МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
ТАГАНРОГСКИЙ ГОСУДАРСТВЕННЫЙ
РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Е.А. Рындин, И.Е. Лысенко

РЕШЕНИЕ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ФИЗИКИ В СИСТЕМЕ MATLAB

УЧЕБНОЕ ПОСОБИЕ

Таганрог 2005

УДК 621.382.8:658.512.2.011.5(076.5)

Р е ц е н з е н т ы :

Таганрогский государственный педагогический институт; Б.Е. Механцев, старший преподаватель.

В.В. Поляков, канд. техн. наук, доцент кафедры технологии микро- и наноэлектронной аппаратуры ТРТУ.

Рындин Е.А., Лысенко И.Е. Решение задач математической физики в системе MatLab. – Таганрог: Изд-во ТРТУ, 2005. – 62 с.

Учебное пособие по освоению студентами методов решения задач математической физики подготовлен сотрудниками кафедры конструирования электронных средств (КЭС) Таганрогского государственного радиотехнического университета (ТРТУ).

В работе излагаются сведения, необходимые для численного решения уравнений математической физики методами конечных разностей и конечных элементов с использованием системы MATLAB. Приведены общие сведения о системе MATLAB, необходимые для создания m-файлов, функций, об использовании пакета PDETOOL для решения уравнений математической физики.

Ил. 29. Библиогр.: 2 назв.

Печатается по решению редакционно-издательского совета Таганрогского радиотехнического университета.

ВВЕДЕНИЕ

Разработка и исследование значительной части элементов современных сверхбольших интегральных схем (СБИС) и микрооптикоэлектромеханических систем (МОЭМС) связаны с решением задач математической физики, к которым относят 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. Нахождение точного аналитического решения, к сожалению, возможно лишь для весьма ограниченного круга одномерных задач при использовании целого ряда допущений. Для решения уравнений математической физики в случае нескольких измерений используют численные методы, позволяющие преобразовать дифференциальные уравнения или их системы в системы алгебраических уравнений. Точность решения определяется шагом координатной сетки, количеством итераций и разрядной сеткой компьютера [1].

В методическом пособии на конкретных примерах рассмотрены методы решения основных уравнений математической физики, а также особенности задания граничных и начальных условий в системе MATLAB.

1. ОБЩИЕ СВЕДЕНИЯ О СИСТЕМЕ MATLAB

Система MATLAB в настоящее время является мощным и универсальным средством решения задач, возникающих в различных областях человеческой деятельности. Спектр проблем, решение которых может быть осуществлено при помощи MATLAB, охватывает: матричный анализ, обработку сигналов и изображений, задачи математической физики, оптимизационные задачи, обработку и визуализацию данных, нейронные сети, нечеткую логику и многие другие [2]. Специализированные средства собраны в пакеты программ, называемые ToolBox и могут быть выбраны при установке MATLAB по желанию пользователя. В состав многих ToolBox входят приложения с графическим интерфейсом, которые обеспечивают быстрый и наглядный доступ к основным функциям.

Обширная и удобная справочная система MATLAB способна удовлетворить потребности как начинающего, так и опытного пользователя. Часто оказываются полезными прилагаемые к MATLAB электронные справочники в формате PDF, которые не только дублируют справочную систему MATLAB, но и содержат теоретические сведения и математическую базу, необходимые для более осознанного использования описываемых программных средств [2].

Важным достоинством MATLAB является открытость кода, что дает возможность опытным пользователям при необходимости изменять запрограммированные алгоритмы. Впрочем, разнообразие набора функций MATLAB допускает решение большинства задач без предварительных модификаций [2].

MATLAB имеет достаточно простой и эффективный встроенный язык программирования, позволяющий пользователю реализовывать собственные алгоритмы. Простота языка программирования компенсируется огромным множеством функций MATLAB. Такое сочетание позволяет достаточно быстро разрабатывать эффективные программы [2].

MATLAB является интерпретатором, т. е. каждая строка программы преобразуется в код и затем выполняется. Разумеется, это существенно увеличивает время работы алгоритмов, содержащих циклически повторяемые действия. Для повышения производительности вычислений в составе MATLAB имеется дополнительный модуль Matlab Compiler, который обеспечивает компиляцию программ, написанных на языке MATLAB. Объектно-ориентированный подход, заложенный в основу MATLAB, обеспечивает современную эффективную технологию программирования [2].

Таким образом, можно сделать вывод о том, что начинающий пользователь MATLAB может в процессе работы совершенствовать свои знания как в области моделирования и численных методов, так и в области программирования и визуализации данных, что полностью отвечает целям данного учебного пособия.

Ниже на конкретных примерах рассмотрены основные подходы к решению уравнений математической физики в системе MATLAB 5.3:

- составление m-файлов;
- составление функций;
- использование PDETool.

2. РЕАЛИЗАЦИЯ АЛГОРИТМОВ РЕШЕНИЯ УРАВНЕНИЙ МАТЕМАТИЧЕСКОЙ ФИЗИКИ В СИСТЕМЕ MATLAB

2.1. Работа с командным окном

При запуске MATLAB на экране появляется командное окно **MATLAB Command Window**, изображенное на рис. 2.1 и состоящее из следующих элементов:

- меню;
- панель с кнопками;
- рабочая область с командной строкой;
- строка состояния.

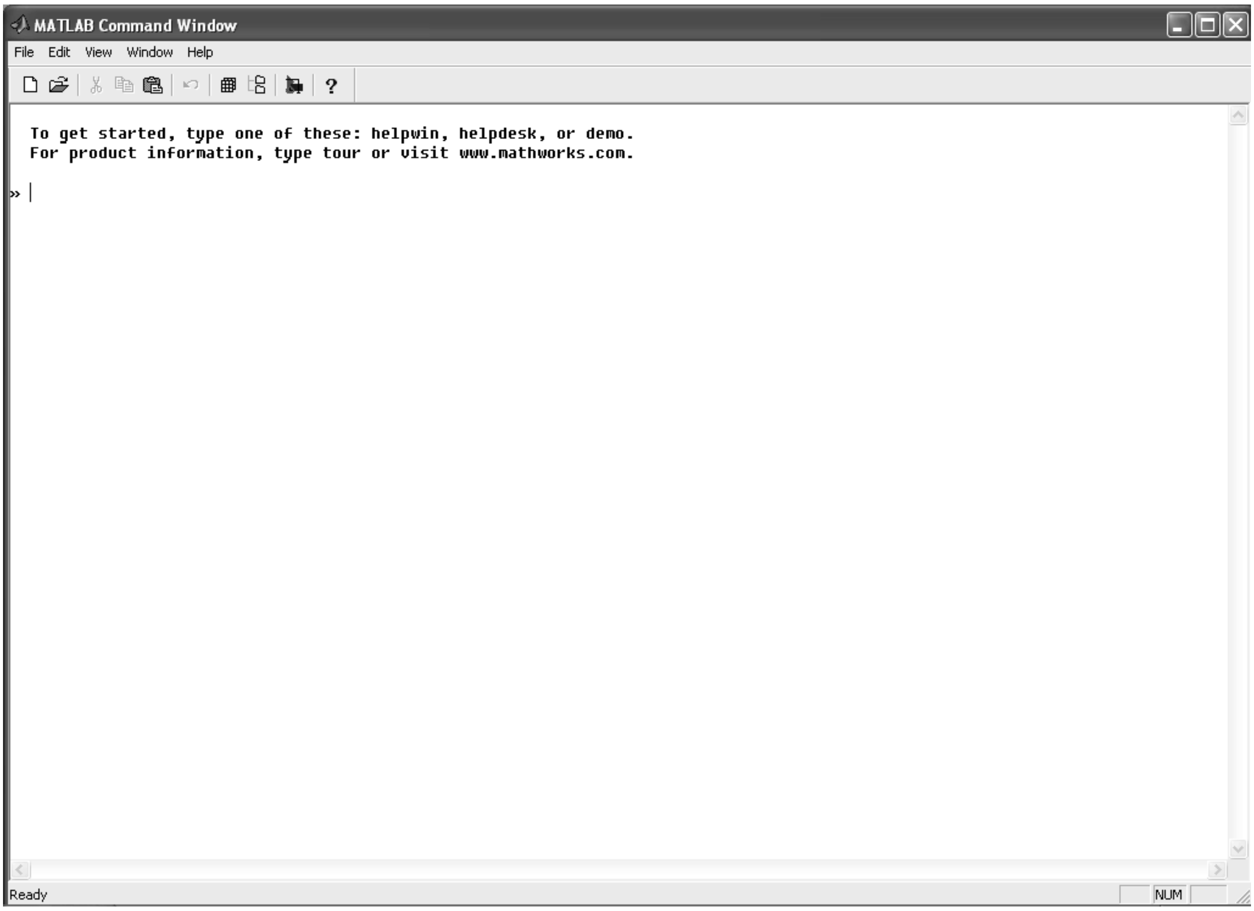


Рис. 2.1. Командное окно MATLAB Command Window

Символ >> означает приглашение командной строки к вводу команды. Набор любой команды или выражения должен сопровождаться нажатием клавиши **<Enter>** для того, чтобы система MATLAB выполнила введенную команду или вычислила выражение.

Встроенные математические функции MATLAB позволяют находить значения различных выражений. Команды для вычисления выражений имеют вид, свойственный всем языкам программирования высокого уровня. Полный перечень встроенных математических функций можно найти в справочной системе MATLAB.

При работе с командной строкой следует помнить следующие особенности:

– при наборе выражения без символа «;» в конце результат вычислений запишется в оперативную память в виде значения соответствующей переменной и будет выведен на экран, в противном случае результат вычислений запишется в оперативную память, но на экран выводиться не будет;
– при наборе выражения с левой и правой частями, разделенными знаком равенства, результат вычислений запишется в переменную левой части выражения. Если выражение не содержит левой части, то ре-

зультат запишется в специальную переменную **ans** и будет храниться в ней до момента записи в данную переменную результата вычисления следующего выражения.

На рис. 2.2 приведен ряд примеров, иллюстрирующих вычисления в командной строке.

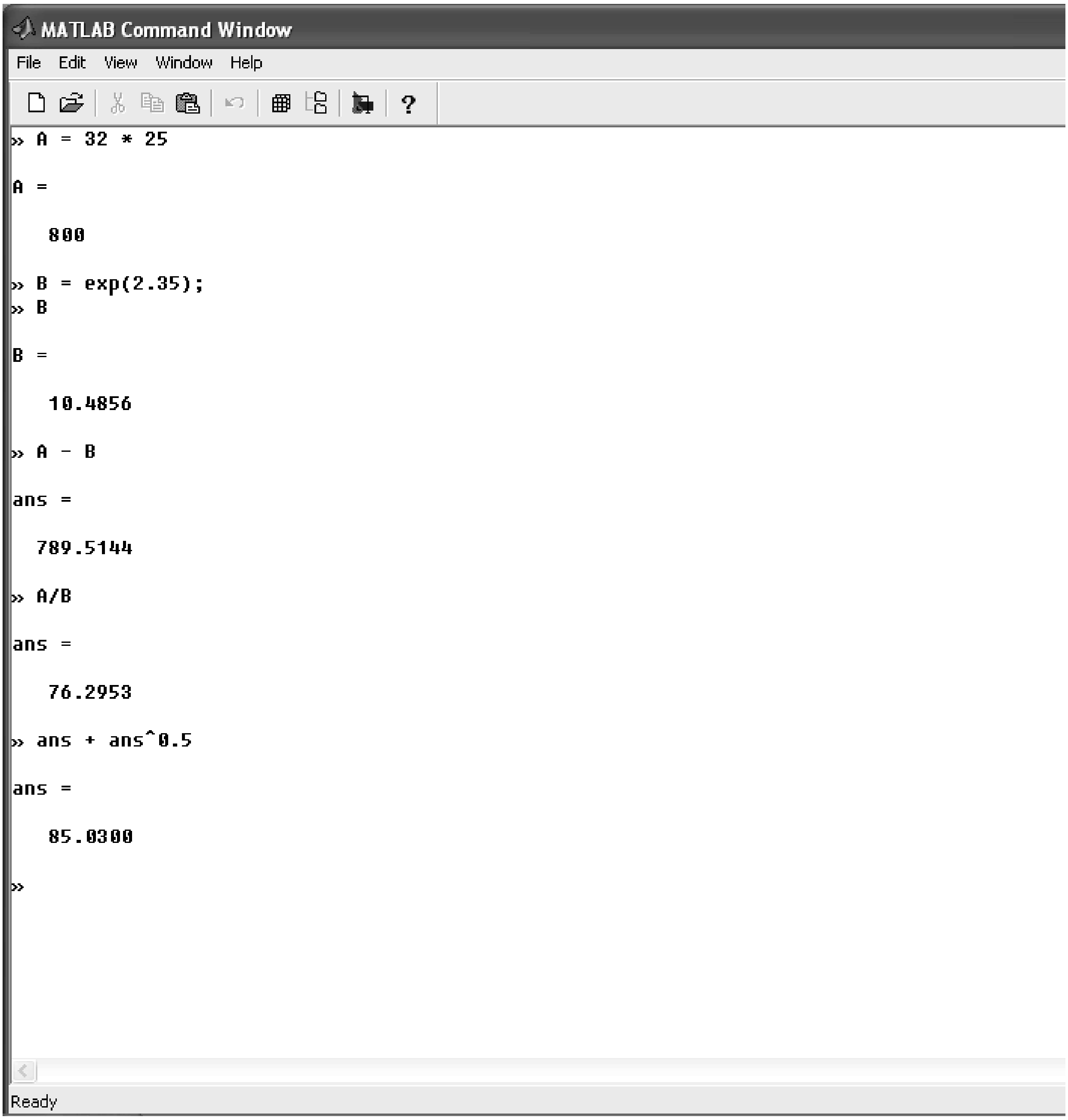


Рис. 2.2. Примеры вычисления выражений в командной строке

При вычислении выражения **A = 32 * 25;** результат был сохранен в переменной **A** и выведен на экран, поскольку в конце выражения отсутствовал символ «;».

При вычислении выражения **B = exp(2.35);** результат был сохранен в переменной **B**, но не выведен на экран, поскольку в конце выражения присутствовал символ «;». Для вывода результата на экран в командной строке было введено имя переменной **B** и нажата клавиша **<Enter>**.

При вводе выражений **A – B** и **A/B** использовались значения переменных, найденные в предыдущих операциях, а результат вычисления выражения был помещен по умолчанию в специальную переменную **ans** и выведен на экран.

Переменная **ans** также может использоваться в процессе вычислений, как это показано в примере **ans + ans^0.5** (см. рис. 2.2). При этом результат вычислений вновь помещается в переменную **ans**.

Следует помнить, что MATLAB различает прописные и строчные символы в именах переменных, функций и команд.

По умолчанию результаты вычислений выводятся на экран с округлением до четвертого знака после десятичной точки в так называемом формате **Short** (см. рис. 2.2).

Если при выводе слишком большого или слишком малого числа результат не укладывается в формат **Short**, вывод осуществляется в экспоненциальной форме (формат **Short E**). Например, результат **0.0000033333** будет выведен на экран в виде **3.3333e-6**, что эквивалентно $\mathbf{3.3333 \cdot 10^{-6}}$.

При необходимости результат вычислений может быть выведен в ином формате. Для этого следует активизировать команду **Preferences** меню **File**. На экране появится диалоговое окно **Preferences**, показанное на рис. 2.3 и позволяющее выбрать требуемый формат из списка.

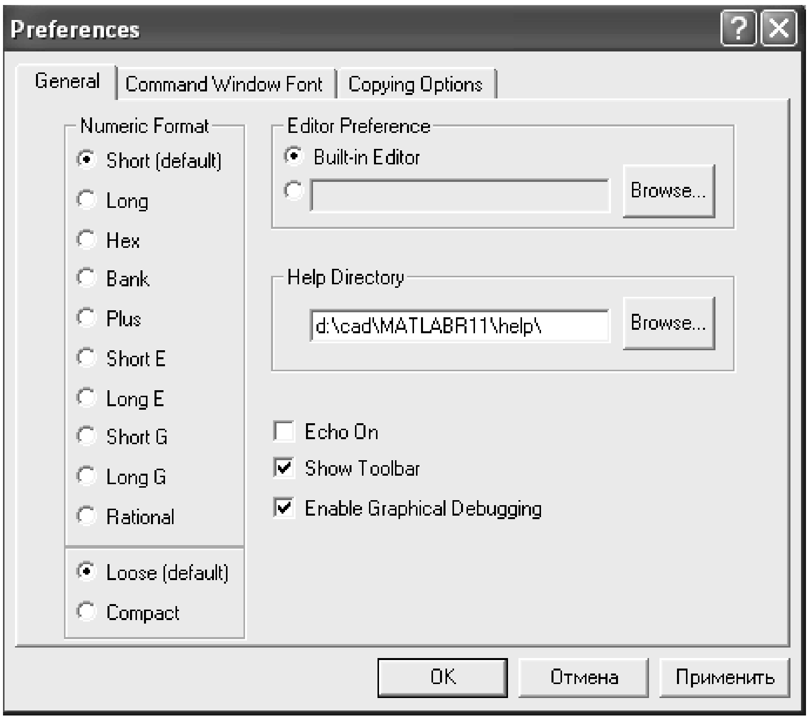


Рис. 2.3. Диалоговое окно Preferences

Например, при выборе формата **Long** результаты всех последующих вычислений будут выводиться с 14 знаками после десятичной точки, а числа, не укладывающиеся в этот формат, будут выводиться в экспоненциальной форме (формат **Long E**).

Ввод матриц в командном окне может быть осуществлен одним из следующих способов:

– поэлементный ввод матриц осуществляется в виде последовательности элементов, заключенной в квадратные скобки **[…]**. Разделителями элементов в строке матрицы являются пробелы, разделителями строк матрицы являются символы «**;**» (рис. 2.4);
– ввод матриц в виде последовательности монотонно возрастающих или убывающих значений элементов (рис. 2.5);
– ввод матриц специального вида (матрицы с нулевыми элементами, матрицы с единичными элементами и др. (рис. 2.6).

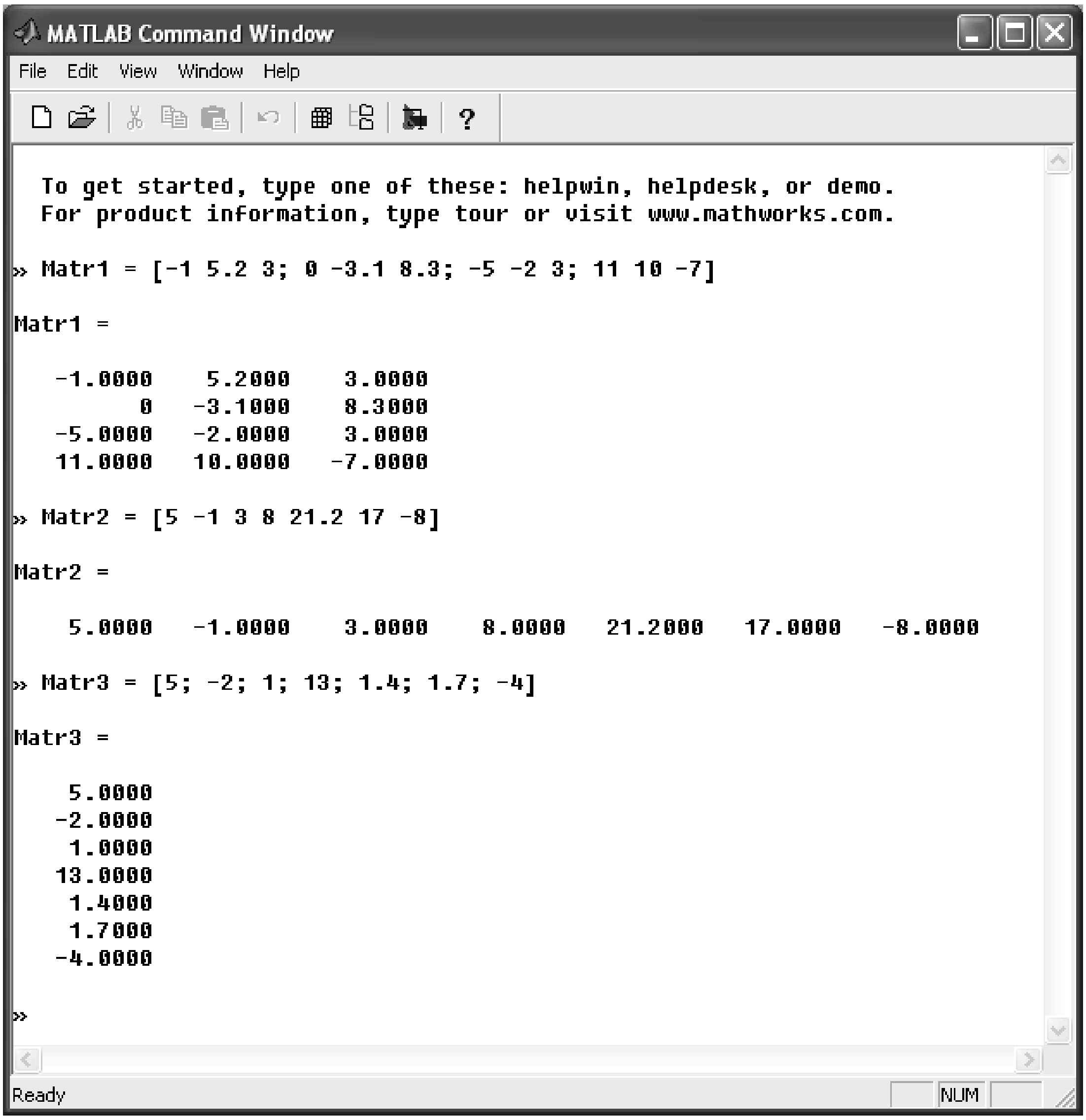


Рис. 2.4. Поэлементный ввод матриц

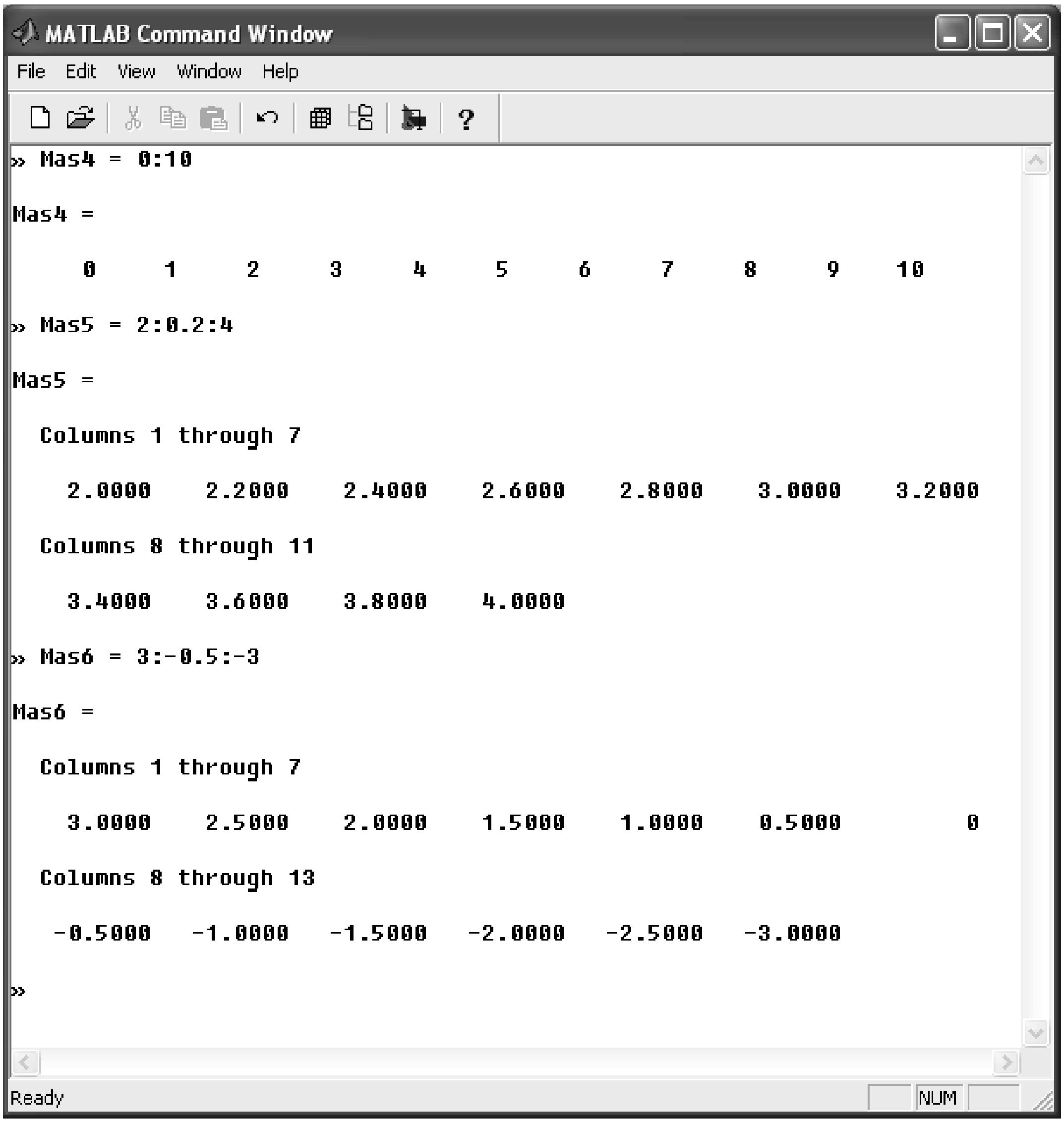


Рис. 2.5. Ввод матриц в виде последовательности монотонно возрастающих (убывающих) значений элементов

При вводе матриц в виде последовательности монотонно возрастающих или убывающих значений элементов (см. рис. 2.5) использовать квадратные скобки не обязательно. Ввод осуществляется в следующей последовательности: начальное значение, двоеточие, приращение, двоеточие, конечное значение. Если приращение равно 1, последовательность ввода может быть упрощена: начальное значение, двоеточие, конечное значение.

Приведенные выше способы ввода матриц в командном окне могут комбинироваться в любых сочетаниях, как показано на примере, приведенном на рис. 2.7.

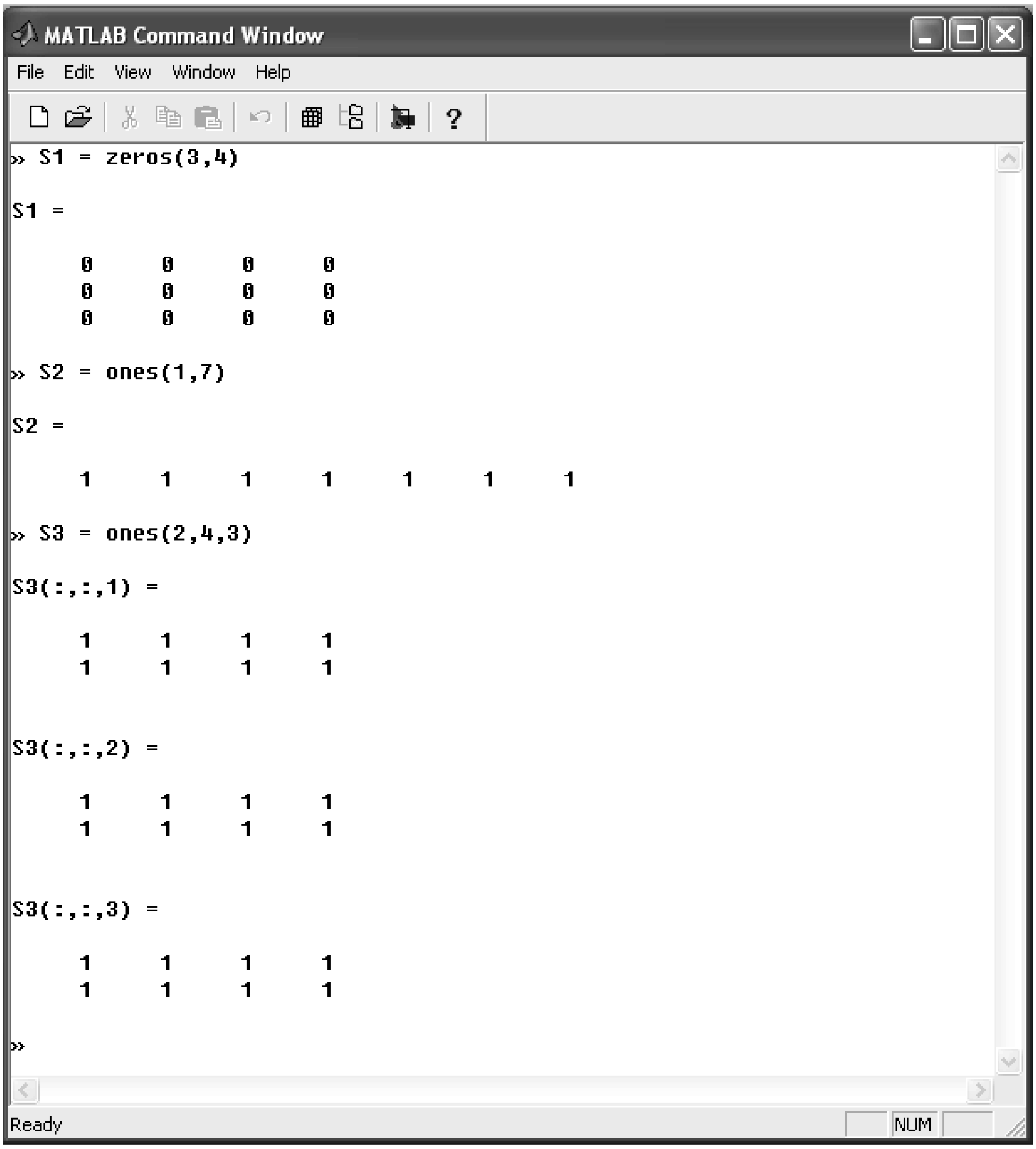


Рис. 2.6. Ввод матриц специального вида

Выражения, содержащие операции с матрицами (сложение, вычитание, умножение, деление матриц), вводятся аналогично арифметическим операциям с числами, но выполняются по соответствующим правилам матричных преобразований. Примеры выполнения таких преобразований приведены на рис. 2.8, 2.9.

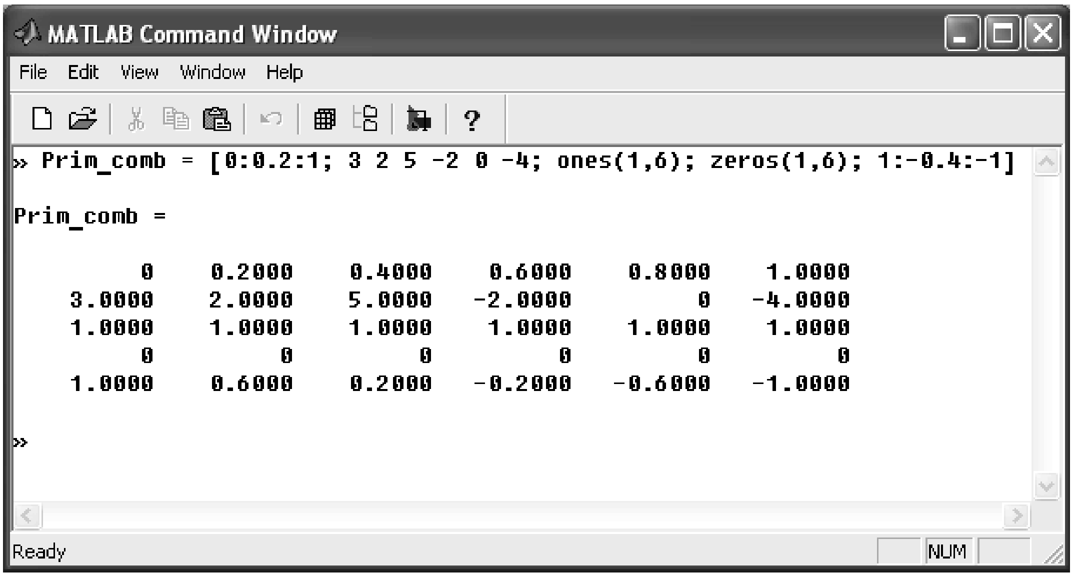


Рис. 2.7. Комбинированный ввод матриц

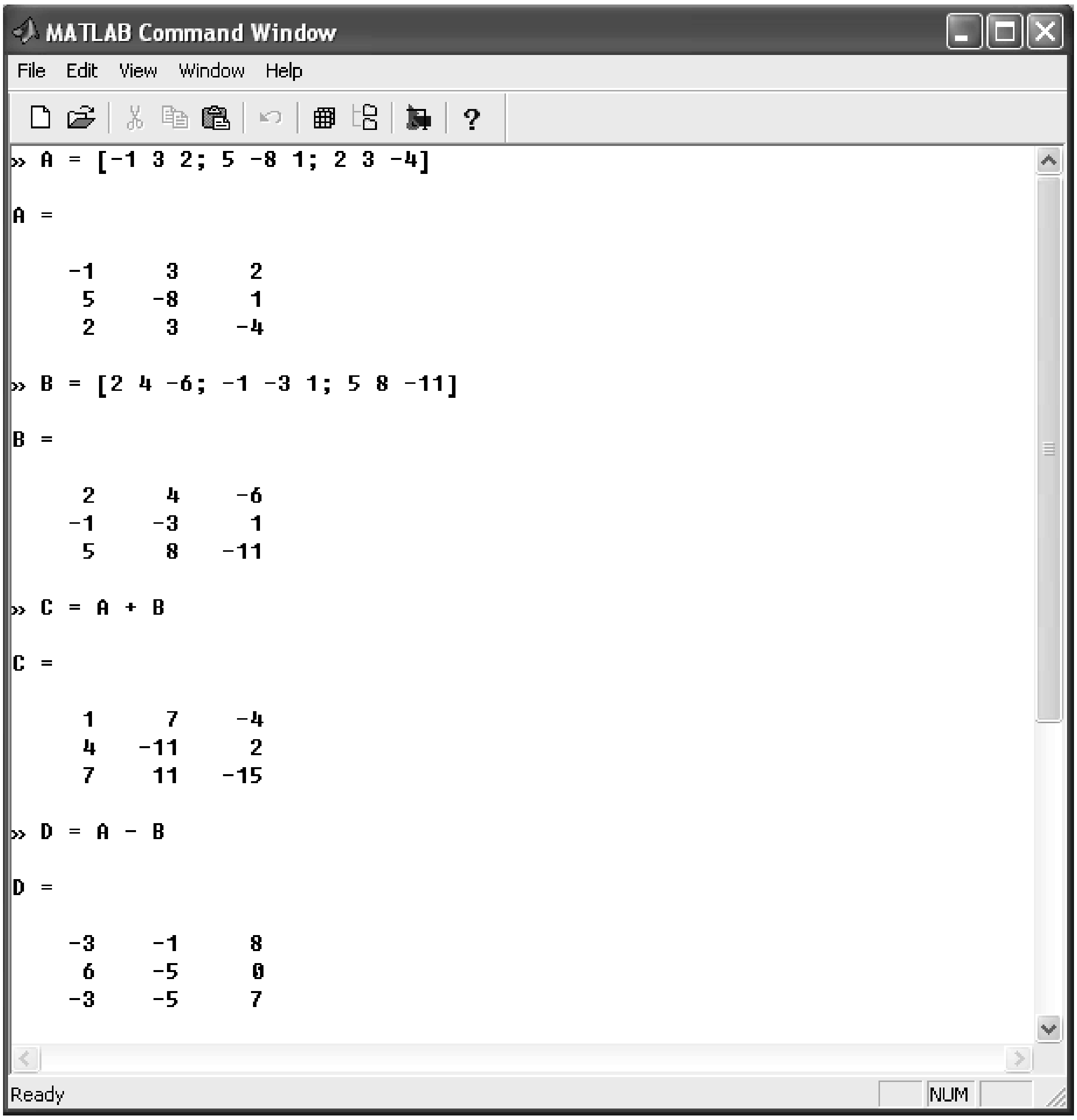


Рис. 2.8. Сложение и вычитание матриц

При выполнении почленного умножения и деления матриц или почленного умножения или деления матрицы на число, необходимо ввести точку перед символом операции, как показано в примерах на рис. 2.10.

При использовании имени матрицы в качестве аргумента встроенной функции MATLAB (например, **sin(A), exp(-B), log2(C), abs(D)**) соответствующая математическая операция применяется почленно к каждому элементу матрицы и результат возвращается в виде матрицы такого же размера, как и исходная матрица.

При выполнении операций сложения, вычитания, умножения и деления матриц следует помнить о необходимости согласования размерностей исходных матриц в соответствии с правилами выполнения данных математических операций. В противном случае MATLAB будет выводить в командном окне сообщение об ошибке.

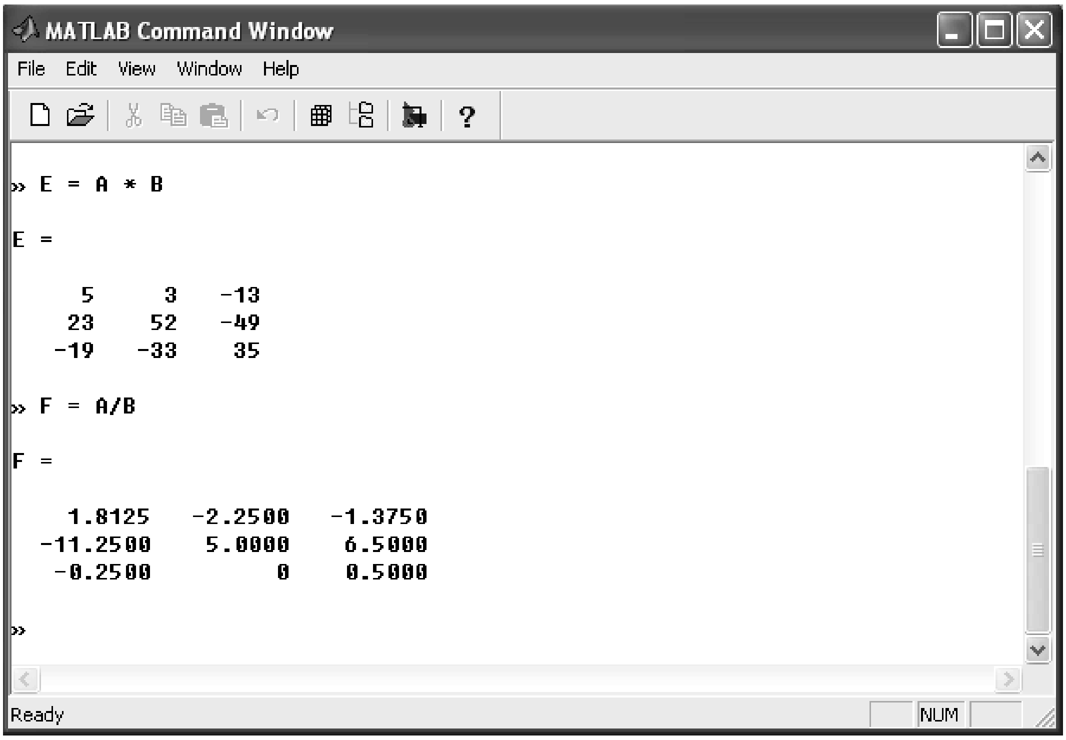


Рис. 2.9. Умножение и деление матриц

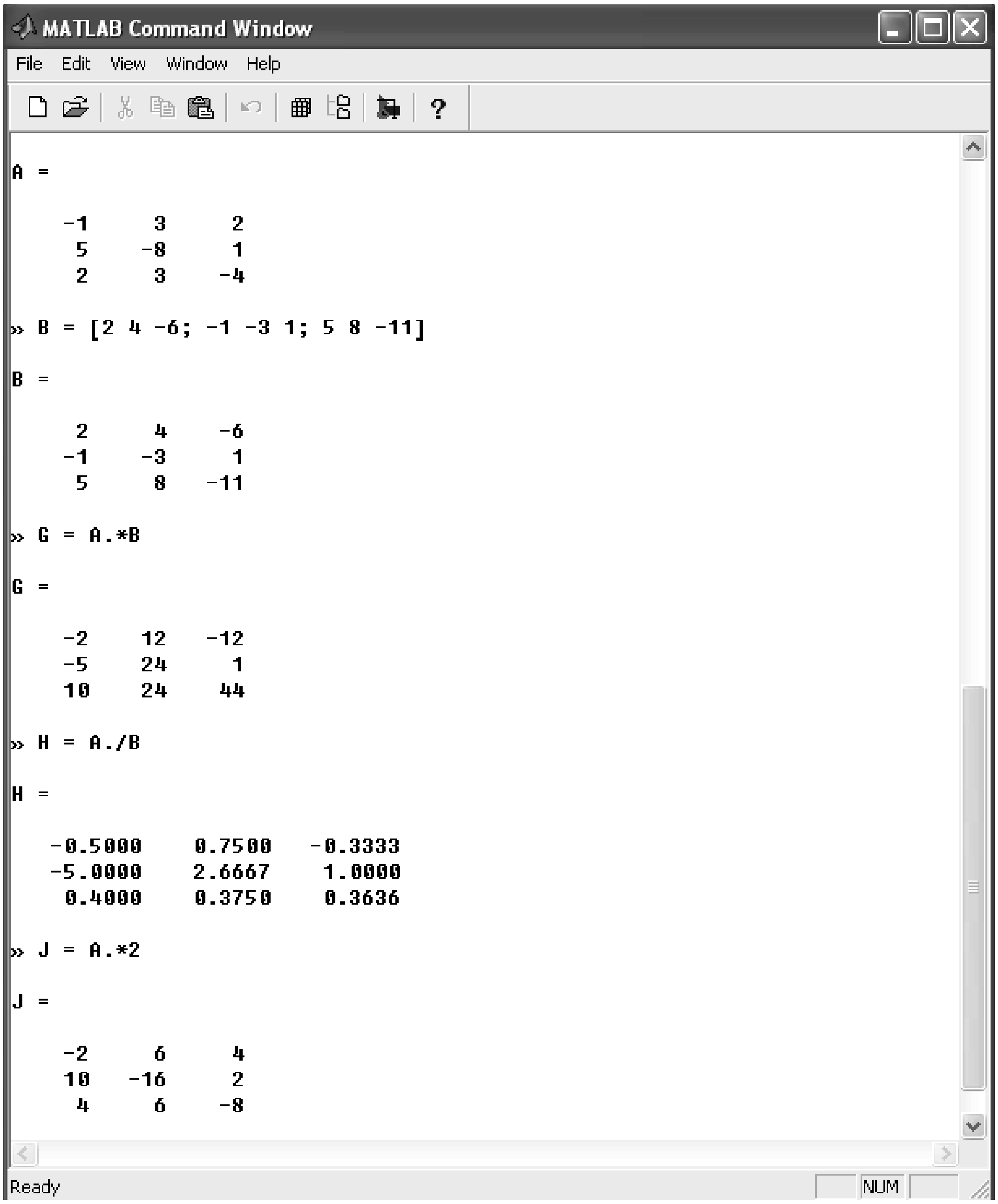


Рис. 2.10. Почленное умножение и деление матриц
и умножение матрицы на число

2.2. Реализация алгоритмов в виде m-файлов

В среде программирования системы MATLAB исходные файлы программ по умолчанию сохраняются с расширением ***.m**. Поэтому их принято называть m-файлами.

Реализацию алгоритмов решения задач математической физики в MATLAB в виде m-файлов, не содержащих функций, созданных пользователем, покажем на примере решения уравнения Пуассона, к решению которого сводятся многие задачи математической физики, например задачи о стационарном распределении температуры в твердом теле, задачи диффузии, задачи о распределении электростатического поля в непроводящей среде при наличии электрических зарядов и многие другие.

Уравнение Пуассона относится к уравнениям эллиптического типа и в одномерном случае имеет вид [1]

$$\frac{\partial}{\partial x}\left(A(x)\,\frac{\partial u}{\partial x}\right)= f(x)\,, \tag{2.1}$$

где x – координата;

$u(x)$ – искомая функция;

$A(x)$, $f(x)$ – некоторые непрерывные функции координаты.

Решим одномерное уравнение Пуассона для случая A = 1, которое при этом принимает вид

$$\frac{\partial^2 u}{\partial x^2}= f(x)\ . \tag{2.2}$$

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

$$\mathbf{x} =\{x_i | i=1,2,...,n\}\ . \tag{2.3}$$

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

$$u\ (x_1) = g_1; \tag{2.4}$$

$$u\ (x_n) = g_2, \tag{2.5}$$

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

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

$$\left.\frac{du}{dx}\right|_{x_1} = g_1; \tag{2.6}$$

$$\left.\frac{du}{dx}\right|_{x_n} = g_2. \tag{2.7}$$

Проводя дискретизацию граничных условий Дирихле на равномерной координатной сетке (2.3) с использованием метода конечных разностей, получим

$$u_1 = g_1; \tag{2.8}$$

$$u_n = g_2, \tag{2.9}$$

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

Проводя дискретизацию граничных условий Неймана на сетке (2.3), получим

$$\frac{u_2 - u_1}{\Delta x} = g_1; \qquad (2.10)$$

$$\frac{u_n - u_{n-1}}{\Delta x} = g_2. \qquad (2.11)$$

Проводя дискретизацию уравнения (2.2) для внутренних точек сетки, получим

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} = f_i, \quad i = 2, \ldots, n-1, \qquad (2.12)$$

где u_i, f_i – значения функций $u(x)$, $f(x)$ в точке сетки с координатой x_i.

Таким образом, в результате дискретизации получим систему линейных алгебраических уравнений размерностью n, содержащую n – 2 уравнения вида (2.12) для внутренних точек области и уравнения (2.8) или (2.10) и (2.9) или (2.11) для двух граничных точек [1].

Ниже приведен один из вариантов m-файла для численного решения уравнения (2.2) с граничными условиями (2.4) – (2.7) на координатной сетке (2.3). В MATLAB строки, начинающиеся символом «%», являются комментариями.

```
% Очистка ранее сохраненных результатов вычислений
% из оперативной памяти

clear all

% Закрытие ранее выведенных графических окон

close all

% Очистка экрана

clc

% Ввод исходных данных с клавиатуры

input('Начальная координата области решения: ');
x0=ans;
input('Конечная координата области решения: ');
xn=ans;
input('Число точек координатной сетки: ');
n=ans;
input('Функция правой части уравнения в одинарных кавычках: ');
f=ans;
```

```
input('Вид ГУ на левой границе (1 - Дирихле, 2 - Неймана): ');
v1=ans;
input('Значение ГУ на левой границе: ');
g1=ans;
input('Вид ГУ на правой границе (1 - Дирихле, 2 - Неймана): ');
v2=ans;
input('Значение ГУ на правой границе: ');
g2=ans;

% Задание равномерной координатной сетки с шагом dx

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');
FF=F(x);

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

a=zeros(n,n);

% Задание матрицы-строки свободных членов СЛАУ размерностью 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;
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;
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)=FF(i);
end

% Решение СЛАУ

u=b/a';

% Построение графика функции правой части f(x)

plot(x,FF,x,ones(size(x)).*(sum(FF)/length(FF)),'r.','LineWidth',1.5)
xlabel('x','FontSize',13)
ylabel('f(x)','FontSize',13)
grid on

% Построение графика искомой функции u(x)

figure
plot(x,u,x,ones(size(x)).*(sum(u)/length(u)),'r.','LineWidth',1.5)
xlabel('x','FontSize',13)
ylabel('U(x)','FontSize',13)
grid on
```

При запуске m-файла на выполнение в оперативной памяти компьютера могут храниться результаты предыдущих вычислений, причем имена переменных и массивов, значения которых хранятся в оперативной памяти, в принципе могут совпасть с именами переменных или матриц запускаемого m-файла, что при определенном стечении обстоятельств может привести к неверному результату производимых вычислений. Это возможно, поскольку все переменные и массивы, используемые в m-файлах, по умолчанию являются глобальными. Во избежание данных нежелательных моментов, первые три команды m-файла производят очистку оперативной памяти от результатов предыдущих вычислений, закрывают все графические окна (если таковые были ранее открыты) и очищают экран от ранее выведенной информации.

В данном варианте программы предусмотрен ввод исходных данных с клавиатуры с помощью функции **input**. Данная функция выводит на экран строку символов, являющуюся ее входным аргументом, позволяет пользователю ввести с клавиатуры произвольный набор символов и после нажатия клавиши **<Enter>** записывает введенные символы в специальную переменную **ans**, о которой упоминалось выше. Далее в m-файле предусмотрен оператор присвоения значения переменной **ans** другой переменной с определенным именем.

Функция правой части уравнения Пуассона **f(x)** задается в данном варианте символьно, а затем преобразуется в вектор значений в соответствии с вектором координатной сетки **x**.

Решение системы линейных алгебраических уравнений (СЛАУ), заданной матрицей коэффициентов **a** и вектором свободных членов **b** и полученной в результате дискретизации уравнения Пуассона на равномерной координатной сетке **x**, производится путем деления вектора-строки **b** на транспонированную матрицу **a**. Операция транспонирования матрицы обозначается символом апострофа после имени матрицы.

Функция **plot** обеспечивает вывод результатов вычислений в виде графиков функций одной переменной. Первым аргументом функции **plot** является вектор, определяющий значения по оси абсцисс, а вторым – вектор, определяющий значения по оси ординат. Таких пар аргументов в функции **plot** может быть несколько. Соответственно в этом случае в графическом окне будет выведено несколько графиков, как в приведенном примере. Кроме того, функция **plot** может содержать (и в приведенном примере содержит) дополнительные аргументы, управляющие цветом, символьным сопровождением и шириной линий графиков.

Функции **xlabel**, **ylabel** позволяют вывести на экран наименования осей координат. Функция **grid** позволяет включить или отключить отображение координатной сетки на графике (при помощи параметров **on** и **off** соответственно).

Для создания m-файла необходимо активизировать команду **New** меню **File** и выбрать в ниспадающем меню команду **M-file**. При этом на экране появится рабочее окно встроенного текстового редактора MATLAB, в котором следует набрать приведенный выше текст программы и сохранить его с произвольным именем. Расширение ***.m** будет присвоено файлу по умолчанию. Также по умолчанию файл будет сохранен в подкаталоге **WORK** корневого каталога MATLAB. При задании имени файла следует использовать только буквы латинского алфавита.

Для вызова m-файла на выполнение необходимо в командном окне MATLAB набрать имя файла без расширения и нажать клавишу **<Enter>**.

Например, при запуске m-файла и вводе исходных данных

```
x0=0;
xn=5;
n=60;
f='2*sin(x.^2)+cos(x.^2)';
v1=1;
g1=0;
v2=1;
g2=-0.5;
```

на экране в отдельных графических окнах появятся график функции правой части уравнения Пуассона и график искомой функции (рис. 2.11, 2.12). Точками на графиках будет отображаться координатная сетка.

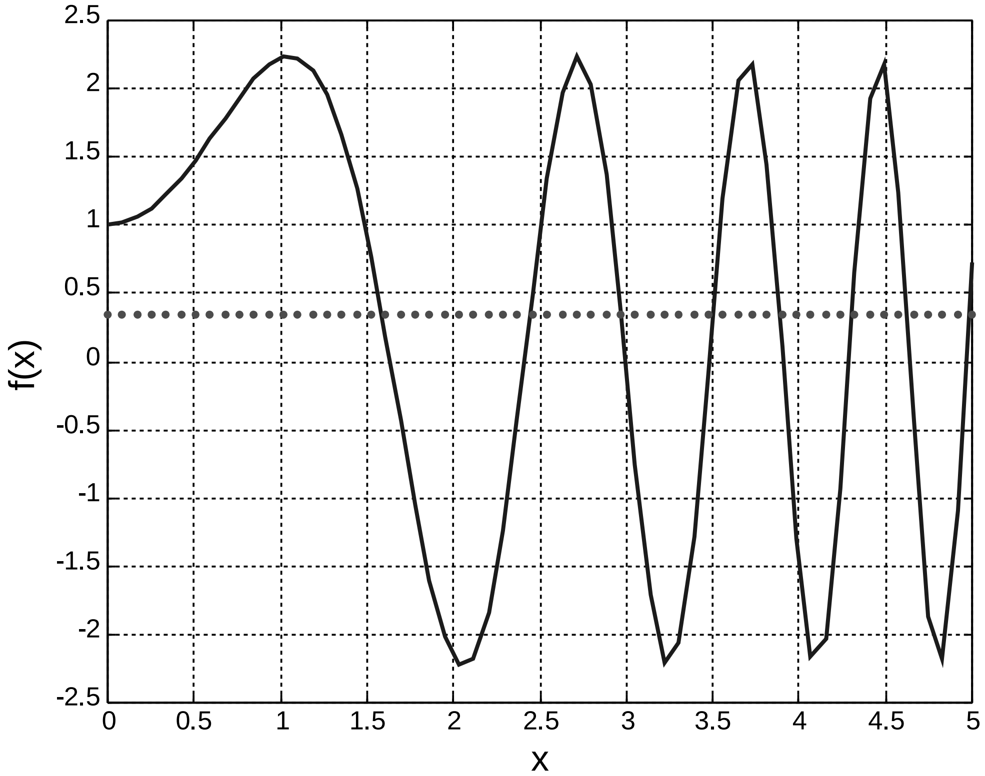


Рис. 2.11. График функции правой части уравнения Пуассона

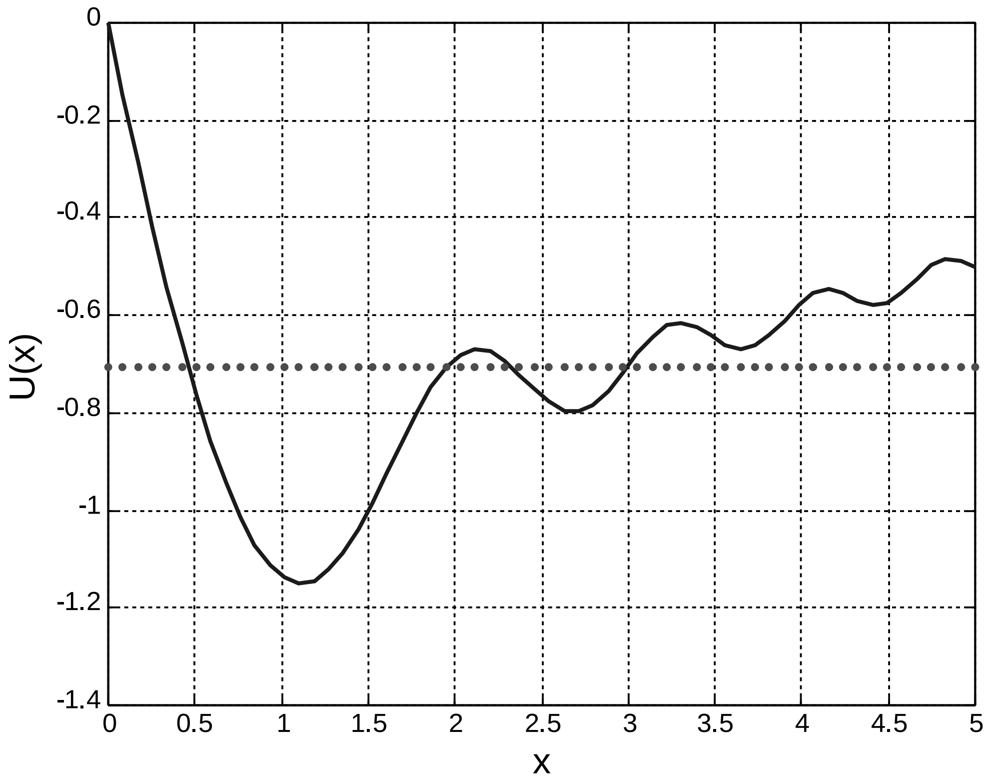


Рис. 2.12. График искомой функции уравнения Пуассона

2.3. Реализация алгоритмов в виде функций

Реализацию алгоритмов решения задач математической физики в MATLAB в виде функций, созданных пользователем, покажем на примере решения волнового уравнения, описывающего незатухающие колебания в некоторой среде.

Волновое уравнение относится к уравнениям гиперболического типа и в двухмерном случае имеет вид [1]

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \qquad (2.13)$$

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

$$\mathbf{x} = \{x_i | i = 1, 2, ..., n\}, \qquad (2.14)$$

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

$$\mathbf{y} = \{y_j | j = 1, 2, ..., m\}, \qquad (2.15)$$

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

$$\mathbf{t} = \{t_l | l = 1, 2, ..., s\}. \qquad (2.16)$$

Векторы, заданные выражениями (2.14) – (2.16), определяют на прямоугольной области равномерную пространственно-временную сетку:

$$G = \{(x_i = i\Delta x,\ y_j = j\Delta y,\ t_l = l\Delta t),\ |\ i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, m,\ l = 1, 2, \ldots, s\}. \qquad (2.17)$$

Граничные условия первого рода (Дирихле) для рассматриваемой задачи представим в виде

$$u(x_1, y, t) = g_1(y); \qquad (2.18)$$

$$u(x_n, y, t) = g_2(y); \qquad (2.19)$$

$$u(x, y_1, t) = g_3(x); \qquad (2.20)$$

$$u(x, y_m, t) = g_4(x), \qquad (2.21)$$

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 представим в виде

$$\left.\frac{\partial u}{\partial x}\right|_{x_1, y, t} = g_1(y) \; ; \tag{2.22}$$

$$\left.\frac{\partial u}{\partial x}\right|_{x_n, y, t} = g_2(y) \; ; \tag{2.23}$$

$$\left.\frac{\partial u}{\partial y}\right|_{x, y_1, t} = g_3(x) \; ; \tag{2.24}$$

$$\left.\frac{\partial u}{\partial y}\right|_{x, y_m, t} = g_4(x) \; . \tag{2.25}$$

Начальные условия первого рода для рассматриваемой задачи запишем в виде

$$u(x, y, t_1) = g_{t1}(x, y) \; ; \tag{2.26}$$

$$u(x, y, t_s) = g_{t2}(x, y) \; , \tag{2.27}$$

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 представляются в виде

$$\left.\frac{\partial u}{\partial t}\right|_{x, y, t_1} = g_{t1}(x, y) \; ; \tag{2.28}$$

$$\left.\frac{\partial u}{\partial t}\right|_{x, y, t_S} = g_{t2}(x, y) \; . \tag{2.29}$$

Проводя дискретизацию граничных условий Дирихле на равномерной сетке (2.17) с использованием метода конечных разностей, получим

$$u_{1,j,l} = g_1(y_j) \; ; \tag{2.30}$$

$$u_{n,j,l} = g_2(y_j) \; ; \tag{2.31}$$

$$u_{i,1,l} = g_3(x_i) \; ; \tag{2.32}$$

$$u_{i,m,l} = g_4(x_i) \; , \tag{2.33}$$

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 (2.17), получим

$$\frac{u_{2,j,l} - u_{1,j,l}}{\Delta x} = g_1(y_j) \; ; \tag{2.34}$$

$$\frac{u_{n,j,l} - u_{n-1,j,l}}{\Delta x} = g_2(y_j) \; ; \tag{2.35}$$

$$\frac{u_{i,2,l} - u_{i,1,l}}{\Delta y} = g_3(x_i) \; ; \tag{2.36}$$

$$\frac{u_{i,m,l} - u_{i,m-1,l}}{\Delta y} = g_4(x_i) \; . \tag{2.37}$$

Проводя дискретизацию начальных условий первого рода, получим

$$u_{i,j,1} = g_{t1}(x_i, y_j) \; ; \tag{2.38}$$

$$u_{i,j,s} = g_{t2}(x_i, y_j) \; , \tag{2.39}$$

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, получим

$$\frac{u_{i,j,2} - u_{i,j,1}}{\Delta t} = g_{t1}(x_i, y_j) \; ; \tag{2.40}$$

$$\frac{u_{i,j,s} - u_{i,j,s-1}}{\Delta t} = g_{t2}(x_i, y_j) \; . \tag{2.41}$$

Проводя дискретизацию волнового уравнения (2.13) для внутренних точек сетки, получим

$$\frac{u_{i,j,l+1}-2u_{i,j,l}+u_{i,j,l-1}}{\Delta t^2}-\frac{u_{i+1,j,l}-2u_{i,j,l}+u_{i-1,j,l}}{\Delta x^2}-$$

$$-\frac{u_{i,j+1,l}-2u_{i,j,l}+u_{i,j-1,l}}{\Delta y^2}=0\,;$$

$$i=2,\ldots,n-1\,;\; j=2,\ldots,m-1\,;\; l=2,\ldots,s-1\,. \qquad (2.42)$$

Таким образом, в результате дискретизации получим систему линейных алгебраических уравнений размерностью $n\times m\times s$.

Ниже приведен один из вариантов функции с комментариями для решения системы (2.30) – (2.42) на равномерной сетке (2.17).

```
% 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]=f_wave2d(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 (ГУ) 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 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 размерностью 1 x s;
% U - матрица значений результирующей функции
%      в узлах координатной сетки размерностью 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 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);
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 размерностью N x N,
% все элементы которой равны 0

a=zeros(N,N);

% Задание матрицы-строки свободных членов СЛАУ размерностью 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
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

% 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','FontSize',13)
   ylabel('x','FontSize',13)
   zlabel('U','FontSize',13)
    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(l)=getframe;
end
for l=s+1:2*s-2
    figure
   surf(y,x,U(:,:,2*s-l))
   xlabel('y','FontSize',13)
   ylabel('x','FontSize',13)
   zlabel('U','FontSize',13)
    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(l)=getframe;
end

% Отображение волнового процесса в динамическом режиме

figure
ans=1;
while ans==1
    movie(M,10,10)
   ans=menu('Повторить просмотр результатов?','ДА','НЕТ');
end
```

Приведенное выше текстовое описание функции сохраняется, как и в предыдущем примере, в виде m-файла с именем **f_wave2d.m**. В отличие от m-файла, приведенного выше, все переменные и массивы в данном случае по умолчанию локальны, т.е. после выполнения функции они автоматически удаляются из оперативной памяти, за исключением возвращаемых функцией значений переменных, имена которых указаны в квадратных скобках после директивы **function**.

Функция может запускаться на выполнение из командной строки или из m-файла. Запуск на выполнение функции осуществляется директивой

```
[x,y,t,U]=f_wave2d(t0,ts,s,x0,xn,n,y0,ym,m,vt1,gt1,vt2,gt2,v1,g1,v2,
        g2,v3,g3,v4,g4);
```

В круглых скобках указываются конкретные значения входных параметров функции. В тексте функции предусмотрены значения входных параметров по умолчанию. Поэтому при вызове без списка входных параметров функция будет выполняться со значениями по умолчанию.

Графики распределений искомой функции по координатам в различные моменты времени отображаются в отдельных графических окнах. После вывода всех графиков в новом окне выводится волновой процесс в динамическом режиме **movie**, после чего на экране появляется меню, приведенное на рис. 2.13, которое предоставляет возможность повторного просмотра результатов в динамике.

На рис. 2.14 представлены графики искомой функции в различные моменты времени для значений входных параметров функции по умолчанию. Результаты вычисляются и выводятся в нормированном (безразмерном) виде. При необходимости возможен вывод результатов решения волнового уравнения в реальных физических единицах. Для этого необходимо входные параметры функции **surf**, выводящей графики на экран, почленно умножить на нормирующие коэффициенты и добавить соответствующие единицы измерения в строковых аргументах функций **xlabel**, **ylabel**, **zlabel**.

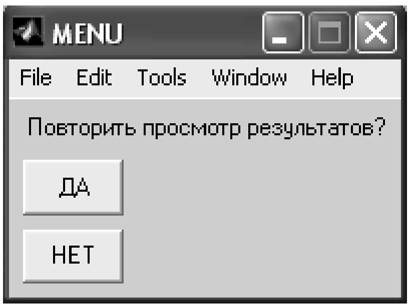


Рис. 2.13. Меню для повторного просмотра результатов

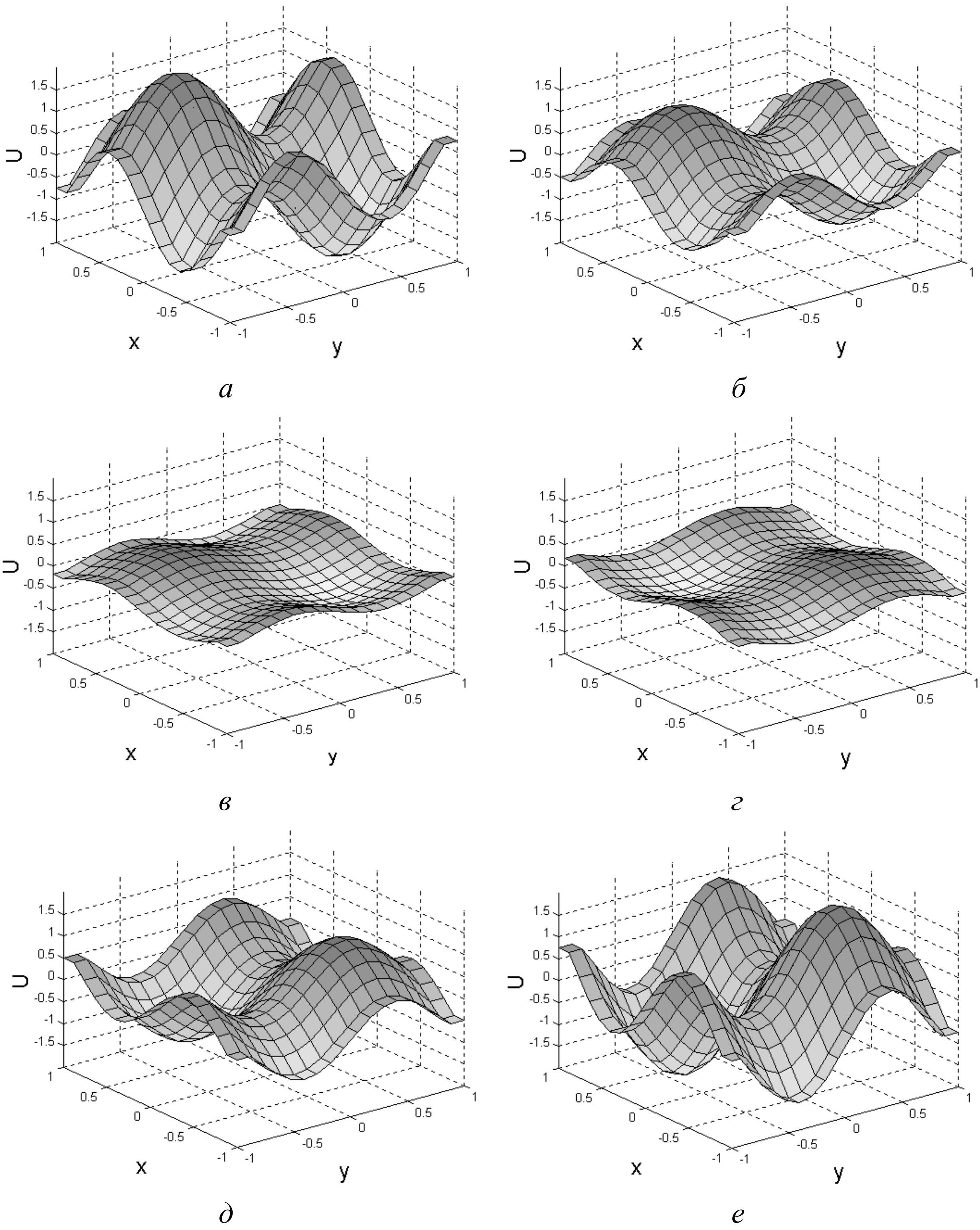


Рис. 2.14. Волновой процесс u(x, y, t) в различные моменты времени:
а) t = 0; б) t = 0,04; в) t = 0,08; г) t = 0,12; д) t = 0,16; е) t = 0,2

MATLAB предоставляет возможность пользователю реализовать разрабатываемую функцию в виде приложения с графическим интерфейсом, содержащим элементы управления (кнопки, списки, переключатели, флаги, полосы скроллинга, области ввода, пользовательские меню), а также координатные оси и текстовые области для вывода полученных результатов.

Создание графического окна, размещение в нем элементов интерфейса и задание связанных с ними команд или функций производится в специальной среде программирования **GUIDE**, переход в которую осуществляется выполнением команды **guide** в командной строке. При этом на экране появляется два окна:

– **Guide Control Panel** (панель управления, рис. 2.15);
– **Figure No**. **1** (заготовка для окна приложения, рис. 2.16).

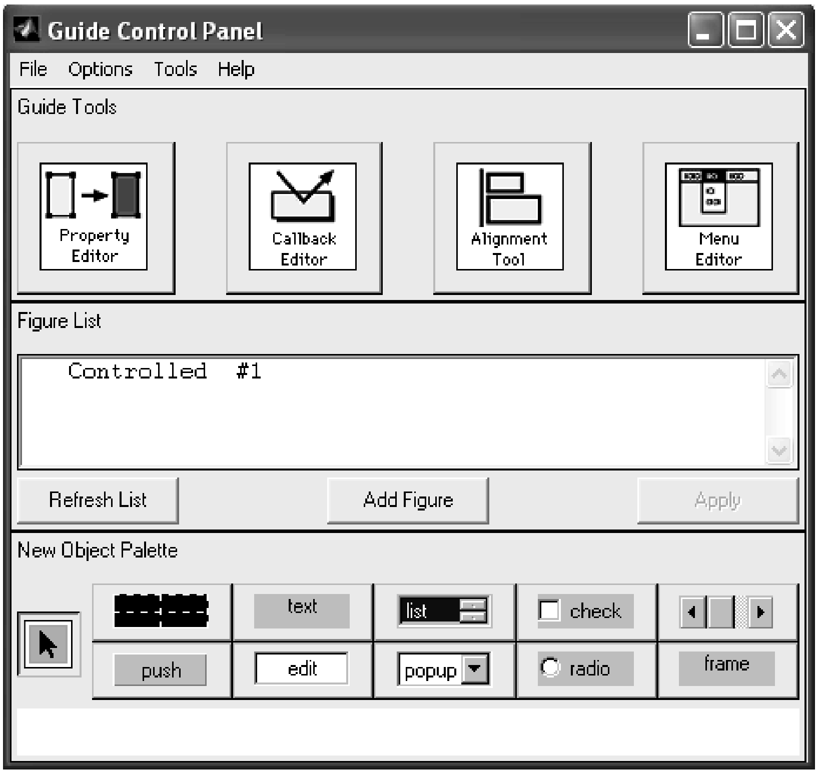


Рис. 2.15. Окно Guide Control Panel

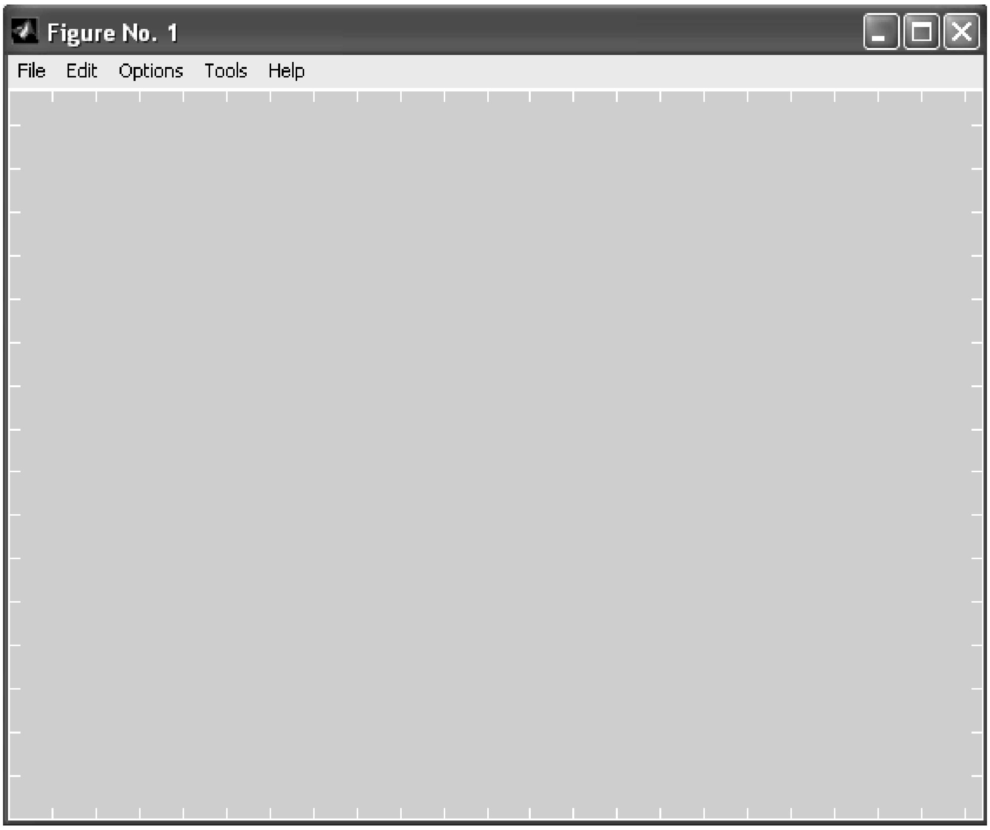


Рис. 2.16. Заготовка для окна приложения пользователя

Создание приложений включает расположение и модификацию требуемых элементов интерфейса из окна **Guide Control Panel** в пределах графического окна заготовки приложения и определение действий (команд, функций), которые выполняются при обращении пользователя к данным элементам интерфейса. Процесс работы над приложением допускает постепенное добавление элементов в графическое окно, запуск и тестирование приложения и возврат в режим редактирования. Конечным результатом является функция с графическим интерфейсом пользователя, содержащаяся в нескольких файлах, запуск которой осуществляется указанием ее имени в командной строке или в другом приложении MATLAB [2].

В качестве примера ниже приводятся исходные тексты функций с графическим интерфейсом пользователя, осуществляющие решение нестационарного уравнения теплопроводности для случая двух измерений:

$$\rho(x,y)\, C(x,y)\, \frac{\partial T}{\partial t} - \nabla(k(x,y)\nabla T) = f(x,y)\,, \qquad (2.43)$$

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 прямоугольной области с граничными условиями Дирихле или Неймана на границах $x = x_{min}$, $x = x_{max}$, $y = y_{min}$, $y = y_{max}$ и с начальными условиями первого или второго рода на отрезке времени $[t_{min}, t_{max}]$.

Зададим на прямоугольной области равномерную пространственно-временную сетку:

$$G=\{(x_i = i\Delta x,\ y_j = j\Delta y,\ t_l = l\Delta t),|\ i = 1, 2, \dots, n,\ j = 1, 2, \dots, m,\ l = 1, 2, \dots, s\}. \quad (2.44)$$

Граничные условия первого рода (Дирихле) для рассматриваемой задачи имеют вид

$$T(x_1, y, t) = g_1(y); \quad (2.45)$$

$$T(x_n, y, t) = g_2(y); \quad (2.46)$$

$$T(x, y_1, t) = g_3(x); \quad (2.47)$$

$$T(x, y_m, t) = g_4(x), \quad (2.48)$$

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 имеют вид

$$\left.\frac{\partial T}{\partial x}\right|_{x_1, y, t} = g_1(y); \quad (2.49)$$

$$\left.\frac{\partial T}{\partial x}\right|_{x_n, y, t} = g_2(y); \quad (2.50)$$

$$\left.\frac{\partial T}{\partial y}\right|_{x, y_1, t} = g_3(x); \quad (2.51)$$

$$\left.\frac{\partial T}{\partial y}\right|_{x, y_m, t} = g_4(x). \quad (2.52)$$

Начальные условия первого рода представляются в виде

$$T(x, y, t_1) = g_t(x, y), \tag{2.53}$$

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

Начальные условия второго рода имеют вид

$$\left.\frac{\partial T}{\partial t}\right|_{x, y, t_1} = g_t(x, y). \tag{2.54}$$

Проводя дискретизацию граничных условий Дирихле на равномерной сетке (2.44), получим

$$T_{1,j,l} = g_1(y_j); \tag{2.55}$$

$$T_{n,j,l} = g_2(y_j); \tag{2.56}$$

$$T_{i,1,l} = g_3(x_i); \tag{2.57}$$

$$T_{i,m,l} = g_4(x_i), \tag{2.58}$$

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 (2.44), получим

$$\frac{T_{2,j,l} - T_{1,j,l}}{\Delta x} = g_1(y_j); \tag{2.59}$$

$$\frac{T_{n,j,l} - T_{n-1,j,l}}{\Delta x} = g_2(y_j); \tag{2.60}$$

$$\frac{T_{i,2,l} - T_{i,1,l}}{\Delta y} = g_3(x_i); \tag{2.61}$$

$$\frac{T_{i,m,l} - T_{i,m-1,l}}{\Delta y} = g_4(x_i). \tag{2.62}$$

Проводя дискретизацию начальных условий первого рода, получим

$$T_{i,j,1} = g_t(x_i, y_j), \tag{2.63}$$

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, получим

$$\frac{T_{i,j,2}-T_{i,j,1}}{\Delta t}=g_t(x_i,y_j)\,. \tag{2.64}$$

Дискретизируя уравнение (2.43) для внутренних точек сетки, получим

$$\rho_{i,j,l}C_{i,j,l}\frac{T_{i,j,l}-T_{i,j,l-1}}{\Delta t}-$$

$$-\frac{1}{\Delta x^2}\left[k_{i,j,l}\left(T_{i+1,j,l}-T_{i,j,l}\right)-k_{i-1,j,l}\left(T_{i,j,l}-T_{i-1,j,l}\right)\right]-$$

$$-\frac{1}{\Delta y^2}\left[k_{i,j,l}\left(T_{i,j+1,l}-T_{i,j,l}\right)-k_{i,j-1,l}\left(T_{i,j,l}-T_{i,j-1,l}\right)\right]=f_{i,j,l},$$

$$i=2,\ldots,n-1\ ;\ j=2,\ldots,m-1\ ;\ l=2,\ldots,s\ , \tag{2.65}$$

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

Ниже приводится исходный текст функции, осуществляющей запуск графического интерфейса пользователя **i_termo**.

```
function fig = i_termo()

load i_termo

h0 = figure('Color',[0.8 0.8 0.8], ...
   'Colormap',mat0, ...
   'FileName','D:\CAD\MATLABR11\work\i_termo.m', ...
   'MenuBar','none', ...
   'Name','Решение нестационарного уравнения теплопроводности', ...
   'NumberTitle','off', ...
   'PaperPosition',[18 180 576 432], ...
   'PaperUnits','points', ...
   'Position',[460 36 560 653], ...
   'Tag','Fig1', ...
   'ToolBar','none');
h1 = axes('Parent',h0, ...
   'Units','pixels', ...
   'CameraUpVector',[0 1 0], ...
   'CameraUpVectorMode','manual', ...
   'Color',[1 1 1], ...
   'ColorOrder',mat1, ...
   'Position',[52 296 353 321], ...
   'Tag','Axes1', ...
   'XColor',[0 0 0], ...
   'YColor',[0 0 0], ...
   'ZColor',[0 0 0]);
h2 = text('Parent',h1, ...
```

```
   'Color',[0 0 0], ...
   'HandleVisibility','off', ...
   'HorizontalAlignment','center', ...
   'Position',[0.497159090909090 -0.0749999999999996
9.160254037844386], ...
   'Tag','Axes1Text4', ...
   'VerticalAlignment','cap');
set(get(h2,'Parent'),'XLabel',h2);
h2 = text('Parent',h1, ...
   'Color',[0 0 0], ...
   'HandleVisibility','off', ...
   'HorizontalAlignment','center', ...
   'Position',[-0.08238636363636363 0.496875 9.160254037844386], ...
   'Rotation',90, ...
   'Tag','Axes1Text3', ...
   'VerticalAlignment','baseline');
set(get(h2,'Parent'),'YLabel',h2);
h2 = text('Parent',h1, ...
   'Color',[0 0 0], ...
   'HandleVisibility','off', ...
   'HorizontalAlignment','right', ...
   'Position',[-0.1477272727272727 1.1125 9.160254037844386], ...
   'Tag','Axes1Text2', ...
   'Visible','off');
set(get(h2,'Parent'),'ZLabel',h2);
h2 = text('Parent',h1, ...
   'Color',[0 0 0], ...
   'HandleVisibility','off', ...
   'HorizontalAlignment','center', ...
   'Position',[0.4971590909090909 1.021875 9.160254037844386], ...
   'Tag','Axes1Text1', ...
   'VerticalAlignment','bottom');
set(get(h2,'Parent'),'Title',h2);
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
   'Callback','f_termo2d2(''press_Run'');', ...
   'ListboxTop',0, ...
   'Position',[20 20 60 30], ...
   'String','Решение', ...
   'Tag','button_Run');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'Callback','f_termo2d2(''press_Quit'');', ...
   'ListboxTop',0, ...
   'Position',[350 20 45 15], ...
   'String','Выход', ...
   'Tag','button_Quit');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'Callback','f_termo2d2(''press_Rep'');', ...
   'Enable','off', ...
   'ListboxTop',0, ...
   'Position',[95 20 60 30], ...
   'String','Повторить', ...
```

```
   'Tag','button_Rep');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'Callback','f_termo2d2(''press_Figures'');', ...
   'Enable','off', ...
   'ListboxTop',0, ...
   'Position',[175 20 60 30], ...
   'String','По кадрам', ...
   'Tag','button_Figures');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'ListboxTop',0, ...
   'Position',[310 425 90 33], ...
   'Style','frame', ...
   'Tag','Frame1');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
   'Callback','f_termo2d2(''press_Grid'');', ...
   'Enable','off', ...
   'ListboxTop',0, ...
   'Position',[320 435 75 15], ...
   'String','Включить', ...
   'Style','checkbox', ...
   'Tag','box_Grid', ...
   'UserData','[ ]');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'ListboxTop',0, ...
   'Position',[310 330 90 65], ...
   'Style','frame', ...
   'Tag','Frame2');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
   'Callback','f_termo2d2(''press_Cool'');', ...
   'Enable','off', ...
   'ListboxTop',0, ...
   'Position',[315 375 75 15], ...
   'String','Оттенки синего', ...
   'Style','radiobutton', ...
   'Tag','Radio_Cool');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'Callback','f_termo2d2(''press_Hot'');', ...
   'Enable','off', ...
   'ListboxTop',0, ...
   'Position',[315 355 75 15], ...
   'String','Оттенки красного', ...
   'Style','radiobutton', ...
   'Tag','Radio_Hot', ...
   'Value',1);
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
```

```
    'Callback','f_termo2d2(''press_Default'');', ...
    'Enable','off', ...
    'ListboxTop',0, ...
    'Position',[315 335 75 15], ...
    'String','Радуга', ...
    'Style','radiobutton', ...
    'Tag','Radio_Default');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[310 460 90 15], ...
    'String','Координатная сетка', ...
    'Style','text', ...
    'Tag','StaticText1');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[310 395 90 15], ...
    'String','Палитра', ...
    'Style','text', ...
    'Tag','StaticText1');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','f_termo2d2(''press_t0'');', ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[23.25 169.5 45 15], ...
    'Style','edit', ...
    'Tag','Edit_t0');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','f_termo2d2(''press_ts'');', ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[24.75 134.25 45 15], ...
    'Style','edit', ...
    'Tag','Edit_ts');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','f_termo2d2(''press_s'');', ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[25 100 45 15], ...
    'Style','edit', ...
    'Tag','Edit_s');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','f_termo2d2(''press_n'');', ...
```

```
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[84.75 99 45 15], ...
    'Style','edit', ...
    'Tag','Edit_n');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','f_termo2d2(''press_xn'');', ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[84.75 134.25 45 15], ...
    'Style','edit', ...
    'Tag','Edit_xn');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','f_termo2d2(''press_x0'');', ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[84.75 169.5 45 15], ...
    'Style','edit', ...
    'Tag','Edit_x0');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','f_termo2d2(''press_y0'');', ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[144.75 169.5 45 15], ...
    'Style','edit', ...
    'Tag','Edit_y0');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','f_termo2d2(''press_ym'');', ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[144.75 134.25 45 15], ...
    'Style','edit', ...
    'Tag','Edit_ym');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','f_termo2d2(''press_m'');', ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[144.75 99 45 15], ...
    'Style','edit', ...
    'Tag','Edit_m');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','f_termo2d2(''press_r'');', ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
```

```
   'Position',[210 99.75 65.25 15], ...
   'Style','edit', ...
   'Tag','Edit_r');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'Callback','f_termo2d2(''press_k'');', ...
   'HorizontalAlignment','left', ...
   'ListboxTop',0, ...
   'Position',[210 134.25 65.25 15], ...
   'Style','edit', ...
   'Tag','Edit_k');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'Callback','f_termo2d2(''press_c'');', ...
   'HorizontalAlignment','left', ...
   'ListboxTop',0, ...
   'Position',[210 169.5 65.25 15], ...
   'Style','edit', ...
   'Tag','Edit_c');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'Callback','f_termo2d2(''press_f'');', ...
   'HorizontalAlignment','left', ...
   'ListboxTop',0, ...
   'Position',[24.75 64.5 165 15], ...
   'Style','edit', ...
   'Tag','Edit_f');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'Callback','f_termo2d2(''press_vt'');', ...
   'HorizontalAlignment','left', ...
   'ListboxTop',0, ...
   'Position',[290 170 25 15], ...
   'Style','edit', ...
   'Tag','Edit_vt');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'Callback','f_termo2d2(''press_gt1'');', ...
   'HorizontalAlignment','left', ...
   'ListboxTop',0, ...
   'Position',[320 170 75 15], ...
   'Style','edit', ...
   'Tag','Edit_gt1');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'Callback','f_termo2d2(''press_g3'');', ...
   'HorizontalAlignment','left', ...
   'ListboxTop',0, ...
   'Position',[320 140 75 15], ...
   'Style','edit', ...
```

```
    'Tag','Edit_g3');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','f_termo2d2(''press_v3'');', ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[290 140 25 15], ...
    'Style','edit', ...
    'Tag','Edit_v3');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','f_termo2d2(''press_v4'');', ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[290 110 25 15], ...
    'Style','edit', ...
    'Tag','Edit_v4');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','f_termo2d2(''press_g4'');', ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[320 110 75 15], ...
    'Style','edit', ...
    'Tag','Edit_g4');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','f_termo2d2(''press_g1'');', ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[320 80 75 15], ...
    'Style','edit', ...
    'Tag','Edit_g1');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','f_termo2d2(''press_v1'');', ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[290 80 25 15], ...
    'Style','edit', ...
    'Tag','Edit_v1');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','f_termo2d2(''press_v2'');', ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[290 50 25 15], ...
    'Style','edit', ...
    'Tag','Edit_v2');
h1 = uicontrol('Parent',h0, ...
```

```
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'Callback','f_termo2d2(''press_g2'');', ...
   'HorizontalAlignment','left', ...
   'ListboxTop',0, ...
   'Position',[320.25 49.5 75 15], ...
   'Style','edit', ...
   'Tag','Edit_g2');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.8 0.8 0.8], ...
   'HorizontalAlignment','left', ...
   'ListboxTop',0, ...
   'Position',[24.75 184.5 45 15], ...
   'String','tmin', ...
   'Style','text', ...
   'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.8 0.8 0.8], ...
   'HorizontalAlignment','left', ...
   'ListboxTop',0, ...
   'Position',[24.75 149.25 45 15], ...
   'String','tmax', ...
   'Style','text', ...
   'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.8 0.8 0.8], ...
   'Callback','f_termo2d2(''press_s'');', ...
   'HorizontalAlignment','left', ...
   'ListboxTop',0, ...
   'Position',[25 115 45 15], ...
   'String','dim t', ...
   'Style','text', ...
   'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.8 0.8 0.8], ...
   'HorizontalAlignment','left', ...
   'ListboxTop',0, ...
   'Position',[85.5 114 45 15], ...
   'String','dim x', ...
   'Style','text', ...
   'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.8 0.8 0.8], ...
   'HorizontalAlignment','left', ...
   'ListboxTop',0, ...
   'Position',[85.5 149.25 45 15], ...
   'String','xmax', ...
   'Style','text', ...
   'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
```

```
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[85.5 184.5 45 15], ...
    'String','xmin', ...
    'Style','text', ...
    'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[144.75 184.5 45 15], ...
    'String','ymin', ...
    'Style','text', ...
    'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[145 150 45 15], ...
    'String','ymax', ...
    'Style','text', ...
    'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[144.75 115.5 45 15], ...
    'String','dim y', ...
    'Style','text', ...
    'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[210 115 65 15], ...
    'String','Плотность', ...
    'Style','text', ...
    'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[210 149.25 75 15], ...
    'String','Теплопроводность', ...
    'Style','text', ...
    'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
```

```
    'ListboxTop',0, ...
    'Position',[210 184.5 65 15], ...
    'String','Теплоемкость', ...
    'Style','text', ...
    'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[25.5 80.25 165 15], ...
    'String','Функция в правой части уравнения', ...
    'Style','text', ...
    'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[290.25 184.5 105 15], ...
    'String','gt(x,y), t=tmin', ...
    'Style','text', ...
    'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[290.25 154.5 105 15], ...
    'String','g1(x), y=ymin', ...
    'Style','text', ...
    'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[290.25 124.5 105 15], ...
    'String','g2(x), y=ymax', ...
    'Style','text', ...
    'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[290.25 94.5 105 15], ...
    'String','g3(y), x=xmin', ...
    'Style','text', ...
    'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[290.25 64.5 105 15], ...
```

```
   'String','g4(y), x=xmax', ...
   'Style','text', ...
   'Tag','StaticText2');
if nargout > 0, fig = h0; end
```

Далее приводится исходный текст функции **f_termo2d2**, осуществляющей выполнение определенных команд под управлением пользовательского интерфейса **i_termo**.

```
% Функция решения двухмерного 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]=f_termo2d2(event)
global t0 ts s x0 xn n y0 ym m r c k f vt gt1 v1 g1 v2 g2 v3 g3 v4 ...
g4 x y T;
switch event
case 'press_t0'
   t0=str2num(get(gcbo,'String'));
case 'press_ts'
   ts=str2num(get(gcbo,'String'));
case 'press_s'
   s=str2num(get(gcbo,'String'));
case 'press_x0'
   x0=str2num(get(gcbo,'String'));
case 'press_xn'
   xn=str2num(get(gcbo,'String'));
case 'press_n'
   n=str2num(get(gcbo,'String'));
case 'press_y0'
   y0=str2num(get(gcbo,'String'));
case 'press_ym'
   ym=str2num(get(gcbo,'String'));
case 'press_m'
   m=str2num(get(gcbo,'String'));
case 'press_r'
   r=get(gcbo,'String');
case 'press_c'
   c=get(gcbo,'String');
case 'press_k'
   k=get(gcbo,'String');
case 'press_f'
   f=get(gcbo,'String');
case 'press_vt'
   vt=str2num(get(gcbo,'String'));
case 'press_gt1'
   gt1=get(gcbo,'String');
case 'press_v1'
   v1=str2num(get(gcbo,'String'));
case 'press_g1'
   g1=get(gcbo,'String');
case 'press_v2'
   v2=str2num(get(gcbo,'String'));
```

```
case 'press_g2'
   g2=get(gcbo,'String');
case 'press_v3'
   v3=str2num(get(gcbo,'String'));
case 'press_g3'
   g3=get(gcbo,'String');
case 'press_v4'
   v4=str2num(get(gcbo,'String'));
case 'press_g4'
   g4=get(gcbo,'String');
case 'press_Grid'
   if get(gcbo,'Value')
      grid on
   else
      grid off
   end
case 'press_Cool'
   colormap('cool');
   HRadio_Hot=findobj('Tag','Radio_Hot');
   set(HRadio_Hot,'Value',0);
   HRadio_Default=findobj('Tag','Radio_Default');
   set(HRadio_Default,'Value',0);
case 'press_Hot'
   colormap('hot');
   HRadio_Cool=findobj('Tag','Radio_Cool');
   set(HRadio_Cool,'Value',0);
   HRadio_Default=findobj('Tag','Radio_Default');
   set(HRadio_Default,'Value',0);
case 'press_Default'
   colormap('default');
   HRadio_Hot=findobj('Tag','Radio_Hot');
   set(HRadio_Hot,'Value',0);
   HRadio_Cool=findobj('Tag','Radio_Cool');
   set(HRadio_Cool,'Value',0);
case 'press_Run'
   HRadio_Style=findobj('Style','radiobutton');
   set(HRadio_Style,'Enable','off');
   HCheckbox_Style=findobj('Style','checkbox');
   set(HCheckbox_Style,'Enable','off');
   Hbutton_Rep=findobj('Tag','button_Rep');
   set(Hbutton_Rep,'Enable','off');
   Hbutton_Figures=findobj('Tag','button_Figures');
   set(Hbutton_Figures,'Enable','off');
   HStyle_Edit=findobj('Style','edit');
   if max(strcmp(get(HStyle_Edit,'String'),''))
      errordlg('Не все исходные данные заданы','Ошибка!');
   else
      HEdit_t0=findobj('Tag','Edit_t0');
      t0=str2num(get(HEdit_t0,'String'));
      HEdit_ts=findobj('Tag','Edit_ts');
     ts=str2num(get(HEdit_ts,'String'));
      HEdit_s=findobj('Tag','Edit_s');
     s=str2num(get(HEdit_s,'String'));
      HEdit_x0=findobj('Tag','Edit_x0');
     x0=str2num(get(HEdit_x0,'String'));
      HEdit_xn=findobj('Tag','Edit_xn');
```

```
xn=str2num(get(HEdit_xn,'String'));
 HEdit_n=findobj('Tag','Edit_n');
n=str2num(get(HEdit_n,'String'));
 HEdit_y0=findobj('Tag','Edit_y0');
y0=str2num(get(HEdit_y0,'String'));
 HEdit_ym=findobj('Tag','Edit_ym');
ym=str2num(get(HEdit_ym,'String'));
 HEdit_m=findobj('Tag','Edit_m');
m=str2num(get(HEdit_m,'String'));
 HEdit_r=findobj('Tag','Edit_r');
r=get(HEdit_r,'String');
 HEdit_c=findobj('Tag','Edit_c');
c=get(HEdit_c,'String');
 HEdit_k=findobj('Tag','Edit_k');
k=get(HEdit_k,'String');
 HEdit_f=findobj('Tag','Edit_f');
f=get(HEdit_f,'String');
 HEdit_vt=findobj('Tag','Edit_vt');
vt=str2num(get(HEdit_vt,'String'));
 HEdit_gt1=findobj('Tag','Edit_gt1');
gt1=get(HEdit_gt1,'String');
 HEdit_v1=findobj('Tag','Edit_v1');
v1=str2num(get(HEdit_v1,'String'));
 HEdit_g1=findobj('Tag','Edit_g1');
g1=get(HEdit_g1,'String');
 HEdit_v2=findobj('Tag','Edit_v2');
v2=str2num(get(HEdit_v2,'String'));
 HEdit_g2=findobj('Tag','Edit_g2');
g2=get(HEdit_g2,'String');
 HEdit_v3=findobj('Tag','Edit_v3');
v3=str2num(get(HEdit_v3,'String'));
 HEdit_g3=findobj('Tag','Edit_g3');
g3=get(HEdit_g3,'String');
 HEdit_v4=findobj('Tag','Edit_v4');
v4=str2num(get(HEdit_v4,'String'));
 HEdit_g4=findobj('Tag','Edit_g4');
g4=get(HEdit_g4,'String');

% 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);
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 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
```

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

      Hbox_Grid=findobj('Tag','box_Grid');
      for l=1:s
         surf(y,x,T(:,:,l))
         xlabel('y, м','FontSize',13)
         ylabel('x, м','FontSize',13)
         zlabel('T, K','FontSize',13)
         axis([min(y) max(y) min(x) max(x) min(min(T(:,:,1))) ...
max(max(T(:,:,1)))])
         if get(Hbox_Grid,'Value')
            grid on
         else
            grid off
         end
         pause(0.1)
         M(l)=getframe;
      end
       movie(M,7,4)
       set(HRadio_Style,'Enable','on');
      set(HCheckbox_Style,'Enable','on');
      set(Hbutton_Rep,'Enable','on');
      set(Hbutton_Figures,'Enable','on');
   end
case 'press_Rep'
   HRadio_Style=findobj('Style','radiobutton');
   set(HRadio_Style,'Enable','off');
   HCheckbox_Style=findobj('Style','checkbox');
   set(HCheckbox_Style,'Enable','off');
   Hbutton_Run=findobj('Tag','button_Run');
   set(Hbutton_Run,'Enable','off');
   Hbutton_Figures=findobj('Tag','button_Figures');
   set(Hbutton_Figures,'Enable','off');
   Hbox_Grid=findobj('Tag','box_Grid');
   for l=1:s
      surf(y,x,T(:,:,l))
      xlabel('y, м','FontSize',13)
      ylabel('x, м','FontSize',13)
      zlabel('T, K','FontSize',13)
      axis([min(y) max(y) min(x) max(x) min(min(T(:,:,1))) ...
max(max(T(:,:,1)))])
```

```
      if get(Hbox_Grid,'Value')
         grid on
      else
         grid off
      end
      pause(0.1)
      M(l)=getframe;
   end
   movie(M,7,4)
   set(HRadio_Style,'Enable','on');
   set(HCheckbox_Style,'Enable','on');
   set(Hbutton_Run,'Enable','on');
   set(Hbutton_Figures,'Enable','on');
case 'press_Figures'
   HRadio_Style=findobj('Style','radiobutton');
   set(HRadio_Style,'Enable','off');
   HCheckbox_Style=findobj('Style','checkbox');
   set(HCheckbox_Style,'Enable','off');
   Hbutton_Run=findobj('Tag','button_Run');
   set(Hbutton_Run,'Enable','off');
   Hbutton_Rep=findobj('Tag','button_Rep');
   set(Hbutton_Rep,'Enable','off');
   Hbox_Grid=findobj('Tag','box_Grid');
   HRadio=findobj('Style','radiobutton');
   HTurn=findobj(HRadio,'Value',1);
   for l=1:s
      figure
      surf(y,x,T(:,:,l))
      xlabel('y, м','FontSize',13)
      ylabel('x, м','FontSize',13)
      zlabel('T, K','FontSize',13)
      axis([min(y) max(y) min(x) max(x) min(min(T(:,:,1))) ...
max(max(T(:,:,1)))])
      if strcmp(get(HTurn,'Tag'),'Radio_Cool')
         colormap('cool')
      elseif strcmp(get(HTurn,'Tag'),'Radio_Hot')
         colormap('hot')
      else
         colormap('default')
      end
      if get(Hbox_Grid,'Value')
         grid on
      else
         grid off
      end
      pause(0.1)
   end
   set(HRadio_Style,'Enable','on');
   set(HCheckbox_Style,'Enable','on');
   set(Hbutton_Run,'Enable','on');
   set(Hbutton_Rep,'Enable','on');
case 'press_Quit'
   button=questdlg('Завершить работу с программой?', ...
'i_termo','Да','Нет','Нет');
```

```
    if strcmp(button,'Да')
       clear all
       close all
       clc
    end
end
```

Приведенные исходные тексты функций необходимо сохранить в виде m-файлов с именами **i_termo.m** и **f_termo2d2.m** соответственно и поместить их в подкаталог **WORK** корневого каталога MATLAB.

Запуск приложения осуществляется вводом имени файла **i_termo** в командной строке. При этом на экране появится окно, показанное на рис. 2.17.

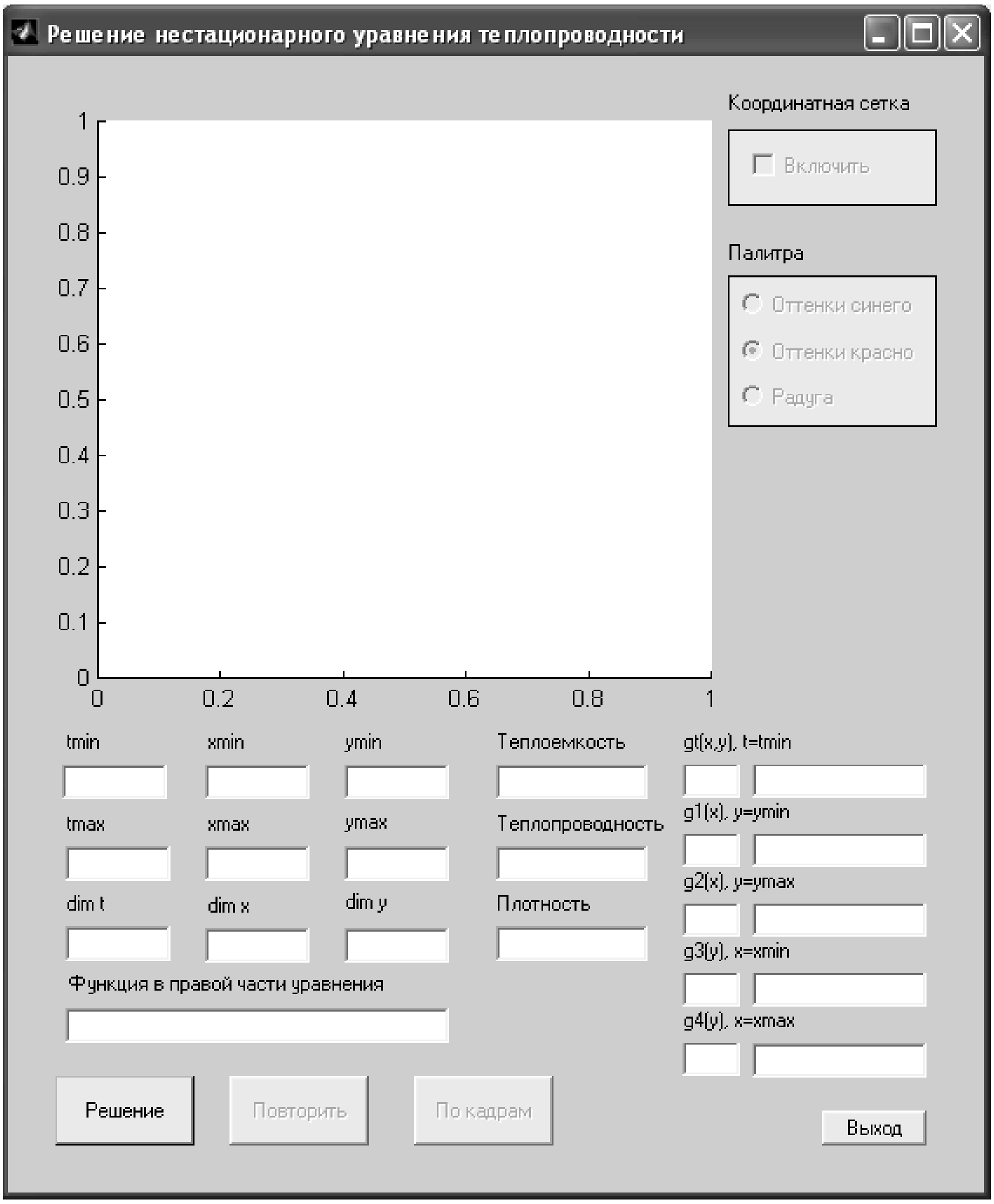


Рис. 2.17. Окно приложения i_termo

Работу с приложением следует начинать с ввода исходных данных в соответствующие текстовые окна. Символьные данные (функция правой части уравнения, функции теплопроводности, теплоемкости, плотности и функции граничных и начального условий) вводятся без апострофов.

В процессе ввода исходных данных доступными являются только кнопки **«Решение»** и **«Выход»**. Однако при активизации кнопки **«Решение»** до окончания ввода всех исходных данных программа выведет предупреждение об ошибке, после чего ввод исходных данных может быть продолжен.

По окончании ввода исходных данных за счет активизации кнопки **«Решение»** осуществится запуск задачи на решение, и через некоторый промежуток времени (в зависимости от вычислительных ресурсов компьютера и заданного числа узлов сетки) в графическом окне отобразится в динамике процесс изменения температуры по координатам и времени.

В качестве примера на рис. 2.18 приведено приложение **i_termo** с введенными исходными данными в процессе вывода на экран одного из графиков. Функция, введенная в текстовом окне начального условия **gt(x, y)**, имеет вид **10*(sign(1e2*x–2)–sign(1e2*x–3)+sign(1e2*y–3)–sign(1e2*y–5))+300**.

После этого становятся доступными остальные элементы интерфейса: флаг «**Включить**» в рамке «**Координатная сетка**», позволяющий выводить или не выводить координатную сетку на графиках; переключатель «**Палитра**», обеспечивающий выбор цветовой палитры для графиков; кнопка «**Повторить**», осуществляющая повторный просмотр результатов моделирования в динамике (без повторного решения задачи); кнопка «**По кадрам**», позволяющая вывести решение для каждой точки временной сетки в отдельном графическом окне для более детального анализа.

При активизации кнопки «**Выход**» текущее приложение и все графические окна закрываются и производится очистка оперативной памяти от всех результатов вычислений и очистка экрана.

2.4. Решение уравнений математической физики в среде PDEtool

Решим двухмерное уравнение Пуассона

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y) \,, \tag{2.66}$$

где *x, y* – координаты; *u*(*x, y*) – искомая функция; *f*(*x, y*) – некоторая непрерывная функция, определяемая выражением

$$f(x,y) = \exp(-x) + \exp(-y) \tag{2.67}$$

на прямоугольной области с граничными условиями

$$u\,(x_{\min}, y) = \sin(y^2) \;; \tag{2.68}$$

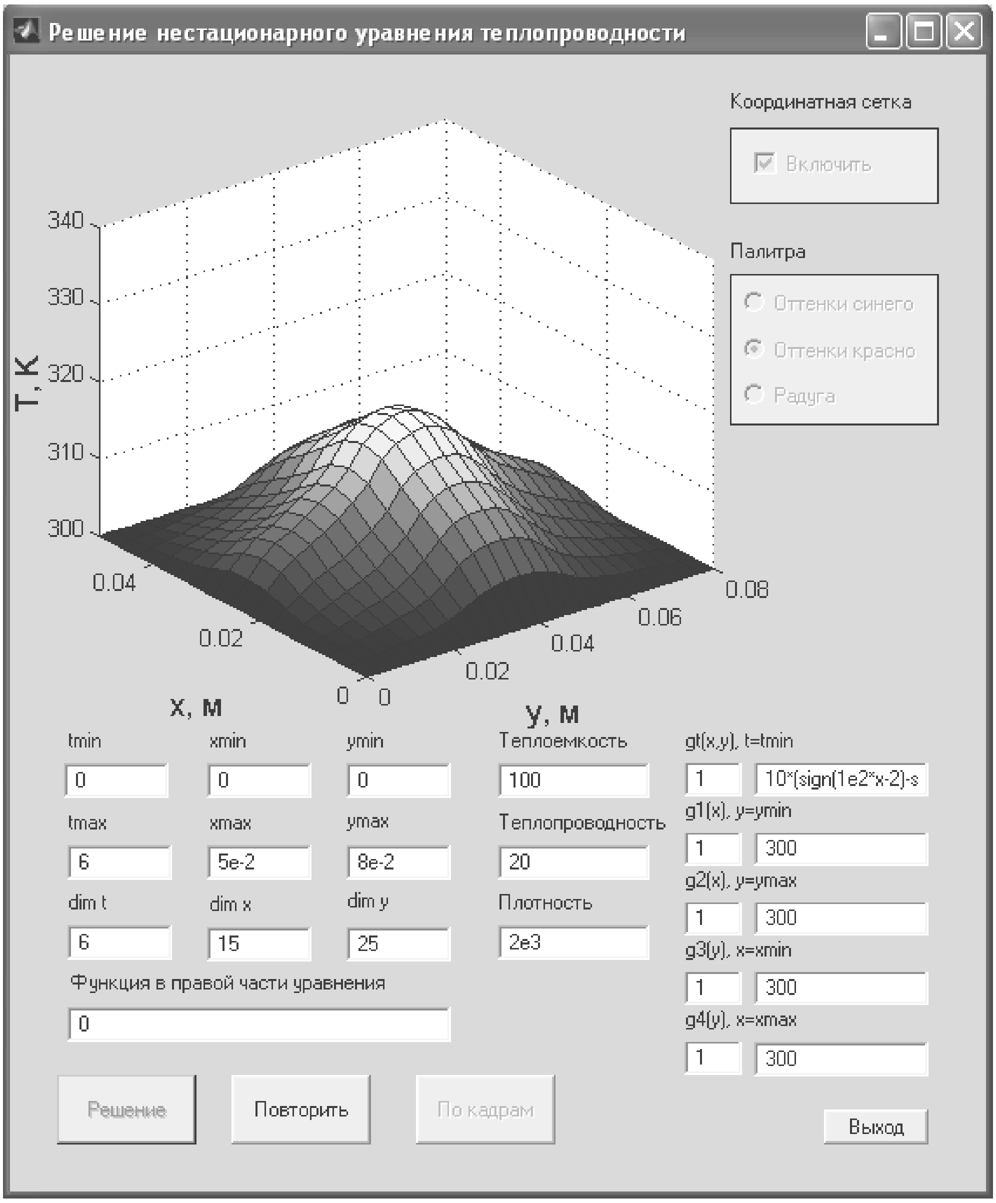


Рис. 2.18. Окно приложения i_termo в процессе вывода результатов

$$u\,(x_{\max}, y\,) = \cos(3y\,)\;; \tag{2.69}$$

$$\left.\frac{du}{dy}\right|_{x\,, y_1} = 10\sin(x^2)\;; \tag{2.70}$$

$$\left.\frac{du}{dy}\right|_{x\,, y_m} = 10\sin(6x) \tag{2.71}$$

на границах $x_{min} = 0$, $x_{max} = 1$, $y_{min} = -1$, $y_{max} = 1$ триангулярной координатной сетке методом конечных элементов с использованием приложения **PDEtool**.

Запуск приложения осуществляется командой **pdetool**. При этом на экране монитора отображается главное окно приложения (рис. 2.19).

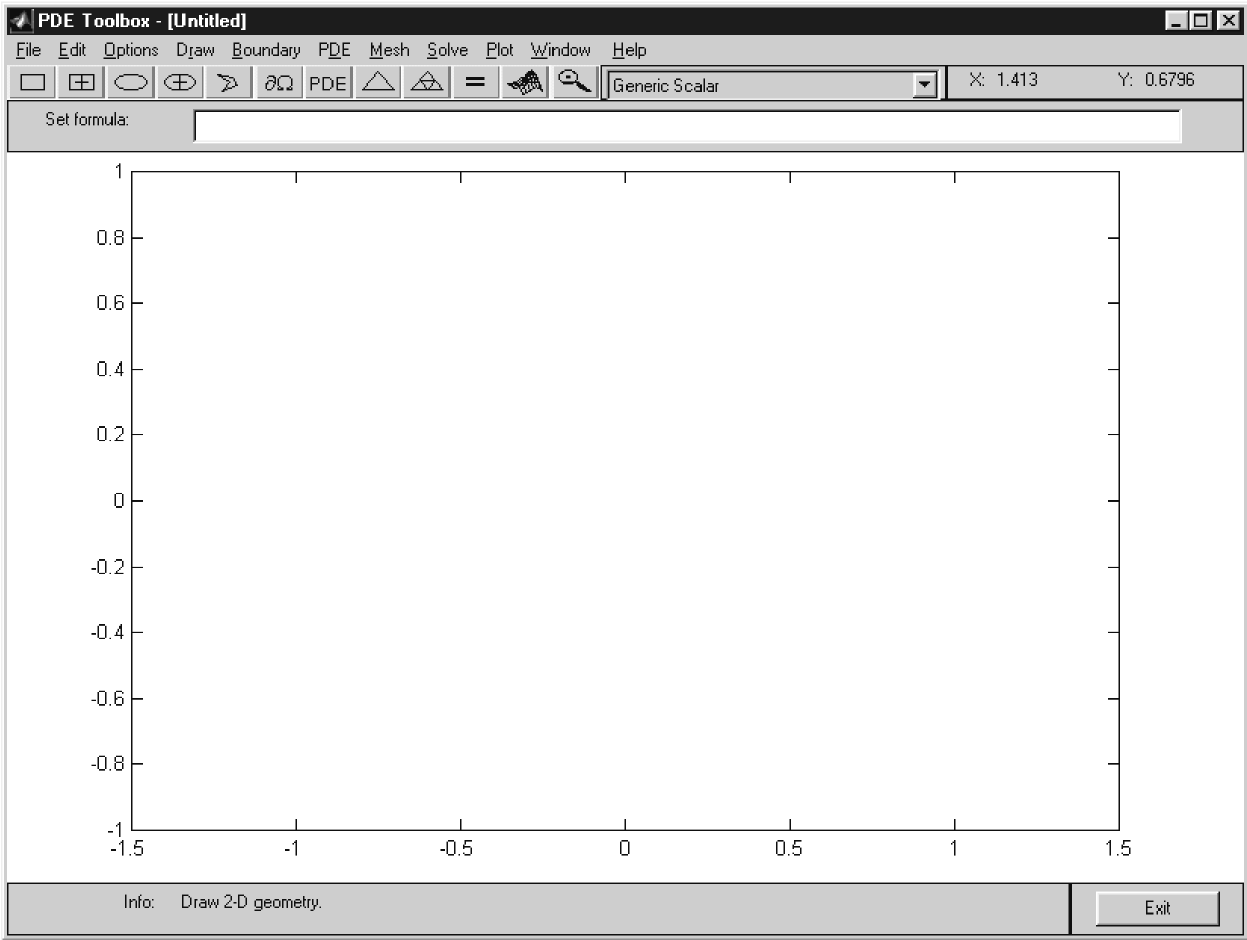


Рис. 2.19. Главное окно приложения pdetool

Для задания прямоугольной области решения необходимо активизировать с помощью «мыши» 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. 2.20).

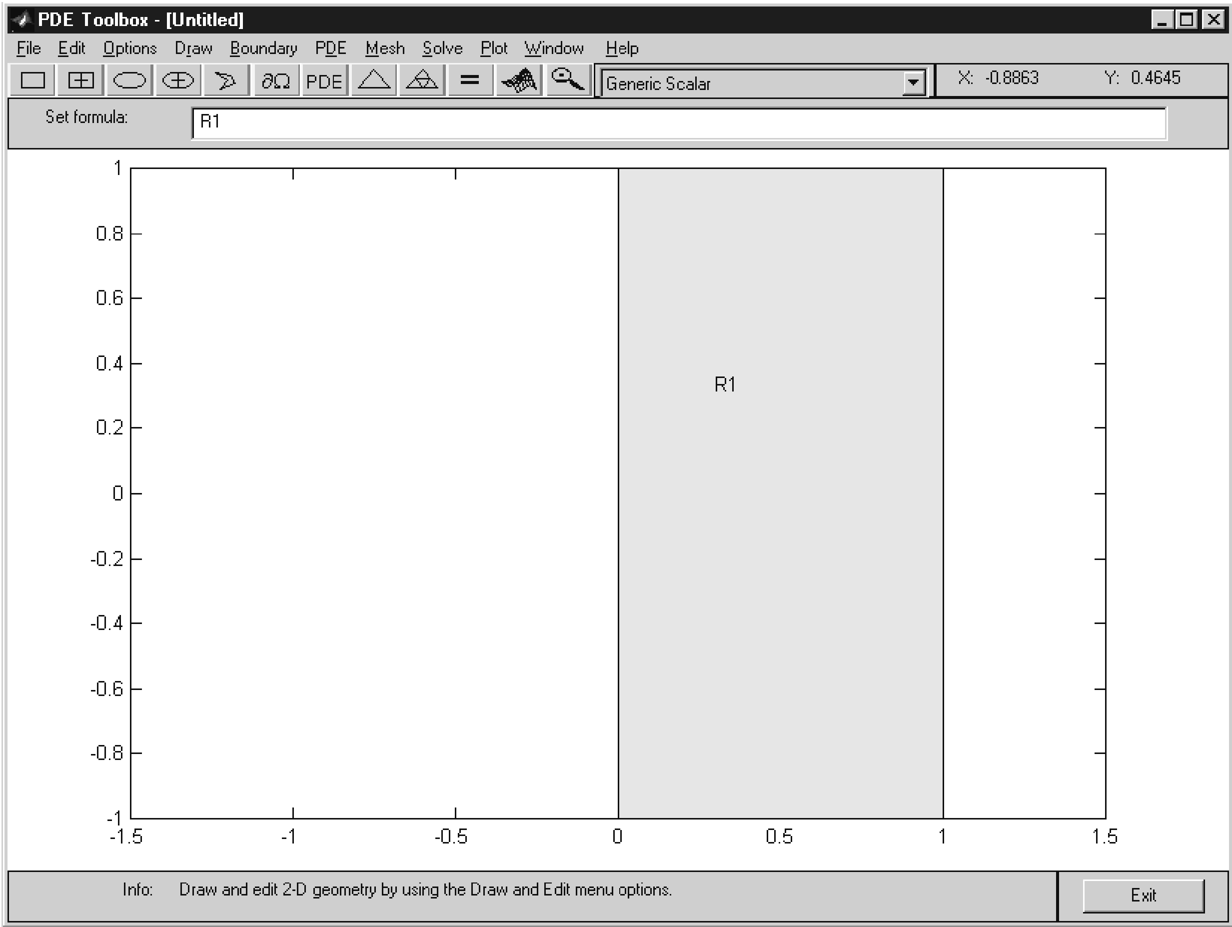


Рис. 2.20. Задание прямоугольной области решения задачи в pdetool

При необходимости корректировки координат и размеров области нужно навести курсор «мыши» на изображение прямоугольника и дважды щелкнуть левой кнопкой. На экране появится окно редактирования параметров области с соответствующими полями (рис. 2.21). 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, имеющих изображения 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 знаком «+» в случае объединения или знаком «–» в случае разности (см. рис. 2.20).

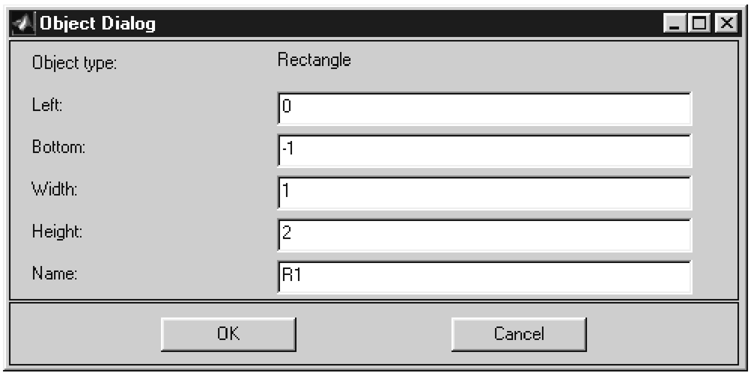


Рис. 2.21. Окно редактирования параметров прямоугольной области решения задачи в приложении pdetool

Для задания граничных условий необходимо активизировать манипулятором «мышь» кнопку с символом **∂ Ω**, в результате чего окно приложения примет вид, показанный на рис. 2.22.

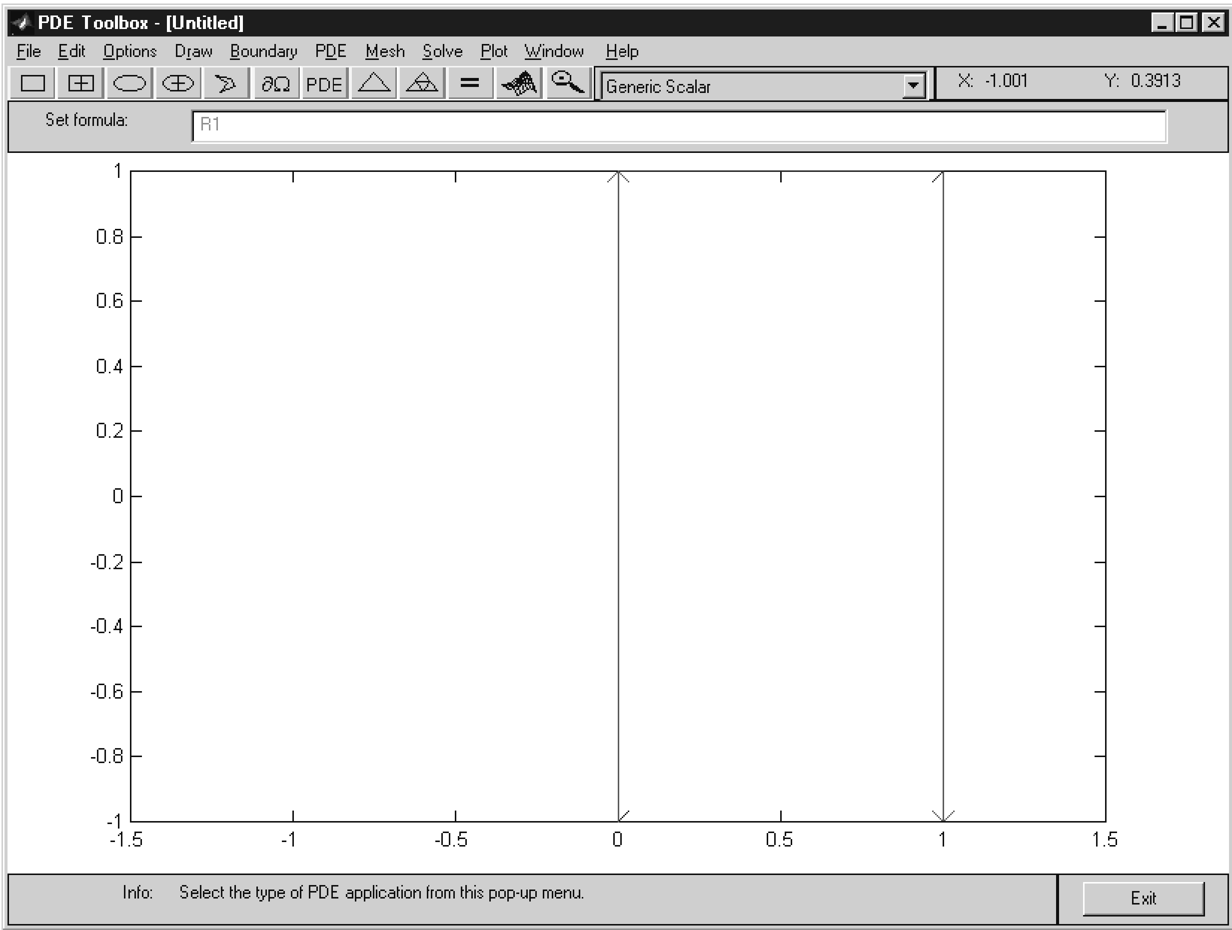


Рис. 2.22. Задание граничных условий в приложении pdetool

Все границы области показаны линиями со стрелками, причем по умолчанию на них заданы условия Дирихле (красные линии на экране). Для редактирования граничных условий необходимо дважды щелкнуть левой кнопкой «мыши» на выбранной границе и внести соответствующие изменения в полях окна редактирования граничных условий (рис. 2.23).

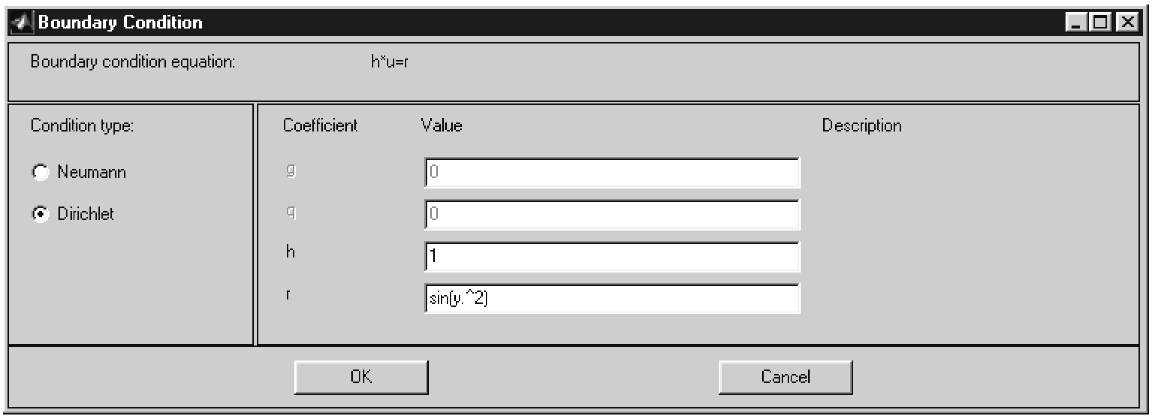


Рис. 2.23. Задание граничных условий

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. 2.24.

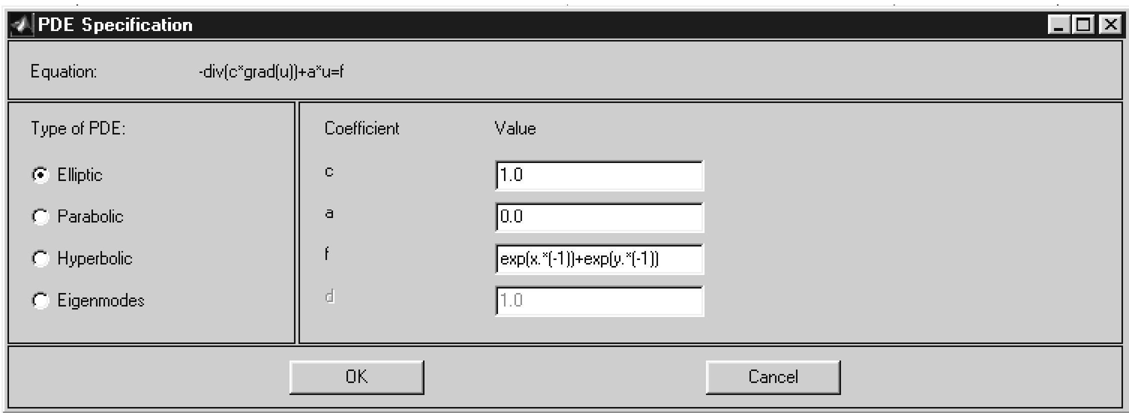


Рис. 2.24. Окно редактирования уравнения задачи в приложении pdetool

Формирование триангулярной сетки с использованием метода Делоне осуществляется за счет активизации кнопки с символом Δ (рис. 2.25). При необходимости увеличения числа узлов сетки следует активизировать кнопку  (рис. 2.26).

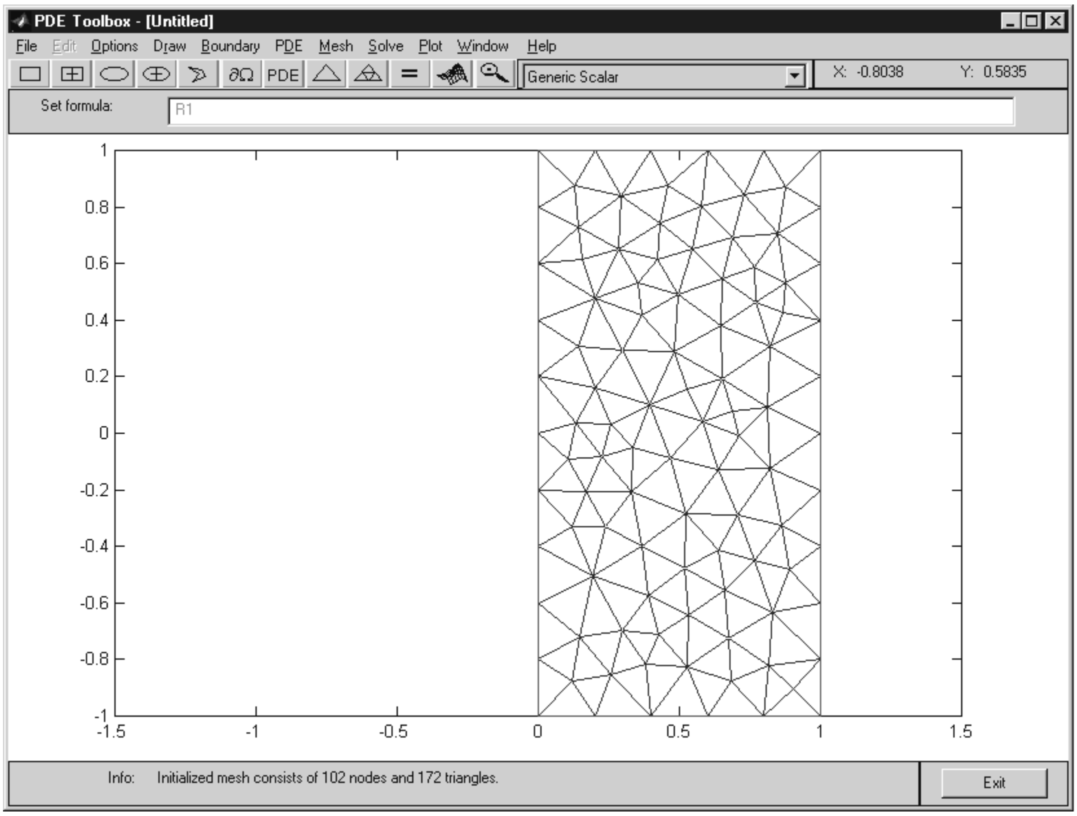


Рис. 2.25. Генерация триангулярной сетки с использованием метода Делоне в приложении pdetool

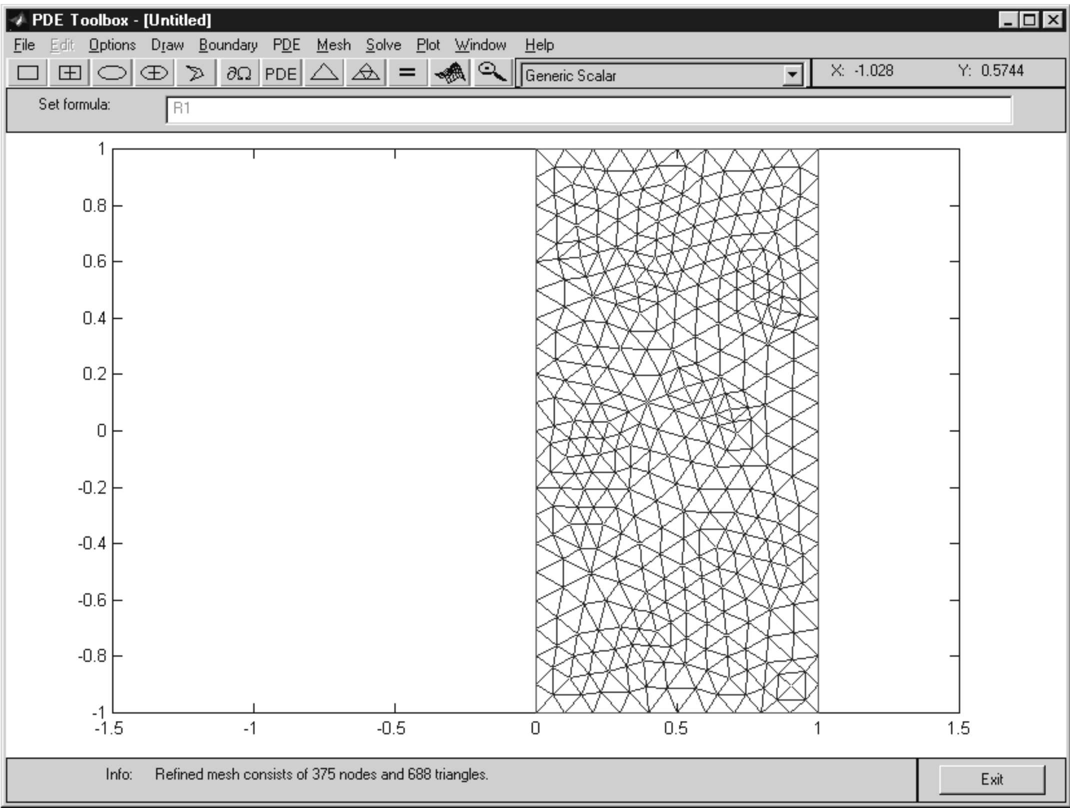
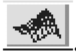


Рис. 2.26. Увеличение числа элементов сетки в приложении pdetool

Решение задачи осуществляется при активизации кнопки с символом «=». По умолчанию значения функции решения выделяются различными цветми (рис. 2.27).

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. 2.28.

При активизации кнопки **Plot** на экран будет выведен график, показанный на рис. 2.29.

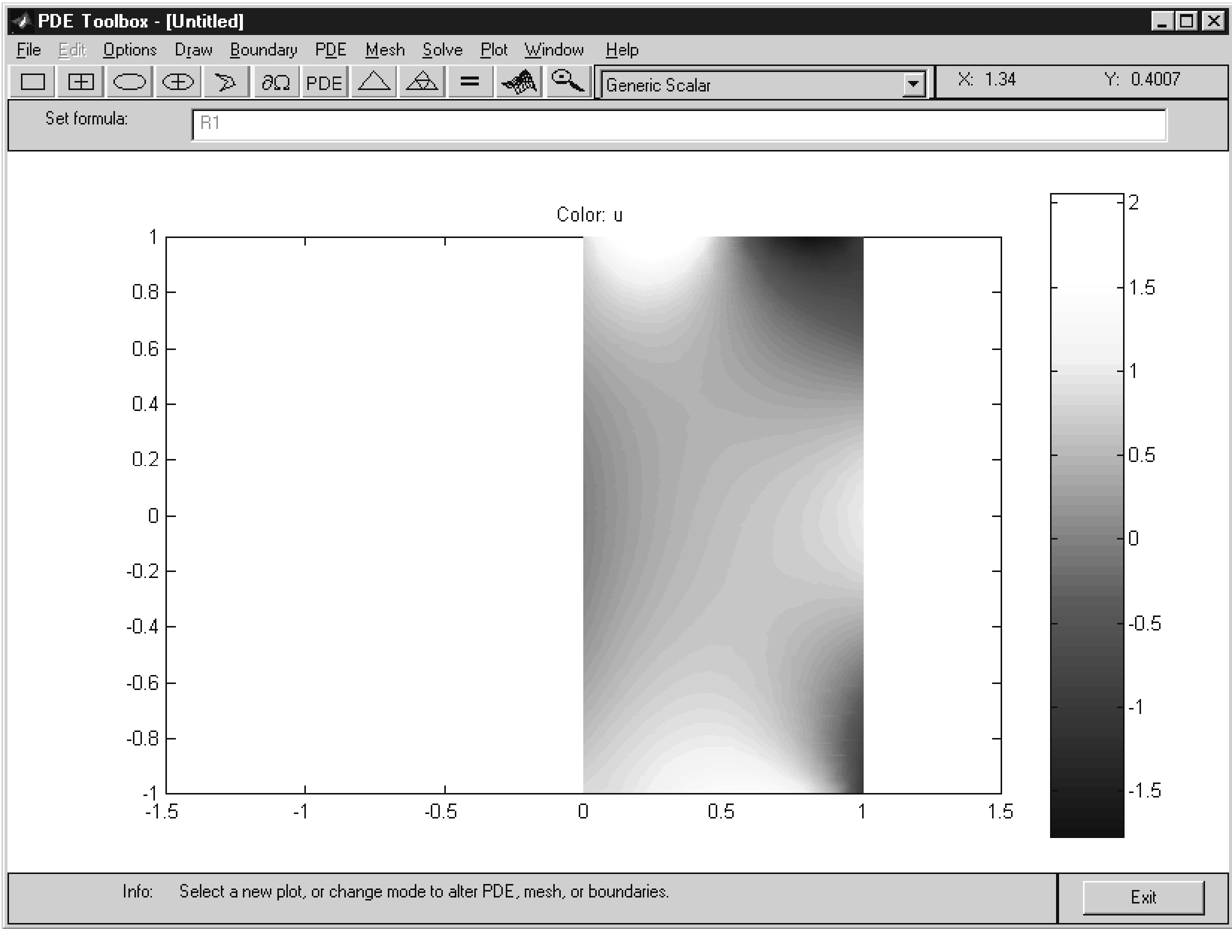


Рис. 2.27. Решение уравнения Пуассона в приложении pdetool

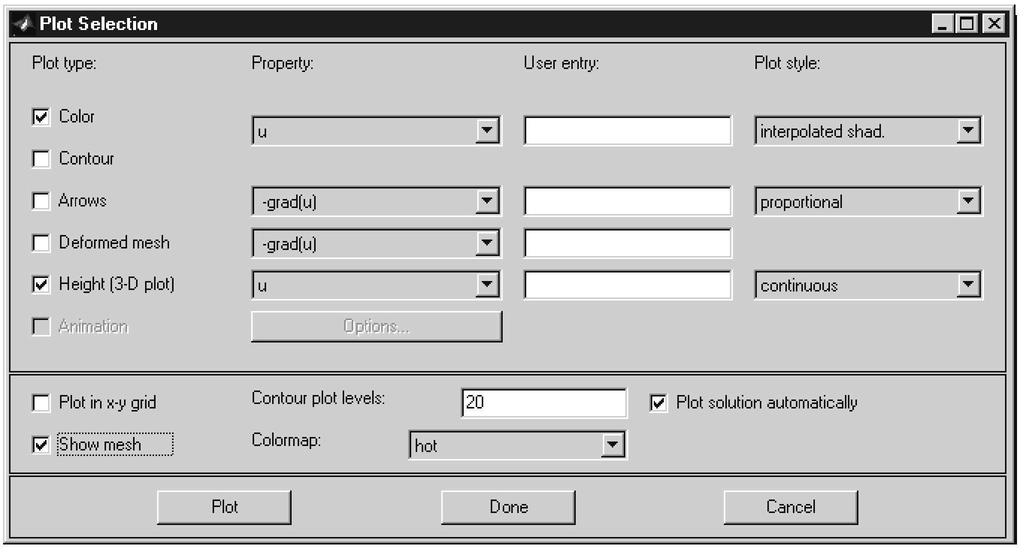


Рис. 2.28. Окно редактирования параметров графического представления решения в приложении pdetool

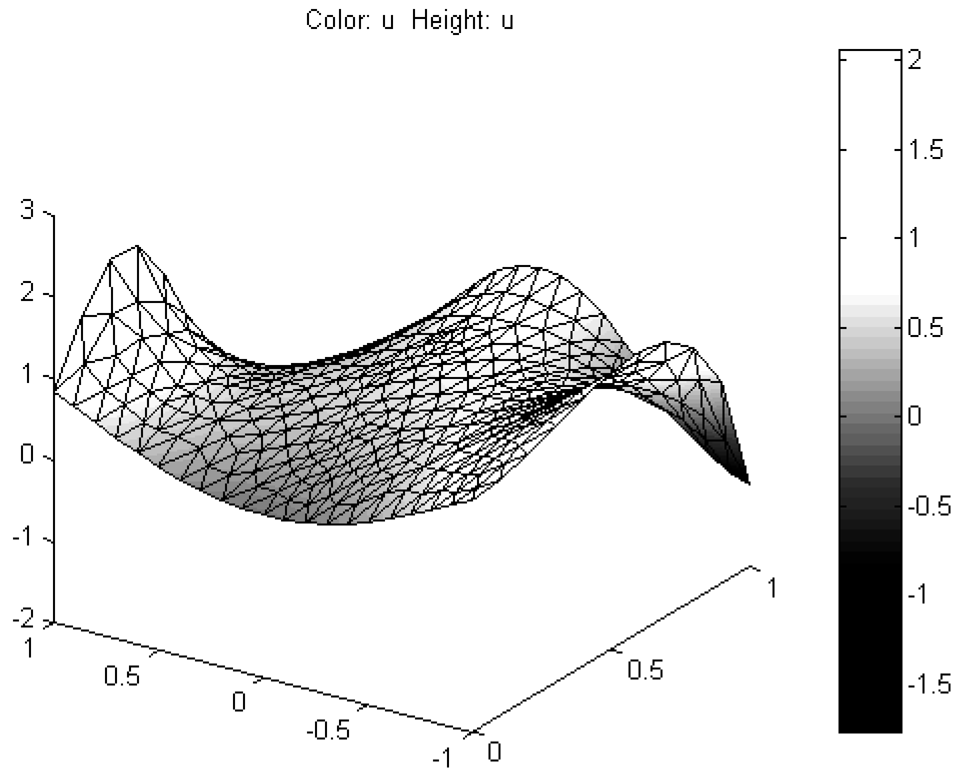


Рис. 2.29. Представление решения уравнения Пуассона в виде 3D-изображения

Более подробно с возможностями приложения **pdetool** можно ознакомиться в справке HELP системы MATLAB или в [2].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Рындин Е.А. Методы решения задач математической физики. – Таганрог: Изд-во ТРТУ, 2003. – 120 с.
2. Ануфриев И.Е. Самоучитель MATLAB 5.3/6.x. – СПб: БХВ-Петербург, 2002. – 736 с.

ОГЛАВЛЕНИЕ

Рындин Евгений Адальбертович
Лысенко Игорь Евгеньевич

Решение задач математической физики в системе MatLab

Учебное пособие

Ответственный за выпуск *Лысенко И.Е.*
Редактор *Маныч Э.И.*
Корректор *Селезнева Н.И.*

ЛР №020565 от 23.06.1997 г.
Подписано к печати 30.10.05
Печать офсетная
Бумага офсетная
Формат $60 * 84_{1/16}$
Усл. п. л. – 4,0
Уч.- изд. л. – 3,8
Заказ № 419
Тираж 150 экз.
"С"

Издательство Таганрогского государственного радиотехнического университета
ГСП 17А, Таганрог, 28, Некрасовский, 44
Типография Таганрогского государственного радиотехнического университета
ГСП 17А, Таганрог, 28, Энгельса, 1