

ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ОБРАЗОВАНИЮ И НАУКЕ
ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ

Абдурагимов Э.И.

Метод сеток решения задачи Дирихле для уравнения Пуассона

Лабораторные задания и методические указания по
численным методам

$$\Delta u = f(x, y)$$

Махачкала
ИИЦ ДГУ
2010

Печатается по решению редакционно-издательского совета
Дагестанского государственного университета

Пособие предназначено для студентов математического факультета специальности 010200—прикладная математика и информатика.

По разделу «Метод сеток решения типичных задач для уравнений математической физики» предусмотрено выполнение студентами двух лабораторных работ. В предлагаемом пособии приводится описание одной из этих лабораторных работ. В нем приведены также краткие теоретические сведения, варианты задания и образец выполнения нулевого варианта лабораторной работы, аналогичного вариантам, которые должны выполнить студенты.

Составитель: *Абдурагимов Э.И.*—доцент кафедры
прикладной математики ДГУ

ИПЦ ДГУ, 2010

Краткие теоретические сведения

1. Построение разностной схемы, аппроксимирующей задачу Дирихле для уравнения Пуассона.

Пусть в области $D = \{0 < x < a, 0 < y < b\}$ задано уравнение Пуассона

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y), \quad (1)$$

и на границе Γ области D – условию Дирихле

$$u_{\Gamma} = \varphi(M), \quad (2)$$

где f и φ – известные функции, M – точка контура Γ .

Предположим, что функции f и φ таковы, что задача (1), (2) имеет единственное решение $u(x, y)$ в области $\bar{D} = D + \Gamma$ и это решение имеет непрерывные частные производные $\frac{\partial^2 u}{\partial x^2}$ и $\frac{\partial^2 u}{\partial y^2}$.

Возьмем прямоугольную сетку, положив

$$x_m = mh, \quad m = 0, 1, \dots, M, \quad h = \frac{a}{M};$$

$$y_n = nl, \quad n = 0, 1, \dots, N, \quad l = \frac{b}{N}.$$

Для аппроксимации уравнения (1) выберем пятиточечный шаблон

$$\begin{array}{c} (m, n+1) \\ | \\ (m-1, n) \text{ --- } (m, n) \text{ --- } (m+1, n) \\ | \\ (m, n-1) \end{array}$$

и к множеству D_h^0 внутренних узлов отнесем все узлы, лежащие в D , а к множеству граничных узлов Γ_h отнесем все узлы, лежащие на Γ .

Разностная схема, аппроксимирующая задачу (2), имеет вид (см. [1])

$$L_h(u^{(h)}) = f^{(h)}, \quad (3)$$

где

$$L_h(u^{(h)}) = \begin{cases} \frac{u_{m+1,n} - 2u_{mn} + u_{m-1,n}}{h^2} + \frac{u_{m,n+1} - 2u_{mn} + u_{m,n-1}}{l^2}, \\ m = 1, 2, \dots, M-1, \quad n = 1, 2, \dots, N-1, \\ u_{mn}, \\ \{m = \overline{0, M}, n = 0; m = \overline{0, M}, n = N; m = 0, n = \overline{0, N}; m = M, n = \overline{0, N}\} \end{cases}$$

$$f^{(h)} = \begin{cases} f(x_m, y_n), \quad m = \overline{1, M-1}, \quad n = \overline{1, N-1} \\ \varphi(x_m, y_n), \\ \{m = \overline{0, M}, n = 0; m = \overline{0, M}, n = N; m = 0, n = \overline{0, N}; m = M, n = \overline{0, N}\} \end{cases}$$

Разностная схема (3) представляет собой из $(M-1)(N-1)$ линейных алгебраических уравнений, столько же неизвестных, это — значения u_{mn} при $m = \overline{1, M-1}, n = \overline{1, N-1}$.

Известно [1], то разностная схема (3) устойчива и аппроксимирует задачу Дирихле (1), (2) с порядком $O(h^2 + l^2)$. Поэтому по теореме о сходимости [1] решение u_{mn} разностной схемы (3) сходится к решению $u(x_m, y_n)$ задачи Дирихле (1), (2) при $h \rightarrow 0, l \rightarrow 0$ с тем же порядком.

2. Численные методы решения разностной схемы

Существуют различные методы решения системы разностных уравнений (3), например, метод Гаусса, итерационные методы, метод матричной прогонки, метод установления и др. Из них рассмотрим только два метода, которые учитывают специфический вид системы (3): а) прямой метод — метод матричной прогонки; б) итерационный метод — метод Либмана.

а) Метод матричной прогонки

Если $M \gg N$ (M значительно больше N), то для применения метода матричной прогонки перепишем систему (3) в виде

$$\begin{cases} u_{m+1,n} - 2u_{mn} + u_{m-1,n} + \alpha(u_{m,n+1} - 2u_{mn} + u_{m,n-1}) = h^2 f(x_m, y_n), \\ m = 1, 2, \dots, M-1, \quad n = 1, 2, \dots, N-1; \\ u_{0n} = \varphi(0, y_n), \quad u_{Mn} = \varphi(a, y_n), \quad n = 0, 1, 2, \dots, N, \\ u_{m0} = \varphi(x_m, 0), \quad u_{mN} = \varphi(x_m, b), \quad m = 0, 1, 2, \dots, M; \quad \alpha = (h/l)^2 > 0. \end{cases} \quad (4)$$

Введем обозначение

$$u_m = \begin{pmatrix} u_{m1} \\ u_{m2} \\ \text{M} \\ u_{m,N-1} \end{pmatrix}, \quad m = 0, 1, \Lambda, M. \quad (5)$$

Положим в формулах (4) $n = 1, 2, \Lambda, N - 1$ и, используя обозначение (5), запишем систему (4) в векторной форме

$$\begin{aligned} u_{m+1} + Au_m + u_{m-1} &= f_m, \quad m = 1, 2, \Lambda, M - 1, \\ u_0 &= \varphi_0, \quad u_M = \varphi_a, \end{aligned} \quad (6)$$

где A – трехдиагональная матрица порядка $N - 1$ вида

$$A = \begin{pmatrix} -(2+2\alpha) & \alpha & 0 & \Lambda & 0 & 0 & 0 \\ \alpha & -(2+2\alpha) & \alpha & \Lambda & 0 & 0 & 0 \\ \Lambda & \Lambda & \Lambda & \Lambda & \Lambda & \Lambda & \\ 0 & 0 & 0 & \alpha & -(2+2\alpha) & \alpha & \\ 0 & 0 & 0 & 0 & \alpha & -(2+2\alpha) & \end{pmatrix},$$

$$f_m = \begin{pmatrix} h^2 f_{m1} - \alpha \varphi(x_m, 0) \\ h^2 f_{m2} \\ \text{M} \\ h^2 f_{m,N-2} \\ h^2 f_{m,N-1} - \alpha \varphi(x_m, b) \end{pmatrix}, \quad \varphi_0 = \begin{pmatrix} \varphi(0, y_1) \\ \varphi(0, y_2) \\ \text{M} \\ \varphi(0, y_{N-1}) \end{pmatrix}, \quad \varphi_a = \begin{pmatrix} \varphi(a, y_1) \\ \varphi(a, y_2) \\ \text{M} \\ \varphi(a, y_{N-1}) \end{pmatrix}.$$

Векторную систему (6) так же, как аналогичную скалярную систему можно решить методом прогонки, который в этом случае называется *методом матричной прогонки*. Алгоритм этого метода состоит из следующих шагов

1. По формуле

$$R_{v+1} = -(A + R_m)^{-1}, \quad m = 1, 2, \Lambda, M - 1,$$

полагая $R_1 = 0$, вычисляем матрицы

$$R_m = (R_{ij}^{(m)}), \quad m = 1, 2, \Lambda, M, \quad i, j = 1, 2, \Lambda, N - 1, \text{ затем по формуле}$$

$$s_{m+1} = R_{m+1}(s_m - f_m), \quad m = 1, 2, \Lambda, M - 1,$$

полагая $s_1 = \varphi_0$, вычисляем векторы

$$s_m = (s_1^{(m)}, s_2^{(m)}, \Lambda, s_{N-1}^{(m)})^T, \quad T - \text{знак транспонирования,}$$

$$m = 1, 2, \Lambda, M - 1.$$

2. По формуле

$$u_{m-1} = R_m u_m + s_m, \quad m = M, M-1, \Lambda, 2,$$

полагая $u_M = \varphi_a$, вычисляем искомые векторы

$$u_M, u_{M-1}, \Lambda, u_1, u_0 = \varphi_0.$$

Известно [1], что описанный здесь метод является устойчивым.

Мы рассмотрели метод матричной прогонки в случае $M \gg N$. Если $N \gg M$, то направление прогонки выбирается совпадающим с направлением оси oy . в обоих случаях сокращается объем вычислений.

б) Итерационный метод Либмана

Умножив обе части системы

$$\frac{u_{m+1,n} - 2u_{mn} + u_{m-1,n}}{h^2} + \frac{u_{m,n+1} - 2u_{mn} + u_{m,n-1}}{l^2} = f(x_m, y_n)$$

на $h^2 l^2$ и разделив полученные равенства на $2(h^2 + l^2)$, имеем

$$u_{mn} = \frac{l^2(u_{m+1,n} + u_{m-1,n}) + h^2(u_{m,n+1} + u_{m,n-1})}{2(h^2 + l^2)} - \frac{h^2 l^2}{2(h^2 + l^2)} f_{mn},$$

где $f_{mn} = f(x_m, y_n)$. Итерационный процесс Либмана для полученной системы имеет вид

$$u_{mn}^{(k)} = \frac{l^2(u_{m+1,n}^{(k)} + u_{m-1,n}^{(k)}) + h^2(u_{m,n+1}^{(k)} + u_{m,n-1}^{(k)})}{2(h^2 + l^2)} - \frac{h^2 l^2}{2(h^2 + l^2)} f_{mn}, \quad (7)$$

$$k = 0, 1, 2, \Lambda$$

Известно, что этот итерационный процесс сходится, т.е. существует $\lim_{k \rightarrow \infty} u_{mn}^{(k)} = u_{mn}$.

Для того чтобы начать вычислять последовательные приближения $u_{mn}^{(k)}$ к u_{mn} по формуле (7) необходимо задать начальное приближение $u_{mn}^{(0)}$. При этом $u_{mn}^{(0)}$ должно удовлетворять граничным условиям $u_{mn} = \varphi(x_m, y_n)$ при $(x_m, y_n) \in \Gamma_h$. Поэтому удобно $u_{mn}^{(0)}$ задать так, чтобы на границе Γ_h выполнялось условие $u_{mn}^{(0)} = u_{mn}$. Обозначим $\Phi(x, y)$ некоторое продолжение по непрерывности граничной функции $\varphi(x, y)$ на всю область \bar{D} . Причем продолжение должно быть таким, чтобы выполнялись в начале условия на решение относительно единственности и

гладкости. Тогда можно положить $u_{mn}^{(0)} = \Phi(x_m, y_n)$,
 $m = 0, 1, \Lambda, M, \quad n = 0, 1, \Lambda, N$.

Таким образом, приближения $u_{mn}^{(k)}$ к u_{mn} можно построить с помощью итерационного процесса (7) по следующей схеме:

1. Функцию $\varphi(x, y)$, заданную на границе Γ области D продолжим по непрерывности на всю область \bar{D} так, чтобы при этом решение задачи Дирихле (1), (2) удовлетворяло перечисленным в начале условиям единственности и гладкости. Продолженную функцию обозначим $\Phi(x, y)$ и положим

$$u_{mn}^{(0)} = \Phi(x_m, y_n), \quad m = 0, 1, \Lambda, M, \quad n = 0, 1, \Lambda, N. \quad (8)$$

2. Полагая $k = 0, 1, \Lambda$, последовательно вычисляем по формулам (7)

$u_{mn}^{(1)}, u_{mn}^{(2)}, \dots, u_{mn}^{(k)}$, $m = 0, 1, \Lambda, M, \quad n = 0, 1, \Lambda, N$. Вычисления $u_{mn}^{(k)}$ проводим до такого значения $k = k_0$, при котором

$$\left| u_{mn}^{(k+1)} - u_{mn}^{(k)} \right| \leq \varepsilon \quad (9)$$

для заданного малого числа ε .

Лабораторная работа

Тема: Метод сеток решения задачи Дирихле для эллиптического уравнения.

Цель: Научиться решать задачу Дирихле для эллиптического уравнения методом сеток, используя ЭВМ.

Задание.

Методом сеток в прямоугольнике $D = \{0 < x < a, 0 < y < b\}$ с границей Γ решить задачу Дирихле

$$\begin{aligned} \Delta u &= f(x, y), & (x, y) \in D, \\ u|_{\Gamma} &= \varphi(x, y), \end{aligned}$$

где $f(x, y), \varphi(x, y)$ —заданные функции.

Для этой цели составить методом сеток разностную схему, аппроксимирующую данную задачу Дирихле с порядком

погрешности $O(h^2 + l^2)$, где h и l – шаги по оси ox и oy соответственно. Найти решение составленной разностной схемы

- а) методом матричной прогонки;
- б) итерационным методом Либмана.

Полученные этими методами приближенные значения решения выписать в точках (x_i, y_j) , где $x_i = ih_0, y_j = jl_0, h_0 = \frac{a}{10}, l_0 = \frac{b}{10}$.

Заполнить соответствующие этим методам 2 таблицы видов

**Решение задачи Дирихле методом матричной прогонки
(итерационным методом Либмана)**

x \ y	0.0	h_0	$2h_0$...	a
0.0				...	
l_0				...	
$2l_0$...	
...
b				...	

При заполнении таблиц в результатах сохранять по два десятичных знака. В расчетах взять $\max(h, l) \leq 0.1$.

Для контроля работы составленных программ найти указанными методами приближенное решение, самостоятельно составленной тестовой задачи Дирихле того же вида, что и данная. Заполнить две таблицы приведенного выше вида, озаглавив их:

- а) Таблица значений решения тестовой задачи методом матричной прогонки;
- б) Таблица точных значений решения тестовой задачи.

Найти максимальную по модулю погрешность метода матричной прогонки для тестовой задачи и выписать ее в форме с плавающей точкой. Указать в какой точке это максимальное значение модуля погрешности достигается.

При применении итерационного метода Либмана число итераций k_0 взять таким, чтобы выполнилось условие (9) с $\varepsilon = 10^{-4}$. Указать чему равно k_0 для своей и тестовой задачи.

Сделать выводы.

К зачету необходимо представить выполненную без ошибок и оформленную по приведенному ниже образцу лабораторную работу. После этого выставляется зачет по представленной лабораторной работе, если студент

а) демонстрирует на компьютере полученные результаты, комментируя их;

б) показывает понимание численных методов, использованных в лабораторной работе;

в) демонстрирует четкое знание алгоритмов выполнения лабораторной работы;

г) умеет, пользуясь своей лабораторной работой, выполнить на ЭВМ за короткое время задания преподавателя, аналогичные заданиям лабораторной работы.

При реализации алгоритмов выполнения заданий можно пользоваться любым языком программирования или (и) пакетом прикладных программ.

ВАРИАНТЫ ЗАДАНИЯ

№	a	b	$f(x, y)$	$\varphi(x, y)$
1	1	0.5	$\frac{\sin(xy)}{1+x^2+y^2}$	$\int_0^{x+y} e^{t^2} dt$
2	0.5	1	$-\sum_{k=0}^{10} \frac{1+x+y}{1+x^k+y^k}$	$1+(x-y)^2$
3	1	0.5	$\frac{e^{\sin(x^2+y^2)}}{1+\int_0^{xy} e^{t^2} dt}$	$x-y$
4	0.2	0.5	$\int_0^{x^2+y^2} \frac{\sin^2 t}{1+t} dt$	0
5	0.5	0.5	$1-\sum_{k=0}^{10} \frac{x^k+y^k}{1+x^k+y^k}$	$\frac{1-xy}{1+(x-y)^2}$
6	1	0.5	$\int_0^x e^{-\frac{t+y}{t+1}} dt$	0

7	0.5	0.5	$\frac{e^{-x-y}-2}{1+\int_0^{xy} \sin t^2 dt}$	$e^{xy-x^2-y^2}$
8	0.8	0.5	$xy+2\int_0^1 \frac{\sin t}{1+t^2} dt$	$e^{-\int_0^{x+y} \frac{\sin t}{1+t} dt}$
9	0.5	0.5	$1+\int_0^{xy} \frac{e^t}{1+t} dt - (x-y)^2$	$xy(x-y)$
10	1	1	$tg(\int_0^{x+y} \sin^2 t dt)$	$\frac{x-y}{1+\sqrt{x+y+xy}}$
11	1	0.5	$\sqrt{x}-\sqrt{y}-\sqrt{x+y+4}$	$\frac{x^2+y^2}{\int_0^1 e^{t+1} dt}$
12	0.8	0.5	$\alpha xy, \text{ где } \alpha - \text{ корень}$ $yp - я \quad e^{-x} = 2x + 5$	$xy \int_0^1 e^{(x+y)t^2} dt$
13	1	0.4	$\sqrt[3]{x^2+xy+2y^2}$	$\alpha(x-1)(y-1), \text{ где}$ $\alpha = \int_0^1 \frac{\cos t}{1+t} dt$
14	1	0.2	$\int_x^{1+y} \frac{t}{e^{1+t}} dt$	$xy(1-x)$
15	0.8	0.4	x^2+y^2-3	$\int_{xy}^{x+y+2} \frac{\sin t}{1+t} dt$
16	0.5	0.5	$\frac{x+1}{y+1} - e^{xy}$	$\int_{x+y}^{x^2+y^2} \frac{e^t}{1+\sin t} dt$
17	1	0.4	$\int_0^1 \frac{\sin(xy t^2)}{1+t(x+y)} dt$	$\alpha x(2y-1), \text{ где}$ $\alpha = \int_0^1 \frac{\sin t^2}{1+t} dt$
18	1	0.5	x^2+y^2-xy-3	$\int_0^{xy} \cos \frac{t}{1+t} dt$
19	0.5	0.5	$x+y - \int_0^{xy} e^{\sin t} dt$	$\sqrt{1+\cos^2 \pi x + \sin^2 \pi y}$

20	1	0.5	$\sqrt{x} - \sqrt{y} - \sqrt{x+y} + 4$	$x^2+y^2 \int_0^1 e^{\frac{t}{1+t}} dt$
21	1	0.5	$tg(1 + \sqrt{x^2 + y^2})$	$x \int_{x-y}^{x+y} \sin \frac{t}{t^2+1} dt$
22	1	0.8	$\frac{1-x-y}{1+\sqrt{x}+\sqrt{y}}$	$x \int_{x-y}^{x+y} \sin \frac{t}{e^t+1} dt$
23	0.4	0.4	$\sin \frac{x-y-xy}{1+x+y+xy} - 1$	$x \int_0^1 e^{\frac{t}{1+t}} dt$
24	1	0.8	$\alpha - x - y - xy, \alpha de$ $\alpha = \int_0^\infty \frac{e^{-t^2}}{t+2} dt$	$xy \int_0^1 e^{\frac{t}{1+t}} dt$
25	1	0.5	$x^2 + y^2 - 10$	$\int_{xy}^{1+xy} e^{\frac{\sqrt{t}}{1+t}} dt$

Образец выполнения лабораторной работы

Студент 4 курса 5 гр. Магомедов М.М.

Вариант 0

Тема: Метод сеток решения задачи Дирихле для эллиптического уравнения.

Цель: Научиться решать задачу Дирихле для эллиптического уравнения методом сеток, используя ЭВМ.

Задание

Методом сеток в прямоугольнике $D = \{0 < x < 0.5, 0 < y < 1.0\}$ с границей Γ решить задачу Дирихле

$$\Delta u = -\int_0^y \frac{\sin \pi x t}{1+x^2 t^2} dt, \quad (x, y) \in D, \quad (0.1)$$

$$u|_{\Gamma} = 10(x - x^2) \cos \pi y$$

Для этой цели составить методом сеток разностную схему, аппроксимирующую данную задачу Дирихле с порядком погрешности $O(h^2 + l^2)$, где h и l – шаги по оси ox и oy соответственно. Найти решение составленной разностной схемы

- б) методом матричной прогонки;
- б) итерационным методом Либмана.

Полученные этими методами приближенные значения решения выписать в точках (x_i, y_j) , где $x_i = ih_0, y_j = jl_0, h_0 = \frac{0.5}{10} = 0.05, l_0 = \frac{1}{10} = 0.1$.
Заполнить соответствующие этим методами 2 таблицы видов

Решение задачи Дирихле методом матричной прогонки (итерационным методом Либмана)

x \ y	0.0	h_0	$2h_0$...	0.5
0.0				...	
l_0				...	
$2l_0$...	
...
1.5				...	

При заполнении таблиц в результатах сохранять по два десятичных знака. В расчетах взять $\max(h, l) \leq 0.1$.

Для контроля работы составленных программ найти указанными методами приближенное решение, самостоятельно составленной тестовой задачи Дирихле того же вида, что и данная. Заполнить две таблицы приведенного выше вида, озаглавив их:

а) Таблица значений решения тестовой задачи методом матричной прогонки;

б) Таблица точных значений решения тестовой задачи.

Найти максимальную по модулю погрешность метода матричной прогонки для тестовой задачи и выписать ее в форме с плавающей точкой. Указать в какой точке это максимальное значение модуля погрешности достигается.

При применении итерационного метода Либмана число итераций k_0 взять таким, чтобы выполнилось условие (9) с $\varepsilon = 10^{-4}$. Указать чему равно k_0 для своей и тестовой задачи.

Сделать выводы.

Выполнение лабораторной работы

Правая часть уравнения содержит интеграл с переменным верхним пределом, первообразная которого не выражается конечным числом арифметических операций с элементарными функциями. Поэтому для его использования в программе опишем его как процедуру-функцию f , пользуясь численными методами интегрирования. Именно воспользуемся численным методом трапеции

$$\int_a^b g(x) dx \approx h \left(\frac{g(a) + g(b)}{2} + \sum_{i=1}^{N-1} g(a + ih) \right), \quad h = \frac{b-a}{N}. \quad (0.2)$$

Порядок точности формулы (0.2) равен $O(h^2)$. Поэтому если возьмем $h = 0.01$, то точность будет порядка 10^{-4} . Можно было бы воспользоваться каким-нибудь пакетом прикладных программ. Тогда функция $f(x, y)$ получилась бы настолько сложной, что в дальнейшем ее использовать в программе было бы довольно трудно.

1. Решение задачи Дирихле методом матричной прогонки

Сначала составим разностную схему, аппроксимирующую задачу (0.1) с порядком $O(h^2 + l^2)$, где h, l – шаги по осям ox, oy соответственно. Она имеет вид

$$\begin{cases} \frac{u_{m+1,n} - 2u_{mn} + u_{m-1,n}}{h^2} + \frac{u_{m,n+1} - 2u_{mn} + u_{m,n-1}}{l^2} = f(x_m, y_n), \\ m = 1, 2, \dots, M-1, \quad n = 1, 2, \dots, N-1; \\ u_{mn} = \varphi(x_m, y_n), \\ \{m = \overline{0, M}, n = 0; m = \overline{0, M}, n = N; m = 0, n = \overline{0, N}; m = M, n = \overline{0, N}, \end{cases} \quad (0.3)$$

где $x_m = mh, \quad y_n = nl, \quad f(x, y) = -\int_0^y \frac{\sin \pi xt}{1+x^2 t^2} dt, \quad \varphi(x, y) = 10(x-x^2) \cos \pi y.$

Это система $(M-1)(N-1)$ линейных алгебраических уравнений со столькими же неизвестными

$$u_{mn} \approx u(x_m, y_n), \quad m = 1, 2, \dots, M-1, \quad n = 1, 2, \dots, N-1.$$

Решение u_{mn} системы разностных уравнений (0.3) найдем по приведенному в [1] алгоритму метода матричной прогонки. Пусть M – число разбиений отрезка $[0, 0.5]$ оси ox , N – число разбиений отрезка $[0, 1]$ оси oy . Решим задачу (0.1) в случае $M \gg N$. Это условие выполнится, если взять $h = 0.01, \quad l = 0.1,$

$$\text{тогда } M = \left[\frac{0.5}{0.01} \right] = 50, \quad N = \left[\frac{1}{0.1} \right] = 10.$$

В методе матричной прогонки [1] система (0.3) сводится к следующей векторной системе относительно неизвестного вектора u_m

$$\begin{cases} u_{m+1} + Au_m + u_{m-1} = f_m, \quad m = 1, 2, \dots, M-1, \\ u_0 = \varphi_0, \quad u_M = \varphi_a, \end{cases} \quad (0.4)$$

где

$$u_m = \begin{pmatrix} u_{m1} \\ u_{m2} \\ \vdots \\ u_{m,N-1} \end{pmatrix}, \quad m = 0, 1, \dots, M,$$

A – трехдиагональная матрица порядка $N-1$ вида

$$A = \begin{pmatrix} -(2+2\alpha) & \alpha & 0 & \Lambda & 0 & 0 & 0 \\ \alpha & -(2+2\alpha) & \alpha & \Lambda & 0 & 0 & 0 \\ \Lambda & \Lambda & \Lambda & \Lambda & \Lambda & \Lambda & \\ 0 & 0 & 0 & \alpha & -(2+2\alpha) & \alpha & \\ 0 & 0 & 0 & 0 & \alpha & -(2+2\alpha) & \end{pmatrix}, \quad \alpha = \frac{h^2}{l^2},$$

$$f_m = \begin{pmatrix} h^2 f_{m1} - \alpha \varphi(x_m, 0) \\ h^2 f_{m2} \\ M \\ h^2 f_{m,N-2} \\ h^2 f_{m,N-1} - \alpha \varphi(x_m, b) \end{pmatrix}, \quad (0.5)$$

$$\varphi_0 = \begin{pmatrix} \varphi(0, y_1) \\ \varphi(0, y_2) \\ M \\ \varphi(0, y_{N-1}) \end{pmatrix}, \quad \varphi_a = \begin{pmatrix} \varphi(a, y_1) \\ \varphi(a, y_2) \\ M \\ \varphi(a, y_{N-1}) \end{pmatrix}.$$

Векторную систему (0.4) также, как аналогичную скалярную систему можно решить методом прогонки, который в этом случае называется *методом матричной прогонки*. Алгоритм этого метода состоит из следующих шагов (см. [1])

3. По формуле

$$R_{m+1} = -(A + R_m)^{-1}, \quad m = 1, 2, \Lambda, M-1, \quad (0.6)$$

полагая $R_1 = 0$, вычисляем матрицы

$R_m = (R_{ij}^{(m)})$, $m = 1, 2, \Lambda, M$, $i, j = 1, 2, \Lambda, N-1$, затем по формуле

$$s_{m+1} = R_{m+1}(s_m - f_m), \quad m = 1, 2, \Lambda, M-1, \quad (0.7)$$

полагая $s_1 = \varphi_0$, вычисляем векторы $s_m = (s_1^{(m)}, s_2^{(m)}, \Lambda, s_{N-1}^{(m)})^T$, T – знак транспонирования, $m = 1, 2, \Lambda, M-1$.

4. По формуле

$$u_{m-1} = R_m u_m + s_m, \quad m = M, M-1, \Lambda, 2, \quad (0.8)$$

полагая $u_M = \varphi_a$, вычисляем искомые векторы

$u_M, u_{M-1}, \Lambda, u_1, u_0 = \varphi_0$.

Для вычисления обратной матрицы A^{-1} применим процедуру *Gauss*(n, a, x, t) решения системы $Ax = b$. В этой процедуре a – расширенная матрица, т.е. матрица $A = (a_{ij})$, $i, j = 1, 2, \Lambda, n$ с

добавлением $n+1$ -го столбца из элементов вектора b , $t=0$, если $\det A=0$, $t=1$, если $\det A \neq 0$. Столбцы $x^{(i)}$ обратной матрицы находим как решения систем

$$Ax^{(i)} = e_i, \quad i = 1, 2, \dots, n,$$

пользуясь процедурой *Gauss*(n, a, x, t). Здесь $e_1 = (1, 0, \dots, 0)^T$, $e_2 = (0, 1, \dots, 0)^T, \dots, e_n = (0, 0, \dots, 1)^T$, T – знак транспонирования.

При составлении программы решения задачи (0.1) методом матричной прогонки по приведенному выше алгоритму сначала составим процедуру, по которой вычисляются матрицы R_m и векторы s_m и f_m , пользуясь формулами (0.6), (0.7) и (0.5) соответственно. Затем, пользуясь формулой (0.8), вычисляем векторы $u_M, u_{M-1}, \dots, u_1, u_0$.

Приведем программу реализации этого метода на языке программирования Турбо-Паскаль. Она снабжена краткими комментариями для облегчения понимания. Работа программы проверяется на примере тестовой задачи Дирихле

$$\Delta u = -10(4 + \pi^2(x - 2x^2))\sin \pi y, \quad (x, y) \in D$$

$$u|_{\Gamma} = 0,$$

с решением $u(x, y) = 10(x - 2x^2)\sin \pi y$. Здесь D та же область, что и в задаче (0.1).

Программа 1

Решения задачи Дирихле методом матричной прогонки

```
uses crt;
const n=9; {n=N-1}    m=50;
type mas= array[1..n,1..n+1] of real;
      matr= array[1..n,1..n] of real;
      vec=array[1..n] of real;
      stb=set of byte;
var x:vec;
    i,j,tt,k1,k,t:integer;
    ss,vv,h,l,al:real;
    r,c,s1,f1,fm,fi0,sm:vec;
    b,r1,r2,rm,a2,a:matr;
    max,i0,j0:real;
```



```

    u,ut:array[0..m,0..n+1] of real;
    aa:mas;
    s:stb;
procedure Gauss(n:integer;a:mas;var x:vec;var t:integer);
var k,m,i,j,j0:integer;
    l:array[1..21] of integer;
begin t:=1;
s:=[];
for i:=1 to n do
begin
    max:=a[i,1]; j0:=1; for j:=1 to n do
    if abs(a[i,j])>abs(max)then
begin
    max:=a[i,j]; j0:=j;
end;
    l[i]:=j0;
    s:=s+[j0];
    if max=0 then t:=0 else
begin
    for k:=1 to n do if k<>i then
    for m:=1 to n+1 do if not (m in s)
    then a[k,m]:=(a[k,m]*a[i,j0]-a[i,m]*a[k,j0])/max;
    for m:=1 to n+1 do if not (m in s)
    then a[i,m]:=a[i,m]/max;for k:=1 to n do a[k,j0]:=0;a[i,j0]:=1;
    end;
end;
end;
for k:=1 to n do begin
i:=l[k];
x[i]:=a[k,n+1]; end;
end;
    procedure obrm(n:integer;a:matr; var a0:matr;var z:integer);
    var x:vec;
        i,j,k,m,z1:integer;
        aa:mas;
begin
    for i:=1 to n do
    for j:=1 to n do
    aa[i,j]:=a[i,j];

```

```

for i:=1 to n do
begin
  for j:=1 to n do if j=i then aa[j,n+1]:=1 else aa[j,n+1]:=0;
  gauss(n,aa,x,t);
  if t=1 then
  begin z:=1;
  for j:=1 to n do a0[j,i]:=x[j];
  for k:=1 to n do
  for m:=1 to n do
  aa[k,m]:=a[k,m];
  end else z:=0
  end;
end;
function rr(x,t:real):real; { подынтегральная функция }
begin
rr:=sin(pi*x*t)/(1+sqr(x*t));
end;
function f(x,y:real):real; { правая часть уравнения }
var
h,s:real;
i,k:integer;
begin
h:=0.01;
k:=round(100*y);
s:=(rr(x,0)+rr(x,y))/2;
for i:=1 to k-1 do
s:=s+rr(x,i*h);
f:=h*s; { f:=-10*(4+pi*pi*(x-2*x*x))*sin(pi*y) — правая часть
уравнения тестовой задачи }
end;
function fi(x,y:real):real;{ граничная функция }
begin
fi:= 10*(x-x*x)*sin(pi*y); { fi:=0— граничная функция тестовой
задачи };
end;
procedure rmsm(m:integer; a:matr;fi0:vec; var rm:matr; var
sm,fm:vec);
{ вычисляет rm,fm,sm }

```

{ В этой процедуре r_1, s_1, f_1 - m -е векторы, r_m, s_m, f_m - $(m+1)$ -е векторы }

```
var h,l,al:real; i,j,k:integer;
begin k:=1;
  h:=0.01; l:=0.1; al:=0.01;
  for i:=1 to n do
    for j:=1 to n do
      r1[i,j]:=0; {R1=0} s1:=f1;
      {формируется вектор fm при m=1}
      for j:=1 to n do
        if j=1 then f1[j]:=h*h*f(h,j*1)-al*fi(h,0)
        else if j=n then f1[j]:=h*h*f(h,j*1)-al*fi(h,1)
        else f1[j]:=h*h*f(h,j*1);
        {вектор fm сформирован}
      if m=1 then
begin
  rm:=r1; {R1=0} sm:=s1; fm:=f1;
end
else
begin
  repeat k:=k+1;
  for i:=1 to n do
    for j:=1 to n do
      a2[i,j]:=a[i,j]+r1[i,j];
      obrm(n,a2,b,t); {образована матрица обратная A+Rm}
      for i:=1 to n do
        for j:=1 to n do
          rm[i,j]:=-b[i,j]; {это матрица Rm+1}
          {формируется вектор fm при m=k>1}
          for j:=1 to n do
            if j=1 then fm[j]:=h*h*f(k*h,j*1)-al*fi(k*h,0)
            else if j=n then fm[j]:=h*h*f(k*h,j*1)-al*fi(k*h,1)
            else fm[j]:=h*h*f(k*h,j*1);
            {вектор fm сформирован}
            {вычисляется вектор sm по формуле (0,7)}
            for i:=1 to n do
begin
  ss:=0;
```

```

for j:=1 to n do
ss:=ss+r1[i,j]*(s1[j]-f1[j]);
sm[i]:=ss;
end;
{ вектор sm вычислен }
s1:=sm; f1:=fm;r1:=rm;
until k=m;
end;
end;{конец процедуры }
begin clrscr; {Начало основной программы }
h:=0.01; l:=0.1; al:=0.01;
{формируется матрица A }
for i:=1 to n do a[i,i]:=-2-2*al;
for i:=1 to n do
for j:=1 to n do
if abs(i-j)=1 then
a[i,j]:=al else if i<>j then
a[i,i]:=-2-2*al;
{ матрица A сформирована }
for j:=1 to n do fi0[j]:=fi(0,j*1); {вектор fi0 }
for j:=0 to n do u[0,j]:=fi(0,j*1); {вектор u0 }
for j:=0 to n do u[m,j]:=fi(0.5,j*1); {вектор uM }
for i:=0 to m-1 do
begin
u[i,0]:=fi(i*h,0);
u[i,n+1]:=fi(i*h,1);
end;
for i:=m downto 2 do
begin rmsm(i,a,fi0,rm,sm,fm); { вычисление векторов  rm,fm,sm
при i=m }
{вычисление ui при i=m }
for j:=1 to n do
begin
vv:=0;
for k:=1 to n do
vv:=vv+rm[j,k]*u[i,k];
u[i-1,j]:=vv+sm[j];
end; {конец вычислений векторов ui при i=m }

```

```

end; { конец вычислений векторов  $u_i$  для  $i$  от  $M$  до нуля назад }
{ вывод на экран решения }
  writeln; writeln;
  write( ' |y\|x|');
  for i:=0 to m do
    if i mod 5=0 then write(h*i:5:2, '|');writeln;
    writeln(' ----- ');
    for j:=0 to n+1 do
begin
  for i:=0 to m do
    if i mod 5=0 then write(u[i,j]:5:2, '|'); writeln
end;
readkey; { конец вывода на экран решения }
  for i:=0 to m do
    for j:=0 to n+1 do
      ut[i,j]:=10*(i*h-2*sqr(i*h))*sin(pi*j*1);
      writeln; writeln;
      writeln('          точное решение тестовой задачи');
      write( ' |y\|x|');
      for i:=0 to m do
        if i mod 5=0 then write(h*i:5:2, '|');writeln;
        writeln(' ----- ');
        for j:=0 to n+1 do
begin
  for i:=0 to m do
    if i mod 5=0 then write(ut[i,j]:5:2, '|'); writeln
end;
readkey; { конец вывода на экран точного решения }
max:=abs(u[0,0]-ut[0,0]);
  for i:=0 to m do
    for j:=0 to n+1 do
      if abs(u[i,j]-ut[i,j])>max then begin i0:=i;j0:=j;
      max:= abs(u[i,j]-ut[i,j]) end;
      writeln(максимальная погрешность =', max:8,
      ' и достигается при x=',i0*h:5:2, ' y=',j0*1:5:2); readkey;
end.

```

По этой программе получены следующие результаты для тестовой задачи

Решение тестовой задачи методом матричной прогонки

$y \backslash x$	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
0.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.1	0.00	0.13	0.24	0.32	0.36	0.38	0.36	0.32	0.24	0.14	0.00
0.2	0.00	0.25	0.45	0.60	0.69	0.72	0.69	0.61	0.46	0.26	0.00
0.3	0.00	0.35	0.62	0.83	0.95	0.99	0.95	0.84	0.64	0.36	0.00
0.4	0.00	0.41	0.73	0.97	1.11	1.16	1.12	0.98	0.75	0.42	0.00
0.5	0.00	0.43	0.77	1.02	1.17	1.22	1.18	1.03	0.79	0.44	0.00
0.6	0.00	0.41	0.73	0.97	1.11	1.16	1.12	0.98	0.75	0.42	0.00
0.7	0.00	0.35	0.62	0.83	0.95	0.99	0.95	0.84	0.64	0.36	0.00
0.8	0.00	0.25	0.45	0.60	0.69	0.72	0.69	0.61	0.46	0.26	0.00
0.9	0.00	0.13	0.24	0.32	0.36	0.38	0.36	0.32	0.24	0.14	0.00
1.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Точное решение тестовой задачи

$y \backslash x$	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
0.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.1	0.00	0.14	0.25	0.32	0.37	0.39	0.37	0.32	0.25	0.14	0.00
0.2	0.00	0.26	0.47	0.62	0.71	0.73	0.71	0.62	0.47	0.26	0.00
0.3	0.00	0.36	0.65	0.85	0.97	1.01	0.97	0.85	0.65	0.36	0.00
0.4	0.00	0.43	0.76	1.00	1.14	1.19	1.14	1.00	0.76	0.43	0.00
0.5	0.00	0.45	0.80	1.05	1.20	1.25	1.20	1.05	0.80	0.45	0.00
0.6	0.00	0.43	0.76	1.00	1.14	1.19	1.14	1.00	0.76	0.43	0.00
0.7	0.00	0.36	0.65	0.85	0.97	1.01	0.97	0.85	0.65	0.36	0.00
0.8	0.00	0.26	0.47	0.62	0.71	0.73	0.71	0.62	0.47	0.26	0.00
0.9	0.00	0.14	0.25	0.32	0.37	0.39	0.37	0.32	0.25	0.14	0.00
1.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Максимальная по модулю погрешность равна $3.0 \cdot 10^{-2}$ и достигается в точке (0.15,0.50). Мы также проверили работу составленной программы на других тестовых задачах. При этом максимальная по модулю погрешность не превосходила $5 \cdot 10^{-2}$. Эту погрешность можно уменьшить, если уменьшить шаги h и l по координатным осям. Зато увеличится объем вычислений,

соответственно время счета. Поэтому сильно уменьшать шаги h и l невыгодно.

По этой же программе получены следующие значения решения задачи (0.1)

**Таблица значений решения задачи Дирихле (0.1)
методом матричной прогонки**

$y \backslash x$	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
0.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.1	0.00	0.05	0.11	0.16	0.23	0.29	0.37	0.45	0.54	0.65	0.77
0.2	0.00	0.10	0.20	0.31	0.43	0.56	0.70	0.85	1.03	1.24	1.47
0.3	0.00	0.14	0.28	0.43	0.59	0.76	0.96	1.17	1.42	1.70	2.02
0.4	0.00	0.16	0.33	0.50	0.69	0.90	1.12	1.38	1.67	2.00	2.38
0.5	0.00	0.17	0.35	0.53	0.73	0.94	1.18	1.45	1.75	2.10	2.50
0.6	0.00	0.16	0.33	0.50	0.69	0.90	1.12	1.38	1.67	2.00	2.38
0.7	0.00	0.14	0.28	0.43	0.59	0.76	0.95	1.17	1.42	1.70	2.02
0.8	0.00	0.10	0.20	0.31	0.43	0.55	0.69	0.85	1.03	1.23	1.47
0.9	0.00	0.05	0.11	0.16	0.22	0.29	0.36	0.45	0.54	0.65	0.77
1.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Выводы

1. Полученное решение положительно в прямоугольнике D .
2. Максимальное значение решения равно 2.50 и достигается в точке $(0.5; 0.5)$ границы Γ прямоугольника D , а минимальное значение решения равно 0 и достигается на левой боковой, нижней и верхней границах прямоугольника D .
3. Уменьшение шагов h и l приводит к увеличению точности решения задачи Дирихле, зато резко возрастает время счета. Поэтому сильно уменьшать шаги h и l невыгодно.

2. Решение задачи Дирихле итерационным методом Либмана.

Рассмотренный нами в п.1 метод матричной прогонки является прямым, т.е. этим методом мы получаем точное решение разностной схемы (0.3) $u_{mn} \approx u(x_m, y_n)$. С помощью итерационного метода Либмана можно найти решение той же

разностной схемы (0.3) с любой наперед заданной точностью, подбирая для этого соответствующее число итераций k . В нашем случае по условию требуется найти значение $k = k_0$ так, чтобы выполнилось неравенство

$$\left| u_{mn}^{(k+1)} - u_{mn}^{(k)} \right| \leq \varepsilon. \quad (0.9)$$

Здесь $u_{mn}^{(k)}$ – последовательные приближения к решению разностной схемы (0.3), которые определяются с помощью итерационного процесса Либмана

$$u_{mn}^{(k+1)} = \frac{l^2(u_{m+1,n}^{(k)} + u_{m-1,n}^{(k)}) + h^2(u_{m,n+1}^{(k)} + u_{m,n-1}^{(k)})}{2(h^2 + l^2)} - \frac{h^2 l^2}{2(h^2 + l^2)} f_{mn} \quad (0.10)$$

$k = 0, 1, 2, \dots, \Lambda$

Чтобы начать итерационный процесс по формуле (0.10) необходимо задать начальное приближение $u_{mn}^{(0)}$ к u_{mn} . Для нашей задачи (0.1) можно взять

$$u_{mn}^{(0)} = \varphi(x_m, y_n) = 10(x_m - 2x_m^2) \sin \pi y_n, \quad m = 0, 1, \dots, M, \quad n = 0, 1, \dots, N.$$

Очевидно, для граничных узлов $u_{mn}^{(0)}$ совпадают со значениями u_{mn} . Теперь остается вычислять последовательные приближения $u_{mn}^{(k)}$ по формулам (0.10). Вычисления $u_{mn}^{(k)}$ будем проводить до тех пор, пока не выполнится условие (0.9) с $\varepsilon = 10^{-4}$.

В соответствии с этим описанием приведем программу решения задачи Дирихле (0.1) на языке Турбо—Паскаль.

Программа 2

Решения задачи Дирихле итерационным методом Либмана.

```
uses crt;
const n=10;    m=50;
var
  i,j,k:integer;
  h,l:real;
  max:real;
  u0,u1,v,fmn:array[0..m,0..n] of real;
function rr(x,t:real):real; { подынтегральная функция }
begin
rr:=sin(pi*x*t)/(1+sqr(x*t));
end;
```



```

function f(x,y:real):real; { правая часть уравнения }
var
h,s:real;
i,k:integer;
begin
h:=0.01;
k:=round(100*y);
s:=(rr(x,0)+rr(x,y))/2;
for i:=1 to k-1 do
s:=s+rr(x,i*h);
f:=h*s; { f:=-10*(4+pi*pi*(x-2*x*x))*sin(pi*y) для тестовой
задачи }
end;

function fi(x,y:real):real;{ граничная функция }
begin
fi:={0} 10*(x-x*x)*sin(pi*y);
end;

begin clrscr;
h:=0.01; l:=0.1;
for i:=0 to m do
for j:=0 to n do
fmn[i,j]:=f(i*h,j*l); { значения функции f(x,y) в узлах сетки }
k:=0;
for i:=0 to m do
for j:=0 to n do
u0[i,j]:=fi(i*h,j*l); { начальные значения u(x,y) во всех узлах
сетки }
u1:=u0;
repeat
k:=k+1;
for i:=1 to m-1 do
for j:=1 to n-1 do
u1[i,j]:=(l*l*(u0[i+1,j]+u0[i-1,j])+h*h*(u0[i,j+1]+u0[i,j-1]))/
(2*(h*h+l*l))-sqr(h*l)/(2*(h*h+l*l))*fmn[i,j];
for i:=0 to m do
for j:=0 to n do
v[i,j]:=abs(u1[i,j]-u0[i,j]);

```


При этом необходимое по условию (0.9) число итераций k_0 оказалось равным 1295.

Выводы

Из сравнения результатов, полученных методом матричной прогонки и итерационным методом Либмана с условием останова (0.9), следует, что эти результаты отличаются не более чем на 0.01. Это объясняется тем, что выполнение условия (0.9) не гарантирует еще требуемой точности.

Литература

1. Крылов В.И., Бобков В.В., Монастырный П.И.
Вычислительные методы, т.2, Москва, 1977, 340 с.