_{0}/4 & ' + 'h_{0}/8 & ' + 'h_{0}/16 '
157                + ' \\\ ' + '\hline' + '\n')
158     for i in range(0, 4):
159         epsil = ep / (10 ** i)
160         for j in range(0, 6):
161             h1 = h_1 / (2 ** (j - 1))
162             h2 = h_2 / (2 ** (j - 1))
163             tm, it = Cross_Jac_with_param(func, h1, h2,
164                                           y0, 0.8,
165                                           epsil)
166             if j == 5:
167                 f.write(
168                     ' ' + "{:1.4G}".format(
169                         it) + ' ' + "{:1.4G}".format(
170                             tm))
171                 f.write(' \\\ ' + '\hline')
172                 f.write('\n')
173                 print("Итераций, time", it, " ", tm)
174             elif j == 0:
175                 f.write('\n' + 'varepsilon_{0}/{ '
176                        + str(10 ** i) + ' } & ')
177                 print(2 ** i)

```

```
178         else:
179             f.write("{:1.4G}".format(
180                 it) + ' ' + "{:1.4G}".format(tm))
181             f.write(' & ')
182         print("Итераций, time ", it, " ", tm)
```
