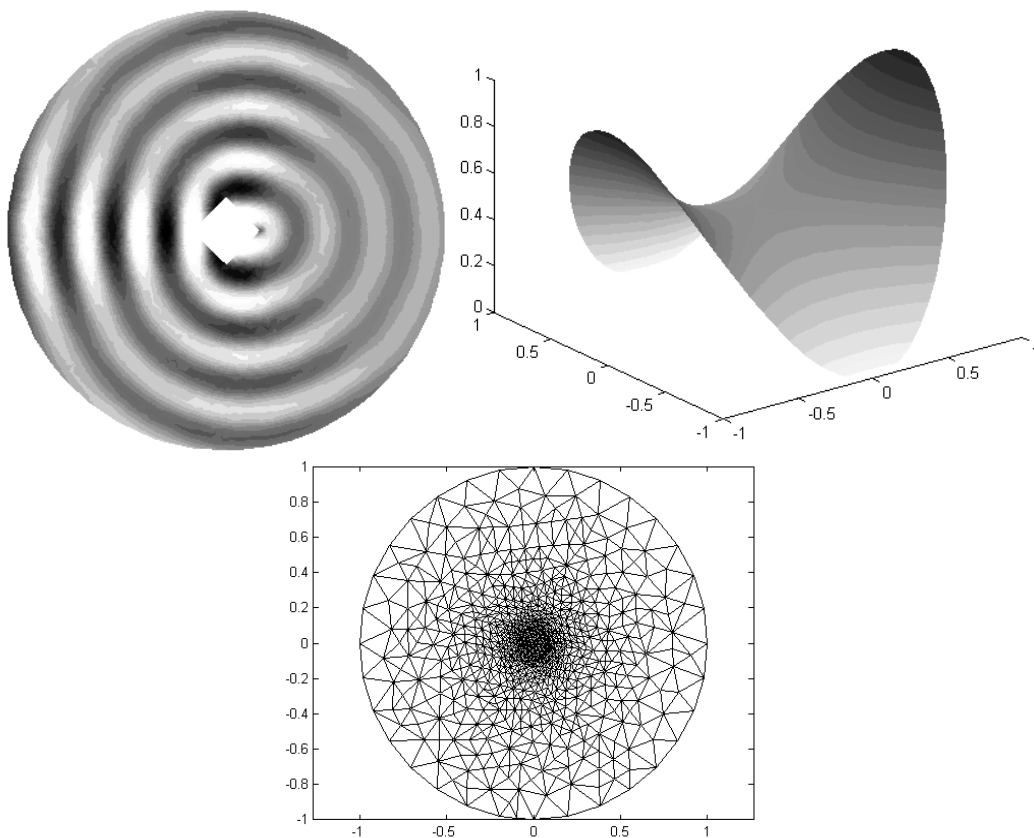




Е.А. Рындин

МЕТОДЫ РЕШЕНИЯ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ФИЗИКИ



СОДЕРЖАНИЕ

ВВЕДЕНИЕ	1
1. УРАВНЕНИЯ МАТЕМАТИЧЕСКОЙ ФИЗИКИ	5
1.1. Эллиптические уравнения	6
1.1.1. Уравнение Лапласа	6
1.1.2. Уравнение Пуассона	9
1.2. Параболические уравнения	11
1.2.1. Уравнение теплопроводности	12
1.2.1. Уравнения непрерывности	13
1.3. Гиперболические уравнения	15
1.3.1. Волновое уравнение	15
1.4. Системы дифференциальных уравнений в частных производных	16
1.4.1. Фундаментальная система уравнений	17
1.4.2. Нормировка	18
1.4.3. Базисы переменных	20
2. ГРАНИЧНЫЕ И НАЧАЛЬНЫЕ УСЛОВИЯ	22
2.1. Граничные условия	23
2.2. Начальные условия	25
3. МЕТОДЫ ДИСКРЕТИЗАЦИИ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ	25
3.1. Метод конечных разностей	26
3.1.1. Конечно-разностные сетки	26
3.1.2. Сеточные функции, конечные разности и шаблоны	27
3.2. Метод конечных элементов	31
3.2.1. Разбиение Дирихле и триангуляция Делоне	31
3.2.2. Метод интегральных тождеств	37
4. МЕТОДЫ РЕШЕНИЯ СИСТЕМ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ	39
4.1. Методы решения систем линейных алгебраических уравнений	39
4.1.1. Метод исключения Гаусса	40
4.1.2. Метод LU-разложения	42
4.1.3. Итерационные методы решения систем линейных алгебраических уравнений	44
4.2. Методы решения систем нелинейных алгебраических уравнений	55
4.2.1. Итерация неподвижной точки	56
4.2.2. Метод Ньютона-Рафсона	61
5. ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ФИЗИКИ В СИСТЕМЕ MATLAB	66
5.1. Примеры решения уравнения Пуассона	66
5.1.1. Решение одномерного уравнения Пуассона методом конечных разностей	67
5.1.2. Решение двумерного уравнения Пуассона методом конечных разностей	73
5.1.3. Решение двумерного уравнения Пуассона методом конечных элементов	83

5.2. Примеры решения уравнения теплопроводности.....	91
5.3. Примеры решения волнового уравнения.....	104
ЗАКЛЮЧЕНИЕ	118
ЛИТЕРАТУРА	119

ВВЕДЕНИЕ

Микроэлектроника является одной из наиболее динамично развивающихся и востребованных отраслей науки и техники. Элементы современных СБИС и микрооптикоэлектромеханических систем (МОЭМС) представляют собой сложные структуры, в основу функционирования которых положены разнообразные физические эффекты. Разработка подобных элементов практически невозможна без решения уравнений математической физики, представляющих собой, как правило, дифференциальные уравнения (ДУ) в частных производных.

Нахождение точного аналитического решения, к сожалению, возможно лишь для весьма ограниченного круга одномерных задач при использовании целого ряда допущений, негативно отражающихся на адекватности полученных результатов. Для решения задач математической физики в случае нескольких измерений необходимо использовать численные методы, позволяющие преобразовать дифференциальные уравнения или их системы в системы алгебраических уравнений. Для решения полученных нелинейных систем алгебраических уравнений или линейных систем большой размерности используют итерационные методы. При этом одной из наиболее сложных проблем является обеспечение сходимости итерационного процесса, в значительной степени определяющей время вычислений. Точность решения определяется шагом координатной сетки, количеством итераций и разрядной сеткой компьютера.

В учебном пособии рассмотрены основные уравнения математической физики, особенности задания граничных и начальных условий, методы дискретизации дифференциальных уравнений в частных производных, методы решения систем алгебраических уравнений, представлены основные этапы решения задач матфизики, включая постановку задачи, выбор базиса переменных, метода дискретизации, формирование координатной сетки, выбор шаблона, метода решения, анализ сходимости и др.

Рассмотренные методы решения уравнений проиллюстрированы примерами для системы MATLAB с комментариями и рекомендациями, позволяющими составить представление об основных правилах и приемах разработки компьютерных программ для решения уравнений математической физики.

Методам решения подобных задач посвящено достаточно много монографий, учебников и учебных пособий [1 - 4]. В данном учебном пособии принята попытка достичь более полного соответствия целям подготовки специалистов в области проектирования электронно-вычислительных средств и микросистем по характеру материала, стилю его изложения и приводимым примерам.

1. УРАВНЕНИЯ МАТЕМАТИЧЕСКОЙ ФИЗИКИ

Разработка и исследование значительной части элементов современных СБИС и МОЭМС связана с решением так называемых задач математической физики (или сокращенно – матфизики), к которым относятся задачи теплопроводности, диффузии, электростатики и электродинамики, задачи о течении жидкости, о распределении плотности электрического тока в проводящей среде, задачи о деформациях твердых тел и многие другие.

Подобные задачи описываются дифференциальными уравнениями в частных производных с дополнительными уравнениями, выражающими граничные и начальные условия. В учебном пособии рассматриваются ДУ в частных производных не выше второго порядка, поскольку эти уравнения охватывают достаточно широкий диапазон физических явлений, положенных в основу функционирования элементов СБИС и МОЭМС и, кроме того, рассмотренные ниже методы решения применимы и к ДУ в частных производных более высоких порядков. В общем случае линейное дифференциальное уравнение в частных производных второго порядка с n независимыми переменными имеет вид

$$\sum_{\alpha, \beta=1}^n A_{\alpha\beta}(\mathbf{x}) \frac{\partial^2 u}{\partial x_\alpha \partial x_\beta} + \sum_{\alpha=1}^n B_\alpha(\mathbf{x}) \frac{\partial u}{\partial x_\alpha} + C(\mathbf{x}) u = f(\mathbf{x}), \quad (1)$$

где $\mathbf{x} = [x_1, x_2, \dots, x_n]$ – вектор (матрица-строка) независимых переменных¹; u – искомая функция независимых переменных; $A_{\alpha\beta}(\mathbf{x})$, $B_\alpha(\mathbf{x})$, $C(\mathbf{x})$, $f(\mathbf{x})$ – некоторые вещественные функции независимых переменных [1, 2].

Уравнение (1) всегда может быть приведено к одной из трех стандартных канонических форм. По соотношению значений $A_{\alpha\beta}(\mathbf{x})$ уравнения относят к эллиптическим, параболическим или гиперболическим в точке \mathbf{x} . В частности, для дифференциальных уравнений в частных производных второго порядка с двумя независимыми переменными x, y , которые могут быть представлены в виде

$$\begin{aligned} &A_{xx}(x, y) \frac{\partial^2 u}{\partial x^2} + A_{xy}(x, y) \frac{\partial^2 u}{\partial x \partial y} + A_{yy}(x, y) \frac{\partial^2 u}{\partial y^2} + \\ &+ B_x(x, y) \frac{\partial u}{\partial x} + B_y(x, y) \frac{\partial u}{\partial y} + C(x, y) u = f(x, y) \end{aligned}, \quad (2)$$

тип ДУ определяется знаком выражения, называемого дискриминантом

$$D(x, y) = A_{xy}^2(x, y) - 4A_{xx}(x, y) A_{yy}(x, y). \quad (3)$$

Если $D(x, y) < 0$, дифференциальное уравнение является эллиптическим в точке (x, y) .

Если $D(x, y) = 0$, дифференциальное уравнение является параболическим в точке (x, y) .

¹ В дальнейшем по тексту векторные величины и матрицы будут выделяться жирным шрифтом.

Если $D(x, y) > 0$, дифференциальное уравнение является гиперболическим в точке (x, y) .

Если коэффициенты A_{xx} , A_{xy} , A_{yy} постоянные и значение D не зависит от x , y , то в зависимости от знака D уравнение является полностью эллиптическим, гиперболическим или параболическим [1].

1.1. Эллиптические уравнения

Рассмотрим некоторые задачи математической физики, приводящие к решению эллиптических уравнений.

1.1.1. Уравнение Лапласа

Многие стационарные, т.е. не изменяющиеся во времени физические процессы описываются уравнениями эллиптического типа, в простейшем случае (однородной среды и отсутствия источников) – уравнением Лапласа [2], которое для трех направлений координат (x, y, z) может быть записано в виде:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = 0, \quad (4)$$

где $u = u(x, y, z)$ – искомая функция координат.

В операторной форме уравнение Лапласа (4) может быть представлено следующим образом:

$$\Delta u = 0, \quad (5)$$

где $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ – оператор Лапласа.

Рассмотрим задачу о стационарном распределении тепла в некотором объеме V , ограниченном замкнутой поверхностью S трехмерного пространства $X = (x, y, z)$.

Процесс теплопроводности или кондукции определяется законом Фурье, согласно которому вектор плотности теплового потока \mathbf{W} пропорционален градиенту температуры $T = T(x, y, z)$ [1]:

$$\mathbf{W} = -k \cdot \text{grad}(T), \quad (6)$$

где $k = k(x, y, z)$ – коэффициент теплопроводности.

Плотность теплового потока равна количеству теплоты, протекающему в единицу времени через единичную площадь изотермической поверхности [1].

Как правило, цель стационарной задачи теплопроводности сводится к необходимости нахождения зависимости температуры от координат (x, y, z) при известном распределении плотности источников тепла $f(x, y, z)$. Поскольку функция $f(x, y, z)$ не входит непосредственно в уравнение Фурье (6), необходимо выполнить ряд предварительных преобразований.

Из приведенного выше определения плотности теплового потока следует, что суммарное количество тепла Q_s , прошедшее в единицу времени через замк-

нутую поверхность S , ограничивающую объем V , в общем случае выражается интегралом

$$Q_s = \iint_S \mathbf{W} \cdot d\mathbf{S} , \quad (7)$$

где $d\mathbf{S}$ – вектор, модуль которого численно равен площади dS соответствующего бесконечно малого элемента поверхности, а направление совпадает с направлением нормали к этому элементу; $\mathbf{W} \cdot d\mathbf{S} = W \cdot dS \cdot \cos(\gamma)$ – скалярное произведение векторов \mathbf{W} и $d\mathbf{S}$; γ - угол между ними.

Суммарное количество тепла Q_V , выделяющегося в единицу времени в объеме V , ограниченном поверхностью S , определяется интегралом

$$Q_V = \iiint_V f(x, y, z) dV . \quad (8)$$

Очевидно, в данном случае уравнение баланса тепла должно отражать факт равенства количества тепла Q_s , прошедшего в единицу времени через замкнутую поверхность S , ограничивающую объем V , и количества тепла Q_V , выделяющегося в единицу времени в этом объеме:

$$\iint_S \mathbf{W} \cdot d\mathbf{S} = \iiint_V f(x, y, z) dV . \quad (9)$$

Согласно теореме Остроградского-Гаусса

$$\iint_S \mathbf{W} \cdot d\mathbf{S} = \iiint_V \operatorname{div} \mathbf{W} dV . \quad (10)$$

Тогда, подставив (10) в (9), получим

$$\iiint_V \operatorname{div} \mathbf{W} dV = \iiint_V f(x, y, z) dV ; \quad (11)$$

$$\operatorname{div} \mathbf{W} = f(x, y, z) . \quad (12)$$

Подставив в уравнение (12) закон Фурье (6), получим уравнение для стационарной задачи теплопроводности в векторной форме

$$\operatorname{div}(k(x, y, z) \cdot \operatorname{grad}(T)) = -f(x, y, z) . \quad (13)$$

Если источники тепла отсутствуют ($f(x, y, z) = 0$) и среда однородна ($k(x, y, z) = \text{const}$), уравнение (13) можно переписать в виде

$$\operatorname{div}(\operatorname{grad}(T)) = 0 . \quad (14)$$

Учитывая, что по определению градиент некоторого скалярного поля $u = u(x, y, z)$ определяется выражением

$$\operatorname{grad}(u) = \frac{\partial u}{\partial x} \mathbf{e}_x + \frac{\partial u}{\partial y} \mathbf{e}_y + \frac{\partial u}{\partial z} \mathbf{e}_z , \quad (15)$$

где $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$ – единичные вектора (орты) в направлениях соответствующих координатных осей, а дивергенция некоторого векторного поля $\mathbf{v} = \mathbf{v}(x, y, z)$ – выражением

$$\operatorname{div}(\mathbf{v}) = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}, \quad (16)$$

где v_x, v_y, v_z - проекции вектора \mathbf{v} на соответствующие оси координат, уравнение (14) можно переписать в частных производных

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = 0 \quad (17)$$

или в операторной форме

$$\Delta T = 0, \quad (18)$$

то есть в виде уравнения Лапласа.

Процессы диффузии вещества во многом аналогичны процессам теплопроводности. При описании диффузии аналогом закона Фурье является закон Нернста, согласно которому вектор плотности потока вещества \mathbf{W} пропорционален градиенту концентрации $N = N(x, y, z)$ [1]:

$$\mathbf{W} = -D \cdot \operatorname{grad}(N), \quad (19)$$

где $D = D(x, y, z)$ – коэффициент диффузии.

Плотность потока вещества равна количеству частиц вещества (атомов, молекул), диффундирующему в единицу времени через единичную площадь поверхности.

Подставляя выражение (19) в уравнение (12), при отсутствии источников диффундирующего вещества ($f(x, y, z) = 0$) и однородной среде ($D(x, y, z) = \text{const}$) получим уравнение Лапласа в векторной форме

$$\operatorname{div}(\operatorname{grad}(N)) = 0, \quad (20)$$

в частных производных

$$\frac{\partial^2 N}{\partial x^2} + \frac{\partial^2 N}{\partial y^2} + \frac{\partial^2 N}{\partial z^2} = 0 \quad (21)$$

и в операторной форме

$$\Delta N = 0. \quad (22)$$

К уравнению Лапласа приводят и многие другие задачи, например, задача о распределении электростатического поля в однородной непроводящей среде в отсутствие электрических зарядов.

В общем виде данная задача описывается уравнениями Максвелла

$$\operatorname{rot}(\mathbf{E}) = 0; \quad (23)$$

$$\operatorname{div}(\varepsilon(x, y, z) \mathbf{E}) = \frac{\rho(x, y, z)}{\varepsilon_0}, \quad (24)$$

где $\mathbf{E} = \mathbf{E}(x, y, z)$ – вектор напряженности электрического поля; $\rho = \rho(x, y, z)$ – объемная плотность электрических зарядов; $\varepsilon = \varepsilon(x, y, z)$ – диэлектрическая

проницаемость среды; ε_0 – электрическая постоянная. Уравнение (23) выражает отсутствие вихревых электрических полей.

Если непроводящая среда однородна ($\varepsilon(x, y, z) = \text{const}$) и электрические заряды в объеме отсутствуют или уравновешены ($\rho(x, y, z) = 0$), уравнение (24) принимает вид

$$\text{div}(\mathbf{E}) = 0 . \quad (25)$$

Поскольку напряженность электрического поля \mathbf{E} связана с электрическим потенциалом φ равенством [1, 4]

$$\mathbf{E} = -\text{grad}(\varphi) , \quad (26)$$

то, подставляя (26) в (25) и учитывая выражения (5), (15) и (16), получим уравнение Лапласа в векторной форме

$$\text{div}(\text{grad}(\varphi)) = 0 , \quad (27)$$

в частных производных

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} = 0 \quad (28)$$

и в операторной форме

$$\Delta \varphi = 0 . \quad (29)$$

1.1.2. Уравнение Пуассона

В общем случае в векторной форме уравнение Пуассона имеет вид [1, 4]

$$\text{div}(A(x, y, z) \cdot \text{grad}(u)) = f(x, y, z) , \quad (30)$$

где $u = u(x, y, z)$ – искомая функция; $A(x, y, z)$, $f(x, y, z)$ – некоторые функции независимых переменных.

Уравнение (30) может быть записано в частных производных как

$$\frac{\partial}{\partial x} \left(A(x, y, z) \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(A(x, y, z) \frac{\partial u}{\partial y} \right) + \frac{\partial}{\partial z} \left(A(x, y, z) \frac{\partial u}{\partial z} \right) = f(x, y, z) , \quad (31)$$

или в операторной форме как

$$\nabla \left(A(x, y, z) \cdot \nabla u \right) = f(x, y, z) , \quad (32)$$

где ∇ - оператор Наббла, определяемый выражением

$$\nabla = \frac{\partial}{\partial x} + \frac{\partial}{\partial y} + \frac{\partial}{\partial z} . \quad (33)$$

Из выражений (30) – (32) видно, что уравнение Пуассона является обобщением уравнения Лапласа для случая отличной от нуля правой части. Покажем это на примерах, приведенных выше.

Рассмотрим задачу о стационарном распределении тепла в некотором объеме V , ограниченном замкнутой поверхностью S трехмерного пространства $X = (x, y, z)$, которая в векторной форме описывается уравнением (13).

При наличии в объеме V источников тепла ($f(x, y, z) \neq 0$) и в случае неоднородной среды ($k = k(x, y, z)$), уравнение (13) в частных производных можно переписать в виде

$$\frac{\partial}{\partial x} \left(k(x, y, z) \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k(x, y, z) \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(k(x, y, z) \frac{\partial T}{\partial z} \right) = -f(x, y, z), \quad (34)$$

или в операторной форме

$$\nabla \left(k(x, y, z) \cdot \nabla T \right) = -f(x, y, z). \quad (35)$$

Если среда однородна ($k(x, y, z) = \text{const}$), то k можно вынести за знак частной производной в выражении (34) или за знак оператора Наббля в выражении (35). В результате получим частный случай уравнения Пуассона в виде

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = -\frac{f(x, y, z)}{k(x, y, z)}, \quad (36)$$

или в операторной форме

$$\Delta T = -\frac{f(x, y, z)}{k(x, y, z)}. \quad (37)$$

Если среда анизотропна, т.е. коэффициент теплопроводности k зависит от направления распространения тепла и является тензором

$$\mathbf{k} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}, \quad (38)$$

то уравнение (34) преобразуется к виду [1]

$$\sum_{i,j=1}^3 \frac{\partial}{\partial x_i} \left(k_{ij} \frac{\partial T}{\partial x_j} \right) = f(x_1, x_2, x_3), \quad (39)$$

где пространство (x_1, x_2, x_3) соответствует (x, y, z) .

Если в тензоре \mathbf{k} только элементы главной диагонали отличны от нуля ($k_{ij} = 0$ для $i \neq j$), то уравнение (39) может быть записано в виде

$$\frac{\partial}{\partial x} \left(k_{11} \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k_{22} \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(k_{33} \frac{\partial T}{\partial z} \right) = -f(x, y, z). \quad (40)$$

Процессы диффузии при наличии источников диффундирующего вещества ($f(x, y, z) \neq 0$) и в случае неоднородной среды ($D = D(x, y, z)$) описываются уравнением Пуассона в векторной форме

$$\text{div}(D(x, y, z) \cdot \text{grad}(N)) = f(x, y, z), \quad (41)$$

в частных производных

$$\frac{\partial}{\partial x} \left(D(x, y, z) \frac{\partial N}{\partial x} \right) + \frac{\partial}{\partial y} \left(D(x, y, z) \frac{\partial N}{\partial y} \right) + \frac{\partial}{\partial z} \left(D(x, y, z) \frac{\partial N}{\partial z} \right) = f(x, y, z), \quad (42)$$

или в операторной форме

$$\nabla \left(D(x, y, z) \cdot \nabla N \right) = f(x, y, z). \quad (43)$$

Для однородной среды ($D(x, y, z) = \text{const}$), аналогично выражениям (36), (37) можно записать

$$\frac{\partial^2 N}{\partial x^2} + \frac{\partial^2 N}{\partial y^2} + \frac{\partial^2 N}{\partial z^2} = \frac{f(x, y, z)}{D(x, y, z)}, \quad (44)$$

или в операторной форме

$$\Delta N = \frac{f(x, y, z)}{D(x, y, z)}. \quad (45)$$

Задача о распределении электростатического поля в непроводящей среде при наличии электрических зарядов описывается уравнениями (23), (24). С учетом выражения (26), в векторной форме можно записать

$$\text{div}(\varepsilon(x, y, z) \cdot \text{grad}(\varphi)) = -\frac{\rho(x, y, z)}{\varepsilon_0}, \quad (46)$$

в частных производных

$$\frac{\partial}{\partial x} \left(\varepsilon(x, y, z) \frac{\partial \varphi}{\partial x} \right) + \frac{\partial}{\partial y} \left(\varepsilon(x, y, z) \frac{\partial \varphi}{\partial y} \right) + \frac{\partial}{\partial z} \left(\varepsilon(x, y, z) \frac{\partial \varphi}{\partial z} \right) = -\frac{\rho(x, y, z)}{\varepsilon_0}, \quad (47)$$

или в операторной форме

$$\nabla \left(\varepsilon(x, y, z) \cdot \nabla \varphi \right) = -\frac{\rho(x, y, z)}{\varepsilon_0}. \quad (48)$$

Для однородной среды ($\varepsilon(x, y, z) = \text{const}$), аналогично выражениям (36), (37) можно записать

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} = -\frac{\rho(x, y, z)}{\varepsilon \varepsilon_0}, \quad (49)$$

или в операторной форме

$$\Delta \varphi = -\frac{\rho(x, y, z)}{\varepsilon \varepsilon_0}. \quad (50)$$

1.2. Параболические уравнения

Рассмотрим некоторые задачи математической физики, приводящие к решению параболических уравнений.

1.2.1. Уравнение теплопроводности

Многие нестационарные, т.е. изменяющиеся во времени физические процессы описываются уравнениями параболического типа. Рассмотрим в качестве примера нестационарное уравнение теплопроводности, которое является более общим случаем уравнения (35) и получается на основании закона Фурье (6) в результате следующих рассуждений.

Рассмотрим задачу о нестационарном распределении тепла в некотором объеме V , ограниченном замкнутой поверхностью S трехмерного пространства $X = (x, y, z)$.

Количество тепла q_V , выделившегося в объеме V , ограниченном поверхностью S , за некоторый промежуток времени dt , можно определить как

$$q_V = Q_V dt, \quad (51)$$

где Q_V – суммарное количество тепла, выделяющегося в единицу времени в объеме V , ограниченном поверхностью S , определяемое интегралом (8).

Учитывая, что рассматривается неравновесное состояние системы, часть тепла $q_T < q_V$ идет на изменение во времени температуры в объеме V и определяется выражением

$$q_T = q dT, \quad (52)$$

где q – суммарное количество тепла, необходимого для изменения температуры объема V на один градус; dT – изменение температуры объема V за промежуток времени dt .

Остальная часть тепла q_S протекает через ограничивающую поверхность площадью S :

$$q_S = Q_S dt, \quad (53)$$

где Q_S – суммарное количество тепла, протекающего в единицу времени через поверхность S , определяемое интегралом (7).

Иными словами, в нестационарном случае должно быть справедливо уравнение

$$q dT + Q_S dt = Q_V dt. \quad (54)$$

Учитывая, что в общем случае неоднородной среды суммарное количество тепла q , необходимого для изменения температуры объема V на один градус определяется выражением

$$q = \iiint_V \rho(x, y, z) C(x, y, z) dV, \quad (55)$$

где $\rho(x, y, z)$ – плотность вещества; $C(x, y, z)$ – удельная теплоемкость вещества, подставляя выражения (55), (7), (8) в уравнение (54) и применяя теорему Остроградского-Гаусса (10), получим

$$\left[\iiint_V \rho(x, y, z) C(x, y, z) dV \right] dT + \left[\iiint_V \operatorname{div} \mathbf{W} dV \right] dt = \left[\iiint_V f(x, y, z) dV \right] dt, \quad (56)$$

откуда, вынося подынтегральные выражения, разделив левую и правую части на dt и подставляя закон Фурье (6), можем записать нестационарное уравнение теплопроводности в векторной форме:

$$\rho(x, y, z) C(x, y, z) \frac{\partial T}{\partial t} - \operatorname{div}(k(x, y, z) \cdot \operatorname{grad}(T)) = f(x, y, z) . \quad (57)$$

В операторной форме уравнение (57) имеет вид

$$\rho(x, y, z) C(x, y, z) \frac{\partial T}{\partial t} - \nabla(k(x, y, z) \cdot \nabla T) = f(x, y, z) . \quad (58)$$

1.2.1. Уравнения непрерывности

Уравнения непрерывности для электронов и дырок представляют собой уравнения параболического типа, очень важные для моделирования процессов переноса заряда в полупроводниках [4]. Они могут быть получены на основе следующих рассуждений.

Рассмотрим задачу о нестационарном распределении концентрации электронов проводимости² в некотором объеме полупроводника V , ограниченном замкнутой поверхностью S трехмерного пространства $X = (x, y, z)$.

Согласно законам сохранения, изменение числа электронов N в рассматриваемом объеме за некоторый промежуток времени dt определяется с одной стороны соотношением скоростей генерации и рекомбинации электронов в данном объеме, а с другой – суммарным числом электронов J_s , прошедших за время dt через замкнутую поверхность S , ограничивающую объем V :

$$\frac{\partial N}{\partial t} = J_s + R_V , \quad (59)$$

где R_V – изменение числа электронов в объеме V за счет разности скоростей генерации и рекомбинации.

Учитывая, что

$$N = \iiint_V n(x, y, z, t) dV , \quad (60)$$

где $n(x, y, z, t)$ – концентрация электронов³,

$$J_s = \oint_S \frac{\mathbf{j}_n}{e} \mathbf{dS} , \quad (61)$$

где \mathbf{j}_n – вектор плотности тока электронов; e – заряд электрона; \mathbf{dS} – вектор, модуль которого численно равен площади dS соответствующего бесконечно малого элемента поверхности, а направление совпадает с направлением нормали к этому элементу,

$$R_V = \iiint_V R(x, y, z, t) dV , \quad (62)$$

² Электроны, энергия которых соответствует зоне проводимости полупроводника.

³ Число электронов, приходящихся на единицу объема.

где $R(x, y, z, t)$ – изменение числа электронов в единице объема за счет разности скоростей генерации и рекомбинации, из уравнения (59) получим

$$\frac{\partial}{\partial t} \iiint_V n(x, y, z, t) dV = \frac{1}{e} \oint_S \mathbf{j}_n \cdot \mathbf{dS} + \iiint_V R(x, y, z, t) dV . \quad (63)$$

Применяя к первому слагаемому в правой части уравнения (63) теорему Остроградского-Гаусса (10), получим

$$\frac{\partial}{\partial t} \iiint_V n(x, y, z, t) dV = \frac{1}{e} \iiint_V \operatorname{div} \mathbf{j}_n dV + \iiint_V R(x, y, z, t) dV ; \quad (64)$$

$$\frac{\partial n(x, y, z, t)}{\partial t} = \frac{1}{e} \operatorname{div} \mathbf{j}_n + R(x, y, z, t) . \quad (65)$$

Уравнение (65) называют уравнением непрерывности для электронов [2, 4]. Оно выражает тот факт, что изменение во времени концентрации электронов определяется соотношением мощностей источников и стоков плотности электронной составляющей тока \mathbf{j}_n и разностью скоростей генерации и рекомбинации подвижных носителей заряда $R(x, y, z, t)$.

При рассмотрении процессов переноса зарядов в полупроводниках, как правило, помимо электронной учитывают также и дырочную составляющую тока, решая уравнения непрерывности для электронов и дырок совместно.

Уравнение непрерывности для дырок может быть получено в результате рассуждений, аналогичных приведенным выше для электронов, и имеет вид

$$\frac{\partial p(x, y, z, t)}{\partial t} = -\frac{1}{e} \operatorname{div} \mathbf{j}_p + R(x, y, z, t) , \quad (66)$$

где $p(x, y, z, t)$ – концентрация дырок; \mathbf{j}_p – вектор плотности дырочной составляющей тока.

Знаки первых слагаемых в правых частях уравнений (65), (66) определяются выбором положительного направления нормали к замкнутой поверхности, ограничивающей рассматриваемый объем, а также условным положительным направлением векторов плотностей тока. Кратко поясним данное утверждение.

Предположим, что скорости генерации и рекомбинации уравновешены ($R(x, y, z, t) = 0$). Увеличение во времени концентрации электронов соответствует положительной левой части уравнения (65). Тогда суммарный поток электронов должен быть направлен внутрь рассматриваемого объема. Поскольку условным направлением тока считается направление тока положительных зарядов, то вектор плотности электронного тока в данном случае будет совпадать с направлением внешней нормали к поверхности, ограничивающей рассматриваемый объем. Принимая направление внешней нормали положительным, получаем положительный знак первого слагаемого правой части (65), совпадающий со знаком левой части.

Поскольку в уравнении (66) рассматриваются положительно заряженные дырки, вектор плотности тока в аналогичной ситуации будет направлен по внутренней нормали к поверхности, ограничивающей рассматриваемый объем,

то есть знак первого слагаемого правой части (66) будет противоположен знаку левой части.

Следует также отметить, что разность скоростей генерации и рекомбинации $R(x, y, z, t)$ в уравнениях (65) и (66) не выделяется соответствующими индексами, поскольку генерация и рекомбинация в полупроводниках осуществляется электронно-дырочными парами [5].

1.3. Гиперболические уравнения

Рассмотрим некоторые задачи матфизики, приводящие к решению гиперболических уравнений.

1.3.1. Волновое уравнение

Многие физические процессы связаны с возникновением колебаний в некоторой среде. Например, колебания струны, колебания мембраны, распространение звуковых колебаний и др. Они описываются волновым уравнением, относящимся к уравнениям гиперболического типа.

Рассмотрим в качестве примера незатухающие колебания вдоль координаты x физической величины $u(x, t)$ в некоторой среде, описываемые выражением

$$u(x, t) = A \sin(kx + s_0) \sin(\omega t + \varphi_0) , \quad (67)$$

где A – амплитуда колебаний; k – волновое число, определяющее период изменения физической величины $u(x, t)$ по координате; ω – круговая (циклическая) частота, определяющая период изменения физической величины $u(x, t)$ во времени t ; s_0 – фаза колебания в точке $x = 0$; φ_0 – начальная фаза колебания в момент времени $t = 0$.

Волновое число определяется выражением

$$k = \frac{2\pi}{\lambda} , \quad (68)$$

где λ – длина волны.

Круговая частота определяется выражением

$$\omega = \frac{2\pi}{T} , \quad (69)$$

где T – период колебаний.

Производная от $u(x, t)$ по координате x равна

$$\frac{\partial u}{\partial x} = Ak \cos(kx + s_0) \sin(\omega t + \varphi_0) . \quad (70)$$

Производная от $u(x, t)$ по времени t равна

$$\frac{\partial u}{\partial t} = A\omega \sin(kx + s_0) \cos(\omega t + \varphi_0) . \quad (71)$$

Вторая производная от $u(x, t)$ по координате x имеет вид

$$\frac{\partial^2 u}{\partial x^2} = -Ak^2 \sin(kx + s_0) \sin(\omega t + \varphi_0) . \quad (72)$$

Вторая производная от $u(x, t)$ по времени t имеет вид

$$\frac{\partial^2 u}{\partial t^2} = -A\omega^2 \sin(kx + s_0) \sin(\omega t + \varphi_0) . \quad (73)$$

Сравнивая выражения (72) и (73) с (67) легко увидеть, что

$$\frac{\partial^2 u}{\partial x^2} = -k^2 u(x, t) ; \quad (74)$$

$$\frac{\partial^2 u}{\partial t^2} = -\omega^2 u(x, t) . \quad (75)$$

Выражая $u(x, t)$ из (74), (75) и приравнявая правые части полученных выражений, имеем уравнение в частных производных

$$\frac{1}{\omega^2} \frac{\partial^2 u}{\partial t^2} = \frac{1}{k^2} \frac{\partial^2 u}{\partial x^2} , \quad (76)$$

называемое волновым [6].

В простейшем случае, при $\omega = 1$ и $k = 1$, получим

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} . \quad (77)$$

Обобщая (77) для случая трех координат, можем записать волновое уравнение в операторной форме

$$\frac{\partial^2 u}{\partial t^2} = \Delta u . \quad (78)$$

В случае затухающих колебаний, когда амплитуда является функцией от координат и времени $A = A(x, t)$, коэффициенты в уравнении (76) будут иметь более сложный вид.

1.4. Системы дифференциальных уравнений в частных производных

Некоторые задачи математической физики описываются системами дифференциальных уравнений в частных производных. Так, например, процессы переноса и накопления зарядов в полупроводниковых приборах при определенных условиях описываются так называемой фундаментальной системой уравнений (ФСУ) в диффузионно-дрейфовом приближении, в которую входят уравнения непрерывности для электронов и дырок (65), (66), уравнения плотностей электронной и дырочной составляющих электрического тока и уравнение Пуассона (46). В ряде случаев ФСУ решается совместно с уравнениями баланса энергии и импульса, уравнением теплопроводности и др. [2, 4].

1.4.1. Фундаментальная система уравнений

В векторной форме фундаментальная система уравнений в диффузионно-дрейфовом приближении может быть представлена следующим образом [2, 4]:

$$\frac{\partial n(x, y, z, t)}{\partial t} = \frac{1}{e} \operatorname{div} \mathbf{j}_n + R(x, y, z, t); \quad (79)$$

$$\frac{\partial p(x, y, z, t)}{\partial t} = -\frac{1}{e} \operatorname{div} \mathbf{j}_p + R(x, y, z, t); \quad (80)$$

$$\mathbf{j}_n = en\mu_n \mathbf{E} + eD_n \operatorname{grad}(n); \quad (81)$$

$$\mathbf{j}_p = ep\mu_p \mathbf{E} - eD_p \operatorname{grad}(p); \quad (82)$$

$$\operatorname{div}(\varepsilon(x, y, z) \cdot \operatorname{grad}(\varphi)) = -\frac{\rho(x, y, z, t)}{\varepsilon_0}, \quad (83)$$

где D_n , D_p – коэффициенты диффузии соответственно электронов и дырок; μ_n , μ_p – подвижности электронов и дырок.

В уравнениях (81), (82) первые слагаемые правых частей выражают дрейфовую составляющую плотности тока, определяемую напряженностью электрического поля \mathbf{E} , а вторые – диффузионную составляющую, определяемую градиентами концентраций электронов и дырок.

Учитывая выражение (26) и соотношения Эйнштейна [5]

$$D_n = \mu_n \varphi_T = \mu_n \frac{kT}{e}; \quad (84)$$

$$D_p = \mu_p \varphi_T = \mu_p \frac{kT}{e}, \quad (85)$$

где φ_T – температурный потенциал; k – постоянная Больцмана; T – абсолютная температура, уравнения (81), (82) можно переписать в виде

$$\mathbf{j}_n = -e\mu_n \left(n \cdot \operatorname{grad}(\varphi) - \varphi_T \operatorname{grad}(n) \right); \quad (86)$$

$$\mathbf{j}_p = -e\mu_p \left(p \cdot \operatorname{grad}(\varphi) + \varphi_T \operatorname{grad}(p) \right). \quad (87)$$

Подставляя (86), (87) в уравнения непрерывности (79), (80), выражая объемную плотность зарядов ρ через концентрации подвижных носителей и ионизированных примесей и считая диэлектрическую проницаемость ε константой, получим ФСУ в векторной форме в виде

$$\frac{\partial n}{\partial t} = -\operatorname{div} \left[\mu_n \left(n \cdot \operatorname{grad}(\varphi) - \varphi_T \operatorname{grad}(n) \right) \right] + R; \quad (88)$$

$$\frac{\partial p}{\partial t} = -\operatorname{div} \left[\mu_p \left(p \cdot \operatorname{grad}(\varphi) + \varphi_T \operatorname{grad}(p) \right) \right] + R; \quad (89)$$

$$\operatorname{div}(\operatorname{grad}(\varphi)) = -\frac{e}{\varepsilon\varepsilon_0} \left(p - n + N_D - N_A \right), \quad (90)$$

где N_D, N_A – концентрации ионов доноров и акцепторов, соответственно.

В операторной форме система уравнений (88) – (90) будет иметь вид

$$\frac{\partial n}{\partial t} = -\nabla \left[\mu_n \left(n \cdot \nabla \varphi - \varphi_T \nabla n \right) \right] + R; \quad (91)$$

$$\frac{\partial p}{\partial t} = -\nabla \left[\mu_p \left(p \cdot \nabla \varphi + \varphi_T \nabla p \right) \right] + R; \quad (92)$$

$$\Delta \varphi = -\frac{e}{\varepsilon\varepsilon_0} \left(p - n + N_D - N_A \right), \quad (93)$$

а в частных производных для случая трех координат

$$\begin{aligned} \frac{\partial n}{\partial t} = & -\frac{\partial}{\partial x} \left[\mu_n \left(n \cdot \frac{\partial \varphi}{\partial x} - \varphi_T \frac{\partial n}{\partial x} \right) \right] - \\ & -\frac{\partial}{\partial y} \left[\mu_n \left(n \cdot \frac{\partial \varphi}{\partial y} - \varphi_T \frac{\partial n}{\partial y} \right) \right] - \frac{\partial}{\partial z} \left[\mu_n \left(n \cdot \frac{\partial \varphi}{\partial z} - \varphi_T \frac{\partial n}{\partial z} \right) \right] + R; \end{aligned} \quad (94)$$

$$\begin{aligned} \frac{\partial p}{\partial t} = & -\frac{\partial}{\partial x} \left[\mu_p \left(p \cdot \frac{\partial \varphi}{\partial x} + \varphi_T \frac{\partial p}{\partial x} \right) \right] - \\ & -\frac{\partial}{\partial y} \left[\mu_p \left(p \cdot \frac{\partial \varphi}{\partial y} + \varphi_T \frac{\partial p}{\partial y} \right) \right] - \frac{\partial}{\partial z} \left[\mu_p \left(p \cdot \frac{\partial \varphi}{\partial z} + \varphi_T \frac{\partial p}{\partial z} \right) \right] + R; \end{aligned} \quad (95)$$

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} = -\frac{e}{\varepsilon\varepsilon_0} \left(p - n + N_D - N_A \right). \quad (96)$$

1.4.2. Нормировка

Решение фундаментальной системы уравнений непосредственно в виде (94) – (96) весьма затруднительно, во-первых, из-за значительной разницы в диапазонах переменных (электрический потенциал изменяется, как правило, в пределах единиц Вольт, а концентрации электронов и дырок – в пределах $10^8 - 10^{21} \text{ см}^{-3}$), а во-вторых, из-за наличия постоянных коэффициентов. Для устранения этих проблем, а также с целью приведения всех физических величин, входящих в систему к безразмерному виду, все переменные и параметры ФСУ нормируются на коэффициенты, основные из которых приведены в табл. 1 [4].

Нормировочные коэффициенты ФСУ

Нормируемая физическая величина	Нормировочный коэффициент
Время	τ_0
Потенциал	φ_T
Концентрация	n_i – собственная концентрация подвижных носителей ⁴
Длина	$L_0 = \sqrt{\frac{\varepsilon \varepsilon_0 \varphi_T}{e n_i}}$
Коэффициент диффузии	$D_0 = \frac{L_0^2}{\tau_0}$
Подвижность	$\mu_0 = \frac{D_0}{\varphi_T}$

Разделив все переменные и параметры в уравнениях (94) – (96) на соответствующие нормировочные коэффициенты (см. табл. 1) и сократив полученные выражения, имеем ФСУ в частных производных в нормированном виде:

$$\frac{\partial n}{\partial t} = -\frac{\partial}{\partial x} \left[\mu_n \left(n \cdot \frac{\partial \varphi}{\partial x} - \frac{\partial n}{\partial x} \right) \right] - \frac{\partial}{\partial y} \left[\mu_n \left(n \cdot \frac{\partial \varphi}{\partial y} - \frac{\partial n}{\partial y} \right) \right] - \frac{\partial}{\partial z} \left[\mu_n \left(n \cdot \frac{\partial \varphi}{\partial z} - \frac{\partial n}{\partial z} \right) \right] + R ; \quad (97)$$

$$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial x} \left[\mu_p \left(p \cdot \frac{\partial \varphi}{\partial x} + \frac{\partial p}{\partial x} \right) \right] - \frac{\partial}{\partial y} \left[\mu_p \left(p \cdot \frac{\partial \varphi}{\partial y} + \frac{\partial p}{\partial y} \right) \right] - \frac{\partial}{\partial z} \left[\mu_p \left(p \cdot \frac{\partial \varphi}{\partial z} + \frac{\partial p}{\partial z} \right) \right] + R ; \quad (98)$$

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} = n - p - N_D + N_A . \quad (99)$$

⁴ Концентрация электронов проводимости или дырок в беспримесном (собственном) полупроводнике.

В результате нормировки все переменные системы (97) – (99) безразмерные и имеют сравнимые по порядку величины диапазоны значений. Все постоянные коэффициенты ФСУ сокращаются.

1.4.3. Базисы переменных

Искомые функциями при решении системы (97) – (99) являются зависимости от координат и времени потенциала $\varphi = \varphi(x, y, z, t)$ и концентраций электронов $n = n(x, y, z, t)$ и дырок $p = p(x, y, z, t)$ для определенных моделей подвижностей $\mu_n = \mu_n(x, y, z, t)$, $\mu_p = \mu_p(x, y, z, t)$ и генерации-рекомбинации $R = R(x, y, z, t)$ при известных распределениях по координатам концентраций доноров $N_D = N_D(x, y, z)$ и акцепторов $N_A = N_A(x, y, z)$.

Множество искомых функций системы уравнений принято называть базисом переменных. В данном случае система (97) – (99) записана для базиса переменных $\{n, p, \varphi\}$ [2, 4].

Важно отметить, что при необходимости система уравнений может быть приведена к другому базису переменных. В случае ФСУ такая необходимость, как правило, связана с резким уменьшением скорости сходимости или, иными словами, с увеличением числа итераций в процессе численного решения, например, при моделировании полупроводниковых приборов в режимах, характеризующихся высокими плотностями токов.

При решении фундаментальной системы уравнений широко используют три базиса переменных:

- $\{n, p, \varphi\}$ – концентрации электронов и дырок, электрический потенциал;
- $\{\varphi_n, \varphi_p, \varphi\}$ – квазиуровни Ферми для электронов и дырок, электрический потенциал;
- $\{\Phi_n, \Phi_p, \varphi\}$ – экспоненты квазиуровней Ферми для электронов и дырок, электрический потенциал.

Переход к иному базису переменных осуществляется с использованием выражений, связывающих переменные различных базисов [4]:

$$\varphi_n = \ln(n) - \varphi ; \quad (100)$$

$$\varphi_p = \ln(p) + \varphi ; \quad (101)$$

$$\Phi_n = \exp(\varphi_n) ; \quad (102)$$

$$\Phi_p = \exp(\varphi_p) ^5. \quad (103)$$

Подставляя выражения (100), (101) в систему (97) – (99), получим ФСУ в частных производных в базисе $\{\varphi_n, \varphi_p, \varphi\}$:

⁵ Выражения (100) – (103) приведены в нормированном виде.

$$\begin{aligned} \frac{\partial}{\partial t} \left[\exp(\varphi_n + \varphi) \right] &= \frac{\partial}{\partial x} \left[\mu_n \exp(\varphi_n + \varphi) \frac{\partial \varphi_n}{\partial x} \right] + \\ &+ \frac{\partial}{\partial y} \left[\mu_n \exp(\varphi_n + \varphi) \frac{\partial \varphi_n}{\partial y} \right] + \frac{\partial}{\partial z} \left[\mu_n \exp(\varphi_n + \varphi) \frac{\partial \varphi_n}{\partial z} \right] + R ; \end{aligned} \quad (104)$$

$$\begin{aligned} \frac{\partial}{\partial t} \left[\exp(\varphi_p - \varphi) \right] &= \frac{\partial}{\partial x} \left[\mu_p \exp(\varphi_p - \varphi) \frac{\partial \varphi_p}{\partial x} \right] + \\ &+ \frac{\partial}{\partial y} \left[\mu_p \exp(\varphi_p - \varphi) \frac{\partial \varphi_p}{\partial y} \right] + \frac{\partial}{\partial z} \left[\mu_p \exp(\varphi_p - \varphi) \frac{\partial \varphi_p}{\partial z} \right] + R ; \end{aligned} \quad (105)$$

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} = \exp(\varphi_n + \varphi) - \exp(\varphi_p - \varphi) + N_A - N_D. \quad (106)$$

В операторной форме система (104) – (106) будет иметь вид:

$$\frac{\partial}{\partial t} \left[\exp(\varphi_n + \varphi) \right] = \nabla \left[\mu_n \exp(\varphi_n + \varphi) \nabla \varphi_n \right] + R ; \quad (107)$$

$$\frac{\partial}{\partial t} \left[\exp(\varphi_p - \varphi) \right] = \nabla \left[\mu_p \exp(\varphi_p - \varphi) \nabla \varphi_p \right] + R ; \quad (108)$$

$$\Delta \varphi = \exp(\varphi_n + \varphi) - \exp(\varphi_p - \varphi) + N_A - N_D. \quad (109)$$

Подставляя выражения (102), (103) в систему (104) – (106), получим ФСУ в частных производных в базисе $\{\Phi_n, \Phi_p, \varphi\}$:

$$\begin{aligned} \frac{\partial}{\partial t} \left[\Phi_n e^\varphi \right] &= \frac{\partial}{\partial x} \left[\mu_n e^\varphi \frac{\partial \Phi_n}{\partial x} \right] + \\ &+ \frac{\partial}{\partial y} \left[\mu_n e^\varphi \frac{\partial \Phi_n}{\partial y} \right] + \frac{\partial}{\partial z} \left[\mu_n e^\varphi \frac{\partial \Phi_n}{\partial z} \right] + R ; \end{aligned} \quad (110)$$

$$\begin{aligned} \frac{\partial}{\partial t} [\Phi_p e^{-\varphi}] = \frac{\partial}{\partial x} \left[\mu_p e^{-\varphi} \frac{\partial \Phi_p}{\partial x} \right] + \\ + \frac{\partial}{\partial y} \left[\mu_p e^{-\varphi} \frac{\partial \Phi_p}{\partial y} \right] + \frac{\partial}{\partial z} \left[\mu_p e^{-\varphi} \frac{\partial \Phi_p}{\partial z} \right] + R ; \end{aligned} \quad (111)$$

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} = \Phi_n e^{\varphi} - \Phi_p e^{-\varphi} + N_A - N_D. \quad (112)$$

В операторной форме система (110) – (112) будет иметь вид:

$$\frac{\partial}{\partial t} [\Phi_n e^{\varphi}] = \nabla \left[\mu_n e^{\varphi} \nabla \Phi_n \right] + R ; \quad (113)$$

$$\frac{\partial}{\partial t} [\Phi_p e^{-\varphi}] = \nabla \left[\mu_p e^{-\varphi} \nabla \Phi_p \right] + R ; \quad (114)$$

$$\Delta \varphi = \Phi_n e^{\varphi} - \Phi_p e^{-\varphi} + N_A - N_D. \quad (115)$$

2. ГРАНИЧНЫЕ И НАЧАЛЬНЫЕ УСЛОВИЯ

Из курса высшей математики известно, что дифференциальные уравнения, как правило, имеют бесконечное множество решений. Это связано с появлением в процессе интегрирования констант, при любых значениях которых решение удовлетворяет исходному уравнению [7].

Решение задач матфизики связано с нахождением зависимостей от координат и времени определенных физических величин, которые, безусловно, должны удовлетворять требованиям однозначности⁶, конечности⁷ и непрерывности [7]. Иными словами, любая задача матфизики предполагает поиск единственного решения (если оно вообще существует). Поэтому математическая формулировка физической задачи должна помимо основных уравнений (дифференциальных уравнений в частных производных), описывающих искомые функции внутри рассматриваемой области, включать дополнительные уравнения (дифференциальные или алгебраические), описывающие искомые функции на границах рассматриваемой области в любой момент времени и во всех внутренних точках области в начальный момент времени. Эти дополнительные уравнения называют соответственно граничными и начальными условиями задачи.

⁶ Физическая величина не может иметь двух или более различных значений в данной точке пространства и в данный момент времени.

⁷ Физическая величина не может иметь бесконечных по модулю значений.

2.1. Граничные условия

Предположим, необходимо решить определенную задачу, описываемую уравнениями матфизики, для некоторой области Θ . Тогда для нахождения единственного решения необходимо задать граничные условия (ГУ), то есть выразить искомые переменные на границе Ω области Θ некоторыми уравнениями.

Если область Θ представляет собой некоторый объем в трехмерном пространстве, то граница Ω будет представлять собой замкнутую поверхность в этом пространстве, ограничивающую заданный объем. Если область Θ представляет собой некоторую поверхность в двухмерном пространстве, то граница Ω будет представлять собой замкнутый контур в этом пространстве, ограничивающий заданную поверхность. И, наконец, если область Θ представляет собой некоторый отрезок в одномерном пространстве, то граница Ω будет представлять собой две точки на границах заданного отрезка.

По виду уравнений, задающих ГУ, различают граничные условия первого рода (условия Дирихле), второго рода (условия Неймана) и третьего рода [1].

Граничные условия первого рода или краевая задача Дирихле имеют вид

$$u(\mathbf{x}, t) = g(\mathbf{x}, t) \text{ при } \mathbf{x} \in \Omega, \quad t \geq 0, \quad (116)$$

где $u(\mathbf{x}, t)$ – искомая функция; $g(\mathbf{x}, t)$ – некоторая заданная на границе Ω функция; \mathbf{x} – координаты граничной точки в пространстве (например, для трехмерного пространства $\mathbf{x} = (x, y, z)$), t – время.

Если иметь ввиду задачу теплопроводности, то ГУ первого рода задают температуру на границе Ω . В задаче о распределении электростатического поля в непроводящей среде ГУ первого рода задают электрический потенциал на границе Ω и т.д.

Граничные условия второго рода или краевая задача Неймана имеют вид

$$\frac{\partial u(\mathbf{x}, t)}{\partial n} = g(\mathbf{x}, t) \text{ при } \mathbf{x} \in \Omega, \quad t \geq 0, \quad (117)$$

где n – внутренняя нормаль к границе Ω .

Иными словами, условия Неймана задают поток на границе, точнее, проекцию вектора потока на внутреннюю нормаль к границе. Например, в задачах теплопроводности ГУ второго рода задают тепловой поток, в задаче о распределении электростатического поля в непроводящей среде – проекцию вектора напряженности электрического поля на нормаль к границе и т.д.

Граничные условия третьего рода являются более общим случаем краевых задач Дирихле и Неймана и имеют вид

$$h(\mathbf{x}, t) \frac{\partial u(\mathbf{x}, t)}{\partial n} + r(\mathbf{x}, t) u(\mathbf{x}, t) = g(\mathbf{x}, t) \text{ при } \mathbf{x} \in \Omega, \quad t \geq 0, \quad (118)$$

где $h(\mathbf{x}, t)$, $r(\mathbf{x}, t)$ – некоторые функции координат и времени.

Например, в тепловых задачах ГУ третьего рода используют для задания на границе конвективного и излучательного теплообмена.

В соответствии с законом Ньютона, плотность теплового потока, отводимого в газовую или жидкую среду (или подводимого из нее) посредством конвекции с поверхности твердого тела в единицу времени, определяется выражением

$$W = -\alpha (T - T_0), \quad (119)$$

где α – коэффициент конвективного теплообмена; T – температура поверхности твердого тела; T_0 – температура окружающей среды [8].

Применяя для плотности теплового потока вдоль нормали к границе закон Фурье (6) и приравнивая правые части уравнений (119) и (6), получим

$$-k \frac{\partial T}{\partial n} = -\alpha (T - T_0); \quad (120)$$

$$k \frac{\partial T}{\partial n} - \alpha T = -\alpha T_0, \quad (121)$$

то есть граничные условия третьего рода (см. выражение (118)).

В соответствии с законом Стефана-Больцмана, плотность теплового потока, отводимого посредством излучения с поверхности твердого тела в единицу времени, определяется выражением

$$W = -\varepsilon(T) \cdot c_0 (T^4 - T_0^4), \quad (122)$$

где c_0 – коэффициент лучеиспускания абсолютно черного тела⁸; ε – относительная излучательная способность или степень черноты тела⁹ [8].

Применяя для плотности теплового потока вдоль нормали к границе закон Фурье (6) и приравнивая правые части уравнений (122) и (6), получим условия третьего рода в виде

$$-k \frac{\partial T}{\partial n} = -\varepsilon(T) \cdot c_0 (T^4 - T_0^4); \quad (123)$$

$$\frac{k}{\varepsilon(T)} \frac{\partial T}{\partial n} - c_0 T^4 = -c_0 T_0^4. \quad (124)$$

В уравнениях (119) – (124) температура является функцией координат $T = T(x, y, z)$ для точек, принадлежащих поверхности тела Ω .

Следует отметить, что количество граничных условий для каждой переменной определяется максимальным порядком производных по координатам в дифференциальных уравнениях [6]: для уравнений первого порядка – одно ГУ,

⁸ Абсолютно черным называют тело, от нагретой поверхности которого не происходит отражения света.

⁹ Отношение излучательных способностей рассматриваемого тела и абсолютно черного тела.

для уравнений второго порядка – два, для уравнений третьего порядка – три ГУ и т.д.

2.2. Начальные условия

Для нахождения единственного решения в задачах, описывающих нестационарные, т.е. изменяющиеся во времени физические процессы, помимо граничных необходимо задавать еще и начальные условия, определяющие значения переменных или их градиентов во всех внутренних точках рассматриваемой области Θ , исключая границу Ω ($\Theta \setminus \Omega$), в начальный момент времени:

$$u(\mathbf{x}, 0) = \xi(\mathbf{x}) \quad \text{при } \mathbf{x} \in \Theta \setminus \Omega ; \quad (125)$$

$$\frac{\partial u(\mathbf{x}, 0)}{\partial t} = \xi(\mathbf{x}) \quad \text{при } \mathbf{x} \in \Theta \setminus \Omega ; \quad (126)$$

$$\delta(\mathbf{x}) \frac{\partial u(\mathbf{x}, 0)}{\partial t} + \sigma(\mathbf{x}) u(\mathbf{x}, 0) = \xi(\mathbf{x}) \quad \text{при } \mathbf{x} \in \Theta \setminus \Omega , \quad (127)$$

где $u(\mathbf{x}, 0)$ – искомая функция в начальный момент времени; $\xi(\mathbf{x})$, $\delta(\mathbf{x})$, $\sigma(\mathbf{x})$ – некоторые функции координат.

Аналогично граничным условиям, количество начальных условий для каждой переменной определяется максимальным порядком производной по времени в дифференциальных уравнениях [6].

3. МЕТОДЫ ДИСКРЕТИЗАЦИИ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ

К сожалению, аналитическое решение уравнений математической физики возможно лишь для весьма ограниченного круга задач. В большинстве случаев решение дифференциальных уравнений в частных производных возможно только с использованием численных итерационных методов [1 – 7].

Суть данных методов состоит в дискретизации дифференциальных уравнений, то есть представлении всех или части производных в виде приближенных выражений (конечных разностей или конечных элементов), что позволяет преобразовать дифференциальные уравнения в системы алгебраических уравнений. Для этого рассматриваемая область Θ покрывается координатной сеткой, а все переменные заменяются сеточными функциями. Иными словами, значения переменных исследуются не для всего бесконечного множества точек области Θ , а лишь для некоторого конечного подмножества G . Причем при решении нестационарных задач помимо координатной сетки вводится сетка времени.

Число алгебраических уравнений в полученной системе (размерность дискретной задачи) определяется произведением числа точек координатной сетки на количество независимых переменных в исходных дифференциальных уравнениях.

Выбор метода решения полученной системы алгебраических уравнений определяется ее размерностью и характером (линейный или нелинейный). Для решения систем линейных алгебраических уравнений (СЛАУ) широко используют метод исключения Гаусса, метод LU-разложения и др. [3]. Для решения систем нелинейных алгебраических уравнений и линейных систем больших размерностей используют итерационные методы Якоби, Зейделя, Ньютона-Рафсона и др. [1, 3].

3.1. Метод конечных разностей

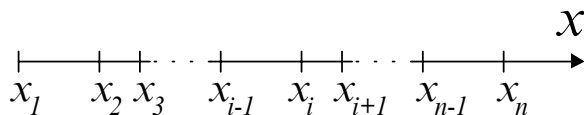
Метод конечных разностей предполагает дискретизацию дифференциальных уравнений на так называемых прямоугольных координатных сетках, то есть на сетках, элементарные ячейки которых представляют собой прямоугольники для двух измерений или параллелепипеды для трех измерений.

3.1.1. Конечно-разностные сетки

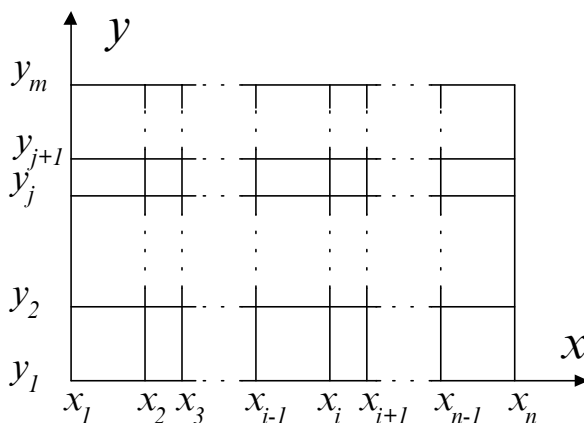
Рассмотрим одномерную область Θ , представляющую собой отрезок $[0, s]$. Разобьем этот отрезок точками $x_i = ih$, $i = 0, 1, 2, \dots, n$ на n равных частей длины $h = s/n$ каждая. Множество точек $G = \{x_i = ih \mid i = 0, 1, 2, \dots, n\}$ называется равномерной одномерной координатной сеткой, а число h – шагом сетки [1].

Отрезок $[0, s]$ можно разбить на n частей, вводя произвольные точки $0 < x_1 < x_2 < \dots < x_{i-1} < x_i < x_{i+1} < \dots < x_{n-1} < s$.

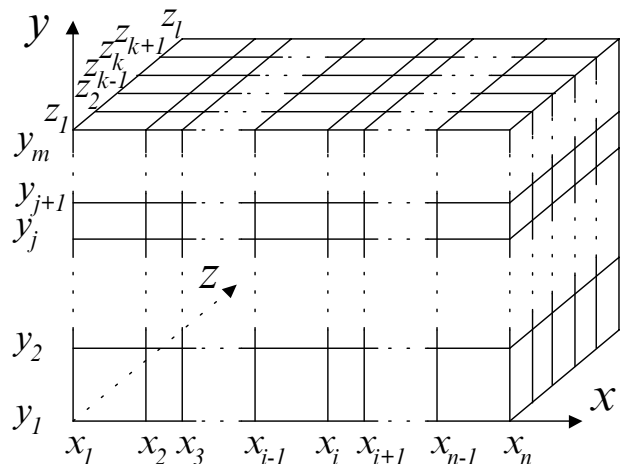
Координатная сетка $G = \{x_i \mid i = 0, 1, 2, \dots, n, x_0 = 0, x_n = s\}$ будет иметь шаг $h_i = x_i - x_{i-1}$, который зависит от номера i узла x_i . Если $h_i \neq h_{i+1}$ хотя бы для одного номера i , координатная сетка G называется неравномерной (рис. 1, а).



а



б



в

Рис. 1. Координатные сетки:

а – одномерная; б – двухмерная; в – трехмерная

Аналогично, двухмерной равномерной сеткой называют множество точек $G = \{(x_i = ih_1, y_j = jh_2) \mid i = 0, 1, 2, \dots, n, j = 0, 1, 2, \dots, m\}$. Неравномерная двухмерная сетка представляет собой множество $G = \{(x_i, y_j) \mid i = 0, 1, 2, \dots, n, j = 0, 1, 2, \dots, m, x_0 = 0, x_n = s_1, y_0 = 0, y_m = s_2\}$ (рис. 1, б).

Соответственно неравномерная трехмерная сетка представляет собой множество $G = \{(x_i, y_j, z_k) \mid i = 0, 1, 2, \dots, n, j = 0, 1, 2, \dots, m, k = 0, 1, 2, \dots, l, x_0 = 0, x_n = s_1, y_0 = 0, y_m = s_2, z_0 = 0, z_l = s_3\}$ (рис. 1, в).

3.1.2. Сеточные функции, конечные разности и шаблоны

Введение для области Θ координатной сетки G предполагает, что значения всех переменных и их производных рассматриваются только в узлах этой сетки (см. рис. 1). С целью выполнения данного требования все переменные задачи заменяются сеточными функциями, а производные любого порядка – конечными разностями [1, 2, 4, 6].

Пусть для некоторой области Θ задана сетка $G = \{(x_i, y_j, z_k) \mid i = 0, 1, 2, \dots, n, j = 0, 1, 2, \dots, m, k = 0, 1, 2, \dots, l, x_0 = 0, x_n = s_1, y_0 = 0, y_m = s_2, z_0 = 0, z_l = s_3\}$. Тогда функцию $\varphi = \varphi(x_i, y_j, z_k)$, $i = 0, 1, 2, \dots, n, j = 0, 1, 2, \dots, m, k = 0, 1, 2, \dots, l$ дискретного аргумента (x_i, y_j, z_k) называют сеточной функцией, определенной на сетке G [1].

Любой непрерывной функции $f(x, y, z)$, заданной в области Θ , можно поставить в соответствие сеточную функцию $\varphi(x_i, y_j, z_k)$ (для удобства обозначают $\varphi_{i,j,k}$), заданную на сетке $G = \{(x_i, y_j, z_k) \mid i = 0, 1, 2, \dots, n, j = 0, 1, 2, \dots, m, k = 0, 1, 2, \dots, l, x_0 = 0, x_n = s_1, y_0 = 0, y_m = s_2, z_0 = 0, z_l = s_3\}$ (спроектировать функцию $f(x, y, z)$ на сетку G), принимая определенное правило соответствия. Например,

$$\varphi_{i,j,k} = f(x_i, y_j, z_k); \quad (128)$$

$$\varphi_{i,j,k} = \frac{1}{(x_{i+1/2} - x_{i-1/2})(y_{j+1/2} - y_{j-1/2})(z_{k+1/2} - z_{k-1/2})} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{z_{k-1/2}}^{z_{k+1/2}} f(x, y, z) dx dy dz, \quad (129)$$

где $x_{i\pm 1/2}$, $y_{j\pm 1/2}$, $z_{k\pm 1/2}$ – координаты срединных точек на соответствующих шагах координатной сетки, определяемые выражениями

$$x_{i+1/2} = \frac{x_{i+1} + x_i}{2}; \quad (130)$$

$$x_{i-1/2} = \frac{x_i + x_{i-1}}{2}; \quad (131)$$

$$y_{j+1/2} = \frac{y_{j+1} + y_j}{2}; \quad (132)$$

$$y_{j-1/2} = \frac{y_j + y_{j-1}}{2}; \quad (133)$$

$$z_{k+1/2} = \frac{z_{k+1} + z_k}{2}; \quad (134)$$

$$z_{k-1/2} = \frac{z_k + z_{k-1}}{2}, \quad (135)$$

или др.

Следует иметь ввиду, что одна и та же сеточная функция, заданная на двух различных, но имеющих общие узлы сетках, не обязательно в этих общих узлах будет иметь одни и те же значения [1].

По определению производная функции непрерывного аргумента x в точке x_0 есть предел отношения приращения функции к приращению аргумента, когда последнее стремится к нулю [7]:

$$\frac{df(x)}{dx} = \lim_{(x-x_0) \rightarrow 0} \frac{f(x) - f(x_0)}{x - x_0}. \quad (136)$$

Исключая предел из выражения (136), производную функции непрерывного аргумента $f(x, y, z)$ можно приближенно заменить (аппроксимировать) разностным выражением, заданным на соответствующей сеточной функции $\varphi(x_i, y_j, z_k)$. Данная аппроксимация может быть осуществлена различными способами [1, 2, 4], например:

$$\frac{\partial f(x, y, z)}{\partial x} \approx \frac{\varphi_{i+1,j,k} - \varphi_{i,j,k}}{\Delta x_i}; \quad (137)$$

$$\frac{\partial f(x, y, z)}{\partial y} \approx \frac{\varphi_{i,j+1,k} - \varphi_{i,j,k}}{\Delta y_j}; \quad (138)$$

$$\frac{\partial f(x, y, z)}{\partial z} \approx \frac{\varphi_{i,j,k+1} - \varphi_{i,j,k}}{\Delta z_k}; \quad (139)$$

$$\frac{\partial f(x, y, z)}{\partial x} \approx \frac{\varphi_{i,j,k} - \varphi_{i-1,j,k}}{\Delta x_{i-1}}; \quad (140)$$

$$\frac{\partial f(x, y, z)}{\partial y} \approx \frac{\varphi_{i,j,k} - \varphi_{i,j-1,k}}{\Delta y_{j-1}}; \quad (141)$$

$$\frac{\partial f(x, y, z)}{\partial z} \approx \frac{\varphi_{i,j,k} - \varphi_{i,j,k-1}}{\Delta z_{k-1}}; \quad (142)$$

$$\frac{\partial f(x, y, z)}{\partial x} \approx \frac{\varphi_{i+1,j,k} - \varphi_{i-1,j,k}}{2\Delta x_{i-1}}; \quad (143)$$

$$\frac{\partial f(x,y,z)}{\partial y} \approx \frac{\varphi_{i,j+1,k} - \varphi_{i,j-1,k}}{2\Delta y_{j-1}}, \quad (144)$$

$$\frac{\partial f(x,y,z)}{\partial z} \approx \frac{\varphi_{i,j,k+1} - \varphi_{i,j,k-1}}{2\Delta z_{k-1}}, \quad (145)$$

где $\Delta x_i, \Delta y_j, \Delta z_k$ – конечные разности координат, определяемые выражениями

$$\Delta x_i = x_{i+1} - x_i; \quad (146)$$

$$\Delta y_j = y_{j+1} - y_j; \quad (147)$$

$$\Delta z_k = z_{k+1} - z_k. \quad (148)$$

Выражения (136) – (138) называют правыми разностями, (139) – (141) – левыми разностями, (142) – (144) – центральными разностями [1].

Каждому из преобразований свойственна определенная погрешность аппроксимации, стремящаяся к нулю при стремлении к нулю шага сетки.

Производные второго порядка аппроксимируются следующим образом:

$$\begin{aligned} \frac{\partial^2 f(x,y,z)}{\partial x^2} &\approx \frac{\frac{\varphi_{i+1,j,k} - \varphi_{i,j,k}}{\Delta x_i} - \frac{\varphi_{i,j,k} - \varphi_{i-1,j,k}}{\Delta x_{i-1}}}{\frac{\Delta x_i + \Delta x_{i-1}}{2}} = \\ &= \frac{2}{\Delta x_i + \Delta x_{i-1}} \left(\frac{\varphi_{i+1,j,k} - \varphi_{i,j,k}}{\Delta x_i} - \frac{\varphi_{i,j,k} - \varphi_{i-1,j,k}}{\Delta x_{i-1}} \right); \end{aligned} \quad (149)$$

$$\frac{\partial^2 f(x,y,z)}{\partial y^2} \approx \frac{2}{\Delta y_j + \Delta y_{j-1}} \left(\frac{\varphi_{i,j+1,k} - \varphi_{i,j,k}}{\Delta y_j} - \frac{\varphi_{i,j,k} - \varphi_{i,j-1,k}}{\Delta y_{j-1}} \right); \quad (150)$$

$$\frac{\partial^2 f(x,y,z)}{\partial z^2} \approx \frac{2}{\Delta z_k + \Delta z_{k-1}} \left(\frac{\varphi_{i,j,k+1} - \varphi_{i,j,k}}{\Delta z_k} - \frac{\varphi_{i,j,k} - \varphi_{i,j,k-1}}{\Delta z_{k-1}} \right). \quad (151)$$

В случае равномерной сетки, т.е. при $\Delta x_i = \Delta x_{i-1} = \Delta x$, $\Delta y_j = \Delta y_{j-1} = \Delta y$, $\Delta z_k = \Delta z_{k-1} = \Delta z$, выражения (148) – (150) примут более простой вид:

$$\frac{\partial^2 f(x,y,z)}{\partial x^2} \approx \frac{\varphi_{i+1,j,k} - 2\varphi_{i,j,k} + \varphi_{i-1,j,k}}{\Delta x^2}; \quad (152)$$

$$\frac{\partial^2 f(x,y,z)}{\partial y^2} \approx \frac{\varphi_{i,j+1,k} - 2\varphi_{i,j,k} + \varphi_{i,j-1,k}}{\Delta y^2}; \quad (153)$$

$$\frac{\partial^2 f(x,y,z)}{\partial z^2} \approx \frac{\varphi_{i,j,k+1} - 2\varphi_{i,j,k} + \varphi_{i,j,k-1}}{\Delta z^2}. \quad (154)$$

Аппроксимирующие выражения для смешанных производных могут быть получены следующим образом:

$$\frac{\partial^2 f(x, y, z)}{\partial x \partial y} \approx \frac{\frac{\varphi_{i+1, j+1, k} - \varphi_{i, j+1, k}}{\Delta x_i} - \frac{\varphi_{i+1, j, k} - \varphi_{i, j, k}}{\Delta x_i}}{\Delta y_j} = \frac{\varphi_{i+1, j+1, k} - \varphi_{i, j+1, k} - \varphi_{i+1, j, k} + \varphi_{i, j, k}}{\Delta x_i \Delta y_j}; \quad (155)$$

$$\frac{\partial^2 f(x, y, z)}{\partial y \partial z} \approx \frac{\frac{\varphi_{i, j+1, k+1} - \varphi_{i, j, k+1}}{\Delta y_j} - \frac{\varphi_{i, j+1, k} - \varphi_{i, j, k}}{\Delta y_j}}{\Delta z_k} = \frac{\varphi_{i, j+1, k+1} - \varphi_{i, j, k+1} - \varphi_{i, j+1, k} + \varphi_{i, j, k}}{\Delta y_j \Delta z_k}; \quad (156)$$

$$\frac{\partial^2 f(x, y, z)}{\partial x \partial z} \approx \frac{\frac{\varphi_{i+1, j, k+1} - \varphi_{i, j, k+1}}{\Delta x_i} - \frac{\varphi_{i+1, j, k} - \varphi_{i, j, k}}{\Delta x_i}}{\Delta z_k} = \frac{\varphi_{i+1, j, k+1} - \varphi_{i, j, k+1} - \varphi_{i+1, j, k} + \varphi_{i, j, k}}{\Delta x_i \Delta z_k}. \quad (157)$$

Аналогичным образом могут быть получены аппроксимирующие выражения для производных более высоких порядков [1].

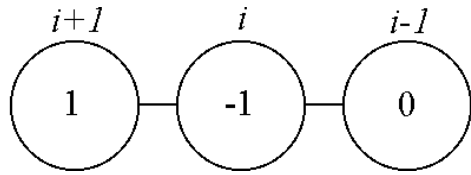
Для наглядного представления аппроксимаций (137) – (157) используют шаблоны [6].

Шаблон представляет собой граф, символически отображающий участок сетки, на котором производится аппроксимация функций и их производных. Вершины графа символизируют точки сетки с индексами (i, j, k) , $(i \pm 1, j \pm 1, k \pm 1)$, $(i \pm 2, j \pm 2, k \pm 2)$ и т.д. в зависимости от вида аппроксимирующих выражений и изображаются, как правило в виде кругов, внутри которых указываются коэффициенты слагаемых числителя выражения.

Например, для правой разности

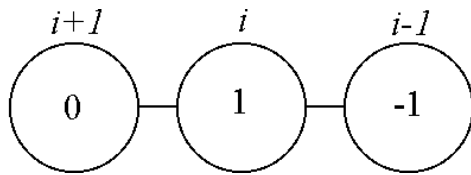
$$\frac{\partial f(x)}{\partial x} \approx \frac{\varphi_{i+1} - \varphi_i}{\Delta x_i} \quad (158)$$

вычислительный шаблон будет иметь вид

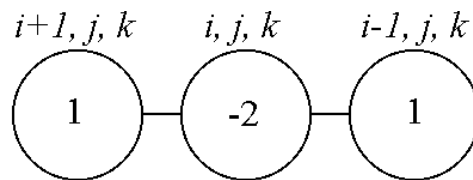


Для левой разности

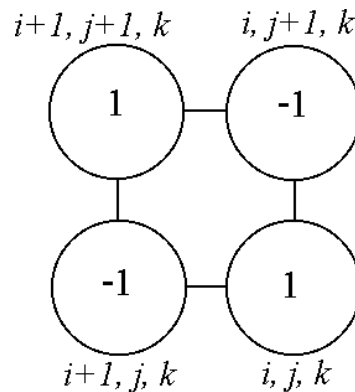
$$\frac{\partial f(x)}{\partial x} \approx \frac{\varphi_i - \varphi_{i-1}}{\Delta x_{i-1}} \quad (159)$$



Шаблон для выражения (152) будет иметь вид



Шаблон для аппроксимации (155) смешанной производной второго порядка



3.2. Метод конечных элементов

Метод конечных элементов предполагает дискретизацию дифференциальных уравнений на так называемых триангулярных координатных сетках, то есть на сетках, элементарные ячейки которых представляют собой треугольники для двух измерений или призмы (тетраэдры) для трех измерений.

3.2.1. Разбиение Дирихле и триангуляция Делоне

Пусть необходимо покрыть некоторую двумерную область Θ триангулярной координатной сеткой.

Рассмотрим некоторый набор точек $\{p_i \mid i = 1, 2, \dots, n\}$ в области Θ . Тогда i -я ячейка Дирихле Ω_i представляет собой множество всех точек области Θ , лежащих ближе к точке p_i , чем к любой другой точке p_j (рис. 2).

Такое множество ячеек $\{\Omega_i \mid i = 1, 2, \dots, n\}$ называется разбиением Дирихле [2]. Оно полностью покрывает область Θ без наложений (рис. 3).



Рис. 2. Ячейка Дирихле

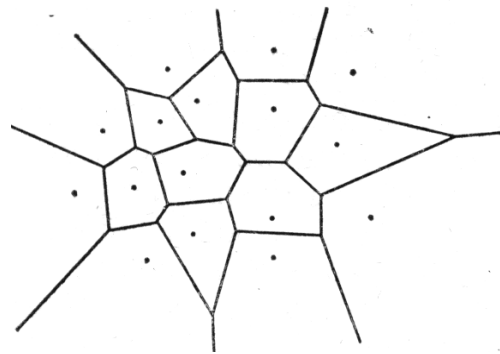
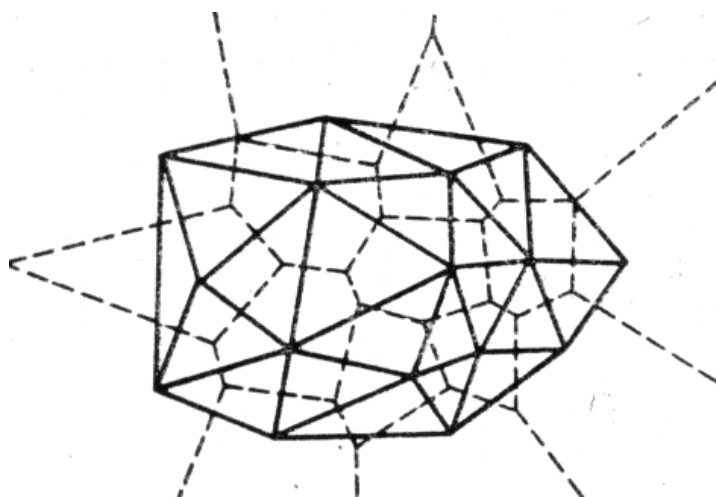


Рис. 3. Разбиение Дирихле

Координатная сетка, на которой удобно проводить дискретизацию уравнений математической физики, получается в результате триангуляции Делоне [2], которая определяется следующим образом.

Пусть две точки сетки называются соседними в смысле Дирихле тогда и только тогда, когда многоугольники Дирихле, содержащие эти точки, имеют общую грань ненулевой длины. Триангуляцией Делоне называется граф, образованный соединением соседних в смысле Дирихле точек отрезками прямых линий (рис. 4), удовлетворяющий следующим условиям [2]:

- 1) вершины любого элемента сетки (треугольника) лежат на некоторой окружности;
- 2) не существует точек сетки внутри окружности, описанной около любого элемента;
- 3) не существует двух элементов с общей описанной окружностью или, иными словами, не существует более 3 точек сетки, принадлежащих одной окружности.



*Рис. 4. Триангуляция Делоне (сплошные линии)
и соответствующее разбиение Дирихле (пунктирные линии)*

Из сопоставления приведенных определений видно, что соответствующие линии разбиения Дирихле и триангуляции Делоне взаимно перпендикулярны (см. рис. 4). Иными словами, построение перпендикуляров в серединах триангуляционных линий дает разбиение Дирихле [2]. Это и лежит в основе процедуры дискретизации дифференциальных уравнений, которая будет описана ниже.

Если какое-либо из приведенных выше условий триангуляции Делоне не выполняется, возникают нежелательные проблемы. Например, если четыре или более точек сетки лежат на одной окружности, то они могут быть триангулированы произвольным образом (рис. 5). В этом случае триангуляция Делоне будет неоднозначной.

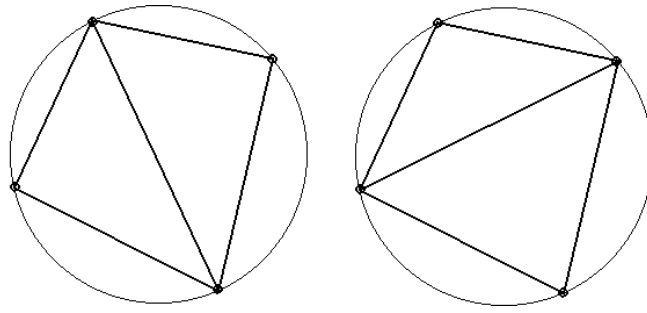


Рис. 5. Два варианта триангуляции четырех точек сетки, лежащих на одной окружности

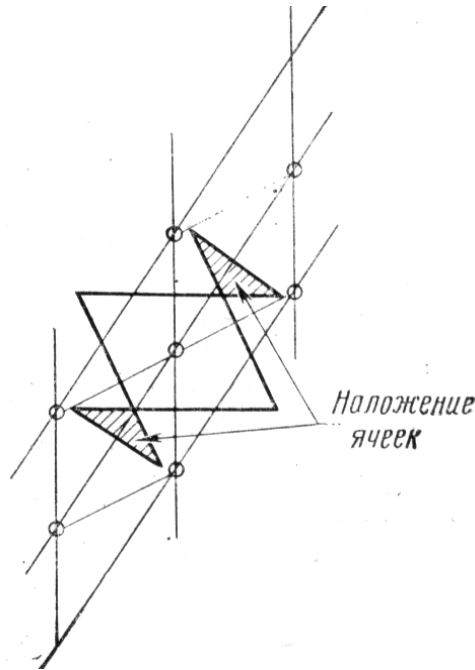


Рис. 6. Наложение ячеек Дирихле

Если при триангуляции образуются тупоугольные треугольники, это может приводить к наложению ячеек Дирихле (рис. 6), что вызовет локальное нарушение законов сохранения при дискретизации дифференциальных уравнений [2].

Разработано достаточно много различных алгоритмов триангуляции Делоне [2]. Все эти алгоритмы являются рекурсивными, то есть точки сетки при построении добавляются по одной. Иными словами, на i -м шаге добавляется i -я точка и получается триангуляция Делоне для первых i точек. При этом предполагается, что предварительно получена триангуляция Делоне для первых $i-1$ точек.

Как правило, используется метод искусственного ограничения [2], состоящий в том, что исходная область Θ (рис. 7), на которой предполагается проводить триангуляцию, покрывается искусственной областью Θ_1 , имеющую размеры, значительно превосходящие Θ . Это делается для того, чтобы каждая последующая точка сетки, включая точки, принадлежащие границе области Θ , являлась внутренней, то есть попадала внутрь области, составленной по предыду-

щим точкам. Триангуляция Делоне для такой области Θ_1 является исходной для рекурсивного алгоритма.

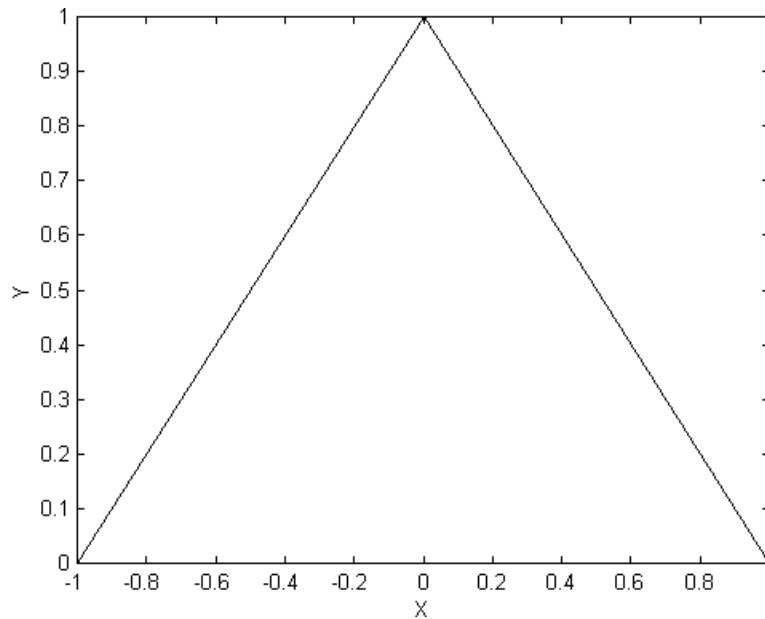


Рис. 7. Исходная область для триангуляции

На первом этапе к искусственным угловым четырем точкам границы области Θ_1 добавляются последовательно реальные угловые точки границы области Θ (рис. 8).

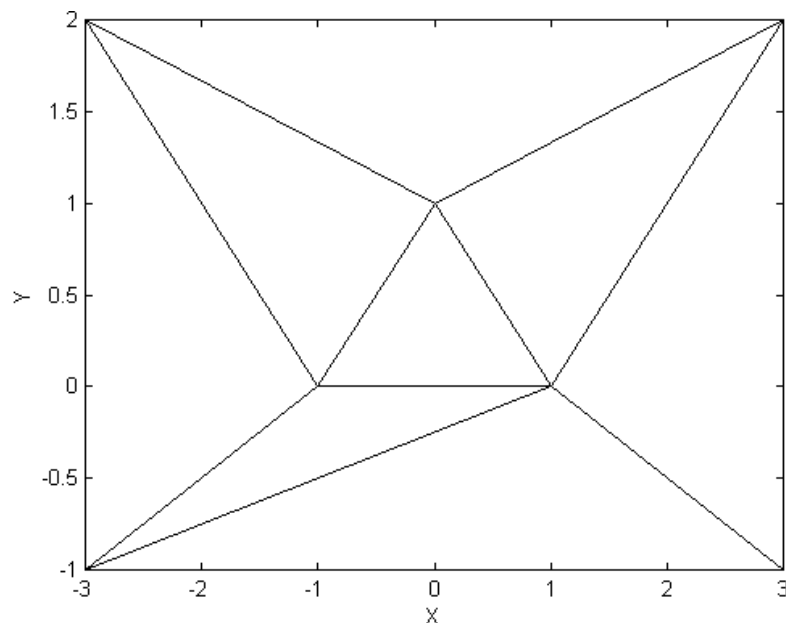


Рис. 8. Начальное разбиение искусственной области на элементы с добавлением угловых точек реальной границы

На втором этапе граничные отрезки области Θ разбиваются на части в соответствии с заданными требованиями к точности (рис. 9).

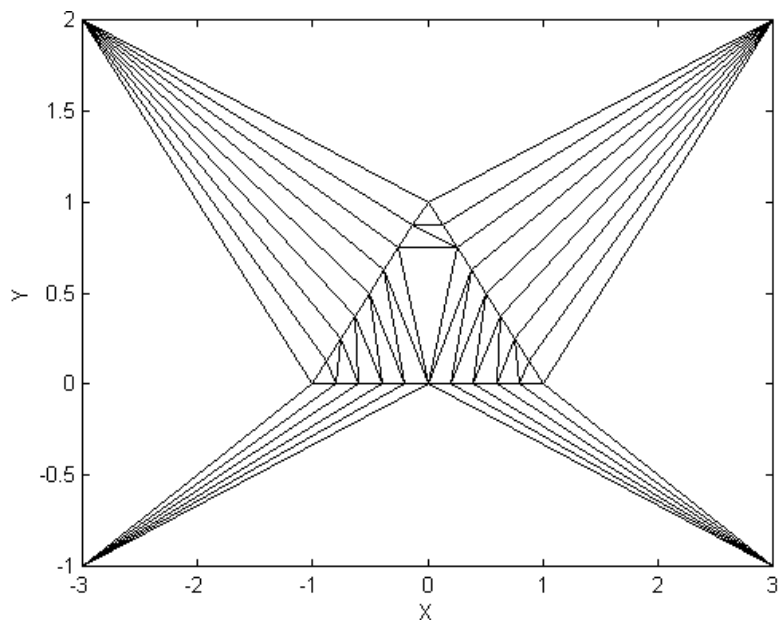


Рис. 9. Триангуляция с разбиением отрезков реальной границы на части

По мере последовательного добавления реальных точек сетки, исходная область Θ разобьется на части, как показано на рис. 10.

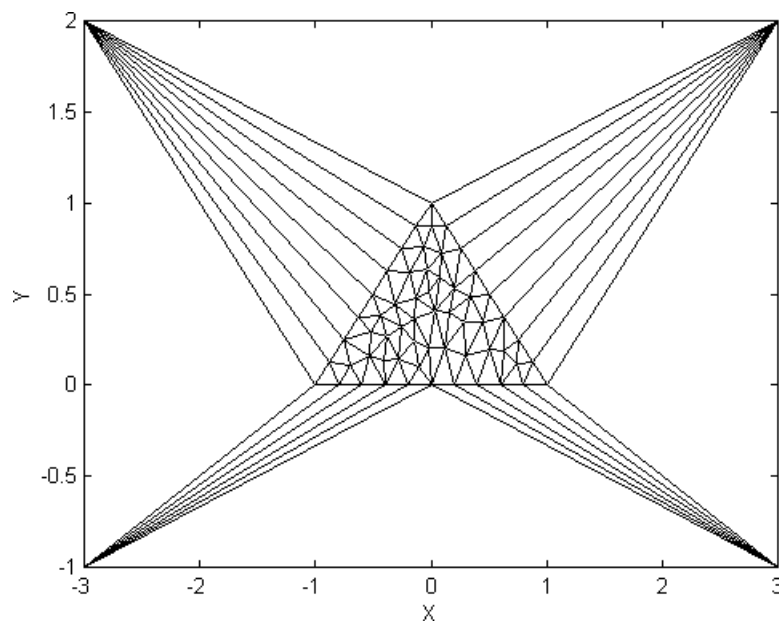


Рис. 10. Триангуляция реальной и искусственной областей

После завершения рекурсивного алгоритма искусственные точки и все связанные с ними отрезки удаляются и получившееся построение считается триангуляцией Делоне для заданного множества точек (рис. 11).

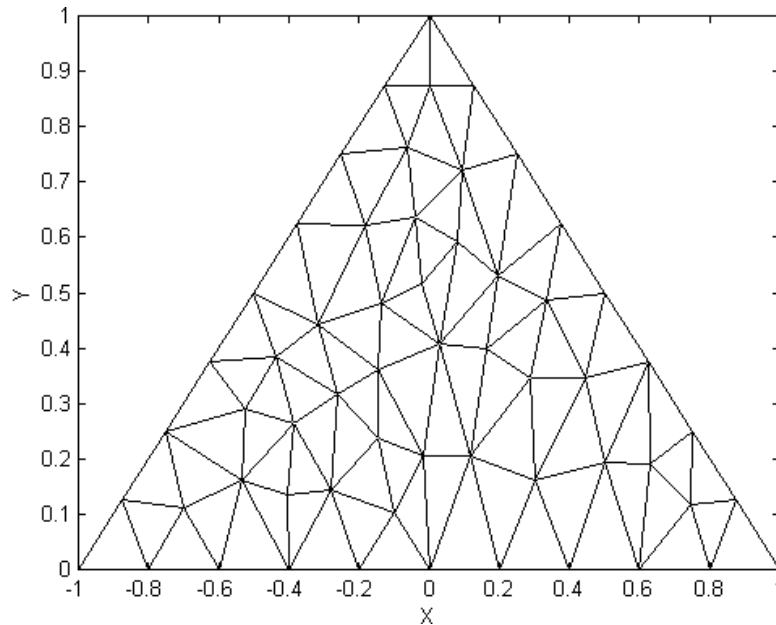


Рис. 11. Триангуляция Делоне для заданного множества точек

В целом рекурсивный алгоритм реализуется в три этапа [2]:

- ранее построенный элемент остается неизменным при добавлении новой точки, если последняя не попадает в описанную около него окружность;
- если новая точка попадает внутрь окружности, описанной около ранее построенного элемента, то новая точка соединяется со всеми его вершинами (при этом ранее построенный элемент рассекается на части);
- существующая между двумя точками линия сетки устраняется тогда и только тогда, когда новая точка попадает внутрь всех окружностей, описанных около любого ранее построенного элемента, границе которого принадлежит данная линия сетки.

На каждом этапе определяются элементы, подлежащие разбиению, то есть содержащие новую точку в описанных около них окружностях. Грань, общая для двух таких элементов, отбрасывается, а каждая из оставшихся граней и новая точка определяют новые элементы и линии сетки. Завершает процесс построение линий сетки по полученным элементам [2].

Триангуляцию Делоне можно получить из любой другой триангуляции по тому же множеству точек, последовательно перестраивая пары соседних треугольников ABC и BCD , не удовлетворяющих условиям Делоне, в пары треугольников ABD и ACD (рис. 12) [9].

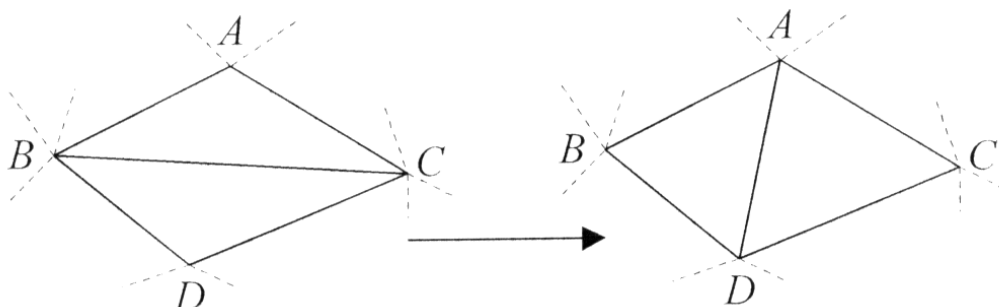


Рис. 12. Перестроение треугольников, не удовлетворяющих условиям Делоне

Такую операцию перестроения часто называют флипом. Она позволяет получать триангуляцию Делоне, построив на начальном этапе некоторую произвольную триангуляцию, а затем последовательно улучшая ее в смысле Делоне [9].

При проверке условий Делоне часто используют следующие теоремы [9].

Теорема 1. Триангуляция Делоне обладает максимальной суммой минимальных углов всех своих треугольников среди всех возможных триангуляций.

Теорема 2. Триангуляция Делоне обладает минимальной суммой радиусов окружностей, описанных около треугольников, среди всех возможных триангуляций.

3.2.2. Метод интегральных тождеств

Дискретизация дифференциальных уравнений в частных производных на триангулярных координатных сетках может быть проведена с использованием метода интегральных тождеств [2]. Суть данного метода состоит в следующем.

Для дискретизации уравнений математической физики прежде всего в соответствии с условиями задачи строится множество точек координатной сетки, проводится триангуляция Делоне и разбиение Дирихле. Дальнейшие шаги рассмотрим на примере задачи о распределении электростатического поля в области Θ объемом V , ограниченной замкнутой поверхностью Ω площадью S , непроводящей среды при наличии электрических зарядов, описываемой уравнениями (46) – (48).

Интегрируя обе части уравнения (46) по объему V , получим

$$\iiint_V \operatorname{div}(\varepsilon(x, y, z) \cdot \operatorname{grad}(\varphi)) dV + \frac{1}{\varepsilon_0} \iiint_V \rho(x, y, z) dV = 0, \quad (160)$$

или в операторной форме

$$\iiint_V \nabla(\varepsilon(x, y, z) \cdot \nabla \varphi) dV + \frac{1}{\varepsilon_0} \iiint_V \rho(x, y, z) dV = 0. \quad (161)$$

Применив теорему Остроградского-Гаусса (10), перепишем уравнение (161) в виде

$$\oint_S \varepsilon(x, y, z) \cdot \nabla \varphi dS + \frac{1}{\varepsilon_0} \iiint_V \rho(x, y, z) dV = 0. \quad (162)$$

Уравнение (162) представляет собой интегродифференциальную форму уравнения (48) и фактически выражает законы сохранения для области решения задачи.

Аппроксимируя уравнение (162) для некоторой внутренней ячейки Дирихле Θ_0 с внешней границей D , построенной вокруг точки p_0 сетки (рис. 13) в предположении, что все переменные в пределах данной ячейки Дирихле неизменны и триангулярная координатная сетка двумерна (все ячейки разбиения Дирихле представляют собой призмы равной высоты H , которая может быть вынесена за знаки интегралов), получим:

$$H \oint_D \varepsilon(x, y, z) \cdot \nabla \varphi dl + \frac{H}{\varepsilon_0} \iint_{S_0} \rho(x, y, z) dS_0 = 0, \quad (163)$$

где dl – элемент контура D ; dS_0 – элемент площади ячейки Дирихле, построенной вокруг точки p_0 (см. рис. 13);

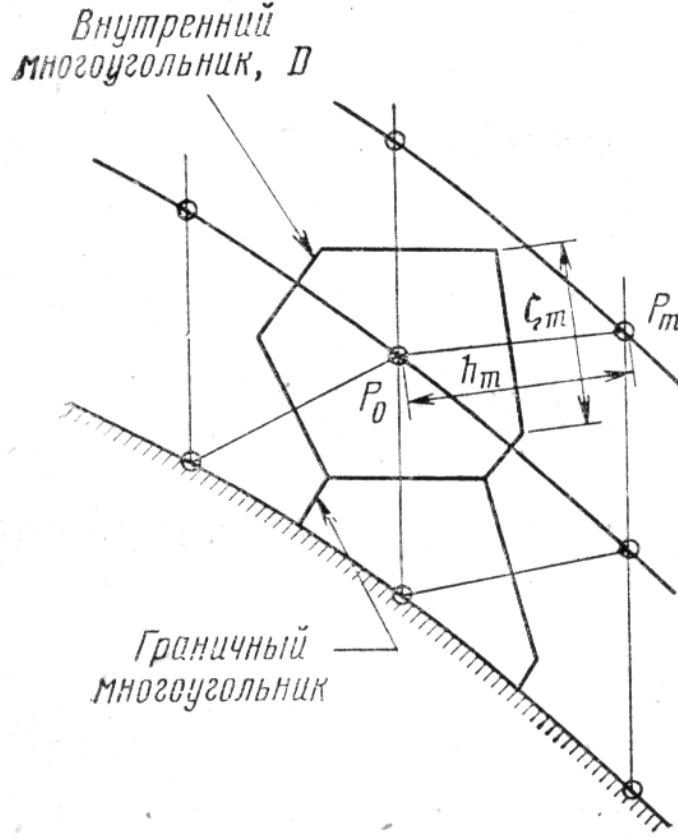


Рис. 13. Параметры m -ой грани ячейки Дирихле

$$\sum_{m=1}^{M_0} \zeta_m \frac{\varepsilon_{p_m} + \varepsilon_{p_0}}{2} \cdot \frac{\varphi_m - \varphi_0}{h_m} + \frac{1}{\varepsilon_0} \rho_0 S_0 = 0, \quad (164)$$

где M_0 – число граней ячейки Дирихле с контуром D ; S_0 – площадь ячейки Дирихле, построенной вокруг точки p_0 (см. рис. 13); φ_0, φ_m – значения электрического потенциала в точках p_0 и p_m , соответственно; $\varepsilon_{p_0}, \varepsilon_{p_m}$ – значения диэлектрической проницаемости среды в точках p_0 и p_m , соответственно; ρ_0 – значение объемной плотности электрических зарядов в точке p_0 ; h_m – расстояние между точками p_0 и p_m ; ζ_m – длина m -й грани ячейки Дирихле.

Обобщая уравнение (164) для всех внутренних ячеек Дирихле области Θ , можно записать

$$\sum_{m=1}^{M_i} \zeta_m \frac{\varepsilon_{p_m} + \varepsilon_{p_i}}{2} \cdot \frac{\varphi_m - \varphi_i}{h_m} + \frac{1}{\varepsilon_0} \rho_i S_i = 0, \quad i=1, 2, \dots, N_{\text{int}}, \quad (165)$$

где i – номер ячейки Дирихле; N_{int} – число внутренних ячеек Дирихле.

Аппроксимация уравнений для граничных ячеек Дирихле (см. рис. 13) производится аналогичным образом, но с учетом соответствующих граничных

условий Дирихле или Неймана. Например, если на границе заданы условия второго рода

$$\frac{\partial \varphi(x, y, z)}{\partial n} = g(x, y, z) \text{ при } (x, y, z) \in \Omega, \quad (166)$$

где n – нормаль к поверхности Ω ; $g(x, y, z)$ – некоторая функция координат, выражения для соответствующих граничных ячеек Дирихле могут быть записаны в виде

$$\sum_{m=1}^{M_{\Omega}} \zeta_m \varepsilon_{Pm} \cdot g_m + \sum_{m=M_{\Omega}+1}^{M_i} \zeta_m \frac{\varepsilon_{Pm} + \varepsilon_{Pi}}{2} \cdot \frac{\varphi_m - \varphi_i}{h_m} + \frac{1}{\varepsilon_0} \rho_i S_i = 0, \quad i=1, 2, \dots, N_{ext}, \quad (167)$$

где M_{Ω} – число граней ячейки Дирихле, совпадающих с поверхностью Ω ; N_{ext} – число граничных ячеек Дирихле, т.е. ячеек, у которых хотя бы одна грань совпадает с поверхностью Ω (см. рис. 13); g_m – значение функции $g(x, y, z)$ на m -й грани ячейки Дирихле.

Нетрудно видеть, что система алгебраических уравнений (165), (167), полученная в результате дискретизации уравнений (48), (166) на триангулярной координатной сетке с использованием метода интегральных тождеств, является линейной. Число уравнений в системе определяется общим числом точек сетки $N = N_{int} + N_{ext}$.

4. МЕТОДЫ РЕШЕНИЯ СИСТЕМ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

Целью дискретизации уравнений математической физики является преобразование дифференциальных уравнений в частных производных в систему алгебраических уравнений (линейных или нелинейных), позволяющую найти решение задачи в узлах прямоугольной (метод конечных разностей) или триангулярной (метод конечных элементов) координатной сетки.

4.1. Методы решения систем линейных алгебраических уравнений

Для решения систем линейных алгебраических уравнений (СЛАУ) широко используют метод исключения Гаусса, метод LU-разложения и др., позволяющими найти точное решение СЛАУ [3]. Использование данных методов предпочтительно, если размерность системы не слишком велика для имеющегося объема оперативной памяти компьютера, на котором предполагается решать поставленную задачу.

В противном случае используют итерационные методы решения, позволяющие оптимизировать процесс решения в зависимости от имеющихся вычислительных ресурсов компьютера. При этом получается приближенное решение. Максимальная точность ограничивается допустимым временем решения задачи и разрядной сеткой компьютера.

В общем случае система n линейных алгебраических уравнений от n переменных имеет вид

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n &= b_1; \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n &= b_2; \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \dots + a_{3n}x_n &= b_3; \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n &= b_n,
\end{aligned} \tag{168}$$

или в матричном представлении

$$\mathbf{A} \mathbf{x} = \mathbf{B}, \tag{169}$$

где $\mathbf{x} = [x_1, x_2, x_3, \dots, x_n]$ – вектор-столбец переменных; $\mathbf{A} = [a_{ij}]$, $i = 1, 2, \dots, n, j = 1, 2, \dots, n$ – матрица коэффициентов СЛАУ размера $n \times n$; $\mathbf{B} = [b_1, b_2, b_3, \dots, b_n]$ – вектор-столбец свободных членов.

Решение системы (169) может быть найдено умножением вектора-столбца свободных членов \mathbf{B} на матрицу \mathbf{A}^{-1} , обратную матрице коэффициентов [7]

$$\mathbf{x} = \mathbf{A}^{-1} \mathbf{B}. \tag{170}$$

4.1.1. Метод исключения Гаусса

Матрица коэффициентов \mathbf{A} называется верхней треугольной, если $a_{ij} = 0$ для всех $i > j$:

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n &= b_1; \\
a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n &= b_2; \\
a_{33}x_3 + \dots + a_{3n}x_n &= b_3; \\
&\vdots \\
a_{nn}x_n &= b_n.
\end{aligned} \tag{171}$$

В случае, если $a_{ij} = 0$ для всех $i < j$, матрица \mathbf{A} называется нижней треугольной:

$$\begin{aligned}
a_{11}x_1 &= b_1; \\
a_{21}x_1 + a_{22}x_2 &= b_2; \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3; \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n &= b_n.
\end{aligned} \tag{172}$$

Соответственно, система линейных алгебраических уравнений (171) с верхней треугольной матрицей коэффициентов называется верхней треугольной системой, а (172) – нижней треугольной системой [3].

Можно показать, что если $\mathbf{Ax} = \mathbf{B}$ – верхняя (или нижняя) треугольная система и не существует равных нулю коэффициентов главной диагонали матрицы \mathbf{A}

$$a_{ii} \neq 0, \quad i=1,2,\dots,n, \quad (173)$$

то система имеет единственное решение [3].

Это решение может быть найдено методом прямой подстановки в случае нижней треугольной системы или обратной подстановки в случае верхней треугольной системы. Например, для верхней треугольной системы (171) n -е уравнение имеет лишь одну переменную x_n , которую из него можно легко найти следующим образом:

$$x_n = \frac{b_n}{a_{nn}}. \quad (174)$$

Используя (174), из $(n-1)$ -го уравнения находим

$$x_{n-1} = \frac{b_{n-1} - a_{(n-1)n}x_n}{a_{(n-1)(n-1)}}. \quad (175)$$

Продолжая делать подобные подстановки далее, получим решение, которое в общем виде может быть записано следующим образом:

$$x_i = \frac{b_i - \sum_{j=i+1}^n a_{ij}x_j}{a_{ii}}, \quad i=n-1, n-2, \dots, 2, 1. \quad (176)$$

Аналогично метод прямой подстановки (начиная с 1-го уравнения) позволит найти решения для нижней треугольной системы.

Пусть требуется решить систему (168) методом исключения Гаусса [3]. Данный метод позволяет привести систему (168) к верхней треугольной системе (171) или нижней треугольной системе (172), после чего находится решение методом обратной подстановки.

Две системы линейных алгебраических уравнений эквивалентны, если они имеют одинаковое множество решений [3].

Метод исключения Гаусса предполагает выполнение трех операций, преобразующих исходную систему в эквивалентную:

- 1) перестановка – произвольное изменение порядка уравнений в системе;
- 2) масштабирование – умножение левой и правой части любого уравнения системы на не равную нулю константу;
- 3) замещение – произвольное уравнение системы можно заменить почленной суммой этого же уравнения и любого другого уравнения системы, умноженного на не равную нулю константу.

Перечисленные операции выполняются до тех пор, пока полученная система, эквивалентная исходной, не будет треугольной.

Проиллюстрируем это на следующем примере [3].

Пусть необходимо найти параболу

$$y = A + Bx + Cx^2, \quad (177)$$

проходящую через три точки (1, 1), (2, -1), (3, 1).

Последовательно подставив координаты точек в выражение (177), найдем для каждой точки уравнение, связывающее значения x и y .

В результате получим систему линейных уравнений, в которой искомыми переменными являются коэффициенты полинома (177):

$$\begin{aligned} A + B + C &= 1; \\ A + 2B + 4C &= -1; \\ A + 3B + 9C &= 1. \end{aligned} \quad (178)$$

Производим замещение третьего уравнения разностью третьего и первого уравнений, а второго уравнения – разностью второго и первого уравнений. В результате из второго и третьего уравнений исключается переменная A :

$$\begin{aligned} A + B + C &= 1; \\ B + 3C &= -2; \\ 2B + 8C &= 0. \end{aligned} \quad (179)$$

Производим замещение третьего уравнения системы (179) разностью третьего уравнения и второго уравнения, умноженного на 2. При этом из третьего уравнения исключается переменная B :

$$\begin{aligned} A + B + C &= 1; \\ B + 3C &= -2; \\ 2C &= 4. \end{aligned} \quad (180)$$

В результате получили верхнюю треугольную систему (180), из которой методом обратной подстановки находим решение: $C = 2$, $B = -8$, $A = 7$. Следовательно, искомая парабола имеет вид

$$y = 7 - 8x + 2x^2. \quad (181)$$

4.1.2. Метод LU-разложения

Пусть необходимо решить систему линейных алгебраических уравнений (168), записанную в матричном виде (169). Допустим, матрица коэффициентов A размера $n \times n$ невырожденная¹⁰. Тогда ее можно представить в виде произведения двух матриц

$$A = LU, \quad (182)$$

где L – нижняя треугольная матрица размера $n \times n$, у которой все элементы главной диагонали – единицы; U – верхняя треугольная матрица размера $n \times n$.

При этом система (169) будет представлена в виде

¹⁰ Матрица размера $n \times n$ называется невырожденной, если ее определитель отличен от нуля [3].

$$\mathbf{LUx} = \mathbf{B} . \quad (183)$$

Введем вектор-столбец \mathbf{Y} , определяемый выражением

$$\mathbf{Ux} = \mathbf{Y} . \quad (184)$$

Подставляя (184) в (183), получим систему

$$\mathbf{LY} = \mathbf{B} . \quad (185)$$

При этом решение системы (183) можно получить последовательным решением линейных систем (185) и (184), причем каждая из них является треугольной и легко решается методами прямой и обратной подстановки [3].

Сама процедура разложения матрицы коэффициентов СЛАУ на две треугольные матрицы выполняется с использованием метода исключения Гаусса, т.е. посредством операций перестановки, масштабирования и замещения. Покажем это на простом примере [3].

Пусть невырожденная матрица коэффициентов СЛАУ имеет вид

$$\mathbf{A} = \begin{bmatrix} 4 & 3 & -1 \\ -2 & -4 & 5 \\ 1 & 2 & 6 \end{bmatrix} . \quad (186)$$

$$\mathbf{A} = \mathbf{EA} , \quad (187)$$

где \mathbf{E} – единичная матрица, все элементы главной диагонали которой единицы, а остальные – нули.

Учитывая (187), первый шаг разложения на треугольные матрицы можно записать в виде

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 4 & 3 & -1 \\ -2 & -4 & 5 \\ 1 & 2 & 6 \end{bmatrix} . \quad (188)$$

Применяя метод исключения Гаусса к правой матрице и занося константы замещения в левую, получим следующую последовательность операций:

– замещение второй строки правой матрицы разностью второй строки и первой строки, умноженной на константу -0.5, которая будет элементом l_{21} левой матрицы

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 4 & 3 & -1 \\ 0 & -2.5 & 4.5 \\ 1 & 2 & 6 \end{bmatrix} ; \quad (189)$$

– замещение третьей строки правой матрицы разностью третьей строки и первой строки, умноженной на константу 0.25, которая будет элементом l_{31} левой матрицы

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0.25 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 4 & 3 & -1 \\ 0 & -2.5 & 4.5 \\ 0 & 1.25 & 6.25 \end{bmatrix}; \quad (190)$$

– замещение третьей строки правой матрицы разностью третьей строки и второй строки, умноженной на константу -0.5, которая будет элементом l_{32} левой матрицы

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0.25 & -0.5 & 1 \end{bmatrix} \times \begin{bmatrix} 4 & 3 & -1 \\ 0 & -2.5 & 4.5 \\ 0 & 0 & 8.5 \end{bmatrix}. \quad (191)$$

В результате получили треугольные матрицы разложения в виде

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0.25 & -0.5 & 1 \end{bmatrix}; \quad (192)$$

$$\mathbf{U} = \begin{bmatrix} 4 & 3 & -1 \\ 0 & -2.5 & 4.5 \\ 0 & 0 & 8.5 \end{bmatrix}. \quad (193)$$

Как видно из приведенного примера, при решении СЛАУ методом LU-разложения основные вычислительные затраты приходятся на разложение матрицы коэффициентов на треугольные матрицы.

Следует отметить, что вычислительная сложность методов исключения Гаусса и метода LU-разложения одинакова [3]. Однако, если необходимо решить несколько систем с одинаковыми матрицами коэффициентов, но различными векторами свободных членов (правая часть СЛАУ), то метод LU-разложения окажется предпочтительным, так как в этом случае нет необходимости производить разложение матрицы коэффициентов многократно. Достаточно лишь сохранить полученные треугольные матрицы в памяти компьютера и, подставляя различные вектора свободных членов, получать решения методами прямой и обратной подстановки. Это позволит значительно сократить объем вычислений по сравнению с методом исключения Гаусса.

4.1.3. Итерационные методы решения систем линейных алгебраических уравнений

Итерационные методы используются, как правило, для решения систем линейных алгебраических уравнений больших размерностей. В частности, при решении многих задач математической физики дискретизация дифференциаль-

ных уравнений в частных производных приводит к СЛАУ, содержащим десятки и сотни тысяч уравнений и более [1, 3].

Поскольку вычислительная сложность методов исключения Гаусса и LU-разложения составляет $O(n^3)$ [3], решение линейных систем таких размерностей данными методами представляет серьезную проблему, решить которую позволяют итерационные методы Якоби, Гаусса-Зейделя и др.

Пусть требуется решить систему линейных уравнений (168). Перепишем ее в следующем виде:

$$\begin{aligned}
 x_1 &= \frac{b_1 - a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n}{a_{11}}, \\
 x_2 &= \frac{b_2 - a_{21}x_1 + a_{23}x_3 + \dots + a_{2n}x_n}{a_{22}}, \\
 &\vdots \\
 x_i &= \frac{b_i - a_{i1}x_1 + \dots + a_{i(i-1)}x_{i-1} + a_{i(i+1)}x_{i+1} + \dots + a_{in}x_n}{a_{ii}}, \\
 &\vdots \\
 x_n &= \frac{b_n - a_{n1}x_1 + a_{n2}x_2 + \dots + a_{n(n-1)}x_{n-1}}{a_{nn}}.
 \end{aligned} \tag{194}$$

Для решения системы (194) может быть использован следующий итерационный метод, получивший название итерации Якоби:

- 1) задание допустимой погрешности решения δ . Допустимая погрешность задается, как правило, в относительных единицах или процентах. Но в некоторых случаях удобнее задавать ее в абсолютных единицах;
- 2) задание начального приближения $\mathbf{x}^{(0)} = [x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}]^{11}$. Начальное приближение к решению может быть задано произвольным образом, либо из определенных соображений. Например, если известна окрестность решения, то есть область в n -мерном пространстве, в которой гарантированно находится решение, в качестве начального приближения может быть выбрана центральная точка этой области;
- 3) нахождение следующего приближения к решению $\mathbf{x}^{(k)} = [x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}]$ путем подстановки текущего приближения $\mathbf{x}^{(k-1)} = [x_1^{(k-1)}, x_2^{(k-1)}, \dots, x_n^{(k-1)}]$ в правую часть системы (194)

$$x_i^{(k)} = \frac{b_i - a_{i1}x_1^{(k-1)} + \dots + a_{i(i-1)}x_{i-1}^{(k-1)} + a_{i(i+1)}x_{i+1}^{(k-1)} + \dots + a_{in}x_n^{(k-1)}}{a_{ii}}, \tag{195}$$

где $i = 1, 2, \dots, n$;

¹¹ Верхний индекс в скобках указывает на номер итерации

- 4) определение погрешности k -го приближения ε . Например, в относительную погрешность приближения часто находят в соответствии с выражением

$$\varepsilon = \frac{\max(|x_1^{(k)} - x_1^{(k-1)}|, |x_2^{(k)} - x_2^{(k-1)}|, \dots, |x_n^{(k)} - x_n^{(k-1)}|)}{\max(|x_1^{(k-1)}|, |x_2^{(k-1)}|, \dots, |x_n^{(k-1)}|)}; \quad (196)$$

- 5) если выполняется неравенство

$$\varepsilon \leq \delta, \quad (197)$$

то найденное приближение к решению удовлетворяет заданной точности и итерационный процесс завершается выводом полученного результата. В противном случае осуществляется переход к п. 3) и выполняется новая итерация.

При правильной реализации вычислительного процесса каждое последующее приближение к решению должно быть точнее предыдущего. Естественно, в этом случае погрешность ε с каждой итерацией уменьшается. Такой итерационный процесс называют сходящимся.

Следует отметить, что в некоторых случаях итерационный процесс с каждой итерацией может не приближать, а наоборот, удалять от истинного решения. При этом погрешность ε с каждой итерацией увеличивается, а итерационный процесс называют расходящимся. Такая ситуация возникает при невыполнении определенных условий, которые называют условиями сходимости. Они зависят от особенностей организации итерационного процесса и характера системы алгебраических уравнений и будут рассмотрены ниже.

Следует иметь в виду, что в случае, когда знаменатель выражения (196) будет стремиться к нулю, относительная погрешность может значительно возрастать при фактическом приближении к решению. Иными словами, в данном случае будет сделан неверный вывод о расходимости фактически сходящегося итерационного процесса.

Во избежание данной проблемы используют отличные от (196) выражения для определения погрешности приближения, например,

$$\varepsilon = \frac{\max(|x_1^{(k)} - x_1^{(k-1)}|, |x_2^{(k)} - x_2^{(k-1)}|, \dots, |x_n^{(k)} - x_n^{(k-1)}|)}{\max(|x_1^{(0)}|, |x_2^{(0)}|, \dots, |x_n^{(0)}|)}, \quad (198)$$

если начальное приближение не приводит знаменатель к нулю.

Иногда итерационный процесс можно ускорить.

Согласно итерационной формуле Якоби (195), для получения элементов приближения $x_i^{(k)}$ используются все, за исключением i -го, элементы $x_j^{(k-1)}$ предыдущего приближения. Но, поскольку на момент определения $x_i^{(k)}$ все элементы с индексами $j < i$ уже определены, их можно использовать для определения $x_i^{(k)}$. При этом возникает итерационный процесс, названный итерацией Гаусса-Зейделя [3]:

- 1) задание допустимой погрешности решения δ ;

- 2) задание начального приближения $\mathbf{x}^{(0)} = [x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}]$;
 3) нахождение следующего приближения к решению $\mathbf{x}^{(k)} = [x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}]$ в соответствии с итерационной формулой

$$x_i^{(k)} = \frac{b_i - a_{i1}x_1^{(k)} - \dots - a_{i(i-1)}x_{i-1}^{(k)} - a_{i(i+1)}x_{i+1}^{(k-1)} - \dots - a_{in}x_n^{(k-1)}}{a_{ii}}, \quad (199)$$

где $i = 1, 2, \dots, n$;

- 4) определение погрешности k -го приближения ε ;
 5) если выполняется неравенство

$$\varepsilon \leq \delta, \quad (200)$$

то найденное приближение к решению удовлетворяет заданной точности и итерационный процесс завершается выводом полученного результата. В противном случае осуществляется переход к п. 3) и выполняется новая итерация.

Как видно из приведенных выше описаний, метод Гаусса-Зейделя отличается от метода Якоби лишь незначительным изменением, внесенным в итерационную формулу (199), но это во многих случаях позволяет ускорить итерационный процесс.

Подтвердим сказанное простым примером.

Решим линейную систему

$$\begin{aligned} 2x_1 - x_2 + x_3 &= 2; \\ 4x_1 - 6x_2 + x_3 &= -4; \\ -2x_1 + x_2 + 8x_3 &= 16 \end{aligned} \quad (201)$$

методом Якоби с использованием начального приближения $\mathbf{x}^{(0)} = [1, 2, 2]$.

Для этого перепишем систему (201) в виде

$$\begin{aligned} x_1 &= \frac{2 + x_2 - x_3}{2}; \\ x_2 &= \frac{-4 - 4x_1 - x_3}{-6}; \\ x_3 &= \frac{16 + 2x_1 - x_2}{8}, \end{aligned} \quad (202)$$

после чего решим систему (202) в среде MATLAB.

Функция для решения СЛАУ произвольной размерности методом Якоби в среде MATLAB может иметь следующий вид:

```
function X=yakobi(A,B,X0,delta,Imax)

% Итерация Якоби.
% A      - невырожденная матрица коэффициентов
%         размера n x n;
```

```

% B      - вектор-столбец свободных членов;
% X0     - вектор-столбец начального приближения;
% delta  - допустимая относительная погрешность;
% Imax   - максимальное число итераций;
% X      - приближенное решение линейной системы AX = B.

n=length(B);
err=5*delta;
X=X0;
ct=0;
while err>delta
    Xp=X;
    for i=1:n
        X(i)=(B(i)-A(i,[1:i-1,i+1:n])*...
            Xp([1:i-1,i+1:n]))/A(i,i);
    end
    if max(abs(X0))==0
        error('Следует изменить начальное приближение')
        break
    end
    err=max(abs(X-Xp))/max(abs(X0));
    ct=ct+1;
    Xe(ct,:)=[X' err];
    if ct>Imax
        error('Требуемое число итераций слишком велико')
        break
    end
end
Xe
ct

```

Получить информацию о функциях и синтаксисе написания программ в системе инженерных и научных расчетов MATLAB можно в [10, 11].

Для запуска описанного выше итерационного процесса с допустимой относительной погрешностью результата 10^{-5} и максимальным числом итераций не более 100, необходимо войти в систему MATLAB, создать файл функции и сохранить его под именем `yakobi.m` в рабочем каталоге `/WORK`, находящемся в корневом каталоге системы MATLAB, после чего в командной строке последовательно набрать команды, приведенные ниже.

```

A=[ 2 -1 1; 4 -6 1; -2 1 8];
B=[2; -4; 16];
X0=[1; 2; 2];
X=yakobi(A,B,X0,1e-5,1e2)

```


Первая из них задает матрицу коэффициентов \mathbf{A} размера 3×3 , вторая – вектор-столбец свободных членов \mathbf{B} размера 3×1 , третья – вектор-столбец начального приближения \mathbf{X}_0 размера 3×1 и четвертая осуществляет запуск итерационного процесса Якоби с заданными параметрами.

Полученные результаты будут представлены на экране монитора в следующем виде

$\mathbf{X_e} =$

1.0000	1.6667	2.0000	0.1667
0.8333	1.6667	2.0417	0.0833
0.8125	1.5625	2.0000	0.0521
0.7813	1.5417	2.0078	0.0156
0.7669	1.5221	2.0026	0.0098
0.7598	1.5117	2.0015	0.0052
0.7551	1.5068	2.0010	0.0025
0.7529	1.5036	2.0004	0.0016
0.7516	1.5020	2.0003	0.0008
0.7509	1.5011	2.0001	0.0005
0.7505	1.5006	2.0001	0.0002
0.7503	1.5003	2.0000	0.0001
0.7501	1.5002	2.0000	0.0001
0.7501	1.5001	2.0000	0.0000
0.7500	1.5001	2.0000	0.0000
0.7500	1.5000	2.0000	0.0000
0.7500	1.5000	2.0000	0.0000

$ct =$

17

$\mathbf{X} =$

0.7500
1.5000
2.0000,

где первые три столбца матрицы $\mathbf{X_e}$ представляют собой приближения переменных x_1, x_2, x_3 к решению, четвертый столбец – погрешности приближений, ct – число итераций, \mathbf{X} – вектор результата.

Таким образом, решение системы (201) с относительной погрешностью 10^{-5} и с использованием начального приближения $\mathbf{x}^{(0)} = [1, 2, 2]$ методом Якоби было получено за 17 итераций.

Функция для решения СЛАУ произвольной размерности методом Гаусса-Зейделя в среде MATLAB может иметь следующий вид:

```
function X=zeydel(A,B,X0,delta,Imax)

% Итерация Гаусса-Зейделя.
% A      - невырожденная матрица коэффициентов
%          размера n x n;
% B      - вектор-столбец свободных членов;
% X0     - вектор-столбец начального приближения;
% delta  - допустимая относительная погрешность;
% Imax   - максимальное число итераций;
% X      - приближенное решение линейной системы AX = B.

n=length(B);
err=5*delta;
X=X0;
ct=0;
while err>delta
    Xp=X;
    for i=1:n
        X(i)=(B(i)-A(i,[1:i-1,i+1:n]))*...
            X([1:i-1,i+1:n]))/A(i,i);
    end
    if max(abs(X0))==0
        error('Следует изменить начальное приближение')
        break
    end
    err=max(abs(X-Xp))/max(abs(X0));
    ct=ct+1;
    Xe(ct,:)=[X' err];
    if ct>Imax
        error('Требуемое число итераций слишком велико')
        break
    end
end
Xe
ct
```

Запуск итерационного процесса Гаусса-Зейделя с допустимой относительной погрешностью результата 10^{-5} и максимальным числом итераций не более 100 осуществляется аналогично итерации Якоби. Создаваемый файл функции сохраняется под именем zeydel.m в рабочем каталоге /WORK, находящемся в корневом каталоге системы MATLAB, после чего в командной строке последовательно набираются команды, приведенные ниже.

```

A=[ 2 -1 1; 4 -6 1; -2 1 8];
B=[2; -4; 16];
X0=[1; 2; 2];
X=zeydel(A,B,X0,1e-5,1e2)

```

Первая из них задает матрицу коэффициентов **A** размера 3×3 , вторая – вектор-столбец свободных членов **B** размера 3×1 , третья – вектор-столбец начального приближения **X0** размера 3×1 и четвертая осуществляет запуск итерационного процесса Гаусса-Зейделя с заданными параметрами.

Полученные результаты будут представлены на экране монитора в следующем виде

Xe =

1.0000	1.6667	2.0417	0.1667
0.8125	1.5486	2.0095	0.0937
0.7695	1.5146	2.0031	0.0215
0.7558	1.5044	2.0009	0.0069
0.7517	1.5013	2.0003	0.0020
0.7505	1.5004	2.0001	0.0006
0.7502	1.5001	2.0000	0.0002
0.7500	1.5000	2.0000	0.0001
0.7500	1.5000	2.0000	0.0000
0.7500	1.5000	2.0000	0.0000

ct =

10

X =

```

0.7500
1.5000
2.0000.

```

Таким образом, решение системы (201) с относительной погрешностью 10^{-5} и с использованием начального приближения $\mathbf{x}^{(0)} = [1, 2, 2]$ методом Гаусса-Зейделя было получено за 10 итераций. То есть в данном случае число итераций в процессе Гаусса-Зейделя оказалось в 1.7 раза меньше, чем при решении методом Якоби.

Перед тем, как сформулировать критерий сходимости итераций Якоби и Гаусса-Зейделя для систем линейных алгебраических уравнений, дадим одно определение.

Матрица $\mathbf{A} = [a_{ij}]$ размера $n \times n$ называется строго диагонально доминирующей, если выполняется условие

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \quad (203)$$

то есть если в каждой строке матрицы модуль элемента главной диагонали больше суммы модулей всех остальных элементов строки [3].

Критерий сходимости итераций Якоби и Гаусса-Зейделя формулируется следующим образом.

Если \mathbf{A} – строго диагонально доминирующая матрица, то линейная система $\mathbf{Ax} = \mathbf{B}$ имеет единственное решение $\mathbf{x} = \mathbf{P}$ и итерационные процессы Якоби и Гаусса-Зейделя порождают последовательность векторов $\{\mathbf{x}^{(k)}\}$, которые сходятся к \mathbf{P} для любого выбора вектора начального приближения $\mathbf{x}^{(0)}$ [3].

Данный критерий является достаточным условием сходимости. Если матрица коэффициентов не является строго диагонально доминирующей, итерационные процессы Якоби и Гаусса-Зейделя могут быть расходящимися.

В качестве примера достаточно в системе (201) поменять местами первое и второе уравнения. При этом матрица коэффициентов будет иметь вид

$$\mathbf{A} = \begin{bmatrix} 4 & -6 & 1 \\ 2 & -1 & 1 \\ -2 & 1 & 8 \end{bmatrix}, \quad (204)$$

то есть не будет строго диагонально доминирующей, поскольку для первых двух строк условие (203) не выполняется.

Результаты, полученные для первых 15 итераций Якоби в этом случае будут иметь вид¹²

$\mathbf{Xe} =$

1.0e+003 *

0.0015	0.0020	0.0020	0.0003
0.0015	0.0030	0.0021	0.0005
0.0030	0.0031	0.0020	0.0007
0.0032	0.0059	0.0024	0.0014
0.0073	0.0067	0.0021	0.0021
0.0086	0.0147	0.0030	0.0040
0.0203	0.0181	0.0023	0.0059

¹² Данная форма представления матрицы \mathbf{Xe} означает, что 1.0e+003 (10^3) является множителем для каждого элемента матрицы.

0.0256	0.0409	0.0048	0.0114
0.0591	0.0541	0.0033	0.0167
0.0793	0.1196	0.0100	0.0327
0.1758	0.1666	0.0069	0.0483
0.2472	0.3565	0.0251	0.0950
0.5275	0.5175	0.0192	0.1402
0.7704	1.0723	0.0692	0.2774
1.5901	1.6080	0.0606	0.4098
2.3959	3.2388	0.1985	0.8154

??? Error using ==> yakobi

Требуемое число итераций слишком велико

Аналогичные результаты для итераций Гаусса-Зейделя будут выглядеть следующим образом:

Xe =

1.0e+007 *

0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000
0.0001	0.0001	0.0000	0.0000
0.0002	0.0003	0.0000	0.0001
0.0005	0.0010	0.0000	0.0003
0.0015	0.0030	0.0000	0.0010
0.0044	0.0089	0.0000	0.0030
0.0133	0.0266	0.0000	0.0089
0.0399	0.0797	0.0000	0.0266
0.1196	0.2391	0.0000	0.0797
0.3587	0.7174	0.0000	0.2391
1.0762	2.1523	0.0000	0.7174

??? Error using ==> zeydel

Требуемое число итераций слишком велико

Обе последовательности расходятся. Из приведенных примеров видно, что более быстрая сходимость метода Гаусса-Зейделя при условии строго диагонально доминирующей матрицы коэффициентов СЛАУ оборачивается более быстрой расходимостью при невыполнении условия (203).

И хотя во многих случаях метод Гаусса-Зейделя сходится быстрее итерации Якоби, иногда метод Якоби будет сходиться даже тогда, когда метод Гаусса-Зейделя расходится [3].

Как было указано выше, итерационные методы используются, как правило, для решения систем линейных алгебраических уравнений больших размерностей. В частности, при решении многих задач математической физики дискретизация дифференциальных уравнений в частных производных приводит к СЛАУ, содержащим десятки и сотни тысяч уравнений и при этом матрицы коэффициентов содержат большое количество нулей [1, 3].

Если объем оперативной памяти компьютера недостаточен для решения системы уравнений большой размерности методами исключения Гаусса или LU-разложения, данная система может быть решена одним из описанных выше итерационных методов с использованием последовательного итерационного решения двух или более подсистем меньшей размерности.

Например, пусть необходимо решить линейную систему

$$\begin{aligned}
 a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n &= b_1; \\
 a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n &= b_2; \\
 a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \dots + a_{3n}x_n &= b_3; \\
 &\vdots \\
 a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n &= b_n.
 \end{aligned} \tag{205}$$

Разобьем систему (205) на 2 подсистемы:

$$\begin{aligned}
 a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n &= b_1; \\
 a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n &= b_2; \\
 a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \dots + a_{3n}x_n &= b_3; \\
 &\vdots \\
 a_{(n/2)1}x_1 + a_{(n/2)2}x_2 + a_{(n/2)3}x_3 + \dots + a_{(n/2)n}x_n &= b_{n/2}
 \end{aligned} \tag{206}$$

и

$$\begin{aligned}
 a_{(n/2+1)1}x_1 + a_{(n/2+1)2}x_2 + a_{(n/2+1)3}x_3 + \dots + a_{(n/2+1)n}x_n &= b_{n/2+1}; \\
 a_{(n/2+2)1}x_1 + a_{(n/2+2)2}x_2 + a_{(n/2+2)3}x_3 + \dots + a_{(n/2+2)n}x_n &= b_{n/2+2}; \\
 a_{(n/2+3)1}x_1 + a_{(n/2+3)2}x_2 + a_{(n/2+3)3}x_3 + \dots + a_{(n/2+3)n}x_n &= b_{n/2+3}; \\
 &\vdots \\
 a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n &= b_n.
 \end{aligned} \tag{207}$$

Далее воспользуемся следующим алгоритмом решения полученной системы:

- 1) задание допустимой погрешности решения δ ;
- 2) задание начального приближения для переменных $\{x_1, x_2, x_3, \dots, x_{n/2}\}$ (произвольным образом или по определенному правилу из известной окрестности решения);
- 3) подстановка полученных значений переменных $\{x_1, x_2, x_3, \dots, x_{n/2}\}$ в подсистему (207);
- 4) решение подсистемы (207) относительно переменных $\{x_{n/2+1}, x_{n/2+2}, x_{n/2+3}, \dots, x_n\}$ одним из известных методов (метод исключения Гаусса, LU-разложения, итерация Якоби и др.);
- 5) подстановка полученных значений переменных $\{x_{n/2+1}, x_{n/2+2}, x_{n/2+3}, \dots, x_n\}$ в подсистему (206);
- 6) решение подсистемы (206) относительно переменных $\{x_1, x_2, x_3, \dots, x_{n/2}\}$ одним из известных методов (метод исключения Гаусса, LU-разложения, итерация Якоби и др.);
- 7) определение относительной погрешности приближения любым из известных способов, например, в соответствии с выражением

$$\varepsilon = \frac{\max(|x_1^{(i)} - x_1^{(i-1)}|, |x_2^{(i)} - x_2^{(i-1)}|, \dots, |x_{n/2}^{(i)} - x_{n/2}^{(i-1)}|)}{\max(|x_1^{(0)}|, |x_2^{(0)}|, \dots, |x_{n/2}^{(0)}|)}, \quad (208)$$

где переменные и числа в скобках в верхних индексах обозначают номер итерации;

- 8) если выполняется неравенство

$$\varepsilon \leq \delta, \quad (209)$$

то найденное приближение к решению удовлетворяет заданной точности и итерационный процесс завершается выводом полученного результата $\{x_1, x_2, x_3, \dots, x_n\}$. В противном случае осуществляется переход к п. 3) и выполняется новая итерация.

Данный алгоритм позволяет примерно в 2 раза сократить требуемый объем оперативной памяти компьютера, поскольку подсистемы (206) и (207) размерностью в 2 раза меньшей, чем исходная система (205), решаются последовательно. При необходимости система (205) может быть разбита на большее число последовательно решаемых подсистем, что позволит дополнительно увеличить ее допустимую размерность.

4.2. Методы решения систем нелинейных алгебраических уравнений

В данном подразделе обобщим итерационные методы Якоби и Гаусса-Зейделя, рассмотренные выше, на случай систем нелинейных алгебраических уравнений, а также рассмотрим метод Ньютона-Рафсона и условия сходимости методов [1, 3].

4.2.1. Итерация неподвижной точки

Пусть задана система нелинейных алгебраических уравнений в виде

$$\begin{aligned} x_1 &= f_1(x_1, x_2, \dots, x_n) \\ x_2 &= f_2(x_1, x_2, \dots, x_n) \\ x_3 &= f_3(x_1, x_2, \dots, x_n) \\ &\vdots \\ x_n &= f_n(x_1, x_2, \dots, x_n) \end{aligned} \quad (210)$$

где f_1, f_2, \dots, f_n – некоторые нелинейные функции от $(x_1, x_2, x_3, \dots, x_n)$.

Неподвижной точкой для системы (210) называется такая точка $\mathbf{P} = (p_1, p_2, \dots, p_n)$, для которой выполняется равенство [3]

$$\begin{aligned} p_1 &= f_1(p_1, p_2, \dots, p_n); \\ p_2 &= f_2(p_1, p_2, \dots, p_n); \\ p_3 &= f_3(p_1, p_2, \dots, p_n); \\ &\vdots \\ p_n &= f_n(p_1, p_2, \dots, p_n). \end{aligned} \quad (211)$$

Иными словами, неподвижная точка фактически является решением системы (210).

Итерационный метод решения систем нелинейных алгебраических уравнений, называемый итерацией неподвижной точки, является обобщением итерации Якоби на случай нелинейных систем состоит в следующем:

- 1) задание допустимой погрешности решения δ ;
- 2) задание начального приближения $\mathbf{x}^{(0)} = [x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}]$;
- 3) нахождение следующего приближения к решению $\mathbf{x}^{(k)} = [x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}]$ путем подстановки текущего приближения $\mathbf{x}^{(k-1)} = [x_1^{(k-1)}, x_2^{(k-1)}, \dots, x_n^{(k-1)}]$ в правую часть системы (210)

$$x_i^{(k)} = f_i(x_1^{(k-1)}, x_2^{(k-1)}, x_3^{(k-1)}, \dots, x_n^{(k-1)}), \quad (212)$$

где $i = 1, 2, \dots, n$;

- 4) определение погрешности k -го приближения ε в соответствии с выражением

$$\varepsilon = \frac{\max(|x_1^{(k)} - x_1^{(k-1)}|, |x_2^{(k)} - x_2^{(k-1)}|, \dots, |x_n^{(k)} - x_n^{(k-1)}|)}{\max(|x_1^{(0)}|, |x_2^{(0)}|, \dots, |x_n^{(0)}|)}; \quad (213)$$

5) если выполняется неравенство

$$\varepsilon \leq \delta, \quad (214)$$

то найденное приближение к решению удовлетворяет заданной точности и итерационный процесс завершается выводом полученного результата. В противном случае осуществляется переход к п. 3) и выполняется новая итерация.

Обобщение итерации Гаусса-Зейделя на случай нелинейных систем будет аналогично рассмотренному выше, за исключением итерационной формулы (212), которая в данном случае будет иметь вид

$$x_i^{(k)} = f_i(x_1^{(k)}, x_2^{(k)}, \dots, x_{i-1}^{(k)}, x_i^{(k-1)}, \dots, x_n^{(k-1)}), \quad (215)$$

где $i = 1, 2, \dots, n$.

Критерий сходимости итерации неподвижной точки к решению может быть сформулирован следующим образом.

Пусть задана нелинейная система (210) и пусть функции f_1, f_2, \dots, f_n и их первые частные производные непрерывны в некоторой области, содержащей неподвижную точку $\mathbf{P} = (p_1, p_2, \dots, p_n)$.

Если начальное приближение $\mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})$ находится в некоторой достаточно малой окрестности неподвижной точки и выполняются неравенства

$$\begin{aligned} & \left| \frac{\partial f_1}{\partial x_1} \right|_{(p_1, p_2, \dots, p_n)} + \left| \frac{\partial f_1}{\partial x_2} \right|_{(p_1, p_2, \dots, p_n)} + \dots + \left| \frac{\partial f_1}{\partial x_n} \right|_{(p_1, p_2, \dots, p_n)} < 1; \\ & \left| \frac{\partial f_2}{\partial x_1} \right|_{(p_1, p_2, \dots, p_n)} + \left| \frac{\partial f_2}{\partial x_2} \right|_{(p_1, p_2, \dots, p_n)} + \dots + \left| \frac{\partial f_2}{\partial x_n} \right|_{(p_1, p_2, \dots, p_n)} < 1; \\ & \left| \frac{\partial f_3}{\partial x_1} \right|_{(p_1, p_2, \dots, p_n)} + \left| \frac{\partial f_3}{\partial x_2} \right|_{(p_1, p_2, \dots, p_n)} + \dots + \left| \frac{\partial f_3}{\partial x_n} \right|_{(p_1, p_2, \dots, p_n)} < 1; \\ & \vdots \\ & \left| \frac{\partial f_n}{\partial x_1} \right|_{(p_1, p_2, \dots, p_n)} + \left| \frac{\partial f_n}{\partial x_2} \right|_{(p_1, p_2, \dots, p_n)} + \dots + \left| \frac{\partial f_n}{\partial x_n} \right|_{(p_1, p_2, \dots, p_n)} < 1, \end{aligned} \quad (216)$$

то итерация, заданная в (212) или в (215), сходится к неподвижной точке.

Критерий (216) является достаточным условием сходимости итерации неподвижной точки. Если условие (216) не выполняется, итерационный процесс может (но не обязательно) расходиться. Обычно это происходит, если сумма модулей частных производных в окрестности неподвижной точки существенно больше 1 [3].

Критерий сходимости для итерации неподвижной точки можно наглядно проиллюстрировать графически на примере итерационного решения нелинейного уравнения одной переменной

$$x = g(x), \quad (217)$$

где $g(x)$ – некоторая нелинейная функция аргумента x .

Нетрудно видеть, что уравнение (217) может быть представлено в виде системы двух уравнений:

$$\begin{cases} y = g(x); \\ y = x. \end{cases} \quad (218)$$

Если систему (218) представить графически, то ее решением, а значит и решением уравнения (217), будет точка P пересечения графиков $y = g(x)$ и $y = x$ (рис. 14).

Выберем некоторое начальное приближение $x = p_0$ (см. рис. 14). Подставив это значение в уравнение $y = g(x)$, найдем соответствующее значение y , подставив которое в уравнение $y = x$, получим новое приближение к решению $x = p_1$ (см. рис. 14).

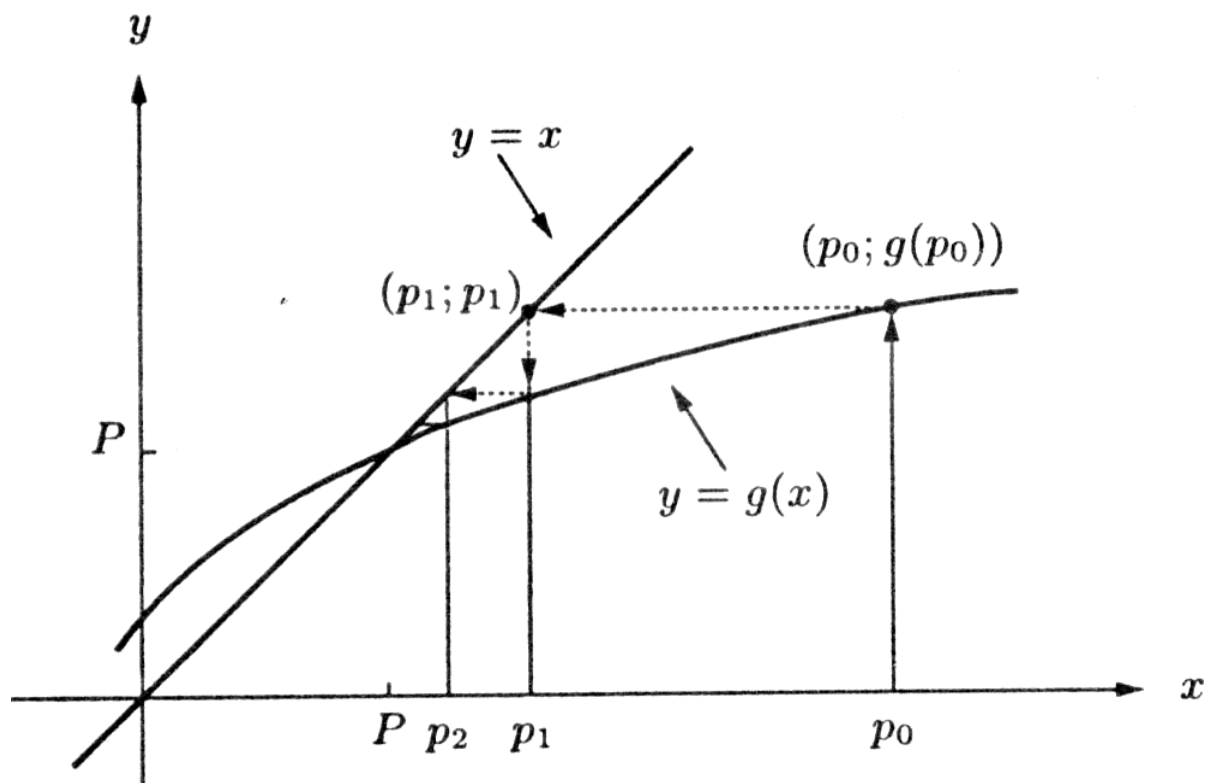


Рис. 14. Монотонно сходящаяся итерация неподвижной точки

Далее, проведя аналогичные операции, получим следующее приближение $x = p_2$, и так далее до тех пор, пока очередное приближение не будет удовлетворять заданной точности, то есть будет достаточно близко к точному решению P (см. рис. 14).

Из приведенных графиков видно, что в данном примере в рассматриваемой окрестности точки P выполняется неравенство

$$\left| \frac{dg(x)}{dx} \right| < 1, \quad (219)$$

что является достаточным условием сходимости. Каждое последующее приближение ближе к решению, чем предыдущее. Шаг между k -м и $(k - 1)$ -м приближениями с каждой итерацией уменьшается, что приводит к уменьшению относительной погрешности приближения, определяемой выражением (213).

Следует отметить, что погрешность k -го приближения должна, строго говоря, определяться относительно точного решения P . Но, поскольку точное решение, являясь целью вычислительного процесса, нам неизвестно, в качестве критерия остановки итерационных процедур используют выражение (213), определяющее погрешность k -го приближения относительно $(k - 1)$ -го приближения. Следует помнить, что при этом истинная погрешность k -го приближения может (но не обязательно) существенно превышать погрешность, определяемую выражением (213).

В приведенном примере (см. рис. 14) модуль разности между текущим приближением и решением P с каждой итерацией уменьшается, а знак этой разности остается неизменным. Такая сходимость называется монотонной [3].

На рис. 15 приведен пример нелинейного уравнения вида (217), для которого в окрестности решения P тоже выполняется условие сходимости (219). При этом модуль разности между текущим приближением и решением P с каждой итерацией также уменьшается, но знак этой разности с каждой последующей итерацией изменяется на противоположный. Такая сходимость называется колеблющейся [3].

Если уравнение (217), представленное в виде системы (218) таково, что в окрестности решения P условие сходимости (219) не выполняется,

$$\left| \frac{dg(x)}{dx} \right| > 1, \quad (220)$$

то, даже если начальное приближение $x = p_0$ достаточно близко к решению P , итерация неподвижной точки дает расходящуюся последовательность. При этом в зависимости от знака производной $dg(x)/dx$ в окрестности решения будет наблюдаться монотонная расходящаяся (в случае $dg(x)/dx > 1$, рис. 16) или расходящиеся колебания (в случае $dg(x)/dx < -1$, рис. 17) [3].

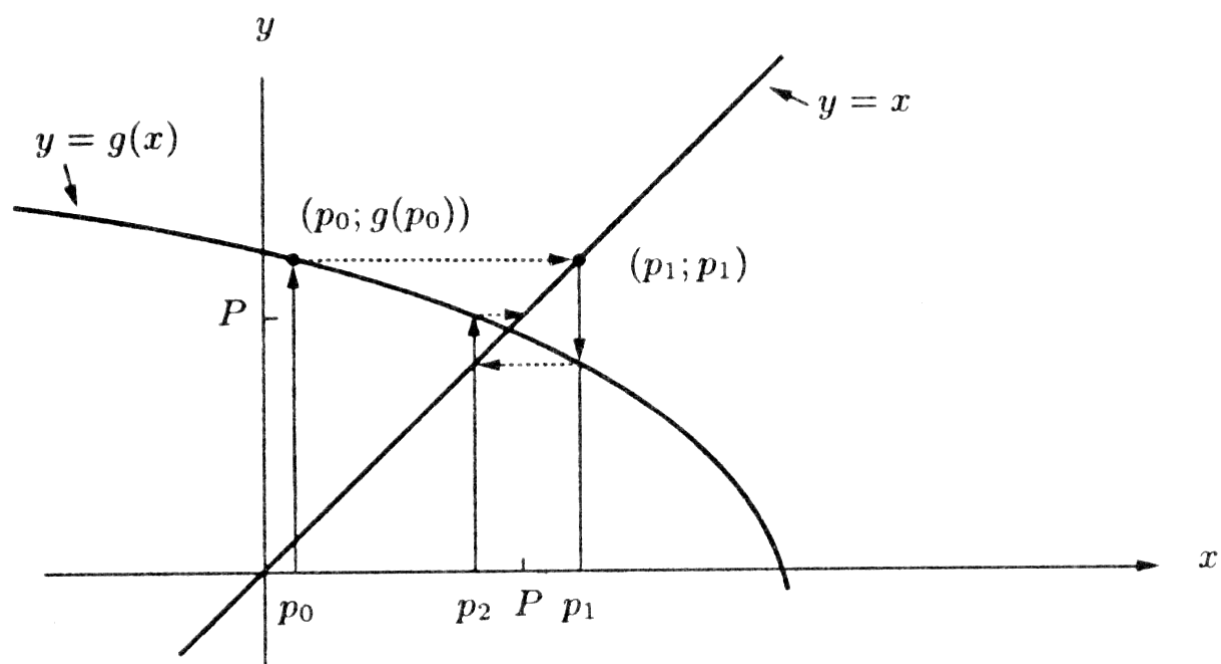


Рис. 15. Колеблущаяся сходимость итерации неподвижной точки

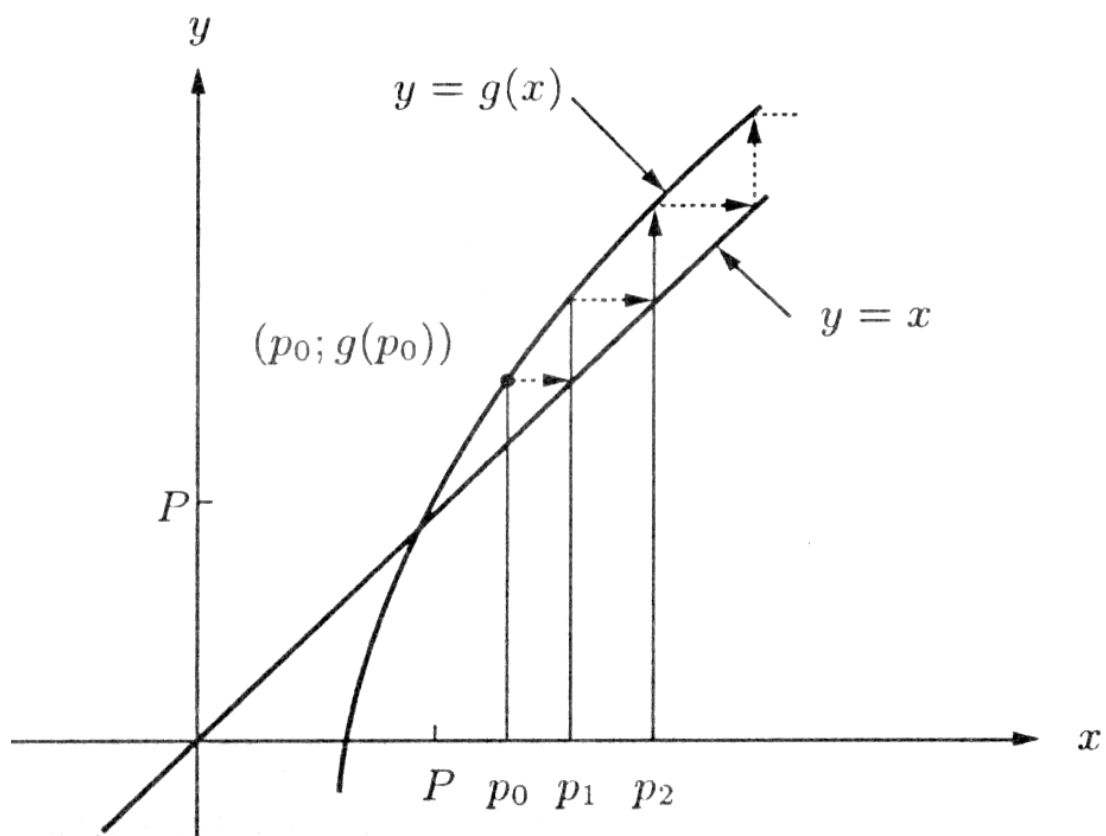


Рис. 16. Монотонная расходимость итерации неподвижной точки

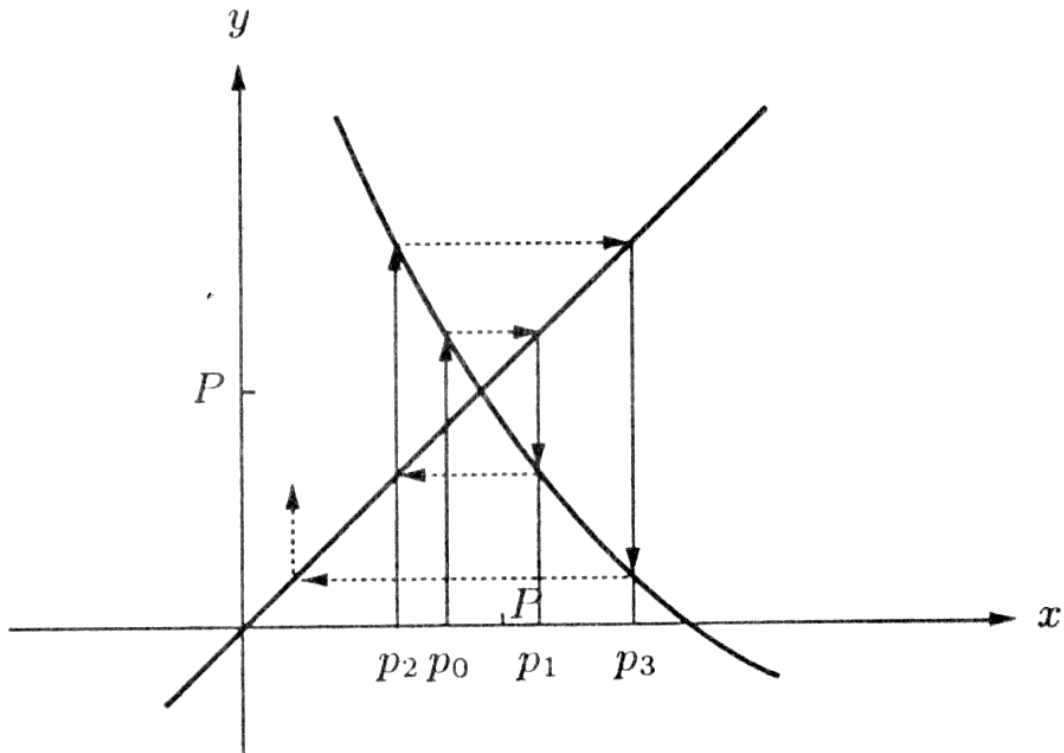


Рис. 17. Расходящиеся колебания в процессе итерации неподвижной точки

4.2.2. Метод Ньютона-Рафсона

Пусть требуется решить нелинейное уравнение (217). Перепишем это уравнение в виде

$$f(x)=0. \quad (221)$$

Если функция $f(x)$, ее первая и вторая производные непрерывны в окрестности решения p , то для решения уравнения (221) может быть использован метод Ньютона-Рафсона (или просто Ньютона), который характеризуется более быстрой сходимостью, чем итерация неподвижной точки [1 – 4].

Если представить уравнение (221) графически, то его решением будет точка p пересечения графика $y=f(x)$ с осью x (рис. 18).

Пусть выбранное начальное приближение $x=p_0$ достаточно близко к решению. Определим следующее приближение $x=p_1$ как точку пересечения с осью x касательной к графику $y=f(x)$ в точке $x=p_0$. Точка $x=p_1$ ближе к решению, чем $x=p_0$ (см. рис. 18).

Соотношение, связывающее p_0 и p_1 , можно получить из выражения для тангенса угла наклона касательной к графику $y=f(x)$, проходящей через точки $x=p_0$ и $x=p_1$:

$$\left. \frac{df(x)}{dx} \right|_{x=p_0} = \frac{f(p_1) - f(p_0)}{p_1 - p_0}. \quad (222)$$

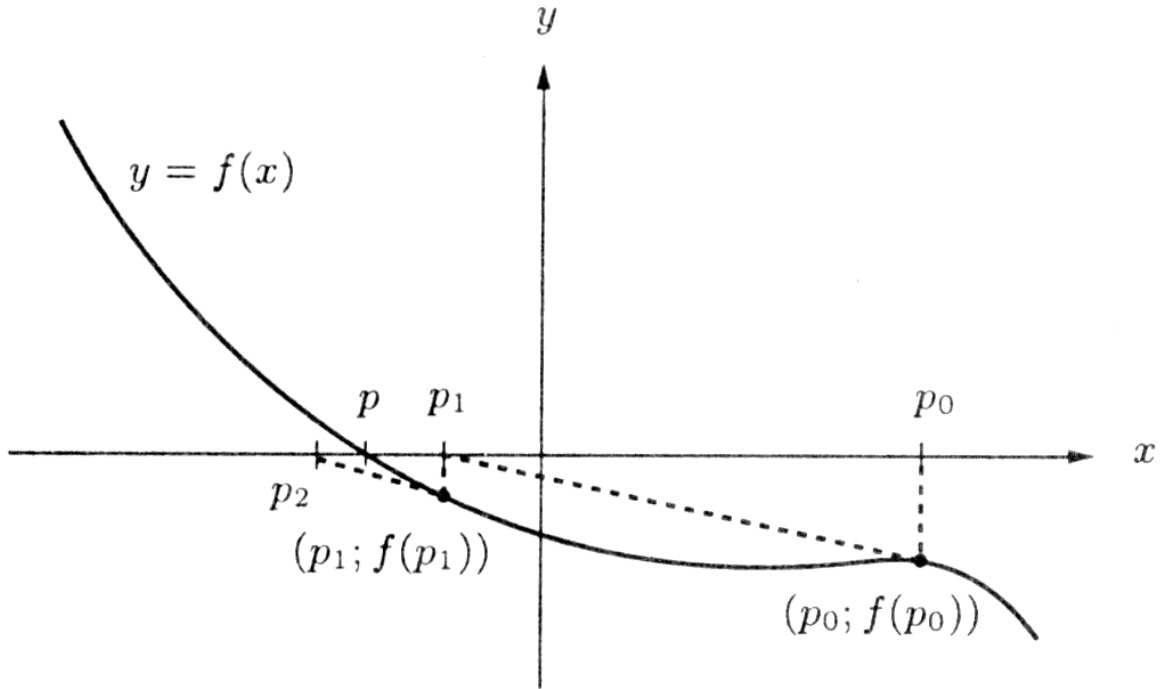


Рис. 18. Графическая интерпретация метода Ньютона-Рафсона

Учитывая, что $f(p_1) = 0$ (см. рис. 18), получим

$$p_1 = p_0 - \frac{f(p_0)}{\left. \frac{df(x)}{dx} \right|_{x=p_0}}. \quad (223)$$

Обобщая выражение (223) для k -й итерации, получим

$$p_k = p_{k-1} - \frac{f(p_{k-1})}{\left. \frac{df(x)}{dx} \right|_{x=p_{k-1}}}. \quad (224)$$

Пусть требуется решить систему нелинейных алгебраических уравнений размерности n , записанную в виде

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0; \\ f_2(x_1, x_2, \dots, x_n) &= 0; \\ f_3(x_1, x_2, \dots, x_n) &= 0; \\ &\vdots \\ f_n(x_1, x_2, \dots, x_n) &= 0. \end{aligned} \quad (225)$$

В матричном представлении система (225) может быть записана как

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}, \quad (226)$$

где $\mathbf{x} = [x_1, x_2, \dots, x_n]$ – вектор-столбец переменных; $\mathbf{0}$ в правой части – вектор-столбец размерности n , все элементы которого равны 0.

Обобщенный метод Ньютона-Рафсона на случай нелинейных систем произвольной размерности n состоит в следующем:

- 1) задание допустимой погрешности решения δ ;
- 2) задание начального приближения $\mathbf{x}^{(0)} = [x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}]$;
- 3) нахождение следующего приближения к решению $\mathbf{x}^{(k)} = [x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}]$ путем подстановки текущего приближения $\mathbf{x}^{(k-1)} = [x_1^{(k-1)}, x_2^{(k-1)}, \dots, x_n^{(k-1)}]$ в итерационную формулу, аналогичную (224):

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} - \left[\left. \frac{d\mathbf{F}(\mathbf{x})}{d\mathbf{x}} \right|_{\mathbf{x}^{(k-1)}} \right]^{-1} \times \mathbf{F}(\mathbf{x}^{(k-1)}), \quad (227)$$

где $\left[\left. \frac{d\mathbf{F}(\mathbf{x})}{d\mathbf{x}} \right|_{\mathbf{x}^{(k-1)}} \right]^{-1}$ – матрица размера $n \times n$, обратная матрице

$$\left[\left. \frac{d\mathbf{F}(\mathbf{x})}{d\mathbf{x}} \right| \right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}, \quad (228)$$

называемой матрицей Якоби [3], в точке $\mathbf{x}^{(k-1)} = [x_1^{(k-1)}, x_2^{(k-1)}, \dots, x_n^{(k-1)}]$;

- 4) определение погрешности k -го приближения ε в соответствии с выражением

$$\varepsilon = \frac{\max(|x_1^{(k)} - x_1^{(k-1)}|, |x_2^{(k)} - x_2^{(k-1)}|, \dots, |x_n^{(k)} - x_n^{(k-1)}|)}{\max(|x_1^{(0)}|, |x_2^{(0)}|, \dots, |x_n^{(0)}|)}; \quad (229)$$

- 5) если выполняется неравенство

$$\varepsilon \leq \delta, \quad (230)$$

то найденное приближение к решению удовлетворяет заданной точности и итерационный процесс завершается выводом полученного результата. В противном случае осуществляется переход к п. 3) и выполняется новая итерация.

Так же, как и итерация неподвижной точки, метод Ньютона-Рафсона может давать расходящиеся последовательности приближений к решению, один из примеров которой показан на рис. 19 [3].

Кроме того, возможны так называемые циклические последовательности, когда получаемые приближения к решению циклически повторяются или почти повторяются (рис. 20) [3].

Критерий сходимости метода Ньютона-Рафсона может быть сформулирован следующим образом [3].

Пусть задана система нелинейных алгебраических уравнений вида (226). Введем матричную функцию

$$\mathbf{G}(\mathbf{x}) = \mathbf{x} - \left[\frac{d\mathbf{F}(\mathbf{x})}{d\mathbf{x}} \right]^{-1} \times \mathbf{F}(\mathbf{x}), \quad (231)$$

называемую интерполяционной функцией Ньютона-Рафсона.

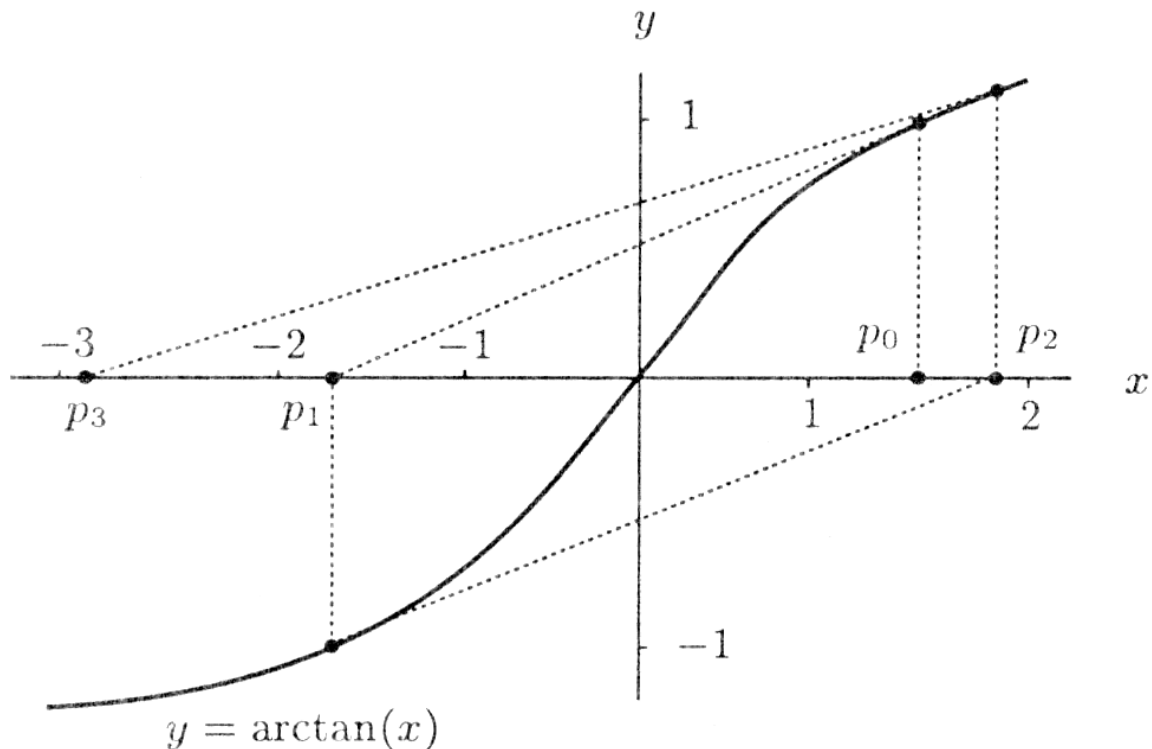


Рис. 19. Расходящаяся последовательность приближений к решению при использовании метода Ньютона-Рафсона

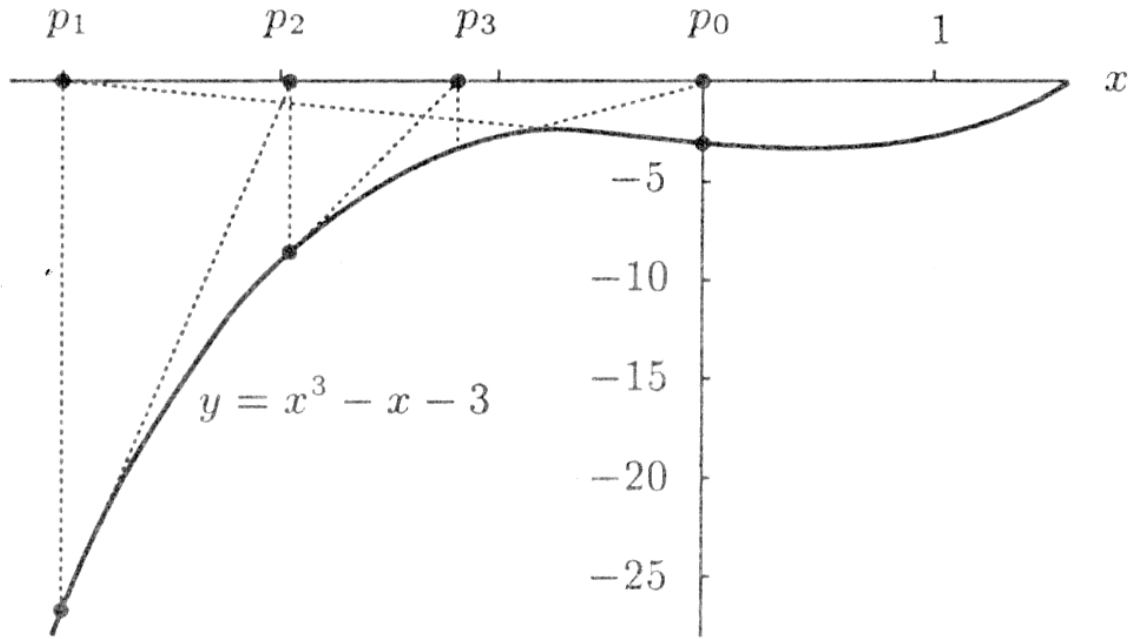


Рис. 20. Циклическая последовательность приближений к решению при использовании метода Ньютона-Рафсона

Пусть $\mathbf{P} = (p_1, p_2, \dots, p_n)$ – решение системы (226). Тогда

$$\mathbf{F}(\mathbf{P}) = \mathbf{0} . \quad (232)$$

Подставляя (232) в интерполяционную функцию (231), получим

$$\mathbf{G}(\mathbf{P}) = \mathbf{P} . \quad (233)$$

Иными словами, метод Ньютона-Рафсона фактически является итерацией неподвижной точки для функции (231), откуда следует, что достаточным условием сходимости метода Ньютона-Рафсона будет выполнение неравенств

$$\begin{aligned}
& \left| \frac{\partial g_1}{\partial x_1} \right|_{(p_1, p_2, \dots, p_n)} + \left| \frac{\partial g_1}{\partial x_2} \right|_{(p_1, p_2, \dots, p_n)} + \dots + \left| \frac{\partial g_1}{\partial x_n} \right|_{(p_1, p_2, \dots, p_n)} < 1 ; \\
& \left| \frac{\partial g_2}{\partial x_1} \right|_{(p_1, p_2, \dots, p_n)} + \left| \frac{\partial g_2}{\partial x_2} \right|_{(p_1, p_2, \dots, p_n)} + \dots + \left| \frac{\partial g_2}{\partial x_n} \right|_{(p_1, p_2, \dots, p_n)} < 1 ; \\
& \left| \frac{\partial g_3}{\partial x_1} \right|_{(p_1, p_2, \dots, p_n)} + \left| \frac{\partial g_3}{\partial x_2} \right|_{(p_1, p_2, \dots, p_n)} + \dots + \left| \frac{\partial g_3}{\partial x_n} \right|_{(p_1, p_2, \dots, p_n)} < 1 ; \\
& \vdots \\
& \left| \frac{\partial g_n}{\partial x_1} \right|_{(p_1, p_2, \dots, p_n)} + \left| \frac{\partial g_n}{\partial x_2} \right|_{(p_1, p_2, \dots, p_n)} + \dots + \left| \frac{\partial g_n}{\partial x_n} \right|_{(p_1, p_2, \dots, p_n)} < 1 .
\end{aligned} \tag{234}$$

В учебных пособиях по математическому анализу [1, 3, 7] показано, что итерация неподвижной точки характеризуется линейной сходимостью, а метод Ньютона-Рафсона – квадратичной сходимостью. Это значит, что при условии сходимости итерации неподвижной точки, ошибка в каждой последующей итерации убывает пропорционально ошибке в предыдущей итерации. А при условии сходимости метода Ньютона-Рафсона, ошибка в каждой последующей итерации убывает пропорционально квадрату ошибки в предыдущей итерации.

5. ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ФИЗИКИ В СИСТЕМЕ MATLAB

В данном разделе приведены примеры решения некоторых задач математической физики в системе инженерных и научных расчетов MATLAB 5.x.

5.1. Примеры решения уравнения Пуассона

Как было показано выше, многие задачи матфизики описываются уравнением Пуассона. К ним относятся задачи о стационарном распределении тепла, о диффузии вещества, о распределении электростатического поля в непроводящей среде при наличии электрических зарядов и многие другие.

5.1.1. Решение одномерного уравнения Пуассона методом конечных разностей

Решим одномерное уравнение Пуассона вида

$$\frac{\partial^2 u}{\partial x^2} = f(x), \quad (235)$$

где x – координата; $u(x)$ – искомая функция; $f(x)$ – некоторая непрерывная функция, на отрезке $[x_{min}, x_{max}]$ с граничными условиями Дирихле или Неймана в точках $x = x_{min}, x = x_{max}$.

Зададим на отрезке $[x_{min}, x_{max}]$ равномерную координатную сетку с шагом Δx :

$$\mathbf{x} = \{x_i | i=1, 2, \dots, n\}. \quad (236)$$

Граничные условия первого рода (Дирихле) для рассматриваемой задачи могут быть представлены в виде

$$u(x_1) = g_1; \quad (237)$$

$$u(x_n) = g_2, \quad (238)$$

где x_1, x_n – координаты граничных точек области $[x_{min}, x_{max}]$; g_1, g_2 – некоторые константы.

Граничные условия второго рода (Неймана) для рассматриваемой задачи могут быть представлены в виде

$$\left. \frac{du}{dx} \right|_{x_1} = g_1; \quad (239)$$

$$\left. \frac{du}{dx} \right|_{x_n} = g_2. \quad (240)$$

Проводя дискретизацию граничных условий Дирихле на равномерной координатной сетке (236) с использованием метода конечных разностей, получим

$$u_1 = g_1; \quad (241)$$

$$u_n = g_2, \quad (242)$$

где u_1, u_n – значения функции $u(x)$ в точках x_1, x_n , соответственно.

Проводя дискретизацию граничных условий Неймана на сетке (236), получим

$$\frac{u_2 - u_1}{\Delta x} = g_1; \quad (243)$$

$$\frac{u_n - u_{n-1}}{\Delta x} = g_2. \quad (244)$$

Проводя дискретизацию уравнения (235) для внутренних точек сетки, получим

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} = f_i, \quad i=2, \dots, n-1, \quad (245)$$

где f_i – значение функции $f(x)$ в точке сетки с координатой x_i .

Таким образом, в результате дискретизации получим систему линейных алгебраических уравнений размерности n , содержащую $n - 2$ уравнений вида (245) для внутренних точек области и уравнения (241) или (243) и (242) или (244) для двух граничных точек.

Ниже приведен один из вариантов функции для численного решения уравнения (235) с граничными условиями (237) – (240) на координатной сетке (236).

% Функция решения одномерного уравнения Пуассона

% $d^2u/dx^2=f(x,y)$

% с граничными условиями Дирихле и/или Неймана

function[x,u]=puass_1d(x0,xn,n,f,v1,g1,v2,g2)

% Входные параметры:

% x0 – начальная координата области решения;

% xn – конечная координата области решения;

% n – число точек координатной сетки;

% f – функция в правой части уравнения Пуассона,

% задаваемая строкой символов, заключенных

% в одинарные кавычки, например, 'exp(-x)+exp(-y)';

% v1 – параметр, значение которого определяет

% тип граничного условия на первой границе

% области $x = x(1)$ (1 – ГУ Дирихле, 2 – ГУ Неймана);

% g1 – граничное условие на первой границе в виде

% значения в одинарных кавычках, например, '0';

% v2 – параметр, значение которого определяет

% тип граничного условия на второй границе

% области $x = x(n)$ (1 – ГУ Дирихле, 2 – ГУ Неймана);

% g2 – граничное условие на второй границе в виде

% значения в одинарных кавычках.

% Выходные параметры:

% x – вектор-строка координатной сетки по оси x

% размерности 1 x n;

% U – матрица результирующих значений функции U

% в узлах координатной сетки размерности 1 x n.

```

% Функции и переменные по умолчанию

if exist('x0')==0
    x0=0;
end
if exist('xn')==0
    xn=5;
end
if exist('n')==0
    n=50;
end
if exist('f')==0
    f='2*sin(x^2)+10*cos(x^2)';
end
if exist('v1')==0
    v1=1;
end
if exist('g1')==0
    g1='0';
end
if exist('v2')==0
    v2=2;
end
if exist('g2')==0
    g2='-0.5';
end

% Задание равномерной координатной сетки

x=x0:(xn-x0)/(n-1):xn;    dx=x(2)-x(1);

% Вычисление значений функций, заданных символьно,
% в узлах координатной сетки

F=inline(f,'x');
G1=inline(g1,'x');
G2=inline(g2,'x');

% Задание матрицы коэффициентов СЛАУ размерности n x n,
% все элементы которой равны 0

a=zeros(n,n);

```

```

% Задание матрицы-строки свободных членов СЛАУ
% размерности 1 x n,
% все элементы которой равны 0

b=zeros(1,n);

% Определение коэффициентов и свободных членов СЛАУ,
% соответствующих граничным условиям и проверка
% корректности значений параметров v1, v2

b(1)=G1(x(1));
if v1==1
    a(1,1)=1;
elseif v1==2
    a(1,1)=-1/dx;
    a(1,2)=1/dx;
else
    error('Parameter v1 have incorrect value');
end
b(n)=G2(x(n));
if v2==1
    a(n,n)=1;
elseif v2==2
    a(n,n)=1/dx;
    a(n,n-1)=-1/dx;
else
    error('Parameter v2 have incorrect value');
end

% Определение коэффициентов и свободных членов СЛАУ,
% соответствующих внутренним точкам области

for i=2:n-1
    a(i,i)=-2/dx^2;
    a(i,i+1)=1/dx^2;
    a(i,i-1)=1/dx^2;
    b(i)=F(x(i));
end

% Решение СЛАУ

u=b/a';

```

```
% Построение графика искомой функции u(x,y)
```

```
plot(x,u)
xlabel('x')
ylabel('u')
grid on
```

Строки, начинающиеся символом %, являются комментариями. Приведенное описание необходимо сохранить в виде текстового файла с именем `puass_1d.m` и поместить его в каталог `/WORK`, находящийся в корневом каталоге системы MATLAB.

Вызов функции `puass_1d` может осуществляться следующими командами:

```
puass_1d;
x=puass_1d;
[x,u]=puass_1d;
[x,u]=puass_1d(x0,xn,n,f,v1,g1,v2,g2).
```

При использовании первой, второй или третьей команд функция будет выводить графически решение задачи при входных параметрах, принятых по умолчанию.

При вызове функции с помощью первой или второй команд без символа ; после текста команды, на экран монитора помимо графического представления решения будет выводиться вектор координат узлов сетки **x**. При использовании символа ; вектор координат узлов сетки выводиться не будет.

При использовании третьей или четвертой команд без символа ; после текста команды, на экран монитора будет выводиться график решения, вектор координат узлов сетки **x** и вектор значений искомой функции в узлах сетки **u**.

Например, при вызове функции командой

```
puass_1d;
```

на экране появится график решения задачи для функции $f(x)$ в правой части уравнения (235), определяемой выражением

$$f(x) = 2\sin(x^2) + 10\cos(x^2), \quad (246)$$

с граничными условиями

$$u(x_{\min}) = 0; \quad (247)$$

$$\left. \frac{du}{dx} \right|_{x_{\max}} = -0.5 \quad (248)$$

на границах $x_{\min} = 0$, $x_{\max} = 5$ на равномерной координатной сетке, содержащей 50 точек (рис. 21).

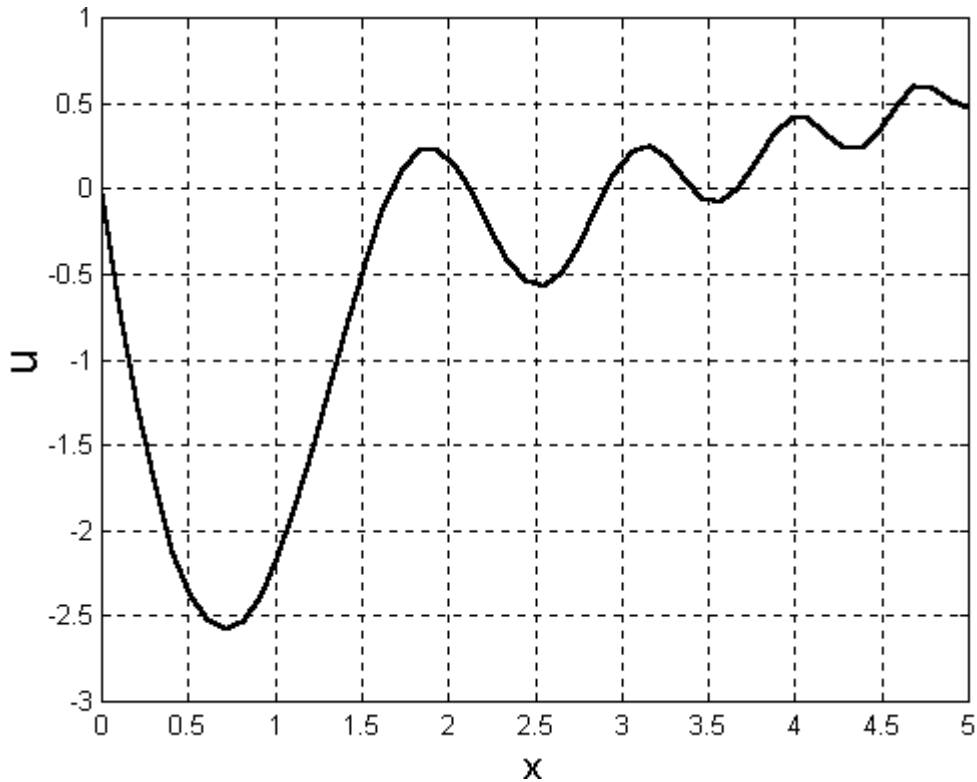


Рис. 21. Решение одномерного уравнения Пуассона при помощи функции *puass_1d* с параметрами по умолчанию

При вызове функции командой

```
[x,u]=puass_1d(-1,6,100,'sin(x)',1,'1',2,'-0.5');
```

на экране появится график решения задачи для функции $f(x)$ в правой части уравнения (235), определяемой выражением

$$f(x) = \sin(x), \quad (249)$$

с граничными условиями

$$u(x_{\min}) = 1; \quad (250)$$

$$\left. \frac{du}{dx} \right|_{x_{\max}} = -0.5 \quad (251)$$

на границах $x_{\min} = -1$, $x_{\max} = 6$ на равномерной координатной сетке, содержащей 100 точек (рис. 22).

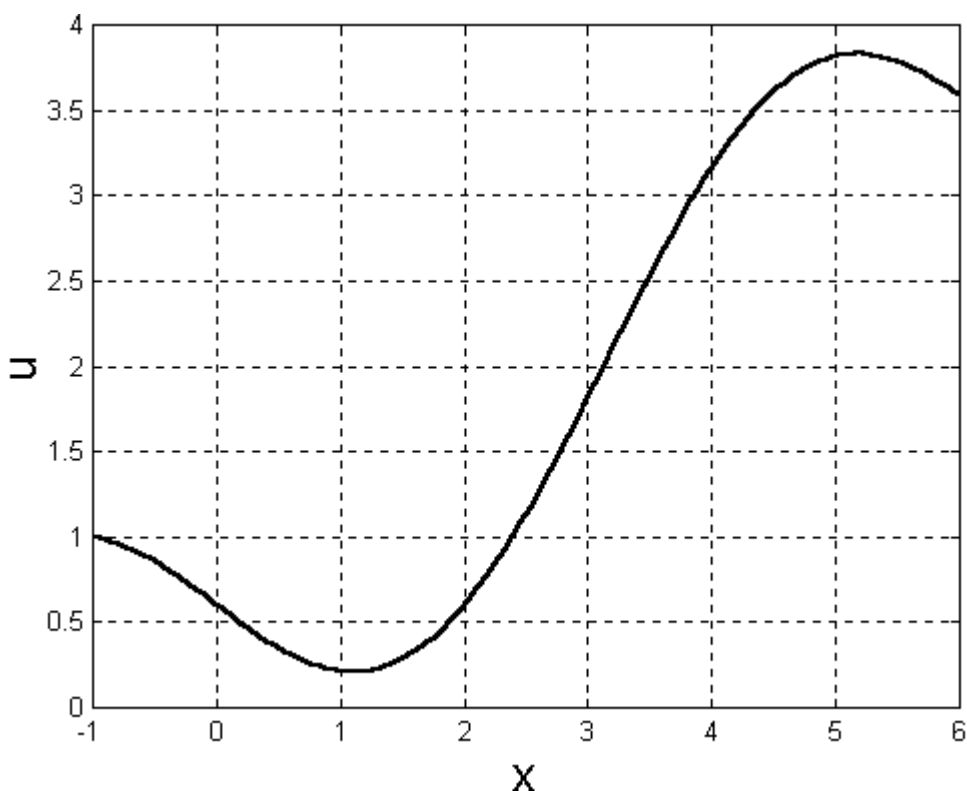


Рис. 22. Решение одномерного уравнения Пуассона при помощи функции `puass_1d` с параметрами, заданными в командной строке

При вызове функции `puass_1d` со списком входных параметров (в круглых скобках), этот список не обязательно должен быть полным. При отсутствии некоторых параметров функция использует их значения по умолчанию.

5.1.2. Решение двумерного уравнения Пуассона методом конечных разностей

Решим двумерное уравнение Пуассона вида

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y), \quad (252)$$

где x, y – координаты; $u(x, y)$ – искомая функция; $f(x, y)$ – некоторая непрерывная функция, на прямоугольной области с граничными условиями Дирихле или Неймана на границах $x = x_{min}$, $x = x_{max}$, $y = y_{min}$, $y = y_{max}$.

Зададим на отрезке $[x_{min}, x_{max}]$ равномерную координатную сетку с шагом Δx :

$$\mathbf{x} = \{x_i | i=1, 2, \dots, n\}, \quad (253)$$

а на отрезке $[y_{min}, y_{max}]$ равномерную координатную сетку с шагом Δy :

$$\mathbf{y} = \{y_j | j=1, 2, \dots, m\}. \quad (254)$$

Вектора, заданные выражениями (253) и (254), определяют на прямоугольной области двумерную равномерную сетку:

$$G = \{(x_i = i\Delta x, y_j = j\Delta y), | i = 1, 2, \dots, n, j = 1, 2, \dots, m \}.$$
 (255)

Граничные условия первого рода (Дирихле) для рассматриваемой задачи могут быть представлены в виде

$$u(x_1, y) = g_1(y);$$
 (256)

$$u(x_n, y) = g_2(y);$$
 (257)

$$u(x, y_1) = g_3(x);$$
 (258)

$$u(x, y_m) = g_4(x),$$
 (259)

где x_1, x_n – координаты граничных точек области x_{min}, x_{max} ; y_1, y_m – координаты граничных точек области y_{min}, y_{max} ; $g_1(y), g_2(y), g_3(x), g_4(x)$ – некоторые непрерывные функции соответствующих координат.

Граничные условия второго рода (Неймана) для рассматриваемой задачи могут быть представлены в виде

$$\left. \frac{\partial u}{\partial x} \right|_{x_1, y} = g_1(y);$$
 (260)

$$\left. \frac{\partial u}{\partial x} \right|_{x_n, y} = g_2(y);$$
 (261)

$$\left. \frac{\partial u}{\partial y} \right|_{x, y_1} = g_3(x);$$
 (262)

$$\left. \frac{\partial u}{\partial y} \right|_{x, y_m} = g_4(x).$$
 (263)

Проводя дискретизацию граничных условий Дирихле на равномерной координатной сетке (255) с использованием метода конечных разностей, получим

$$u_{1,j} = g_1(y_j);$$
 (264)

$$u_{n,j} = g_2(y_j);$$
 (265)

$$u_{i,1} = g_3(x_i);$$
 (266)

$$u_{i,m} = g_4(x_i), \quad (267)$$

где $u_{1,j}$, $u_{n,j}$, $u_{i,1}$, $u_{i,m}$ – значения функции $u(x, y)$ в точках (x_1, y_j) , (x_n, y_j) , (x_i, y_1) , (x_i, y_m) , соответственно.

Проводя дискретизацию граничных условий Неймана на сетке (255), получим

$$\frac{u_{2,j} - u_{1,j}}{\Delta x} = g_1(y_j); \quad (268)$$

$$\frac{u_{n,j} - u_{n-1,j}}{\Delta x} = g_2(y_j); \quad (269)$$

$$\frac{u_{i,2} - u_{i,1}}{\Delta y} = g_3(x_i); \quad (270)$$

$$\frac{u_{i,m} - u_{i,m-1}}{\Delta y} = g_4(x_i). \quad (271)$$

Проводя дискретизацию уравнения (252) для внутренних точек сетки, получим

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} = f_{i,j},$$

$$i=2, \dots, n-1; \quad j=2, \dots, m-1, \quad (272)$$

где $f_{i,j}$ – значение функции $f(x, y)$ в точке сетки с координатами (x_i, y_j) .

Таким образом, в результате дискретизации получим систему линейных алгебраических уравнений размерности $n \cdot m$, содержащую $(n-2)(m-2)$ уравнений вида (272) для внутренних точек области и $2n + 2(m-2)$ уравнений вида (264) или (268), (265) или (269), (266) или (270) и (267) или (271) для граничных точек.

Ниже приведен один из вариантов функции для численного решения уравнения (252) с граничными условиями (256) – (263) на координатной сетке (255).

```
% Функция решения двухмерного уравнения Пуассона
% d2u/dx2+d2u/dy2=f(x,y)
% на прямоугольной области с граничными условиями
% Дирихле и/или Неймана
```

```
function[x,y,U]=...
puass_2d(x0,xn,n,y0,ym,m,f,v1,g1,v2,g2,v3,g3,v4,g4)
% Входные параметры:
```

```

% x0 - начальная координата области решения по оси x;
% xn - конечная координата области решения по оси x;
% y0 - начальная координата области решения по оси y;
% ym - конечная координата области решения по оси y;
% n - число точек координатной сетки вдоль оси x;
% m - число точек координатной сетки вдоль оси y;
% f - функция в правой части уравнения Пуассона,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки, например, 'exp(-x)+exp(-y)';
% v1 - параметр, значение которого определяет
%      тип граничного условия на первой границе
%      области  $x = x(1)$  (1 - ГУ Дирихле, 2 - ГУ Неймана);
% g1 - функция в правой части граничного условия
%      на первой границе,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки;
% v2 - параметр, значение которого определяет
%      тип граничного условия на второй границе
%      области  $x = x(n)$  (1 - ГУ Дирихле, 2 - ГУ Неймана);
% g2 - функция в правой части граничного условия
%      на второй границе,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки;
% v3 - параметр, значение которого определяет
%      тип граничного условия на третьей границе
%      области  $y = y(1)$  (1 - ГУ Дирихле, 2 - ГУ Неймана);
% g3 - функция в правой части граничного условия
%      на третьей границе,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки;
% v4 - параметр, значение которого определяет
%      тип граничного условия на четвертой границе
%      области  $y = y(m)$  (1 - ГУ Дирихле, 2 - ГУ Неймана);
% g4 - функция в правой части граничного условия
%      на четвертой границе,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки.

% Выходные параметры:
% x - вектор-строка координатной сетки по оси x
%      размерности 1 x n;
% y - вектор-строка координатной сетки по оси y
%      размерности 1 x m;
% U - матрица результирующих значений функции U
%      в узлах координатной сетки размерности n x m.

```

% Функции и переменные по умолчанию

```

if exist('x0')==0
    x0=0;
end
if exist('xn')==0
    xn=1;
end
if exist('n')==0
    n=10;
end
if exist('y0')==0
    y0=0;
end
if exist('ym')==0
    ym=2;
end
if exist('m')==0
    m=20;
end
if exist('f')==0
    f='exp(-x)+exp(-y)';
end
if exist('v1')==0
    v1=1;
end
if exist('g1')==0
    g1='sin(y^2)';
end
if exist('v2')==0
    v2=1;
end
if exist('g2')==0
    g2='cos(3*y)';
end
if exist('v3')==0
    v3=2;
end
if exist('g3')==0
    g3='10*sin(x^2)';
end
if exist('v4')==0
    v4=2;
end

```

```

if exist('g4')==0
    g4='10*sin(6*x)';
end

% Задание равномерной координатной сетки

x=x0:(xn-x0)/(n-1):xn;    dx=x(2)-x(1);
y=y0:(ym-y0)/(m-1):ym;    dy=y(2)-y(1);

% Вычисление значений функций, заданных символьно,
% в узлах координатной сетки

F=inline(f,'x','y');
G1=inline(g1,'y');
G2=inline(g2,'y');
G3=inline(g3,'x');
G4=inline(g4,'x');

% Определение размерности СЛАУ

N=n*m;

% Задание матрицы коэффициентов СЛАУ размерности N x N,
% все элементы которой равны 0

a=zeros(N,N);

% Задание матрицы-строки свободных членов СЛАУ
% размерности 1 x N, все элементы которой равны 0

b=zeros(1,N);

% Определение коэффициентов и свободных членов СЛАУ,
% соответствующих граничным условиям и проверка
% корректности значений параметров v1, v2, v3, v4

for j=1:m
    b(j)=G1(y(j));
    if v1==1
        a(j,j)=1;
    elseif v1==2
        a(j,j)=-1/dx;
        a(j,m+j)=1/dx;
    else
        error('Parameter v1 have incorrect value');
    end
end

```

```

end
b(m*(n-1)+j)=G2(y(j));
if v2==1
    a(m*(n-1)+j,m*(n-1)+j)=1;
elseif v2==2
    a(m*(n-1)+j,m*(n-1)+j)=1/dx;
    a(m*(n-1)+j,m*(n-2)+j)=-1/dx;
else
    error('Parameter v2 have incorrect value');
end
end
for i=2:n-1
    b(m*(i-1)+1)=G3(x(i));
    if v3==1
        a(m*(i-1)+1,m*(i-1)+1)=1;
    elseif v3==2
        a(m*(i-1)+1,m*(i-1)+1)=-1/dy;
        a(m*(i-1)+1,m*(i-1)+2)=1/dy;
    else
        error('Parameter v3 have incorrect value');
    end
    b(m*(i-1)+m)=G4(x(i));
    if v4==1
        a(m*(i-1)+m,m*(i-1)+m)=1;
    elseif v4==2
        a(m*(i-1)+m,m*(i-1)+m)=1/dy;
        a(m*(i-1)+m,m*(i-1)+m-1)=-1/dy;
    else
        error('Parameter v4 have incorrect value');
    end
end
end

% Определение коэффициентов и свободных членов СЛАУ,
% соответствующих внутренним точкам области

for i=2:n-1
    for j=2:m-1
        a(m*(i-1)+j,m*(i-1)+j)=-2/dx^2-2/dy^2;
        a(m*(i-1)+j,m*(i)+j)=1/dx^2;
        a(m*(i-1)+j,m*(i-2)+j)=1/dx^2;
        a(m*(i-1)+j,m*(i-1)+j+1)=1/dy^2;
        a(m*(i-1)+j,m*(i-1)+j-1)=1/dy^2;
        b(m*(i-1)+j)=F(x(i),y(j));
    end
end
end

```

```
% Решение СЛАУ
```

```
u=b/a' ;
```

```
% Преобразование вектора-строки значений искомой функции  
% в узлах координатной сетки в матрицу размерности n x m,  
% удобную для представления результатов  
% в графическом виде
```

```
for i=1:n  
    for j=1:m  
        U(i,j)=u(m*(i-1)+j);  
    end  
end
```

```
% Построение графика искомой функции U(x,y)
```

```
surf(y,x,U)  
xlabel('y')  
ylabel('x')  
zlabel('U')  
grid on
```

Приведенное описание необходимо сохранить в виде текстового файла с именем `puass_2d.m` и поместить его в каталог `/WORK`, находящийся в корневом каталоге системы MATLAB.

Вызов функции `puass_2d` может осуществляться следующими командами:

```
puass_2d;  
x=puass_2d;  
[x,y]=puass_2d;  
[x,y,U]=puass_2d;  
[x,y,U]=puass_2d(x0,xn,n,y0,ym,m,f,v1,g1,v2,g2,v3,g3,v4,g4) .
```

При использовании первой, второй, третьей или четвертой команд функция будет выводить графически решение задачи при входных параметрах, принятых по умолчанию.

При вызове функции с помощью первой или второй команд без символа ; после текста команды, на экран монитора помимо графического представления решения будет выводиться вектор координат узлов сетки x . При использовании символа ; вектор координат узлов сетки выводиться не будет.

При использовании третьей команды без символа ; после текста команды, на экран монитора будет выводиться график решения, вектор координат узлов сетки x и вектор координат узлов сетки y .

При использовании третьей или четвертой команд без символа ; после текста команды, на экран монитора будет выводиться график решения, вектор координат узлов сетки x , вектор координат узлов сетки y и вектор значений искомой функции в узлах сетки U .

Например, при вызове функции командой

```
puass_2d;
```

на экране появится график решения задачи для функции $f(x, y)$ в правой части уравнения (252), определяемой выражением

$$f(x, y) = \exp(-x) + \exp(-y), \quad (273)$$

с граничными условиями

$$u(x_{\min}, y) = \sin(y^2); \quad (274)$$

$$u(x_{\max}, y) = \cos(3y); \quad (275)$$

$$\left. \frac{du}{dy} \right|_{x, y_1} = 10 \sin(x^2); \quad (276)$$

$$\left. \frac{du}{dy} \right|_{x, y_m} = 10 \sin(6x) \quad (277)$$

на границах $x_{\min} = 0$, $x_{\max} = 1$, $y_{\min} = 0$, $y_{\max} = 2$ на равномерной координатной сетке, содержащей 200 точек (10×20) (рис. 23).

При вызове функции командой

```
[x,y,U]=puass_2d(-1,2,20,0.35,5,30,'x^2+3*y',1,...  
'2*y*sin(3*y)',1,'3*y*cos(4*y)',2,'2-x',2,'3*x^2');
```

на экране появится график решения задачи для функции $f(x, y)$ в правой части уравнения (252), определяемой выражением

$$f(x, y) = x^2 + 3y, \quad (278)$$

с граничными условиями

$$u(x_{\min}, y) = 2y \cdot \sin(3y); \quad (279)$$

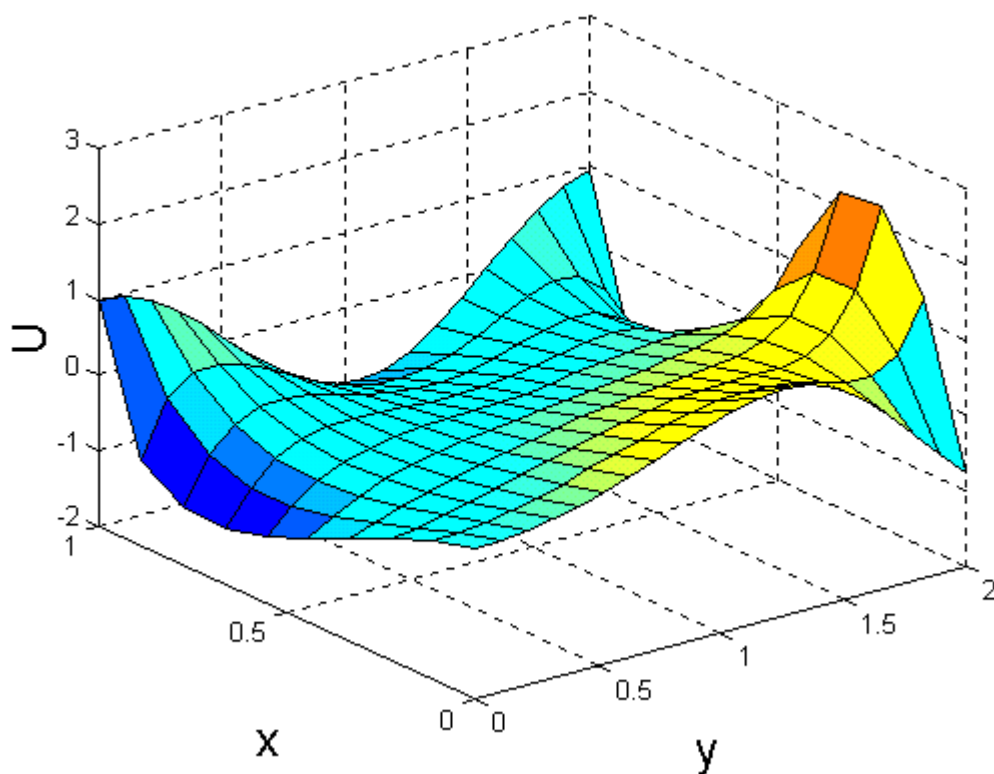


Рис. 23. Решение двухмерного уравнения Пуассона при помощи функции `puass_2d` с параметрами по умолчанию

$$u(x_{\max}, y) = 3y \cdot \cos(4y) ; \quad (280)$$

$$\left. \frac{du}{dy} \right|_{x, y_1} = 2 - x ; \quad (281)$$

$$\left. \frac{du}{dy} \right|_{x, y_m} = 3x^2 \quad (282)$$

на границах $x_{\min} = -1$, $x_{\max} = 2$, $y_{\min} = 0.35$, $y_{\max} = 5$ на равномерной координатной сетке, содержащей 600 точек (20×30) (рис. 24).

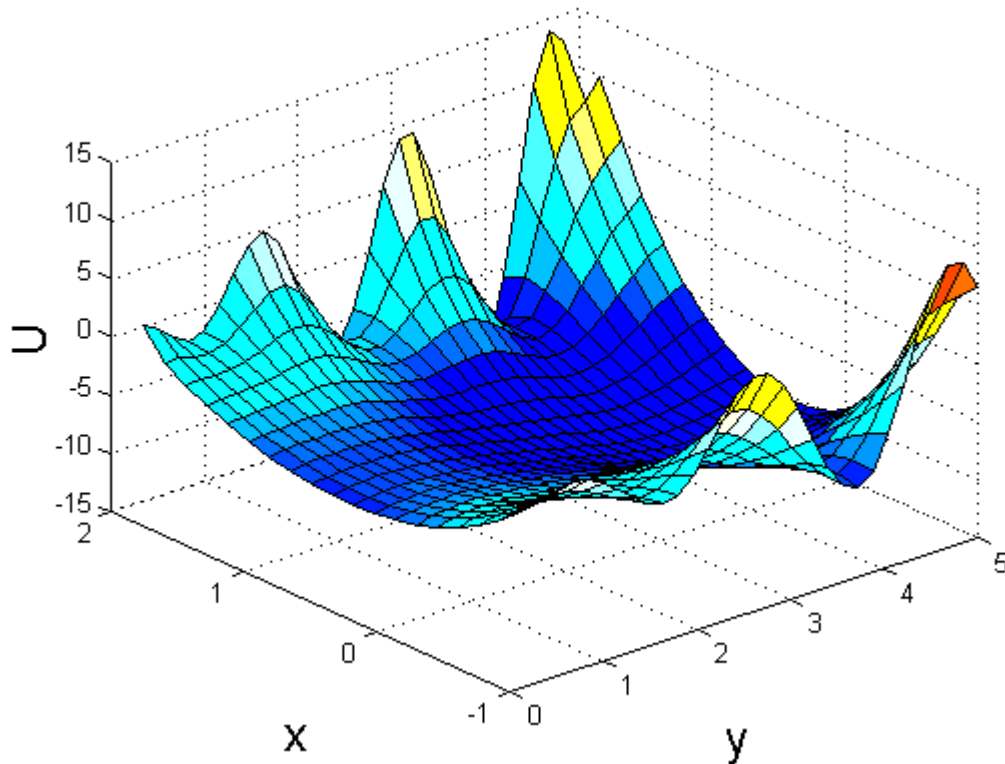


Рис. 24. Решение двухмерного уравнения Пуассона при помощи функции `puass_2d` с параметрами, заданными в командной строке

При вызове функции `puass_2d` список входных параметров не обязательно должен быть полным. При отсутствии некоторых параметров функция использует их значения по умолчанию.

5.1.3. Решение двухмерного уравнения Пуассона методом конечных элементов

Решим двухмерное уравнение Пуассона (252) с правой частью, определяемой выражением (273) и граничными условиями (274) – (277) на границах $x_{\min} = 0$, $x_{\max} = 1$, $y_{\min} = -1$, $y_{\max} = 1$ на триангулярной координатной сетке методом конечных элементов.

Для решения дифференциальных уравнений в частных производных методом конечных элементов в системе MATLAB 5.x воспользуемся приложением `pdetool`.

Запуск приложения осуществляется командой

`pdetool.`

При этом на экране монитора отображается главное окно приложения (рис. 25).

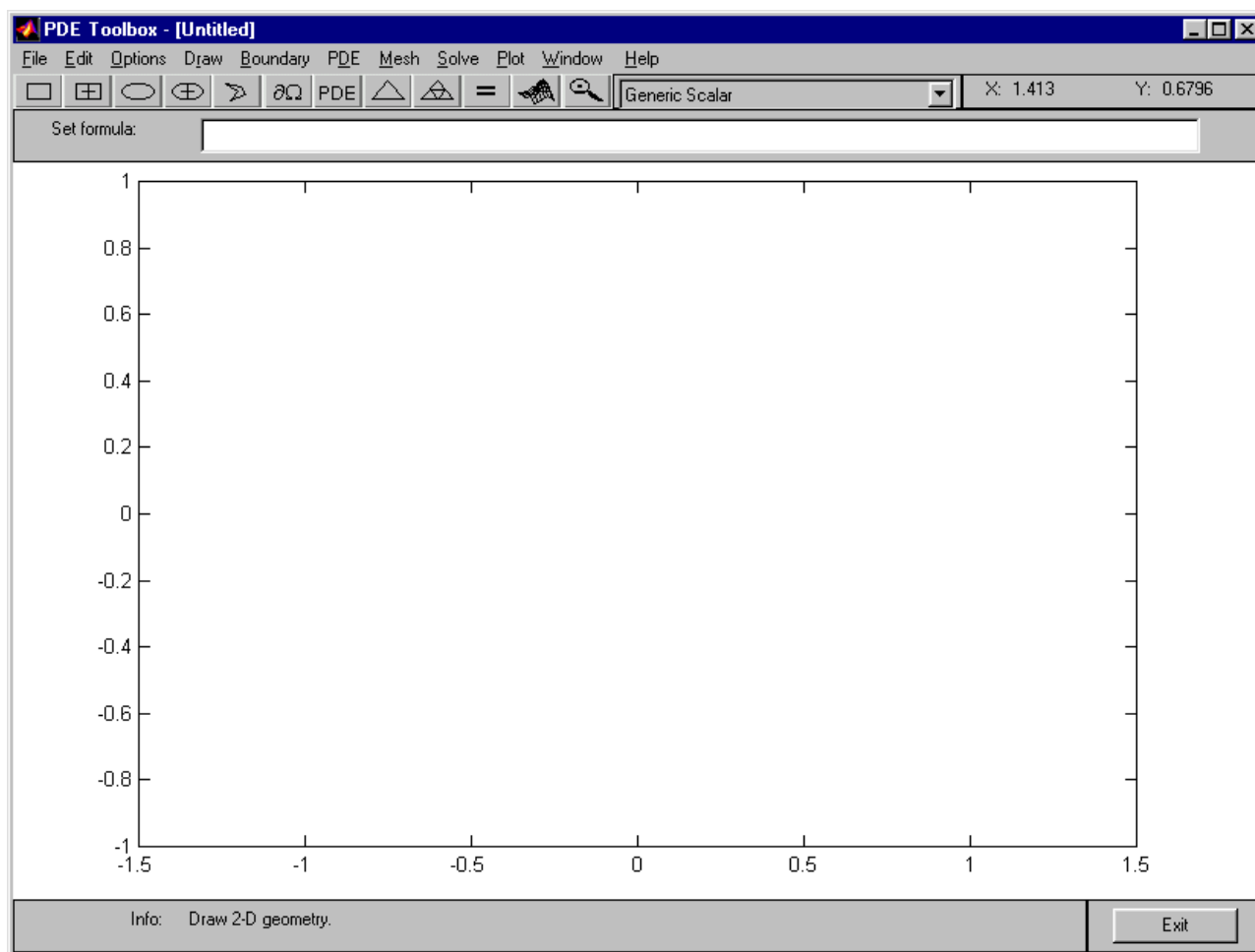



Рис. 25. Главное окно приложения *pdetool*

Для задания прямоугольной области решения задачи необходимо активизировать манипулятором «мышь» кнопку с символом , после чего навести курсор «мыши» на рабочее поле редактора, нажать левую кнопку «мыши» в левом верхнем углу $(0, 1)$ задаваемой прямоугольной области, переместить курсор в правый нижний угол $(1, -1)$ области, удерживая левую кнопку, после чего отпустить ее. Прямоугольная область будет зафиксирована (рис. 26).

При необходимости корректировки координат и размеров области, необходимо навести курсор «мыши» на изображение прямоугольника и дважды щелкнуть левой кнопкой. На экране появится окно редактирования параметров области с соответствующими полями (рис. 27). В первом и втором полях отображаются координаты левой нижней точки прямоугольника по осям x и y , соответственно. В третьем поле – ширина прямоугольника. В четвертом – высота. В пятом – условное обозначение.

Области решения произвольной формы могут быть заданы аналогичным образом с использованием кнопок с изображениями прямоугольников, эллипсов и полигона. При этом результирующая область может быть определена как объединение или разность нескольких областей простой формы. Для этого в поле *Set formula* указываются условные обозначения областей, связанные знаками «+» в случае объединения или «-» в случае разности (см. рис. 26).

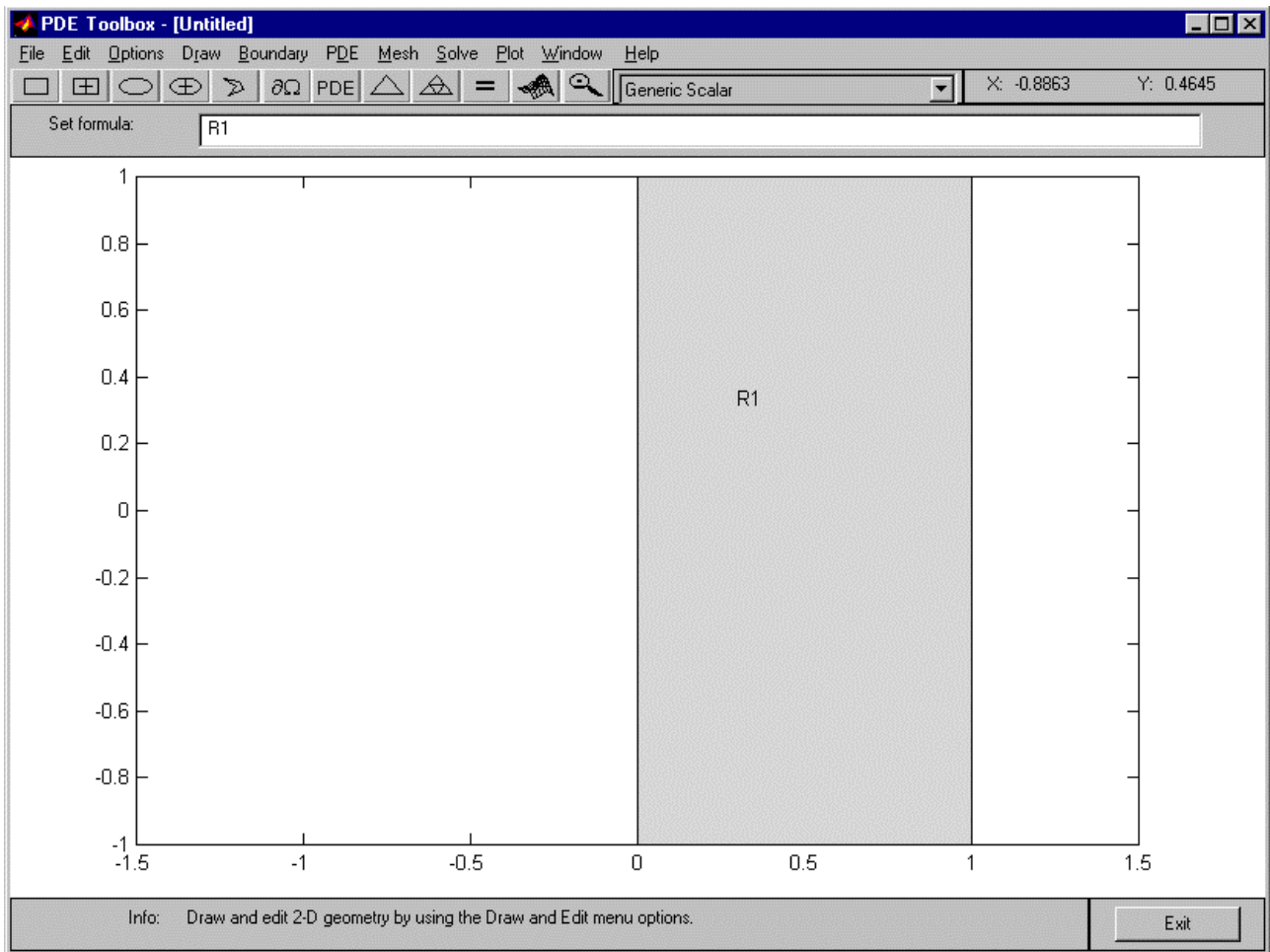


Рис. 26. Задание прямоугольной области решения задачи в приложении *pdetool*

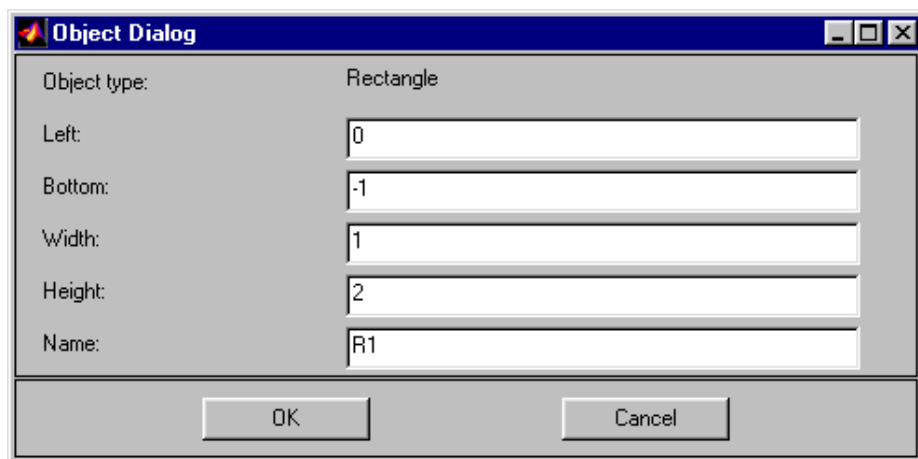


Рис. 27. Окно редактирования параметров прямоугольной области решения задачи в приложении *pdetool*

Для задания граничных условий необходимо активизировать манипулятором «мышь» кнопку с символом $\partial\Omega$, в результате чего окно приложения примет вид, показанный на рис. 28.

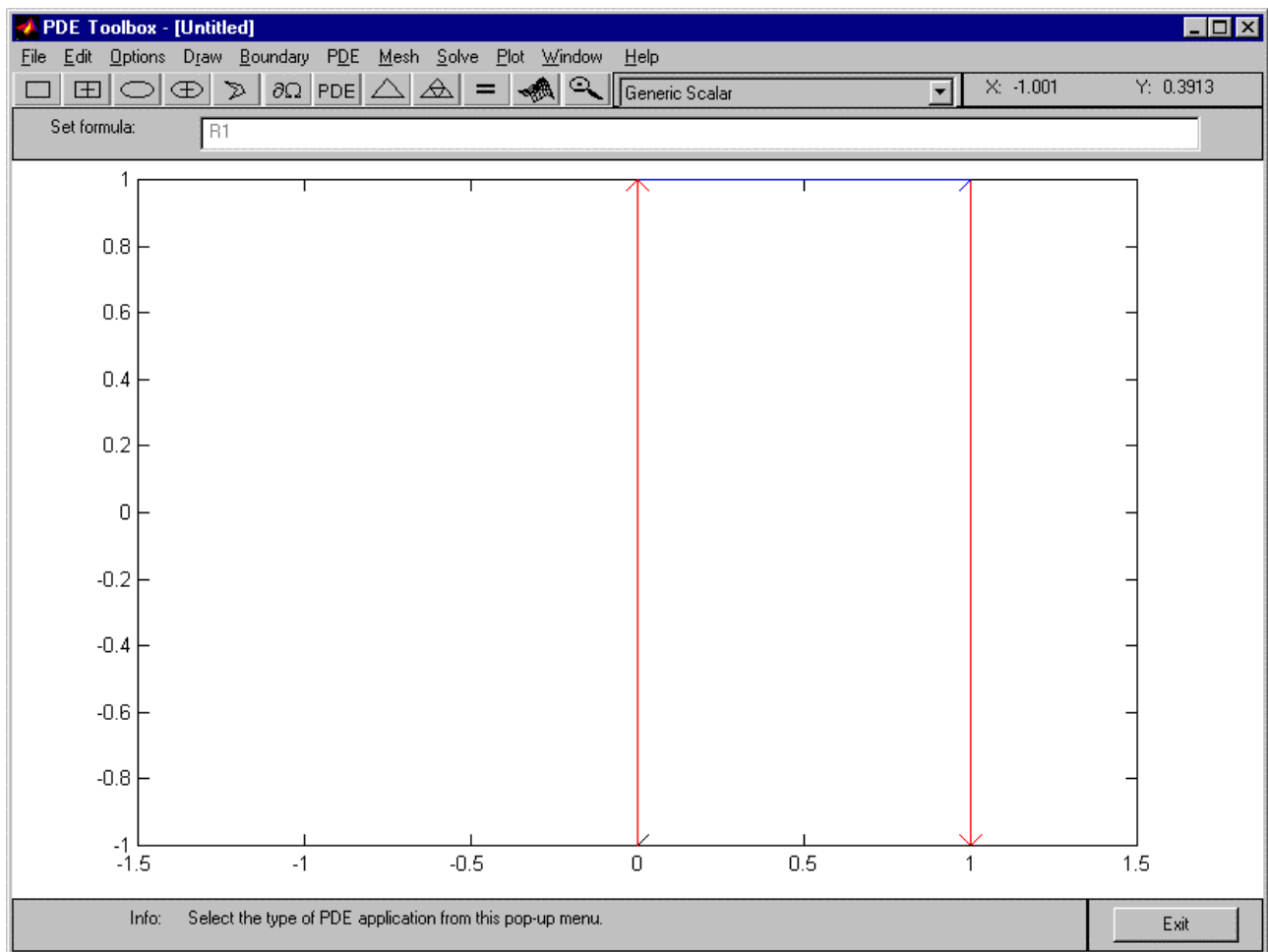


Рис. 28. Задание граничных условий в приложении pde tool

Все границы области показаны линиями со стрелками, причем по умолчанию на них заданы условия Дирихле (красные линии). Для редактирования граничных условий необходимо дважды щелкнуть левой кнопкой «мыши» на выбранной границе и внести соответствующие изменения в полях окна редактирования граничных условий (рис. 29 – 32).

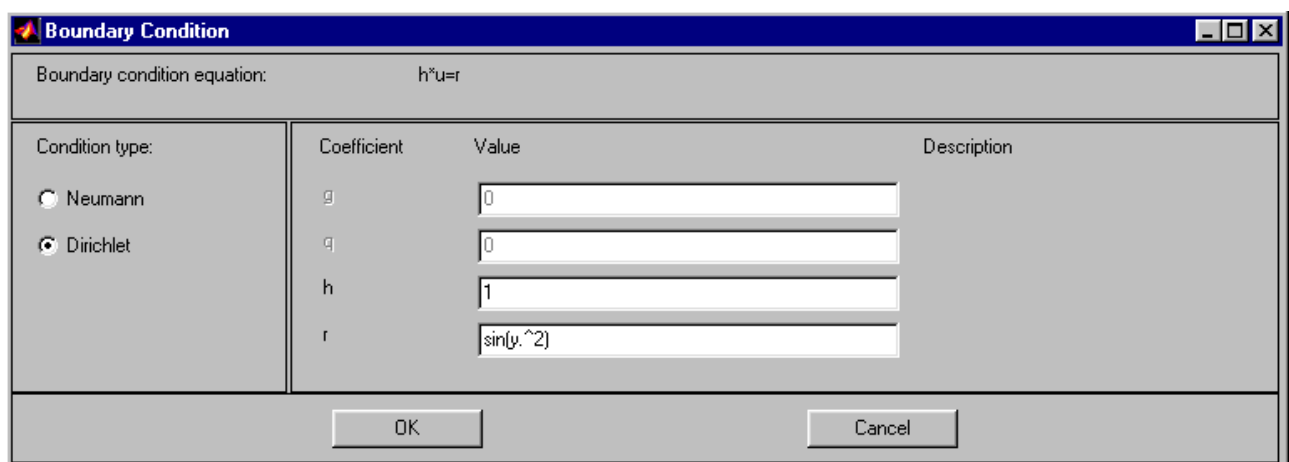


Рис. 29. Задание граничных условий на левой границе

Точки после символов x и y означают, что действие, знак которого стоит после точки, применяется к каждому элементу матрицы x или y .

Для редактирования вида дифференциального уравнения и ввода его функций и коэффициентов, необходимо активизировать манипулятором «мышь» кнопку с символами PDE, после чего внести соответствующие изменения в полях окна редактирования, показанного на рис. 33.

Boundary Condition dialog box. Boundary condition equation: $h*u=r$. Condition type: ☒ Dirichlet. Coefficients and values:

Coefficient	Value	Description
g	0	
q	0	
h	1	
r	$\cos(y.*3)$	

Buttons: OK, Cancel

Рис. 30. Задание граничных условий на правой границе

Boundary Condition dialog box. Boundary condition equation: $n*c*\text{grad}(u)+qu=g$. Condition type: ☒ Neumann. Coefficients and values:

Coefficient	Value	Description
g	$\sin(x.^2).*10$	
q	0	
h	1	
r	0	

Buttons: OK, Cancel

Рис. 31. Задание граничных условий на нижней границе

Boundary Condition dialog box. Boundary condition equation: $n*c*\text{grad}(u)+qu=g$. Condition type: ☒ Neumann. Coefficients and values:

Coefficient	Value	Description
g	$\sin(x.*6).*10$	
q	0	
h	1	
r	0	

Buttons: OK, Cancel

Рис. 32. Задание граничных условий на верхней границе

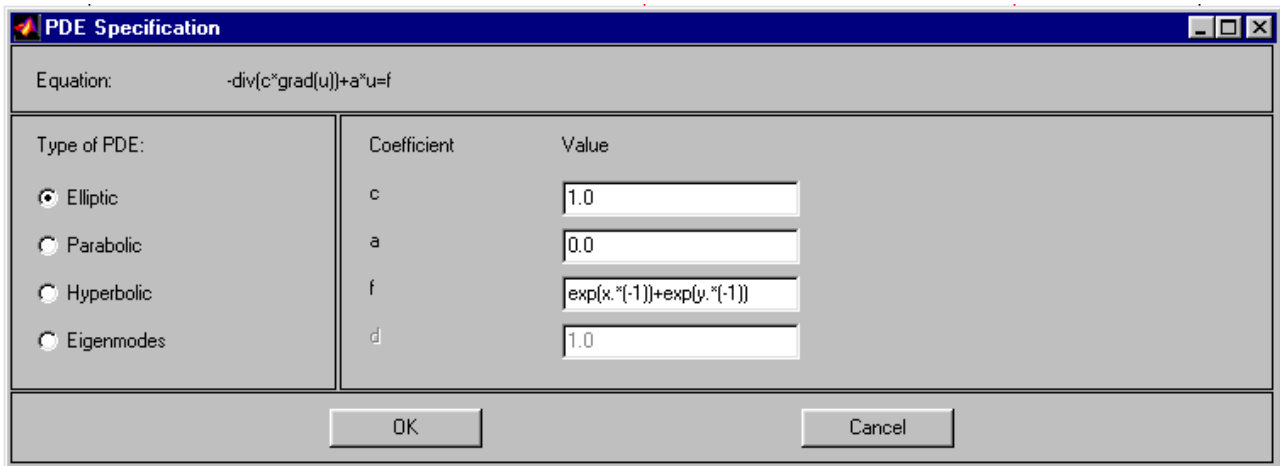



Рис. 33. Окно редактирования уравнения задачи в приложении pde tool

Формирование триангулярной сетки с использованием метода Делоне осуществляется активизацией кнопки с символом Δ (рис. 34). При необходимости увеличения числа узлов сетки следует активизировать кнопку  (рис. 35).

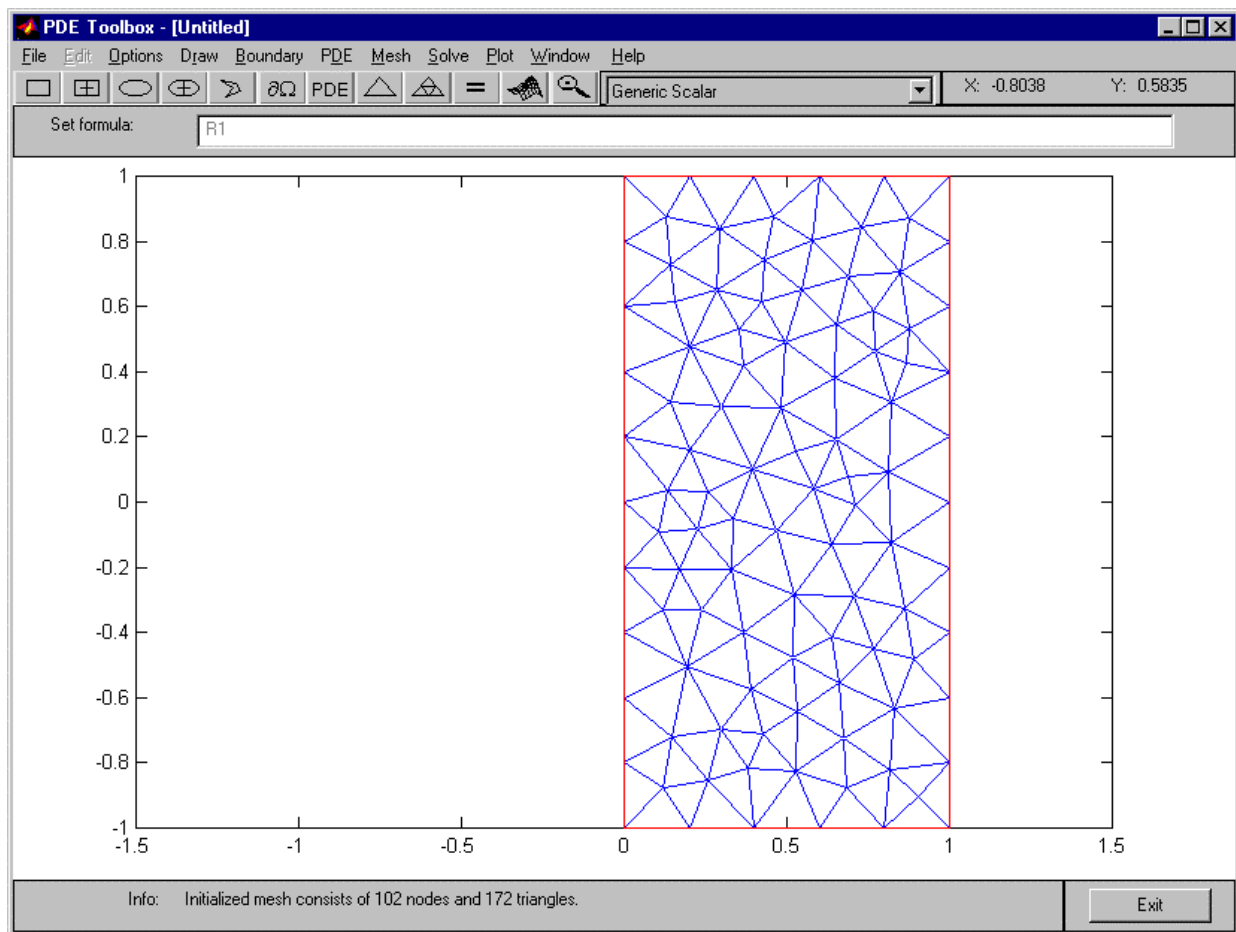


Рис. 34. Генерация триангулярной сетки с использованием метода Делоне в приложении pde tool

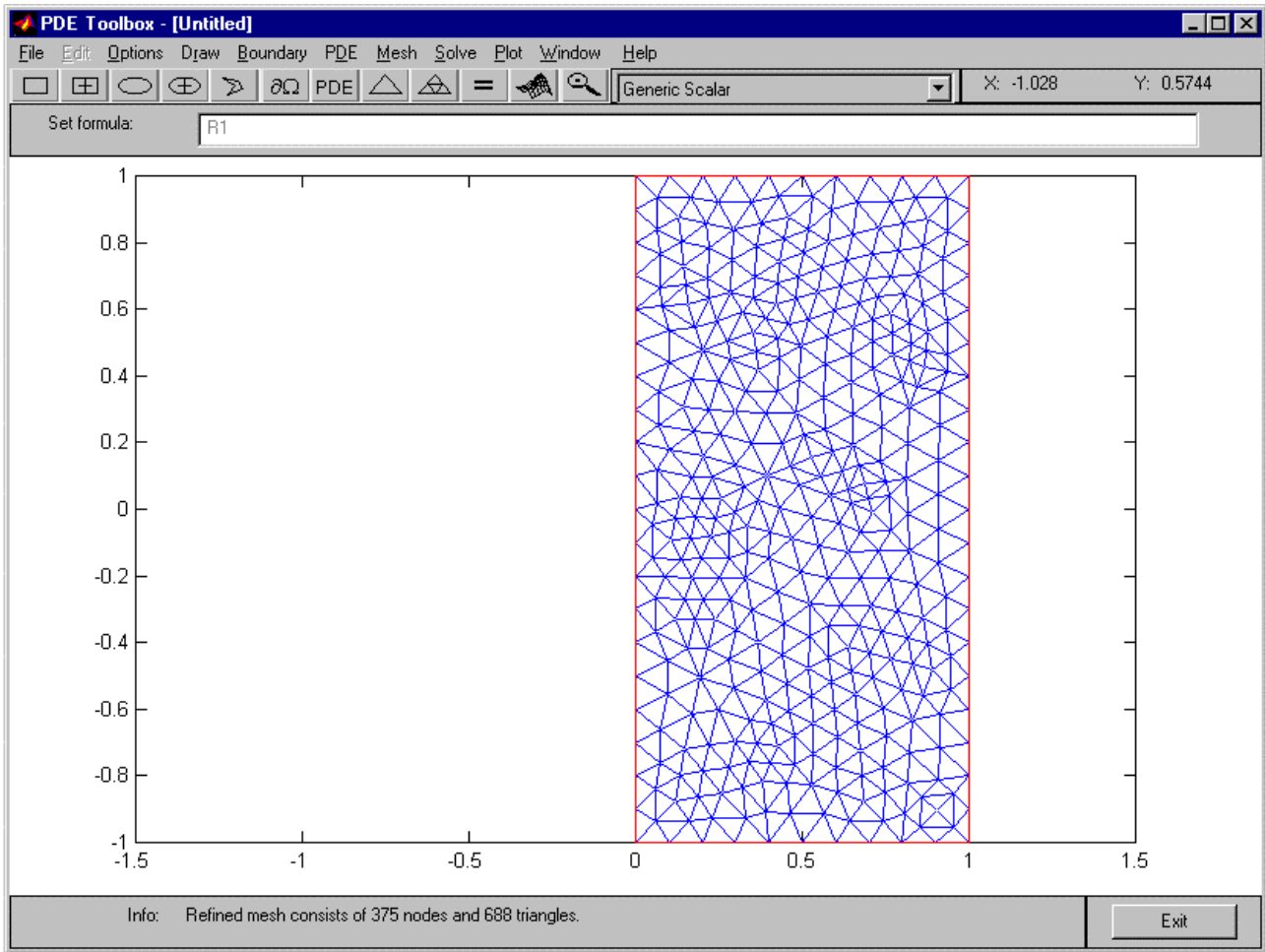



Рис. 35. Увеличение числа элементов сетки в приложении *pdetool*

Решение задачи осуществляется при активизации кнопки с символом « \Rightarrow ». По умолчанию значения функции решения выделены цветом (рис. 36).

Для графического вывода решения задачи в виде трехмерного (3D) изображения, следует активизировать кнопку , после чего в появившемся окне редактирования параметров изображения внести в соответствующие поля данные, как показано на рис. 37.

При активизации кнопки Plot на экран будет выведен график, показанный на рис. 38.

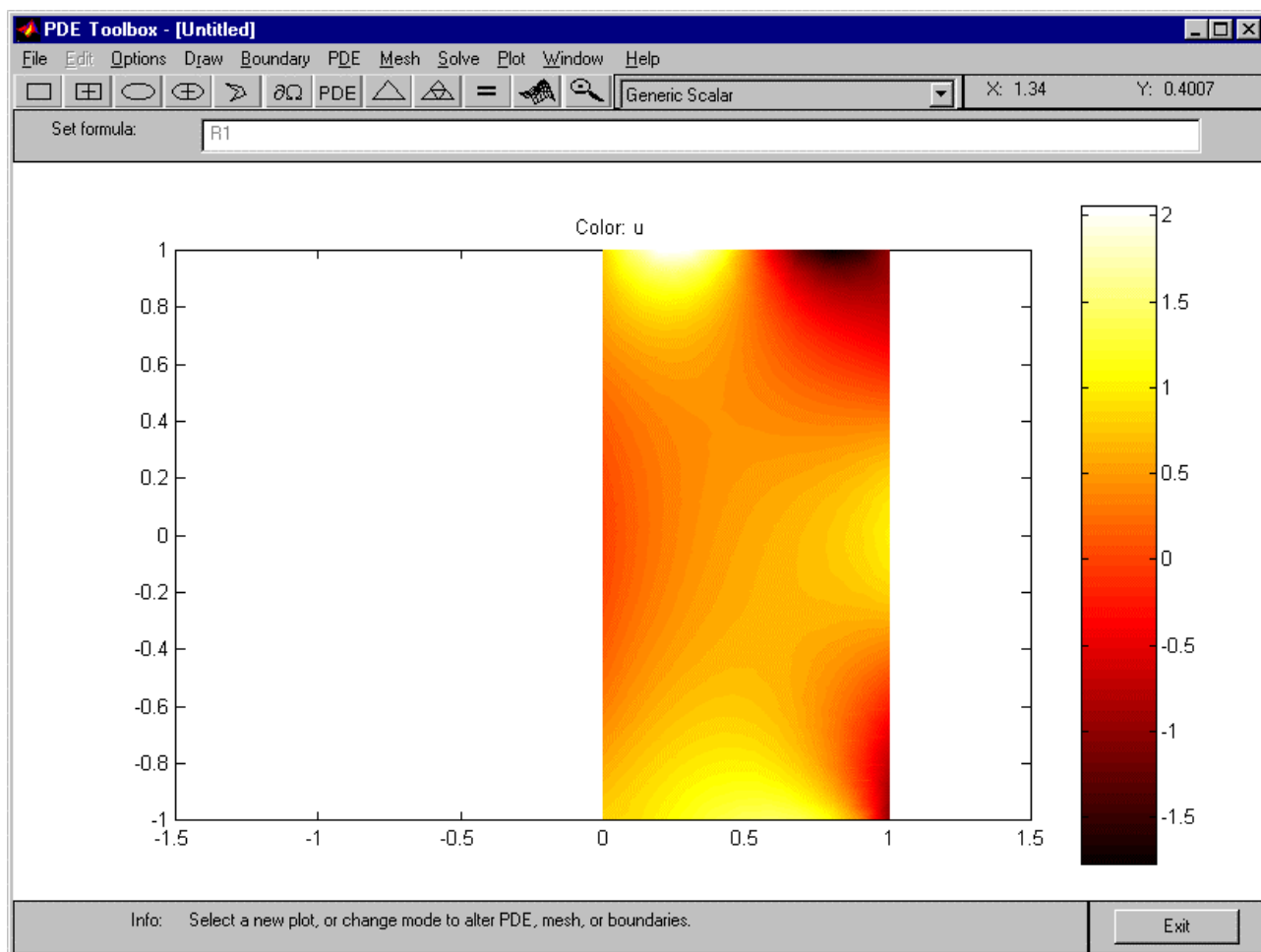


Рис. 36. Решение уравнения Пуассона в приложении *pdetool*

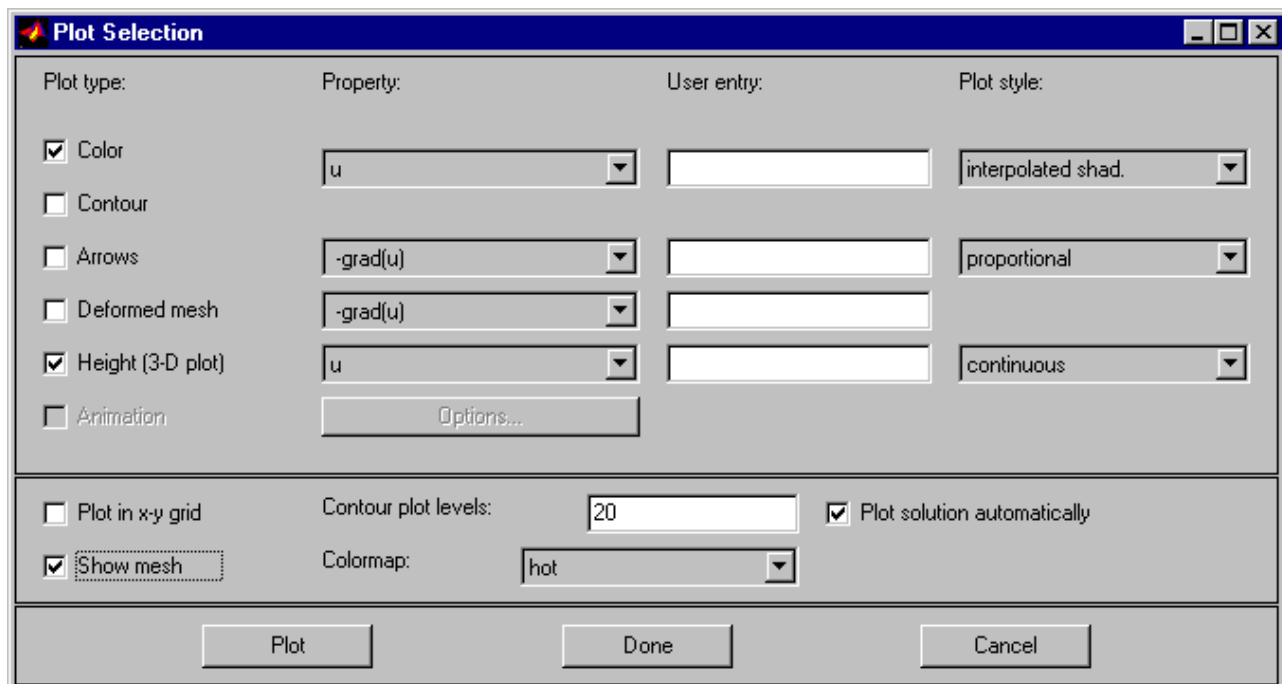


Рис. 37. Окно редактирования параметров графического представления решения в приложении *pdetool*

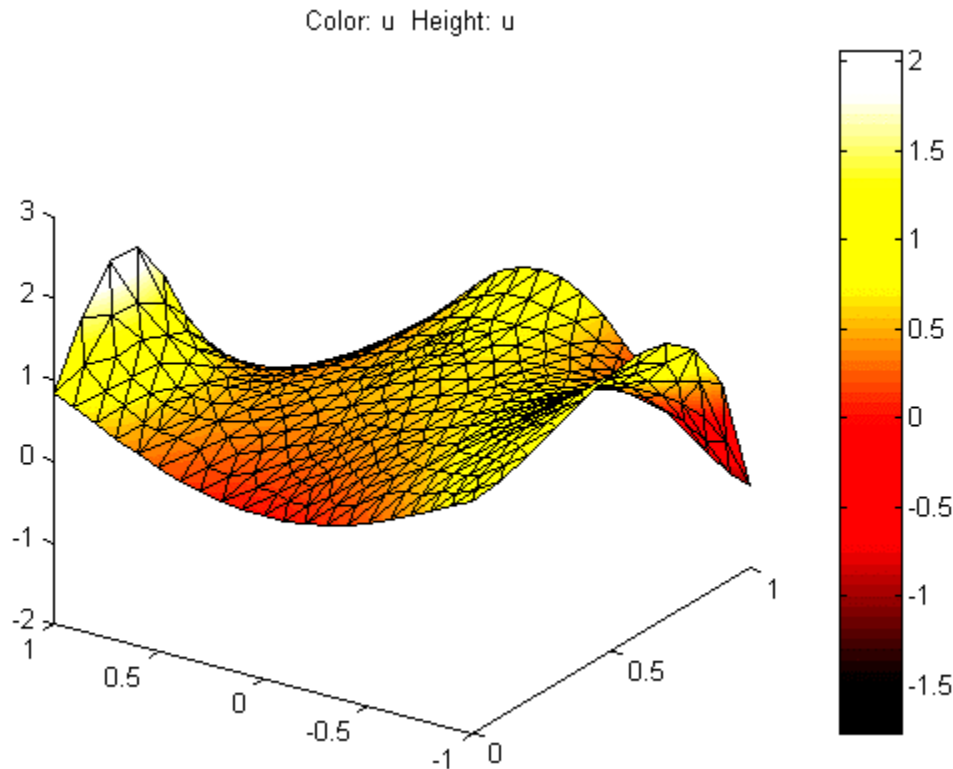


Рис. 38. Представление решения уравнения Пуассона в виде 3D-изображения

Более подробно с возможностями приложения pdetool можно ознакомиться в справке HELP системы MATLAB или в [10, 11].

5.2. Примеры решения уравнения теплопроводности

В качестве примера решения уравнений параболического типа рассмотрим нестационарное уравнение теплопроводности

$$\rho(x, y) C(x, y) \frac{\partial T}{\partial t} - \nabla(k(x, y) \cdot \nabla T) = f(x, y) \quad (283)$$

где t – время; x, y – координаты; $T(x, y)$ – искомая функция распределения абсолютной температуры по координатам; $\rho(x, y)$ – плотность вещества; $C(x, y)$ – удельная теплоемкость вещества; $k(x, y)$ – коэффициент теплопроводности вещества; $f(x, y)$ – плотность мощности источников тепла, на прямоугольной области с граничными условиями Дирихле или Неймана на границах $x = x_{min}, x = x_{max}, y = y_{min}, y = y_{max}$ и начальными условиями первого или второго рода на отрезке времени $[t_{min}, t_{max}]$.

Зададим на отрезке $[x_{min}, x_{max}]$ равномерную координатную сетку с шагом Δx :

$$\mathbf{x} = \{x_i | i=1, 2, \dots, n\}, \quad (284)$$

на отрезке $[y_{min}, y_{max}]$ равномерную координатную сетку с шагом Δy :

$$\mathbf{y} = \{y_j | j=1,2,\dots,m\}, \quad (285)$$

на отрезке $[t_{min}, t_{max}]$ равномерную сетку с шагом Δt :

$$\mathbf{t} = \{t_l | l=1,2,\dots,s\}. \quad (286)$$

Вектора, заданные выражениями (284) – (286), определяют на прямоугольной области равномерную пространственно-временную сетку:

$$G = \{(x_i = i\Delta x, y_j = j\Delta y, t_l = l\Delta t), | i = 1, 2, \dots, n, j = 1, 2, \dots, m, l = 1, 2, \dots, s\}. \quad (287)$$

Граничные условия первого рода (Дирихле) для рассматриваемой задачи могут быть представлены в виде

$$T(x_1, y, t) = g_1(y); \quad (288)$$

$$T(x_n, y, t) = g_2(y); \quad (289)$$

$$T(x, y_1, t) = g_3(x); \quad (290)$$

$$T(x, y_m, t) = g_4(x), \quad (291)$$

где x_1, x_n – координаты граничных точек области x_{min}, x_{max} ; y_1, y_m – координаты граничных точек области y_{min}, y_{max} ; $g_1(y), g_2(y), g_3(x), g_4(x)$ – некоторые непрерывные функции соответствующих координат.

Граничные условия второго рода (Неймана) для рассматриваемой задачи могут быть представлены в виде

$$\left. \frac{\partial T}{\partial x} \right|_{x_1, y, t} = g_1(y); \quad (292)$$

$$\left. \frac{\partial T}{\partial x} \right|_{x_n, y, t} = g_2(y); \quad (293)$$

$$\left. \frac{\partial T}{\partial y} \right|_{x, y_1, t} = g_3(x); \quad (294)$$

$$\left. \frac{\partial T}{\partial y} \right|_{x, y_m, t} = g_4(x). \quad (295)$$

Начальные условия первого рода для рассматриваемой задачи могут быть представлены в виде

$$T(x, y, t_1) = g_t(x, y), \quad (296)$$

где t_1 – начальный момент времени; $g_t(x, y)$ – некоторая непрерывная функция соответствующих координат.

Начальные условия второго рода для рассматриваемой задачи могут быть представлены в виде

$$\left. \frac{\partial T}{\partial t} \right|_{x, y, t_1} = g_t(x, y). \quad (297)$$

Проводя дискретизацию граничных условий Дирихле на равномерной сетке (287) с использованием метода конечных разностей, получим

$$T_{1,j,l} = g_1(y_j); \quad (298)$$

$$T_{n,j,l} = g_2(y_j); \quad (299)$$

$$T_{i,1,l} = g_3(x_i); \quad (300)$$

$$T_{i,m,l} = g_4(x_i), \quad (301)$$

где $T_{1,j,l}$, $T_{n,j,l}$, $T_{i,1,l}$, $T_{i,m,l}$ – значения функции $T(x, y, t)$ в точках (x_1, y_j, t_l) , (x_n, y_j, t_l) , (x_i, y_1, t_l) , (x_i, y_m, t_l) , соответственно.

Проводя дискретизацию граничных условий Неймана на сетке (287), получим

$$\frac{T_{2,j,l} - T_{1,j,l}}{\Delta x} = g_1(y_j); \quad (302)$$

$$\frac{T_{n,j,l} - T_{n-1,j,l}}{\Delta x} = g_2(y_j); \quad (303)$$

$$\frac{T_{i,2,l} - T_{i,1,l}}{\Delta y} = g_3(x_i); \quad (304)$$

$$\frac{T_{i,m,l} - T_{i,m-1,l}}{\Delta y} = g_4(x_i). \quad (305)$$

Проводя дискретизацию начальных условий первого рода на равномерной сетке (287), получим

$$T_{i,j,1} = g_t(x_i, y_j), \quad (306)$$

где $T_{i,j,1}$ – значения функции $T(x, y, t)$ в точке (x_i, y_j, t_1) .

Проводя дискретизацию начальных условий второго рода на сетке (287), получим

$$\frac{T_{i,j,2} - T_{i,j,1}}{\Delta t} = g_t(x_i, y_j). \quad (307)$$

Проводя дискретизацию уравнения (283) для внутренних точек сетки, получим

$$\begin{aligned} & \rho_{i,j,l} C_{i,j,l} \frac{T_{i,j,l} - T_{i,j,l-1}}{\Delta t} - \\ & - \frac{1}{\Delta x^2} [k_{i,j,l} (T_{i+1,j,l} - T_{i,j,l}) - k_{i-1,j,l} (T_{i,j,l} - T_{i-1,j,l})] - \\ & - \frac{1}{\Delta y^2} [k_{i,j,l} (T_{i,j+1,l} - T_{i,j,l}) - k_{i,j-1,l} (T_{i,j,l} - T_{i,j-1,l})] = f_{i,j,l} \end{aligned} \quad (308)$$

где $f_{i,j,l}$ – значение функции $f(x, y, t)$ в точке сетки с координатами (x_i, y_j, t_l) .

Таким образом, в результате дискретизации получим систему линейных алгебраических уравнений размерности $n \cdot m \cdot s$.

Ниже приведен один из вариантов функции для численного решения уравнения (283) с граничными условиями (288) – (295) и начальными условиями (296) или (297) на равномерной сетке (287) с подробными комментариями.

```
% Функция решения двумерного нестационарного
% уравнения теплопроводности
% r(x,y)C(x,y)dT/dt-d/dx(k(x,y)dT/dx)-
% - d/dy(k(x,y)dT/dy)=f(x,y)
% на прямоугольной области с граничными условиями
% Дирихле и/или Неймана
```

```
function[x,y,t,T]= ...
termo_2d(t0,ts,s,x0,xn,n,y0,ym,m,r,c,k,f, ...
vt,gt1,v1,g1,v2,g2,v3,g3,v4,g4)
```

```
% Входные параметры:
% t0 - начальный момент времени, с;
% ts - конечный момент времени, с;
% x0 - начальная координата области решения по оси x, м;
```

```

% xn - конечная координата области решения по оси x, м;
% y0 - начальная координата области решения по оси y, м;
% ym - конечная координата области решения по оси y, м;
% n - число точек координатной сетки вдоль оси x;
% m - число точек координатной сетки вдоль оси y;
% s - число точек сетки вдоль оси времени t;
% r - функция плотности вещества,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки, например, 'x+2*y', кг/м^3;
% c - функция теплоемкости вещества,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки, Дж/(кг К);
% k - функция теплопроводности вещества,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки, Вт/(м К);
% f - функция плотности мощности источников тепла,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки, Вт/м^3;
% vt - параметр, значение которого определяет
%      тип начального условия
%      (1 - Дирихле, 2 - Неймана);
% gt1 - функция в правой части начального условия,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки, К или К/с;
% v1 - параметр, значение которого определяет
%      тип граничного условия на первой границе
%      области x = x(1) (1 - ГУ Дирихле, 2 - ГУ Неймана);
% g1 - функция в правой части граничного условия
%      на первой границе,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки, К или К/м;
% v2 - параметр, значение которого определяет
%      тип граничного условия на второй границе
%      области x = x(n) (1 - ГУ Дирихле, 2 - ГУ Неймана);
% g2 - функция в правой части граничного условия
%      на второй границе,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки, К или К/м;
% v3 - параметр, значение которого определяет
%      тип граничного условия на третьей границе
%      области y = y(1) (1 - ГУ Дирихле, 2 - ГУ Неймана);
% g3 - функция в правой части граничного условия
%      на третьей границе,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки, К или К/м;

```

```

% v4 - параметр, значение которого определяет
%      тип граничного условия на четвертой границе
%      области  $y = y(m)$  (1 - ГУ Дирихле, 2 - ГУ Неймана);
% g4 - функция в правой части граничного условия
%      на четвертой границе,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки, K или K/м.

% Выходные параметры:
% x - вектор-строка координатной сетки по оси x
%     размерности 1 x n;
% y - вектор-строка координатной сетки по оси y
%     размерности 1 x m;
% t - вектор-строка сетки по оси времени
%     размерности 1 x s;
% T - матрица результирующих значений температуры
%     в узлах координатной сетки размерности n x m x s.

% Функции и переменные по умолчанию

if exist('t0')==0
    t0=0;
end
if exist('ts')==0
    ts=6;
end
if exist('s')==0
    s=6;
end
if exist('x0')==0
    x0=0;
end
if exist('xn')==0
    xn=5e-2;
end
if exist('n')==0
    n=10;
end
if exist('y0')==0
    y0=0;
end
if exist('ym')==0
    ym=8e-2;
end
if exist('m')==0

```



```

    m=20;
end
if exist('r')==0
    r='2e3';
end
if exist('c')==0
    c='100';
end
if exist('k')==0
    k='20';
end
if exist('f')==0
    f='0';
end
if exist('vt')==0
    vt=1;
end
if exist('gt1')==0
    gt1='10*(sign(1e2*x-2)-sign(1e2*x-3))+ ...
        sign(1e2*y-3)-sign(1e2*y-5))+300';
end
if exist('v1')==0
    v1=1;
end
if exist('g1')==0
    g1='300';
end
if exist('v2')==0
    v2=1;
end
if exist('g2')==0
    g2='300';
end
if exist('v3')==0
    v3=1;
end
if exist('g3')==0
    g3='300';
end
if exist('v4')==0
    v4=1;
end
if exist('g4')==0
    g4='300';
end

```

```
% Задание равномерной сетки
```

```
x=x0:(xn-x0)/(n-1):xn;    dx=x(2)-x(1);
y=y0:(ym-y0)/(m-1):ym;    dy=y(2)-y(1);
t=t0:(ts-t0)/(s-1):ts;    dt=t(2)-t(1);
```

```
% Вычисление значений функций, заданных символьно,
% в узлах координатной сетки
```

```
F=inline(f, 'x', 'y');
R=inline(r, 'x', 'y');
C=inline(c, 'x', 'y');
K=inline(k, 'x', 'y');
GT=inline(gt1, 'x', 'y');
G1=inline(g1, 'y');
G2=inline(g2, 'y');
G3=inline(g3, 'x');
G4=inline(g4, 'x');
```

```
% Определение размерности СЛАУ
```

```
N=s*n*m;
```

```
% Задание матрицы коэффициентов СЛАУ размерности N x N,
% все элементы которой равны 0
```

```
a=zeros(N,N);
```

```
% Задание матрицы-строки свободных членов СЛАУ
% размерности 1 x N, все элементы которой равны 0
```

```
b=zeros(1,N);
```

```
% Определение коэффициентов и свободных членов СЛАУ,
% соответствующих граничным условиям и проверка
% корректности значений параметров vt, v1, v2, v3, v4
```

```
for i=1:n
    for j=1:m
        b(m*(i-1)+j)=GT(x(i),y(j));
        if vt==1
            a(m*(i-1)+j,m*(i-1)+j)=1;
        elseif vt==2
            a(m*(i-1)+j,m*(i-1)+j)=-1/dt;
```

```

        a(m*(i-1)+j,n*m+m*(i-1)+j)=1/dt;
    else
        error('Parameter vt have incorrect value');
    end
end
end
for l=1:s
    for j=1:m
        b(n*m*(l-1)+j)=G1(y(j));
        if v1==1
            a(n*m*(l-1)+j,n*m*(l-1)+j)=1;
        elseif v1==2
            a(n*m*(l-1)+j,n*m*(l-1)+j)=-1/dx;
            a(n*m*(l-1)+j,n*m*(l-1)+m+j)=1/dx;
        else
            error('Parameter v1 have incorrect value');
        end
        b(n*m*(l-1)+m*(n-1)+j)=G2(y(j));
        if v2==1
            a(n*m*(l-1)+m*(n-1)+j,n*m*(l-1)+m*(n-1)+j)=1;
        elseif v2==2
            a(n*m*(l-1)+m*(n-1)+j,n*m*(l-1)+m*(n-1)+j)=1/dx;
            a(n*m*(l-1)+m*(n-1)+j,n*m*(l-1)+m*(n-2)+j)= ...
                -1/dx;
        else
            error('Parameter v2 have incorrect value');
        end
    end
end
for i=2:n-1
    b(n*m*(l-1)+m*(i-1)+1)=G3(x(i));
    if v3==1
        a(n*m*(l-1)+m*(i-1)+1,n*m*(l-1)+m*(i-1)+1)=1;
    elseif v3==2
        a(n*m*(l-1)+m*(i-1)+1,n*m*(l-1)+m*(i-1)+1)= ...
            -1/dy;
        a(n*m*(l-1)+m*(i-1)+1,n*m*(l-1)+m*(i-1)+2)=1/dy;
    else
        error('Parameter v3 have incorrect value');
    end
    b(n*m*(l-1)+m*(i-1)+m)=G4(x(i));
    if v4==1
        a(n*m*(l-1)+m*(i-1)+m,n*m*(l-1)+m*(i-1)+m)=1;
    elseif v4==2
        a(n*m*(l-1)+m*(i-1)+m,n*m*(l-1)+m*(i-1)+m)=1/dy;
        a(n*m*(l-1)+m*(i-1)+m,n*m*(l-1)+m*(i-1)+m-1)= ...

```

```

        -1/dy;
    else
        error('Parameter v4 have incorrect value');
    end
end
end

% Определение коэффициентов и свободных членов СЛАУ,
% соответствующих внутренним точкам области

for l=2:s
    for i=2:n-1
        for j=2:m-1
            a(n*m*(l-1)+m*(i-1)+j,n*m*(l-1)+m*(i-1)+j)=...
                +R(x(i),y(j))*C(x(i),y(j))/dt+...
                (K(x(i),y(j))+K(x(i-1),y(j)))/dx^2+...
                (K(x(i),y(j))+K(x(i),y(j-1)))/dy^2;
            a(n*m*(l-1)+m*(i-1)+j,n*m*(l-1)+m*i+j)=...
                -K(x(i),y(j))/dx^2;
            a(n*m*(l-1)+m*(i-1)+j,n*m*(l-1)+m*(i-2)+j)=...
                -K(x(i-1),y(j))/dx^2;
            a(n*m*(l-1)+m*(i-1)+j,n*m*(l-1)+m*(i-1)+j+1)=...
                -K(x(i),y(j))/dy^2;
            a(n*m*(l-1)+m*(i-1)+j,n*m*(l-1)+m*(i-1)+j-1)=...
                -K(x(i),y(j-1))/dy^2;
            a(n*m*(l-1)+m*(i-1)+j,n*m*(l-2)+m*(i-1)+j)=...
                -R(x(i),y(j))*C(x(i),y(j))/dt;
            b(n*m*(l-1)+m*(i-1)+j)=F(x(i),y(j));
        end
    end
end

% Решение СЛАУ

u=b/a';

% Преобразование вектора-строки значений искомой функции
% в узлах координатной сетки в матрицу размерности n x m,
% удобную для представления результатов
% в графическом виде

for l=1:s
    for i=1:n
        for j=1:m
            T(i,j,l)=u(n*m*(l-1)+m*(i-1)+j);

```

```

        end
    end
end

% Построение графиков искомой функции T(x,y,t)

for l=1:s
    figure
    surf(y,x,T(:,:,l))
    xlabel('y, м')
    ylabel('x, м')
    zlabel('T, К')
    grid on
    colormap('cool')
    axis([min(y) max(y) min(x) max(x) ...
        min(min(T(:,:,l))) max(max(T(:,:,l)))])
    pause(0.1)
    M(l)=getframe;
end

figure
movie(M,10,3)

```

Приведенное описание необходимо сохранить в виде текстового файла с именем `termo_2d.m` и поместить его в каталог `/WORK`, находящийся в корневом каталоге системы MATLAB.

Вызов функции `termo_2d` может осуществляться следующими командами:

```

termo_2d;
x=termo_2d;
[x,y]=termo_2d;
[x,y,t]=termo_2d;
[x,y,t,T]=termo_2d(t0,ts,s,x0,xn,n,y0,ym,m,r,c,k,f, ...
    vt,gt1,v1,g1,v2,g2,v3,g3,v4,g4).

```

При использовании первой, второй, третьей или четвертой команд функция будет выводить графически решение задачи при входных параметрах, принятых по умолчанию.

Графики распределений температуры по координатам в различные моменты времени будут выводиться в отдельных окнах. После вывода всех графиков в новом окне будет показано изменение температуры во всех точках координатной сетки в динамическом режиме (`movie`).

Правила вывода информации при вызове функции `termo_2d` аналогичны соответствующим правилам для функции `puass_2d`.

Например, при вызове функции командой

`termo_2d;`

на экране появится график решения задачи для следующих значений параметров, заданных в данной функции по умолчанию:

- начальный момент времени – 0 с;
- конечный момент времени – 6 с;
- число точек сетки по оси времени – 6;
- $x_{\min} = 0$;
- $x_{\max} = 5$ см;
- число точек сетки по оси x – 10;
- $y_{\min} = 0$;
- $y_{\max} = 8$ см;
- число точек сетки по оси y – 20;
- плотность вещества – 2000 кг/м³;
- удельная теплоемкость вещества – 100 Дж/(кг·К);
- коэффициент теплопроводности вещества – 20 Вт/(м·К);
- плотность мощности источников (стоков) тепла – 0 Вт/м³,

с начальными условиями первого рода

$$T(x, y, t_1) = 10[\text{sign}(100x-2) - \text{sign}(100x-3) + \text{sign}(100y-3) - \text{sign}(100y-5)] + 300 \quad (309)$$

и граничными условиями Дирихле

$$T(x_{\min}, y, t) = 300 \text{ K}; \quad (310)$$

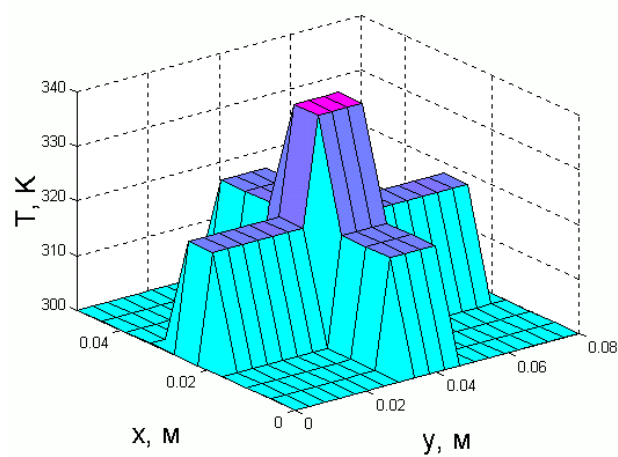
$$T(x_{\max}, y, t) = 300 \text{ K}; \quad (311)$$

$$T(x, y_{\min}, t) = 300 \text{ K}; \quad (312)$$

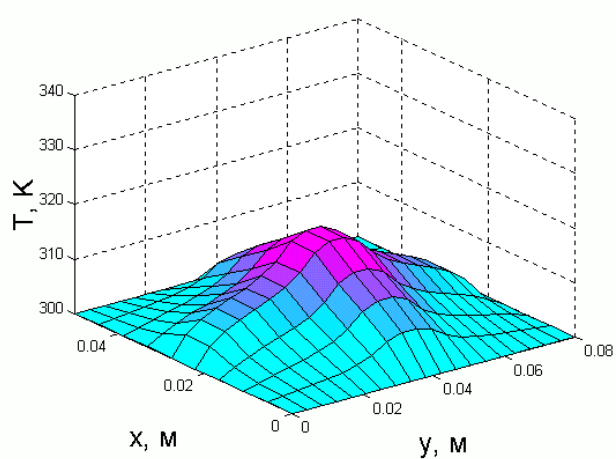
$$T(x, y_{\max}, t) = 300 \text{ K}. \quad (313)$$

на равномерной пространственно-временной сетке, содержащей 1200 точек (10 × 20 × 6) (рис. 39).

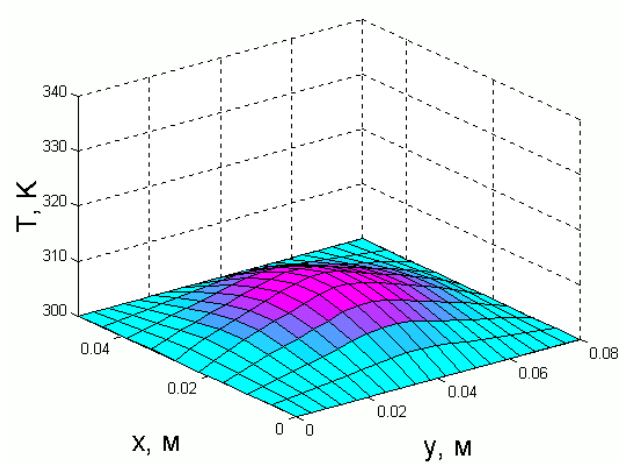
Пусть необходимо решить задачу о нестационарном распределении температуры в медном стержне прямоугольного профиля длиной 20 см, шириной 1 см (плотность 8900 кг/м³, удельная теплоемкость 380 Дж/(кг·К), коэффициент теплопроводности 385 Вт/(м·К) [12]) на промежутке времени 10 с при условии отсутствия источников и стоков тепла в объеме стержня, если температура на концах стержня равна 300 К и тепловой поток через боковые границы равен нулю для начального распределения температуры, показанного на рис. 40, а.



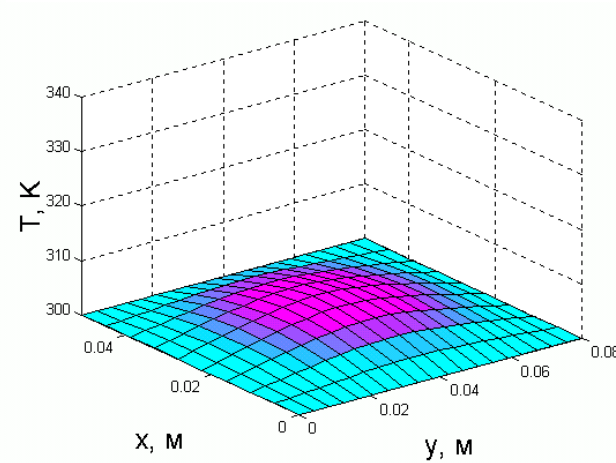
а



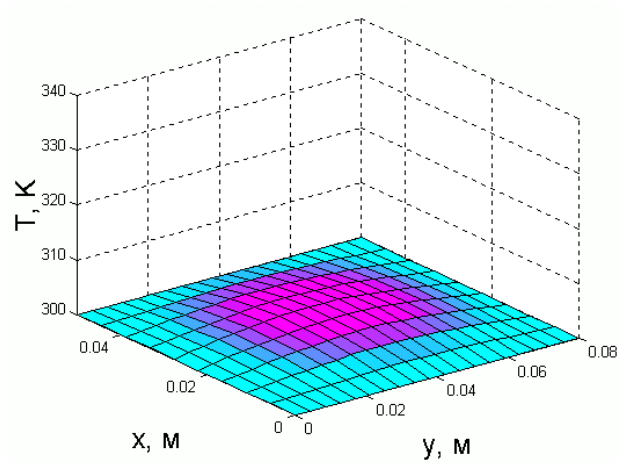
б



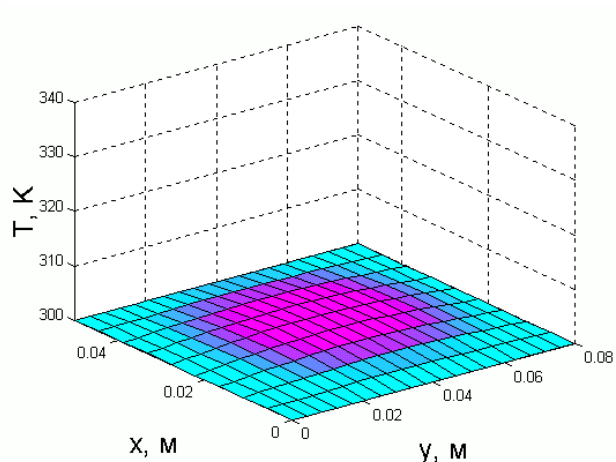
в



г



д



е

Рис. 39. Распределение температуры по координатам в различные моменты времени: а) $t = 0$; б) $t = 1$ с; в) $t = 2$ с; г) $t = 3$ с; д) $t = 4$ с; е) $t = 5$ с.

Для решения данной задачи с помощью функции `termo_2d` следует использовать следующую командную строку

```
[x,y,t,T]=termo_2d(0,10,6,0,1e-2,6,0,20e-2,50,'8900',...
'380','385','0',1,'20*(sign(1e2*y-5)-sign(1e2*y-7))+...
sign(1e2*y-12)-sign(1e2*y-14))+300',2,'0',2,'0',1,...
'300',1,'300')
```

или изменить в исходном файле `termo_2d.m` значения соответствующих параметров по умолчанию.

В результате дискретизации задачи на пространственно-временной сетке, содержащей 1800 узлов ($6 \times 50 \times 6$) и решения СЛАУ получим результаты, представленные на рис. 40.

5.3. Примеры решения волнового уравнения

В качестве примера решения уравнений гиперболического типа рассмотрим волновое уравнение:

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \quad (314)$$

где t – время; x, y – координаты; $u(x, y, t)$ – искомая функция координат и времени, на прямоугольной области с граничными условиями Дирихле или Неймана на границах $x = x_{min}, x = x_{max}, y = y_{min}, y = y_{max}$ и начальными условиями первого или второго рода на отрезке времени $[t_{min}, t_{max}]$.

Зададим на отрезке $[x_{min}, x_{max}]$ равномерную координатную сетку с шагом Δx :

$$\mathbf{x} = \{x_i | i=1, 2, \dots, n\}, \quad (315)$$

на отрезке $[y_{min}, y_{max}]$ равномерную координатную сетку с шагом Δy :

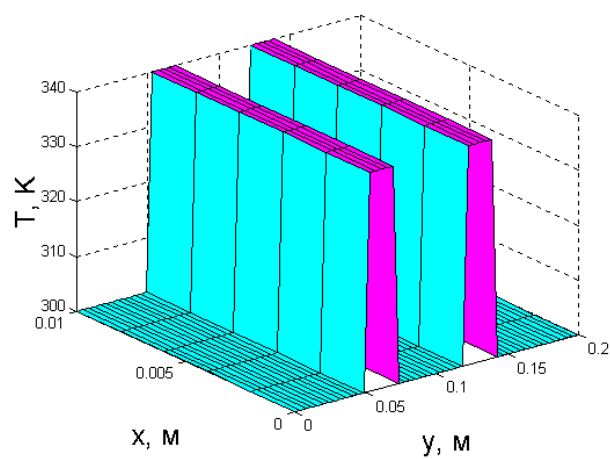
$$\mathbf{y} = \{y_j | j=1, 2, \dots, m\}, \quad (316)$$

на отрезке $[t_{min}, t_{max}]$ равномерную сетку с шагом Δt :

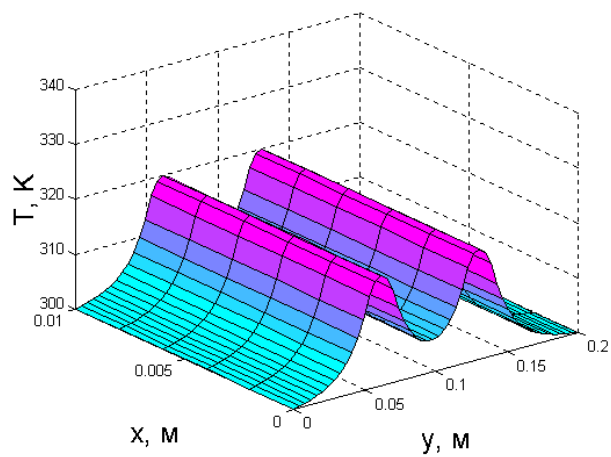
$$\mathbf{t} = \{t_l | l=1, 2, \dots, s\}. \quad (317)$$

Вектора, заданные выражениями (315) – (317), определяют на прямоугольной области равномерную пространственно-временную сетку:

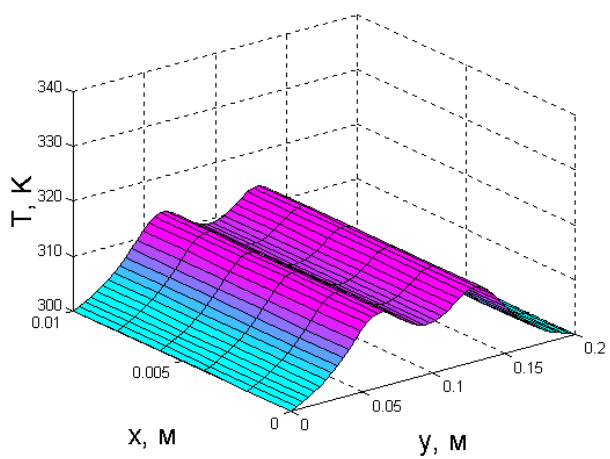
$$G = \{(x_i = i\Delta x, y_j = j\Delta y, t_l = l\Delta t), | i = 1, 2, \dots, n, j = 1, 2, \dots, m, l = 1, 2, \dots, s\}. \quad (318)$$



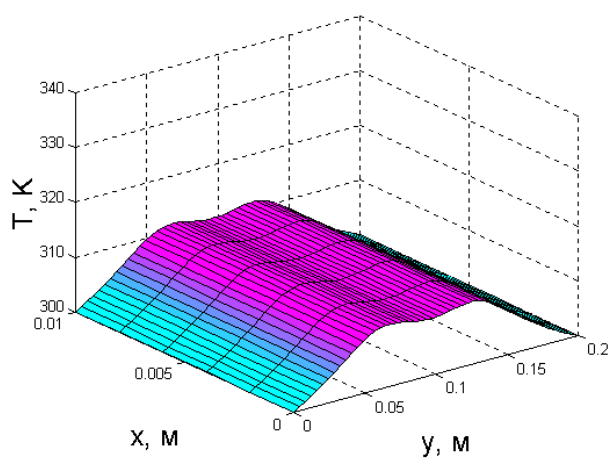
а



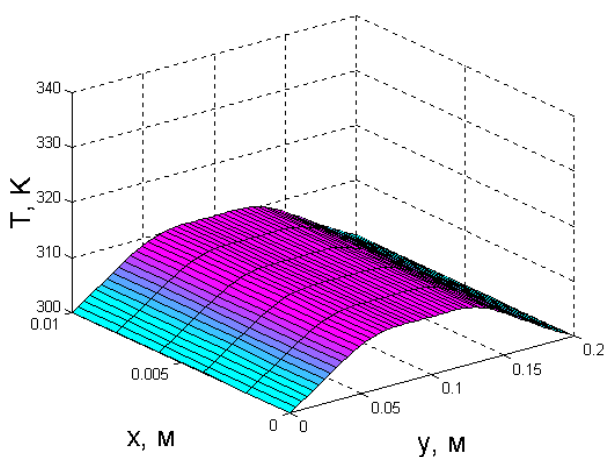
б



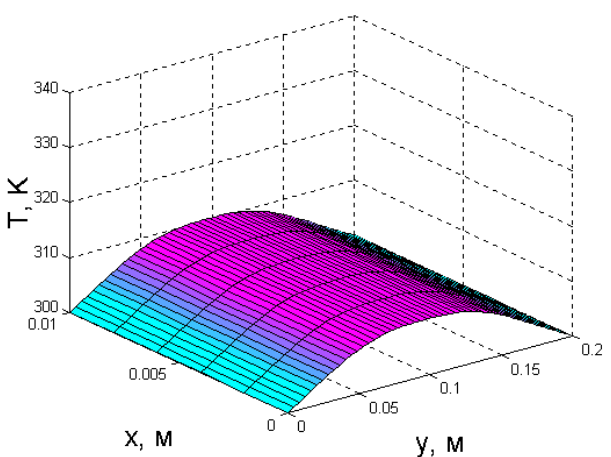
в



г



д



е

Рис. 40. Распределение температуры в медном стержне в различные моменты времени: а) $t = 0$; б) $t = 2$ с; в) $t = 4$ с; г) $t = 6$ с; д) $t = 8$ с; е) $t = 10$ с.

Граничные условия первого рода (Дирихле) для рассматриваемой задачи могут быть представлены в виде

$$u(x_1, y, t) = g_1(y); \quad (319)$$

$$u(x_n, y, t) = g_2(y); \quad (320)$$

$$u(x, y_1, t) = g_3(x); \quad (321)$$

$$u(x, y_m, t) = g_4(x), \quad (322)$$

где x_1, x_n – координаты граничных точек области x_{min}, x_{max} ; y_1, y_m – координаты граничных точек области y_{min}, y_{max} ; $g_1(y), g_2(y), g_3(x), g_4(x)$ – некоторые непрерывные функции соответствующих координат.

Граничные условия второго рода (Неймана) для рассматриваемой задачи могут быть представлены в виде

$$\left. \frac{\partial u}{\partial x} \right|_{x_1, y, t} = g_1(y); \quad (323)$$

$$\left. \frac{\partial u}{\partial x} \right|_{x_n, y, t} = g_2(y); \quad (324)$$

$$\left. \frac{\partial u}{\partial y} \right|_{x, y_1, t} = g_3(x); \quad (325)$$

$$\left. \frac{\partial u}{\partial y} \right|_{x, y_m, t} = g_4(x). \quad (326)$$

Начальные условия первого рода для рассматриваемой задачи могут быть представлены в виде

$$u(x, y, t_1) = g_{t1}(x, y); \quad (327)$$

$$u(x, y, t_s) = g_{t2}(x, y), \quad (328)$$

где t_1 – начальный момент времени; t_s – конечный момент времени; $g_{t1}(x, y), g_{t2}(x, y)$ – некоторые непрерывные функции соответствующих координат.

Начальные условия второго рода для рассматриваемой задачи могут быть представлены в виде

$$\left. \frac{\partial u}{\partial t} \right|_{x, y, t_1} = g_{t1}(x, y) ; \quad (329)$$

$$\left. \frac{\partial u}{\partial t} \right|_{x, y, t_s} = g_{t2}(x, y) . \quad (330)$$

Проводя дискретизацию граничных условий Дирихле на равномерной сетке (318) с использованием метода конечных разностей, получим

$$u_{1,j,l} = g_1(y_j) ; \quad (331)$$

$$u_{n,j,l} = g_2(y_j) ; \quad (332)$$

$$u_{i,1,l} = g_3(x_i) ; \quad (333)$$

$$u_{i,m,l} = g_4(x_i) , \quad (334)$$

где $u_{1,j,l}$, $u_{n,j,l}$, $u_{i,1,l}$, $u_{i,m,l}$ — значения функции $u(x, y, t)$ в точках (x_1, y_j, t_l) , (x_n, y_j, t_l) , (x_i, y_1, t_l) , (x_i, y_m, t_l) , соответственно.

Проводя дискретизацию граничных условий Неймана на сетке (318), получим

$$\frac{u_{2,j,l} - u_{1,j,l}}{\Delta x} = g_1(y_j) ; \quad (335)$$

$$\frac{u_{n,j,l} - u_{n-1,j,l}}{\Delta x} = g_2(y_j) ; \quad (336)$$

$$\frac{u_{i,2,l} - u_{i,1,l}}{\Delta y} = g_3(x_i) ; \quad (337)$$

$$\frac{u_{i,m,l} - u_{i,m-1,l}}{\Delta y} = g_4(x_i) . \quad (338)$$

Проводя дискретизацию начальных условий первого рода на равномерной сетке (318), получим

$$u_{i,j,1} = g_{t1}(x_i, y_j) ; \quad (339)$$

$$u_{i,j,s} = g_{t2}(x_i, y_j) , \quad (340)$$

где $u_{i,j,1}$ — значения функции $u(x, y, t)$ в точке (x_i, y_j, t_1) .

Проводя дискретизацию начальных условий второго рода на сетке (318), получим

$$\frac{u_{i,j,2} - u_{i,j,1}}{\Delta t} = g_{t1}(x_i, y_j); \quad (341)$$

$$\frac{u_{i,j,s} - u_{i,j,s-1}}{\Delta t} = g_{t2}(x_i, y_j). \quad (342)$$

Проводя дискретизацию уравнения (314) для внутренних точек сетки, получим

$$\begin{aligned} & \frac{u_{i,j,l+1} - 2u_{i,j,l} + u_{i,j,l-1}}{\Delta t^2} - \frac{u_{i+1,j,l} - 2u_{i,j,l} + u_{i-1,j,l}}{\Delta x^2} - \\ & - \frac{u_{i,j+1,l} - 2u_{i,j,l} + u_{i,j-1,l}}{\Delta y^2} = 0; \\ & i=2, \dots, n-1; j=2, \dots, m-1; l=2, \dots, s-1. \end{aligned} \quad (343)$$

Таким образом, в результате дискретизации получим систему линейных алгебраических уравнений размерности $n \cdot m \cdot s$.

Ниже приведен один из вариантов функции с комментариями для численного решения уравнения (314) с граничными условиями (319) – (326) и начальными условиями (327) или (330) на равномерной сетке (318).

```
% Функция решения волнового уравнения
% d2U/dt2=d2U/dx2+d2U/dy2
% на прямоугольной области с граничными условиями
% Дирихле и/или Неймана

function[x,y,t,U]= ...
wave_2d(t0,ts,s,x0,xn,n,y0,ym,m, ...
vt1,gt1,vt2,gt2,v1,g1,v2,g2,v3,g3,v4,g4)

% Входные параметры:
% t0 - начальный момент времени;
% ts - конечный момент времени;
% x0 - начальная координата области решения по оси x;
% xn - конечная координата области решения по оси x;
% y0 - начальная координата области решения по оси y;
% ym - конечная координата области решения по оси y;
% n - число точек координатной сетки вдоль оси x;
% m - число точек координатной сетки вдоль оси y;
```

```

% s - число точек сетки вдоль оси времени t;
% vt1- параметр, значение которого определяет
%      тип начального условия в момент времени t(1)
%      (1 - Дирихле, 2 - Неймана);
% gt1- функция в правой части начального условия
%      в момент времени t(1),
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки;
% vt2- параметр, значение которого определяет
%      тип начального условия в момент времени t(s)
%      (1 - Дирихле, 2 - Неймана);
% gt2- функция в правой части начального условия
%      в момент времени t(s),
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки;
% v1 - параметр, значение которого определяет
%      тип граничного условия на первой границе
%      области  $x = x(1)$  (1 - ГУ Дирихле, 2 - ГУ Неймана);
% g1 - функция в правой части граничного условия
%      на первой границе,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки;
% v2 - параметр, значение которого определяет
%      тип граничного условия на второй границе
%      области  $x = x(n)$  (1 - ГУ Дирихле, 2 - ГУ Неймана);
% g2 - функция в правой части граничного условия
%      на второй границе,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки;
% v3 - параметр, значение которого определяет
%      тип граничного условия на третьей границе
%      области  $y = y(1)$  (1 - ГУ Дирихле, 2 - ГУ Неймана);
% g3 - функция в правой части граничного условия
%      на третьей границе,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки;
% v4 - параметр, значение которого определяет
%      тип граничного условия на четвертой границе
%      области  $y = y(m)$  (1 - ГУ Дирихле, 2 - ГУ Неймана);
% g4 - функция в правой части граничного условия
%      на четвертой границе,
%      задаваемая строкой символов, заключенных
%      в одинарные кавычки.

```

```
% Выходные параметры:
% x - вектор-строка координатной сетки по оси x
%     размерности 1 x n;
% y - вектор-строка координатной сетки по оси y
%     размерности 1 x m;
% t - вектор-строка сетки по оси времени
%     размерности 1 x s;
% U - матрица значений результирующей функции
%     в узлах координатной сетки размерности n x m x s.
```

```
% Функции и переменные по умолчанию
```

```
if exist('t0')==0
    t0=0;
end
if exist('ts')==0
    ts=0.2;
end
if exist('s')==0
    s=6;
end
if exist('x0')==0
    x0=-1;
end
if exist('xn')==0
    xn=1;
end
if exist('n')==0
    n=18;
end
if exist('y0')==0
    y0=-1;
end
if exist('ym')==0
    ym=1;
end
if exist('m')==0
    m=18;
end
if exist('vt1')==0
    vt1=1;
end
if exist('gt1')==0
    gt1='sin(4*x)-sin(4*y)';
end
```

```

if exist('vt2')==0
    vt2=1;
end
if exist('gt2')==0
    gt2='sin(4*y)-sin(4*x)';
end
if exist('v1')==0
    v1=2;
end
if exist('g1')==0
    g1='0';
end
if exist('v2')==0
    v2=2;
end
if exist('g2')==0
    g2='0';
end
if exist('v3')==0
    v3=2;
end
if exist('g3')==0
    g3='0';
end
if exist('v4')==0
    v4=2;
end
if exist('g4')==0
    g4='0';
end

```

```

% Задание равномерной сетки

```

```

x=x0:(xn-x0)/(n-1):xn;    dx=x(2)-x(1);
y=y0:(ym-y0)/(m-1):ym;    dy=y(2)-y(1);
t=t0:(ts-t0)/(s-1):ts;    dt=t(2)-t(1);

```

```

% Вычисление значений функций, заданных символьно,
% в узлах координатной сетки

```

```

GT1=inline(gt1,'x','y');
GT2=inline(gt2,'x','y');
G1=inline(g1,'y');
G2=inline(g2,'y');
G3=inline(g3,'x');

```

```

G4=inline(g4,'x');

% Определение размерности СЛАУ

N=s*n*m;

% Задание матрицы коэффициентов СЛАУ размерности N x N,
% все элементы которой равны 0

a=zeros(N,N);

% Задание матрицы-строки свободных членов СЛАУ размерно-
% сти 1 x N,
% все элементы которой равны 0

b=zeros(1,N);

% Определение коэффициентов и свободных членов СЛАУ,
% соответствующих начальным и граничным условиям
% и проверка корректности
% значений параметров vt1, vt2, v1, v2, v3, v4

for i=1:n
    for j=1:m
        b(m*(i-1)+j)=GT1(x(i),y(j));
        if vt1==1
            a(m*(i-1)+j,m*(i-1)+j)=1;
        elseif vt1==2
            a(m*(i-1)+j,m*(i-1)+j)=-1/dt;
            a(m*(i-1)+j,n*m+m*(i-1)+j)=1/dt;
        else
            error('Parameter vt1 have incorrect value');
        end
        b(n*m*(s-1)+m*(i-1)+j)=GT2(x(i),y(j));
        if vt2==1
            a(n*m*(s-1)+m*(i-1)+j,n*m*(s-1)+m*(i-1)+j)=1;
        elseif vt2==2
            a(n*m*(s-1)+m*(i-1)+j,n*m*(s-1)+m*(i-1)+j)=1/dt;
            a(n*m*(s-1)+m*(i-1)+j,n*m*(s-2)+m*(i-1)+j)= ...
                -1/dt;
        else
            error('Parameter vt2 have incorrect value');
        end
    end
end
end

```



```

for l=1:s
    for j=1:m
        b(n*m*(l-1)+j)=G1(y(j));
        if v1==1
            a(n*m*(l-1)+j,n*m*(l-1)+j)=1;
        elseif v1==2
            a(n*m*(l-1)+j,n*m*(l-1)+j)=-1/dx;
            a(n*m*(l-1)+j,n*m*(l-1)+m+j)=1/dx;
        else
            error('Parameter v1 have incorrect value');
        end
        b(n*m*(l-1)+m*(n-1)+j)=G2(y(j));
        if v2==1
            a(n*m*(l-1)+m*(n-1)+j,n*m*(l-1)+m*(n-1)+j)=1;
        elseif v2==2
            a(n*m*(l-1)+m*(n-1)+j,n*m*(l-1)+m*(n-1)+j)=1/dx;
            a(n*m*(l-1)+m*(n-1)+j,n*m*(l-1)+m*(n-2)+j)= ...
                -1/dx;
        else
            error('Parameter v2 have incorrect value');
        end
    end
    for i=2:n-1
        b(n*m*(l-1)+m*(i-1)+1)=G3(x(i));
        if v3==1
            a(n*m*(l-1)+m*(i-1)+1,n*m*(l-1)+m*(i-1)+1)=1;
        elseif v3==2
            a(n*m*(l-1)+m*(i-1)+1,n*m*(l-1)+m*(i-1)+1)= ...
                -1/dy;
            a(n*m*(l-1)+m*(i-1)+1,n*m*(l-1)+m*(i-1)+2)=1/dy;
        else
            error('Parameter v3 have incorrect value');
        end
        b(n*m*(l-1)+m*(i-1)+m)=G4(x(i));
        if v4==1
            a(n*m*(l-1)+m*(i-1)+m,n*m*(l-1)+m*(i-1)+m)=1;
        elseif v4==2
            a(n*m*(l-1)+m*(i-1)+m,n*m*(l-1)+m*(i-1)+m)=1/dy;
            a(n*m*(l-1)+m*(i-1)+m,n*m*(l-1)+m*(i-1)+m-1)= ...
                -1/dy;
        else
            error('Parameter v4 have incorrect value');
        end
    end
end
end

```

```
% Определение коэффициентов и свободных членов СЛАУ,
% соответствующих внутренним точкам области
```

```
for l=2:s-1
    for i=2:n-1
        for j=2:m-1
            a(n*m*(l-1)+m*(i-1)+j,n*m*(l-1)+m*(i-1)+j)=...
                -2/dt^2+2/dx^2+2/dy^2;
            a(n*m*(l-1)+m*(i-1)+j,n*m*(l-1)+m*i+j)=-1/dx^2;
            a(n*m*(l-1)+m*(i-1)+j,n*m*(l-1)+m*(i-2)+j)=...
                -1/dx^2;
            a(n*m*(l-1)+m*(i-1)+j,n*m*(l-1)+m*(i-1)+j+1)=...
                -1/dy^2;
            a(n*m*(l-1)+m*(i-1)+j,n*m*(l-1)+m*(i-1)+j-1)=...
                -1/dy^2;
            a(n*m*(l-1)+m*(i-1)+j,n*m*l+m*(i-1)+j)=1/dt^2;
            a(n*m*(l-1)+m*(i-1)+j,n*m*(l-2)+m*(i-1)+j)=...
                1/dt^2;
        end
    end
end
```

```
% Решение СЛАУ
```

```
u=b/a' ;
```

```
% Преобразование вектора-строки значений искомой функции
% в узлах координатной сетки в матрицу
% размерности n x m x s, удобную для представления
% результатов в графическом виде
```

```
for l=1:s
    for i=1:n
        for j=1:m
            U(i,j,l)=u(n*m*(l-1)+m*(i-1)+j);
        end
    end
end
```

```
% Построение графика искомой функции U(x,y,t)
```

```
for l=1:s
    figure
    surf(y,x,U(:, :, l))
```

```

xlabel('y')
ylabel('x')
zlabel('U')
grid on
colormap('cool')
axis([min(y) max(y) min(x) max(x) min(min(min(U))) ...
      max(max(max(U)))])
pause(0.1)
M(1)=getframe;
end

% Отображение волнового процесса в динамическом режиме

figure
movie(M,20,10)

```

Приведенное описание необходимо сохранить в виде текстового файла с именем `wave_2d.m` и поместить его в каталог `/WORK`, находящийся в корневом каталоге системы MATLAB.

Вызов функции `wave_2d` осуществляется аналогично функциям `puass_2d` и `termo_2d` следующими командами:

```

wave_2d;
x=wave_2d;
[x,y]=wave_2d;
[x,y,t]=wave_2d;
[x,y,t,U]=wave_2d(t0,ts,s,x0,xn,n,y0,ym,m,...
    vt1,gt1,vt2,gt2,v1,g1,v2,g2,v3,g3,v4,g4).

```

Графики распределений искомой функции по координатам в различные моменты времени будут выводиться в отдельных окнах. После вывода всех графиков в новом окне будет показан волновой процесс в динамическом режиме.

При использовании первой, второй, третьей или четвертой команд функция будет выводить графически решение задачи при входных параметрах, принятых по умолчанию:

- начальный момент времени – 0;
- конечный момент времени – 0.2;
- число точек сетки по оси времени – 6;
- $x_{\min} = -1$;
- $x_{\max} = 1$;
- число точек сетки по оси x – 18;
- $y_{\min} = -1$;
- $y_{\max} = 1$;
- число точек сетки по оси y – 18.

с начальными условиями первого рода

$$u(x, y, t_1) = \sin(4x) - \sin(4y); \quad (344)$$

$$u(x, y, t_1) = \sin(4y) - \sin(4x); \quad (345)$$

и граничными условиями Неймана

$$\left. \frac{\partial u}{\partial x} \right|_{x_{\min}, y, t} = 0; \quad (346)$$

$$\left. \frac{\partial u}{\partial x} \right|_{x_{\max}, y, t} = 0; \quad (347)$$

$$\left. \frac{\partial u}{\partial y} \right|_{x, y_{\min}, t} = 0; \quad (348)$$

$$\left. \frac{\partial u}{\partial y} \right|_{x, y_{\max}, t} = 0; \quad (349)$$

на равномерной пространственно-временной сетке, содержащей 1944 узла ($18 \times 18 \times 6$) (рис. 41).

Функция `wave_2d` выводит результаты в нормированных единицах. При умножении на соответствующие коэффициенты нормировки можно получить графическое отображение волнового процесса изменения конкретных физических величин на прямоугольной области с размерами, выраженными в удобных единицах измерения, в моменты времени, выраженные в секундах.

Для этого необходимо осуществить вызов функции `wave_2d` при помощи одной из следующих команд:

```
[x, y, t, U] = wave_2d;
[x, y, t, U] = wave_2d(t0, ts, s, x0, xn, n, y0, ym, m, ...
    vt1, gt1, vt2, gt2, v1, g1, v2, g2, v3, g3, v4, g4);
```

В этом случае при выходе из функции матрицы точек пространственно-временной сетки x , y , t и трехмерная матрица U результирующих значений функции $u(x, y, t)$ будут сохранены в оперативной памяти и доступны для операций умножения на коэффициенты нормировки. При этом в первом случае будут использованы входные параметры по умолчанию.

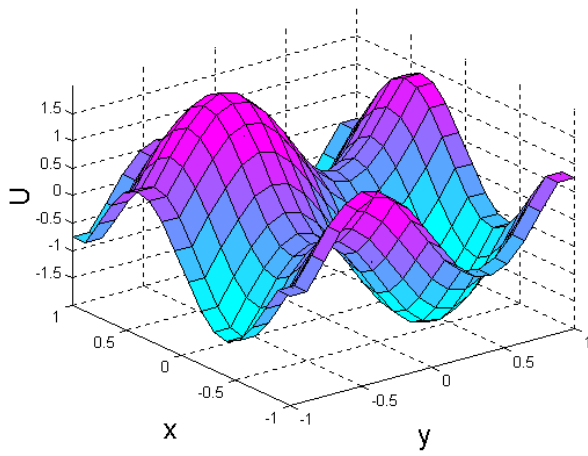
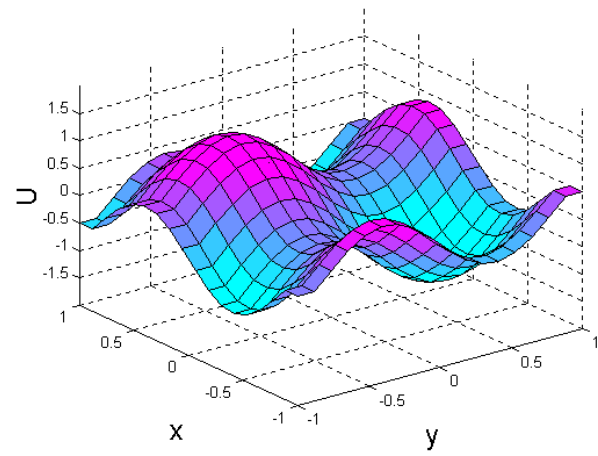
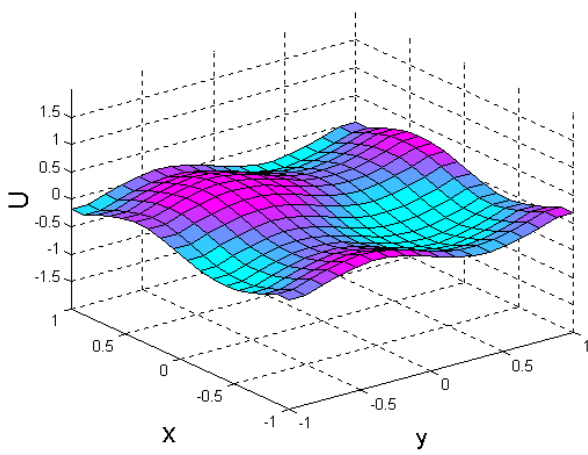
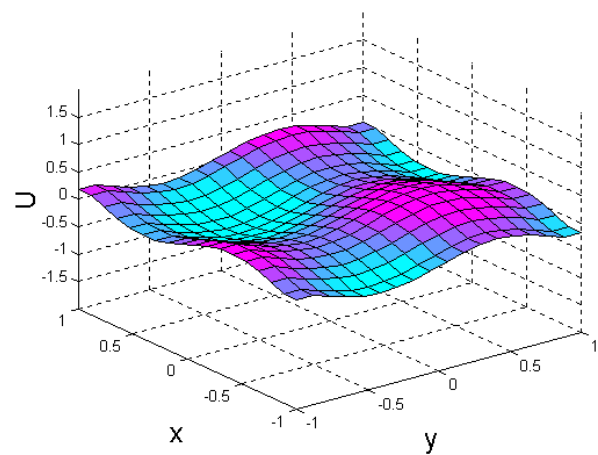
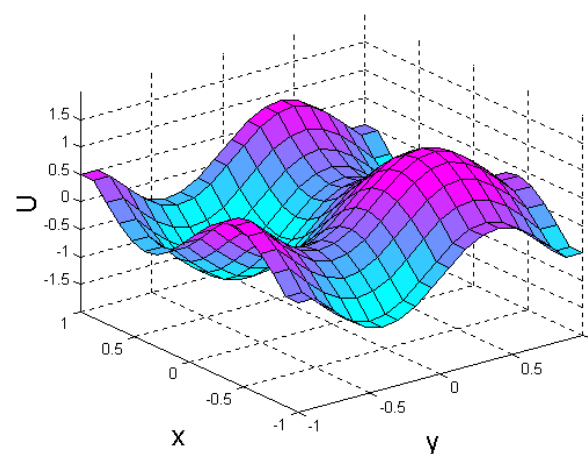
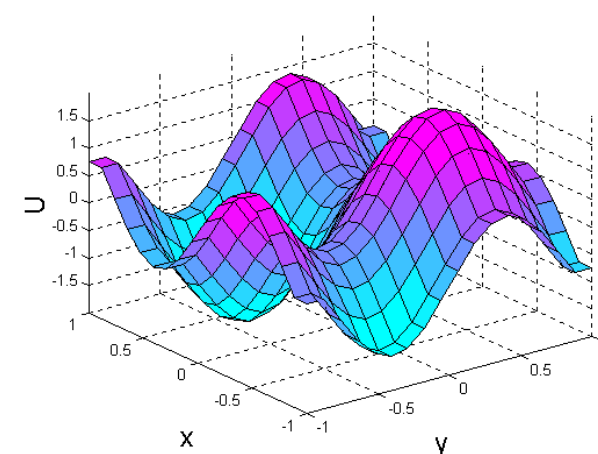
*a**б**в**г**д**е*

Рис. 41. Волновой процесс $u(x, y, t)$ в различные моменты времени:
a) $t = 0$; *б)* $t = 0.04$; *в)* $t = 0.08$; *г)* $t = 0.12$; *д)* $t = 0.16$; *е)* $t = 0.2$.

ЗАКЛЮЧЕНИЕ

Следует отметить, что круг задач, связанных с созданием приборов микро- и наноэлектроники и описываемых уравнениями математической физики, чрезвычайно широк. В настоящее время он все более расширяется в значительной степени благодаря микросистемной технике, быстрые темпы развития которой обусловлены созданием интегральных приборов (сенсоров и актюаторов), характеризующихся новыми принципами функционирования и позволяющих использовать самые различные физические явления и эффекты.

В учебном пособии рассмотрены лишь самые основные уравнения математической физики, наиболее широко используемые в процессе создания элементной базы микросхем и микросистем, а также основные особенности задания граничных и начальных условий, методы дискретизации дифференциальных уравнений в частных производных, методы решения систем алгебраических уравнений, основные этапы решения задач матфизики.

Рассмотренные методы решения уравнений проиллюстрированы примерами для системы MATLAB с исходными описаниями функций, имеющих подробные комментарии и рекомендации по их использованию, которые позволят читателю детально представить себе весь процесс решения основных уравнений математической физики и приобрести практические навыки решения подобных задач.

ЛИТЕРАТУРА

1. Самарский А.А., Андреев В.Б. Разностные методы для эллиптических уравнений. – М.: Наука, 1976. 352 с.
2. Моделирование полупроводниковых приборов и технологических процессов. Последние достижения: Пер. с англ. / Под ред. Д. Миллера. – М.: Радио и связь, 1989. 280 с.
3. Мэтьюз Д.Г., Финк К.Д. Численные методы. Использование MATLAB. 3-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2001. 720 с.
4. Бубенников А.Н., Садовников А.Д. Физико-технологическое проектирование биполярных элементов кремниевых БИС. – М.: Радио и связь, 1991. 288 с.
5. Степаненко И.П. Основы теории транзисторов и транзисторных схем. – М.: Энергия, 1973. – 608 с.
6. Данилина Н.И., Дубровская Н.С., Кваша О.П., Смирнов Г.Л., Феклистов Г.И. Численные методы. – М.: Высш. школа, 1976. 368 с.
7. Бронштейн И.Н., Семендяев К.А. Справочник по математике для инженеров и учащихся втузов. – М.: Государственное издательство физико-математической литературы, 1959. 608 с.
8. Яншин А.А. Теоретические основы конструирования, технологии и надежности ЭВА: Учеб. пособие для вузов. – М.: Радио и связь, 1983. 312 с.
9. Скворцов А.В. Обзор алгоритмов построения триангуляции Делоне // Вычислительные методы и программирование. 2002. Т. 3. С. 14 – 39.
10. Потемкин В.Г. Система инженерных и научных расчетов MATLAB 5.X. В 2-х томах. Т. 1. – М.: ДИАЛОГ-МИФИ, 1999. 366 с.
11. Потемкин В.Г. Система инженерных и научных расчетов MATLAB 5.X. В 2-х томах. Т. 2. – М.: ДИАЛОГ-МИФИ, 1999. 304 с.
12. Элементарный учебник физики / Под ред. Г.С. Ландсберга. Т. 1. – М.: Наука, 1967. 576 с.