



Московский государственный университет имени М.В. Ломоносова

Казахстанский филиал

Факультет вычислительной математики и кибернетики

Отчет по практикуму

Задача Дирихле для уравнения Пуассона

Составил: студент уuuuu

Проверил: преподаватель NNNNN N.N.

Нур-Султан 2020

Содержание

1 Постановка задачи

Рассматривается задача Дирихле для эллиптического уравнения [1]

$$-Lu = f(x, y), \quad (x, y) \in G, \quad (1)$$

$$u = \mu(x, y), \quad (x, y) \in \Gamma, \quad (2)$$

Пусть $\bar{G} = G \cup \Gamma = \{0 \leq x \leq 1, 0 \leq y \leq 1\}$ - квадрат, а

$$Lu = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \quad (3)$$

1.1 Построение сетки, равномерной по каждому из направлений

Разобьем отрезок $[0, 1]$ на n равных частей. Обозначим $h = 1/n$, $x_i = ih, y_j = jh$, $0 \leq i \leq n, 0 \leq j \leq n$. Построим сетку узлов

$$\bar{\omega}_h = \{(x_i, y_j), \quad 0 \leq i \leq n, \quad 0 \leq j \leq n\}$$

Узлы (x_i, y_j) , $1 \leq i \leq n-1, 1 \leq j \leq n-1$ - внутренние, остальные, лежащие на границе прямоугольника, - граничные. Внутренний узел (x_i, y_j) будем называть приграничным, если хотя бы один из узлов $(x_{i-1}, y_j), (x_{i+1}, y_j), (x_i, y_{j-1}), (x_i, y_{j+1})$ является граничным.

1.2 Разностная аппроксимация задачи Дирихле

Обозначим $u_{ij} \approx u(x_i, y_j)$. Заменяем оператор L во всех внутренних узлах разностным оператором [2]

$$L_h u_{ij} = \frac{u_{i+1j} - u_{ij}}{h_x^2} - \frac{u_{ij} - u_{i-1j}}{h_x^2} + \frac{u_{ij+1} - u_{ij}}{h_y^2} - \frac{u_{ij} - u_{ij-1}}{h_y^2}, \quad (4)$$

$$1 \leq i \leq n-1; \quad 1 \leq j \leq n-1.$$

Если $u(x, y)$ имеет не менее четырех непрерывных ограниченных в рассматриваемой области производных по x и по y , то разностный оператор L_h аппроксимирует дифференциальный L со вторым порядком, т.е.

$$Lu - L_h u = O(|h|^2).$$

Задаче (1), (2) ставим в соответствие разностную задачу: найти сеточную функцию, удовлетворяющую во внутренних узлах уравнениям

$$-L_h u_{ij} = f_{ij}, \quad 1 \leq i \leq n-1; \quad 1 \leq j \leq n-1, \quad f_{ij} = f(x_i, y_j), \quad (5)$$

и принимающую в граничных узлах заданные значения

$$\begin{cases} u_{i0} = \mu(x_i, 0), & 0 \leq i \leq n \\ u_{in} = \mu(x_i, 1), & 0 \leq i \leq n \\ u_{0j} = \mu(0, y_j), & 1 \leq j \leq n-1 \\ u_{nj} = \mu(1, y_j), & 1 \leq j \leq n-1 \end{cases} \quad (6)$$

Для решения системы могут быть использованы следующие методы:

- 1) Метод Якоби.
- 2) Метод Зейделя.
- 3) Метод верхней релаксации.

1.3 Расчетные формулы для методов

1.3.1 Метод Якоби

Систему $AU = F$ сводим к системе вида $U = HU + g$ так, чтобы $\rho(H) < 1$, где $\rho(H)$ - максимальное по модулю собственное число матрицы H (спектральный радиус матрицы). Пусть $H = E - D^{-1}A$, $g = D^{-1}F$, где D -диагональная часть матрицы A . Расчетная формула имеет следующий вид:

$$u_{ij}^{(k+1)} = \frac{1}{4}(u_{i+1j}^{(k)} + u_{i-1j}^{(k)} + u_{ij+1}^{(k)} + u_{ij-1}^{(k)} + h^2 f_{ij}) \quad (7)$$

$$i, j = 1, \dots, n-1, k = 0, 1, \dots$$

Для оценки точности будем использовать C -норму.

$$|x^{k+1} - x^k| = \|x^{k+1} - x^k\|_\infty = \max_{1 \leq i, j \leq n-1} |x_{ij}^{k+1} - x_{ij}^k|$$

Необходимое и достаточное условие сходимости метода $\rho(H) < 1$ выполнено, так как в рассматриваемом случае [3]:

$$\lambda_{\max}(H) = 1 - \frac{h^2}{4} \lambda_{\min}(A) = 1 - \frac{h^2}{4} \frac{8}{h^2} \sin^2 \frac{\pi h}{2} = 1 - 2 \sin^2 \frac{\pi h}{2} = \cos \pi h$$

$$\lambda_{min}(H) = 1 - \frac{h^2}{4} \lambda_{max}(A) = 1 - \frac{h^2}{4} \frac{8}{h^2} \cos^2 \frac{\pi h}{2} = 1 - 2 \cos^2 \frac{\pi h}{2} = -\cos \pi h$$

Скорость сходимости метода определяется спектральным радиусом $\rho(H)$.

Ввиду того, что $\rho(H) = 1 - O(h^2)$ при уменьшении шага сетки h сходимость замедляется, и близость двух соседних приближений, часто ошибочно воспринимается за признак сходимости метода.

В качестве апостериорной оценки погрешности здесь часто допустимо использовать следующую [2]:

$$\|U^k - u^*\| \leq \frac{\rho(H)}{1 - \rho(H)} \|U^k - U^{k-1}\|$$

1.3.2 Метод Зейделя

Известно, что если матрица A положительно определена, то метод Зейделя для системы $AU = F$ сходится, причем вдвое быстрее, чем метод Якоби [3].

Расчетная формула метода Зейделя имеет вид

$$u_{ij}^{(k+1)} = \frac{1}{4}(u_{i-1j}^{(k+1)} + u_{ij-1}^{(k+1)} + u_{i+1j}^{(k)} + u_{ij+1}^{(k)} + h^2 f_{ij}) \quad (8)$$

$$1 \leq i \leq n-1, \quad 1 \leq j \leq n-1$$

1.3.3 Метод верхней релаксации (SOR)

Расчетная формула имеет вид

$$u_{ij}^k = u_{ij}^{k-1} + \omega \frac{h^2 f_{ij} + u_{i-1j}^k + u_{i+1j}^{k-1} + u_{ij-1}^k + u_{ij+1}^{k-1} - 4u_{ij}^{k-1}}{4} \quad (9)$$

$$1 \leq i \leq n-1; \quad 1 \leq j \leq n-1.$$

Метод будет сходиться, если матрица A симметрическая, положительно определенная и кроме того $0 < \omega < 2$ [3].

Скорость сходимости релаксационного циклического процесса определяется наибольшим модулем собственных значений матрицы $S_\omega = (D + \omega L)^{-1}(D - \omega D - \omega R)$, где D, L и R диагональная, поддиагональная и наддиагональная части матрицы A . Оптимальным значением ω является

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho^2(H)}} \quad (10)$$

Для нашей задачи

$$\omega_{opt} = \frac{2}{1 + \sin(\pi h)}, \quad \rho(S_{\omega_{opt}}) = \frac{1 - \sin(\pi h)}{1 + \sin(\pi h)} \approx 1 - 2\pi h.$$

1.4 Выбор точности

Разностная схема аппроксимирует исходную задачу(1).(2) со вторым порядком относительно шага сетки. Заданная точность приближенного решения должна быть согласована с порядком аппроксимации. Обычно судят о точности решения по его относительной погрешности $\|U^k - u^*\|/\|U^0 - u^*\|$ или, так как точное решение u^* неизвестно, по величине относительной невязки $\frac{\|AU^k - F\|}{\|AU^0 - F\|}$. Пусть $\varepsilon > 0$ - заданная относительная погрешность, с которой надо найти приближенное решение задачи. Вычисления прекращают, если выполнено условие $\frac{\|U^m - u^*\|}{\|U^0 - u^*\|} < \varepsilon$ или $\frac{\|AU^m - F\|}{\|AU^0 - F\|} < \varepsilon$. Можно получить [3], что число итераций необходимых для достижения заданной точности K :

для метода Якоби $K \geq 2^{\frac{\ln(1/\varepsilon)}{(\pi h)^2}}$;

для метода Зейделя $K \geq \frac{\ln(1/\varepsilon)}{(\pi h)^2}$;

для метода верхней релаксации $K \geq 2^{\frac{\ln(1/\varepsilon)}{(\pi h)^2}}$.

В качестве критерия конца вычислений может быть выбран $\|U^k - u^*\| < \varepsilon$.

2 Результаты расчетов

Для сопоставления численного решения с точным использовалось аналитическое решение вида: $u(x, y) = x^2 + y^2$.

Таблица расчетов для метода Якоби, точность 10^{-2} :

$K = 2333$

k	Uk-aU
100	0.93484
200	0.70043
300	0.55657
400	0.45029

500	0.36582
600	0.29808
700	0.24341
800	0.19923
900	0.16325
1000	0.13379
1100	0.10968
1200	0.09002
1300	0.07389
1400	0.06064
1500	0.04977
1600	0.04085
1700	0.03353
1800	0.02752
1900	0.02259
2000	0.01854
2100	0.01522
2200	0.01249
2300	0.01025
2333	0.00960
time =	22.030576705932617

Графики для метода Якоби:

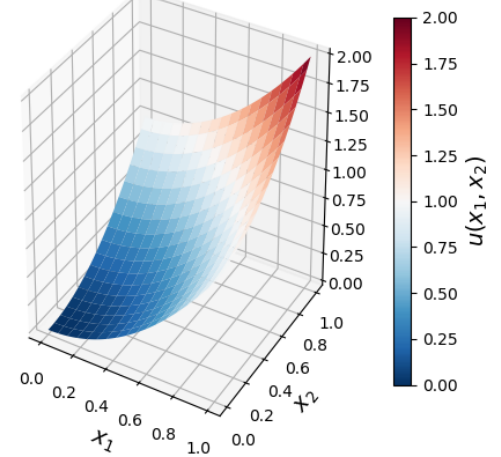
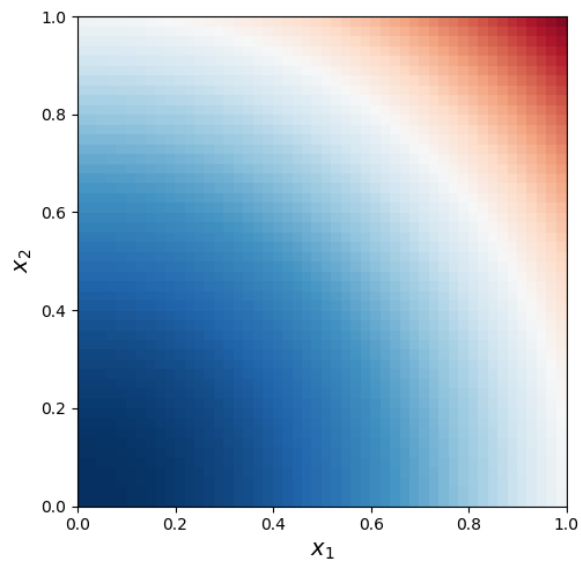


Таблица расчетов для метода Зейделя, точность 10^{-3} :

K = 1749

k	Uk-aU
100	0.98309
200	0.73456
300	0.58222
400	0.47043
500	0.38214
600	0.31167
700	0.25467
800	0.20867
900	0.17095
1000	0.14033
1100	0.11520
1200	0.09456
1300	0.07761
1400	0.06370
1500	0.05229


```

1600    0.04291
1700    0.03522
1749    0.03191
time =  22.171183109283447

```

Графики для метода Зейделя:

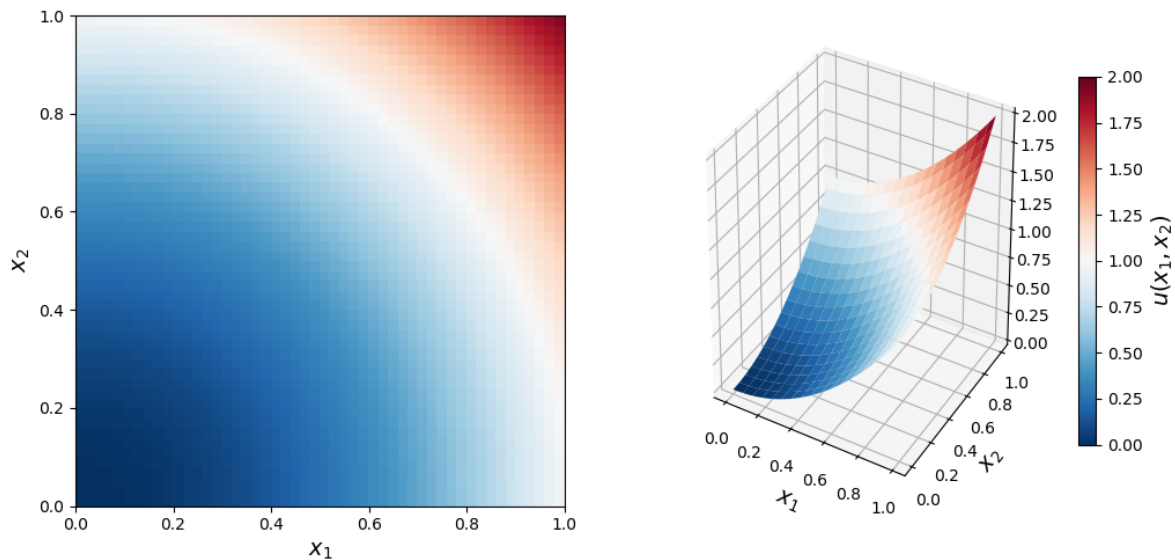


Таблица расчетов для метода SOR, точность 10^{-4} .

```
omega =  1.8818383898322273
```

```
K =  293
```

k	Uk-aU
10	1.60433
20	1.30687
30	1.02577
40	0.76526
50	0.53462
60	0.39768
70	0.30391
80	0.23116

```
90      0.17253
100     0.12531
110     0.08661
120     0.05620
130     0.03359
140     0.01771
150     0.00817
160     0.00458
170     0.00418
180     0.00396
190     0.00336
200     0.00070
210     0.00030
220     0.00016
230     0.00008
240     0.00005
250     0.00003
260     0.00002
270     0.00001
280     0.00001
290     0.00000
293     0.00000
time = 4.499859094619751
```

График идентичен предыдущим.

3 Листинги программ

```
def asolve(x,y):
    return x**2+y**2
def f(x, y):
```

```

        return -4
def wst(y):
    return y**2
def est(y):
    return 1.0+y**2
def sth(x):
    return x**2
def nrd(x):
    return 1.0+x**2
def SOR(n,om,u,f,h2,au):
    a = 0.0
    for i in range(1,n):
        for j in range(1,n):
            u[i][j] = u[i][j] + om*( (u[i-1][j]+u[i+1][j]+u[i][j-1]
            +u[i][j+1]+h2*f[i][j])/4 - u[i][j] )
            b = abs(u[i][j]-au[i][j])
            a = max(a,b)
    return a
n = 50; np1 = n+1; nm1 = n-1
h = 1.0 / n; h2 = h**2
x = np.linspace(0.0, 1.0, n + 1)
y = np.linspace(0.0, 1.0, n + 1)

F = np.zeros((np1, np1))
aU = np.zeros((np1, np1))
U = np.zeros((np1, np1))
V = np.zeros((np1, np1))

for i in range(np1):
    for j in range(np1):

```

```

        F[i][j] = f(x[i], y[j])
        aU[i][j] = asolve(x[i],y[j])

for i in range(1,n):
    U[0][i] = V[0][i] = wst(y[i])
    U[n][i] = V[n][i] = est(y[i])
    U[i][0] = V[i][0] = sth(x[i])
    U[i][n] = V[i][n] = nrd(x[i])

p=4 #eps = 1e-p
K = int(p*log(10)*n/pi*2+0.5)
rsJ = cos(pi*h)
om = 2.0/(1.0+sqrt(1.0-rsJ**2))
print("omega = ",om)
print("K = ",K)
pk = 10
start = time.time()
u = U; v = V
print("    k    Uk-aU")
for k in range(1,K+1):
    euk = SOR(n,om,u,F,h2,aU)
    if k % pk == 0:
        print("{0:5d} {1:10.5f}".format(k,euk))
    t = u; u = v; v = t
print("{0:5d} {1:10.5f}".format(K,euk))
end = time.time()
print(" time = ",end - start)

```

Список литературы

- [1] Уравнения математической физики: учебник для студ. высш. учеб. заведений./ Е.В.Захаров, И.В.Дмитриева, С.И.Орлик. - М.: Издат. центр Академия, 2010. - 320с.
- [2] Самарский А.А., Гулин А.В. Численные методы математической физики. – М.: Научный мир, 2000. – 316 с.
- [3] Самарский А.А. Теория разностных схем. – М.: Издат. Наука, 1977. – 656 с.