

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ РАСТИТЕЛЬНЫХ ПОЛИМЕРОВ»**

Кафедра высшей математики

Краевые задачи в системе Matlab

**Методическое пособие
для студентов направления
«Прикладная математика и информатика»**

**Санкт-Петербург
2015**

УДК 51(07.07)

Краевые задачи в системе Matlab: методическое пособие/ сост.:
З.Л.Абжандадзе, О.Е.Куляхтина, М.Э.Юдовин; СПбГТУРП – СПб., 2015.-26
с.

Приводится теоретический материал с разобранными решениями
типичных примеров. Предназначено для студентов направления 01.03.02
«Прикладная математика и информатика».

Рецензент: профессор кафедры прикладной математики и информатики,
д-р физ.-мат. наук С.П.Помыткин.

Подготовлено и рекомендовано к печати кафедрой высшей математики
Санкт-Петербургского государственного технологического университета
растительных полимеров (протокол № 9 от 26.12.2014).

Утверждено к изданию методической комиссией факультета
автоматических систем управления технологическими процессами СПбГТУРП
(протокол № 8 от 29.12.2014).

ВВЕДЕНИЕ

Цель данных методических указаний – познакомить студентов с методом конечных элементов и техникой его применения при решении краевых задач для уравнений с частными производными с помощью инструмента **pdetool** системы **Matlab**.

Дальнейшее изложение проводится на примере простейшего уравнения эллиптического типа.

УРАВНЕНИЯ ЭЛЛИПТИЧЕСКОГО ТИПА

С помощью таких уравнений описываются стационарные явления, например, электростатическое поле, прогиб мембраны или пластинки. В простейшем двумерном случае уравнение имеет вид (уравнение Пуассона)

$$\Delta u = f(x, y), \quad (x, y) \in \Omega. \quad (1)$$

Здесь $u(x, y)$ – неизвестная функция, $\Delta u = u_{xx} + u_{yy}$ – оператор Лапласа, $f(x, y)$ – заданная функция, Ω – область на плоскости.

Уравнение (1) – частный случай более общего уравнения

$$-\operatorname{div}(c \cdot \operatorname{gradu}) + a \cdot u = f, \quad (2)$$

где $c(x, y)$ – симметричная положительно определенная 2×2 -матрица, а $f(x, y)$, $a(x, y)$ – заданные функции.

КРАЕВЫЕ УСЛОВИЯ

Краевые условия заключаются в задании значений решения или его производной или их комбинации на границе области Ω . Для уравнения (1) они обычно имеют один из следующих видов:

$$u|_{\partial\Omega} = r(x, y), \quad (3)$$

$$\left. \frac{\partial u}{\partial n} \right|_{\partial\Omega} = g(x, y), \quad (4)$$

$$\left(\frac{\partial u}{\partial n} + q(x, y)u \right) \Big|_{\partial\Omega} = g(x, y). \quad (5)$$

Здесь $\partial\Omega$ – граница области Ω , а $\frac{\partial u}{\partial n}$ – производная по направлению внешней нормали к $\partial\Omega$.

Таким образом, получаем три типа краевых задач, они называются соответственно задачами Дирихле, Неймана и смешанной задачей. Возможна также задача, в которой на разных частях границы ставятся разные краевые условия. Далее рассмотрим основную идею метода на примере задачи Дирихле и для простоты положим $r(x, y) \equiv 0$.

КЛАССИЧЕСКОЕ И ОБОБЩЕННОЕ РЕШЕНИЯ

Уточним постановку краевой задачи. Задача поставлена полностью, если даны: а) уравнение; б) область, в которой определено решение; в) краевые условия на границе области; г) функциональное пространство, в котором ищется решение.

Введем обозначения:

- $C(\bar{\Omega})$ – нормированное пространство непрерывных в $\bar{\Omega}$ функций с нормой $\|u\|_C = \max_{(x,y) \in \bar{\Omega}} |u(x,y)|$;
- $C_{2,0}(\bar{\Omega})$ – нормированное пространство непрерывных в $\bar{\Omega}$ функций, равных 0 на границе области и дважды непрерывно дифференцируемых в Ω и.

Задача (1)-(3) в классической постановке заключается в следующем. Пусть область Ω ограничена, $\bar{\Omega} = \Omega \cup \partial\Omega$, $\partial\Omega$ – кусочно-гладкая замкнутая кривая и $f(x,y) \in C(\bar{\Omega})$.

Функция $u(x,y) \in C_{2,0}(\bar{\Omega})$ называется **классическим решением** задачи (1)-(3), если выполняется условие

$$\Delta u = f(x,y), \quad (x,y) \in \Omega.$$

Можно существенно обобщить классическую формулировку, если ослабить ограничения на $f(x,y)$, а решение искать в более широком классе функций. При этом теоремы существования и единственности решения остаются верными. В частности, для $f(x,y)$ достаточно потребовать, чтобы выполнялось условие

$$\iint_{\Omega} |f(x,y)|^2 dx dy < +\infty. \quad (6)$$

При этом функция $u(x,y)$ уже не обязана принадлежать $C_{2,0}(\bar{\Omega})$ и удовлетворять уравнению (1) во всех точках области и, значит, не является классическим решением задачи. Поэтому необходимо определить новое пространство, в котором ищется решение, и заново определить само понятие решения краевой задачи.

Пусть $W_{2,0}^1(\Omega)$ – пространство функций, почти везде дифференцируемых в Ω , равных нулю на $\partial\Omega$ и удовлетворяющих неравенству

$$\iint_{\Omega} (|\nabla u|^2 + |u|^2) dx dy < +\infty.$$

Функция $u(x, y) \in W_{2,0}^1(\Omega)$ называется **обобщенным решением** задачи (1)-(3), если для любой функции $v \in W_{2,0}^1(\Omega)$ выполняется условие

$$\iint_{\Omega} (u_x v_x + u_y v_y - 2f \cdot v) dx dy = 0. \quad (7)$$

Уравнение (7) получается из (1) умножением на $v(x, y)$ и интегрированием по частям. Отсюда следует, что классическое решение является и обобщенным. Обратное утверждение неверно. Однако можно доказать (см., например, [1]), что если обобщенное решение принадлежит $C_{2,0}(\bar{\Omega})$, то оно является решением и в классическом смысле.

В самом определении обобщенного решения не предполагается существования вторых производных, тем не менее, если выполнено условие (7), то вторые производные существуют почти везде (см. [1]). Польза такого обобщения задачи будет видна из дальнейшего.

ВАРИАЦИОННЫЙ ПОДХОД К РЕШЕНИЮ КРАЕВОЙ ЗАДАЧИ

Рассмотрим функционал

$$I(u) = \iint_{\Omega} (u_x^2 + u_y^2 - 2f \cdot u) dx dy = \iint_{\Omega} (|\nabla u|^2 - 2f \cdot u) dx dy$$

на множестве функций $u(x, y) \in W_{2,0}^1(\Omega)$. В вариационном исчислении доказывается, что условие (6) является необходимым условием экстремума для функционала $I(u)$. На этом факте основан вариационный метод приближенного решения краевой задачи.

Пусть функции $\varphi_k(x, y) \in W_{2,0}^1(\Omega)$, $k = 1, 2, \dots$, линейно независимы и таковы, что множество их линейных комбинаций плотно в $W_{2,0}^1(\Omega)$. Это означает, что для любой функции $u(x, y) \in W_{2,0}^1(\Omega)$ существует

последовательность линейных комбинаций функций $\varphi_k(x, y)$, стремящаяся по норме пространства $W_{2,0}^1(\Omega)$ к $u(x, y)$.

Ищем приближенное решение в виде $u_n(x, y) = \sum_{k=1}^n \bar{u}_k \varphi_k(x, y)$, где

$\bar{u}_k$ – произвольные числа, а n фиксировано. Рассмотрим $I(u_n)$. Очевидно, $I(u_n)$ – функция от n переменных $\bar{u}_i, i = 1, 2, \dots, n$. Необходимое условие экстремума для нее имеет вид

$$\frac{\partial I(u_n)}{\partial \bar{u}_i} = 0, i = 1, 2, \dots, n. \quad (8)$$

После стандартных преобразований получаем линейную систему n уравнений для n коэффициентов $\bar{u}_k$:

$$BU_n = F, \quad (9)$$

где

$$U_n = \begin{pmatrix} \bar{u}_1 \\ \dots \\ \bar{u}_n \end{pmatrix},$$

$$B = \{b_{ij}, i, j = 1, \dots, n\}, \quad b_{ik} = \iint_{\Omega} \nabla \varphi_i \cdot \nabla \varphi_k dx dy,$$

$$F = \begin{pmatrix} f_1 \\ \dots \\ f_n \end{pmatrix}, \quad f_i = \iint_{\Omega} f \cdot \varphi_i dx dy.$$

Справедлива следующая теорема.

Теорема. $\lim_{n \rightarrow +\infty} u_n(x, y)$ существует и является обобщенным решением задачи

(1)-(3). Здесь имеется в виду предел по норме пространства $W_{2,0}^1(\Omega)$.

Основная трудность при реализации этой схемы состоит в построении функций $\varphi_i(x, y)$. Готовое решение имеется лишь для простейших областей (прямоугольник или круг). Если область произвольной формы, то обычно задача решается с помощью метода конечных элементов.

МЕТОД КОНЕЧНЫХ ЭЛЕМЕНТОВ

Реализация метода состоит из нескольких шагов:

1. Триангуляция

Разобьем область $\bar{\Omega}$ на треугольники $\Delta_i, i = 1, 2, \dots, n$. Такое разбиение называется триангуляцией. Треугольники, у которых две вершины находятся на границе области, могут иметь криволинейную сторону. Эту сторону заменим отрезком прямой и получившуюся область обозначим через $\bar{\Omega}_n$. Вершины треугольников называются узлами триангуляции, занумеруем их и пусть D_1 – множество узлов, принадлежащих границе, D_2 – множество внутренних узлов, $D = D_1 \cup D_2$ – множество всех узлов триангуляции, а J_1, J_2, J – количество узлов в D_1, D_2, D соответственно.

2. Построение конечных элементов

Рассмотрим любой треугольник $\Delta_i \subset \bar{\Omega}_n$. Для каждой вершины этого треугольника определим функцию $\psi_j^i(x, y)$, равную единице в этой вершине и равную нулю в других вершинах (здесь i – номер треугольника, j – номер вершины в множестве всех узлов D). Простейшей такой функцией является линейная функция вида $z = ax + by + c$, причем $z = 1$ в одной из вершин треугольника и $z = 0$ в двух других вершинах.

Пример. Пусть, например, треугольник Δ_7 имеет вершины $P_2(-1; 2), P_5(1; 0), P_{11}(3; 4)$. Тогда для него определены три функции $\psi_2^7(x, y), \psi_5^7(x, y), \psi_{11}^7(x, y)$, верхний индекс обозначает номер треугольника, а нижний – номер вершины треугольника (узла триангуляции). Здесь функция $\psi_2^7(x, y)$ задает плоскость, проходящую через точки $(-1; 2; 1), (1; 0; 0), (3; 4; 0)$, функция $\psi_5^7(x, y)$ задает плоскость, проходящую через точки $(-1; 2; 0), (1; 0; 1), (3; 4; 0)$, а функция $\psi_{11}^7(x, y)$ задает плоскость, проходящую через точки $(-1; 2; 0), (1; 0; 0), (3; 4; 1)$. Выпишем эти функции, используя стандартное

уравнение плоскости, проходящей через три данные точки. После простых преобразований получаем:

$$\psi_2^7(x, y) = (4 - y) / 2; \quad \psi_5^7(x, y) = (x - 2y + 5) / 6; \quad \psi_{11}^7(x, y) = (x + y - 1) / 6.$$

Итак, каждому треугольнику Δ_i соответствуют три линейные функции $\psi_{j1}^i(x, y), \psi_{j2}^i(x, y), \psi_{j3}^i(x, y)$, где i - номер треугольника, а $j1, j2, j3$ - номера его вершин.

Сопоставим j -му узлу произвольный параметр $\bar{u}_j, j = 1, \dots, J$. Пусть вершины треугольника Δ_i имеют в множестве D номера $j1, j2, j3$. Определяем новую функцию

$$\varphi_i(x, y) = \begin{cases} \bar{u}_{j1}\psi_{j1}^i(x, y) + \bar{u}_{j2}\psi_{j2}^i(x, y) + \bar{u}_{j3}\psi_{j3}^i(x, y), & (x, y) \in \Delta_i \\ 0, & (x, y) \notin \Delta_i \end{cases}. \quad (10)$$

В нашем примере получаем

$$\varphi_7(x, y) = \bar{u}_2\psi_2^7 + \bar{u}_5\psi_5^7 + \bar{u}_{11}\psi_{11}^7 = ax + by + c,$$

где коэффициенты a, b, c вычисляются по формулам:

$$a = \frac{1}{6}(\bar{u}_5 + \bar{u}_{11}); \quad b = -\frac{1}{2}\bar{u}_2 - \frac{1}{3}\bar{u}_5 + \frac{1}{6}\bar{u}_{11}; \quad c = 2\bar{u}_2 + \frac{5}{6}\bar{u}_5 - \frac{1}{6}\bar{u}_{11}.$$

Из построения функций $\psi_j^i(x, y)$ следует, что значение $\varphi_i(x, y)$ в узле с номером j равно $\bar{u}_j$.

Функция $\varphi_i(x, y)$ называется **конечным элементом**.

3. Построение системы уравнений

Определим в $\bar{\Omega}_n$ функцию $u_n(x, y) = \varphi_i(x, y), (x, y) \in \Delta_i, i = 1, 2, \dots, n$.

Из определения видно, что внутри каждого треугольника функция $u_n(x, y)$ дифференцируема, так как совпадает с линейной функцией φ_i . Кроме того, $u_n(x_j, y_j) = \bar{u}_j$. Если треугольники Δ_{i1} и Δ_{i2} имеют общую сторону или вершину, то на этой стороне или вершине $\varphi_{i1}(x, y) = \varphi_{i2}(x, y)$. Поэтому $u_n(x, y)$ непрерывна и кусочно-дифференцируема на всей области $\bar{\Omega}_n$.

Если узел с номером j принадлежит границе области, то, учитывая граничное условие, положим

$$\bar{u}_j = 0. \quad (11)$$

Тогда функция $u_n(x, y)$ равна 0 на границе области $\bar{\Omega}_n$ и $u_n(x, y) \in W_{2,0}^1(\bar{\Omega}_n)$. Ее графиком является поверхность, составленная из плоских треугольников. В определении функции $u_n(x, y)$ участвуют J переменных $\bar{u}_j$. Из них J_1 переменных определяются граничным условием (11). Требуется выбрать значения остальных J_2 переменных так, чтобы функция $u_n(x, y)$ была в каком-то смысле близка к решению исходной краевой задачи. Рассмотрим выражение

$$I_n(u_n) = \iint_{\Omega_n} (|\nabla u_n|^2 - 2f \cdot u_n) dx dy. \quad (12)$$

Очевидно, $I_n(u_n)$ - функция от J_2 переменных $\bar{u}_j$. Индекс n в I_n указывает на отличие от функционала I , так как в (12) область интегрирования $\bar{\Omega}_n$, а не $\bar{\Omega}$.

Из необходимого условия экстремума получаем для всех внутренних узлов

$$\frac{\partial I_n(u_n)}{\partial \bar{u}_j} = 0, \quad j = 1, \dots, J_2. \quad (13)$$

Система (13) линейна, так как $I_n(u_n)$ - квадратичная функция. Порядок системы равен J_2 .

Ввиду очевидной связи между функцией $I_n(u_n)$ и функционалом $I(u)$ естественно предположить, что $\lim_{n \rightarrow +\infty} u_n(x, y) = u(x, y)$. Доказательство этого утверждения имеется, например, в [2].

4. Структура матрицы системы (13).

Запишем систему (13) в виде

$$BU = F, \quad (14)$$

где $B = \{b_{jk}, j, k = 1, 2, \dots, J\}$, $F = \{f_j, j = 1, 2, \dots, J\}$.

Так как $\bar{\Omega}_n = \bigcup_{i=1}^n \Delta_i$, то $I_n(u_n)$ можно представить в следующей форме:

$$I_n(u_n) = \sum_{i=1}^n L_i(\varphi_i), \quad (15)$$

где $L_i(\varphi_i) = \iint_{\Delta_i} (|\nabla \varphi_i|^2 - 2f\varphi_i) dx dy$.

Заметим, что $\bar{u}_j$ присутствует только в тех слагаемых суммы (15), для которых область интегрирования, т.е. Δ_i , содержит j -й узел. Поэтому после дифференцирования по $\bar{u}_j$ в сумме (15) исчезнут все слагаемые, не содержащие $\bar{u}_j$. Значит, каждое уравнение системы (13) содержит только относительно небольшую часть от общего числа неизвестных. Поэтому большая часть элементов матрицы B равна 0. Такие матрицы называются **разреженными** (sparse). Для решения систем с разреженными матрицами разработаны специальные методы.

5. Вычисление элементов матрицы системы (13).

Рассмотрим одно слагаемое в (15). Имеем

$$\varphi_i(x, y) = \bar{u}_{j1}\psi_{j1}^i(x, y) + \bar{u}_{j2}\psi_{j2}^i(x, y) + \bar{u}_{j3}\psi_{j3}^i(x, y), \quad (x, y) \in \Delta_i.$$

Учитывая линейность функций $\psi_j^i(x, y)$, запишем их в виде

$$\psi_j^i(x, y) = a_j^i x + b_j^i y + c_j^i.$$

Отсюда получаем: $\frac{\partial \psi_j^i}{\partial x} = a_j^i$, $\frac{\partial \psi_j^i}{\partial y} = b_j^i$ и

$$\frac{\partial \varphi_i}{\partial x} = \bar{u}_{j1}a_{j1}^i + \bar{u}_{j2}a_{j2}^i + \bar{u}_{j3}a_{j3}^i, \quad \frac{\partial \varphi_i}{\partial y} = \bar{u}_{j1}b_{j1}^i + \bar{u}_{j2}b_{j2}^i + \bar{u}_{j3}b_{j3}^i.$$

Отсюда получаем :

$$\begin{aligned}
|\nabla \varphi_i|^2 &= \left(\frac{\partial \varphi_i}{\partial x} \right)^2 + \left(\frac{\partial \varphi_i}{\partial y} \right)^2 = \\
&= (\bar{u}_{j1} a_{j1}^i + \bar{u}_{j2} a_{j2}^i + \bar{u}_{j3} a_{j3}^i)^2 + (\bar{u}_{j1} b_{j1}^i + \bar{u}_{j2} b_{j2}^i + \bar{u}_{j3} b_{j3}^i)^2 \\
\frac{\partial}{\partial u_{j1}} |\nabla \varphi_i|^2 &= \\
&= 2 \left[(\bar{u}_{j1} a_{j1}^i + \bar{u}_{j2} a_{j2}^i + \bar{u}_{j3} a_{j3}^i) a_{j1}^i + (\bar{u}_{j1} b_{j1}^i + \bar{u}_{j2} b_{j2}^i + \bar{u}_{j3} b_{j3}^i) b_{j1}^i \right].
\end{aligned} \tag{16}$$

После приведения подобных членов выражение принимает вид:

$$\frac{\partial}{\partial u_{j1}} |\nabla \varphi_i|^2 = 2(A_1^i \bar{u}_{j1} + A_2^i \bar{u}_{j2} + A_3^i \bar{u}_{j3}), \tag{17}$$

где: $A_1^i = (a_{j1}^i)^2 + (b_{j1}^i)^2$, $A_2^i = a_{j2}^i a_{j1}^i + b_{j2}^i b_{j1}^i$, $A_3^i = a_{j3}^i a_{j1}^i + b_{j3}^i b_{j1}^i$.

Кроме того, из (10) следует, что $\frac{\partial}{\partial u_{j1}} (f \varphi_i) = f \psi_{j1}^i$. Заметим, что

$$\frac{\partial}{\partial \bar{u}_{j1}} L_i(\varphi_i) = \iint_{\Delta_i} \frac{\partial}{\partial \bar{u}_{j1}} (|\nabla \varphi_i|^2 - 2f \cdot \varphi_i) dx dy.$$

Поэтому с учетом формулы (17) получим

$$\frac{\partial}{\partial \bar{u}_{j1}} L_i(\varphi_i) = 2 \iint_{\Delta_i} (A_1^i \bar{u}_{j1} + A_2^i \bar{u}_{j2} + A_3^i \bar{u}_{j3} - f \cdot \psi_{j1}^i) dx dy. \tag{18}$$

Аналогично вычисляются производные по $\bar{u}_{j1}$ от всех слагаемых в сумме (15), в которых используется узел с номером $j1$. То же рассуждение верно и для любого узла. Элементы $b_{j,k}$ матрицы B из уравнения (14) и f_j вектора F вычисляем следующим образом:

- j -е уравнение системы получаем, дифференцируя функционал $I_n(u_n)$ по переменной $\bar{u}_j$ и используя формулы (15) и (18);
- в полученном уравнении находим слагаемые, содержащие $\bar{u}_k$;
- элемент b_{jk} равен сумме коэффициентов при $\bar{u}_k$;

- элемент f_j равен сумме всех интегралов вида $\iint_{\Delta_i} f \cdot \psi_j^i dx dy$,
входящих в j -е уравнение.

После того, как матрица B и вектор F вычислены, решение системы осуществляется с помощью стандартных компьютерных процедур.

Из вышеизложенного очевидно, что и триангуляция и вычисление коэффициентов весьма трудоемки даже при малых значениях n . Поэтому разработаны удобные инструменты для численного решения достаточно широкого круга краевых задач математической физики. Одним из таких инструментов является **pdetool** из пакета **Matlab**.

PDETOOL

В пакете Matlab имеется набор инструментов для решения разнообразных краевых задач для уравнений и систем уравнений с частными производными. Они содержатся в подкаталоге **toolbox\pde**. Овладение всеми инструментами требует изучения их многочисленных свойств и умения программировать на языке Matlab. Однако один из этих инструментов – **pdetool** – позволяет решать достаточно широкий класс краевых задач с помощью визуальных средств, почти не прибегая к написанию программного кода. Покажем, как это делается на примере простой задачи

$$\begin{cases} -\Delta u + (x^2 + y)u = \sin(4x)y, & (x; y) \in \Omega, \\ u|_{y=0} = x^2 + x; & x \in [0; 1], \\ u|_{x=0} = -y^2; & y \in [0; 1], \\ \frac{\partial u}{\partial n}|_{x^2+y^2=1} = x; & x \in [0; 1]. \end{cases} \quad (19)$$

Здесь Ω - сектор единичного круга, содержащийся в первой четверти.

Запускаем Matlab и вводим команду **pdetool**. На дисплее появляется основная форма (рис.1). В верхней части формы находятся две панели с кнопками, которые отчасти дублируют друг друга. Ниже панелей расположено

текстовое окно **set formula**, о котором речь пойдет ниже, а в основной части формы – рабочее окно, в котором изображены оси координат, причем по умолчанию: $-1,5 \leq x \leq 1,5$; $-1 \leq y \leq 1$ (эти границы можно при необходимости изменить).

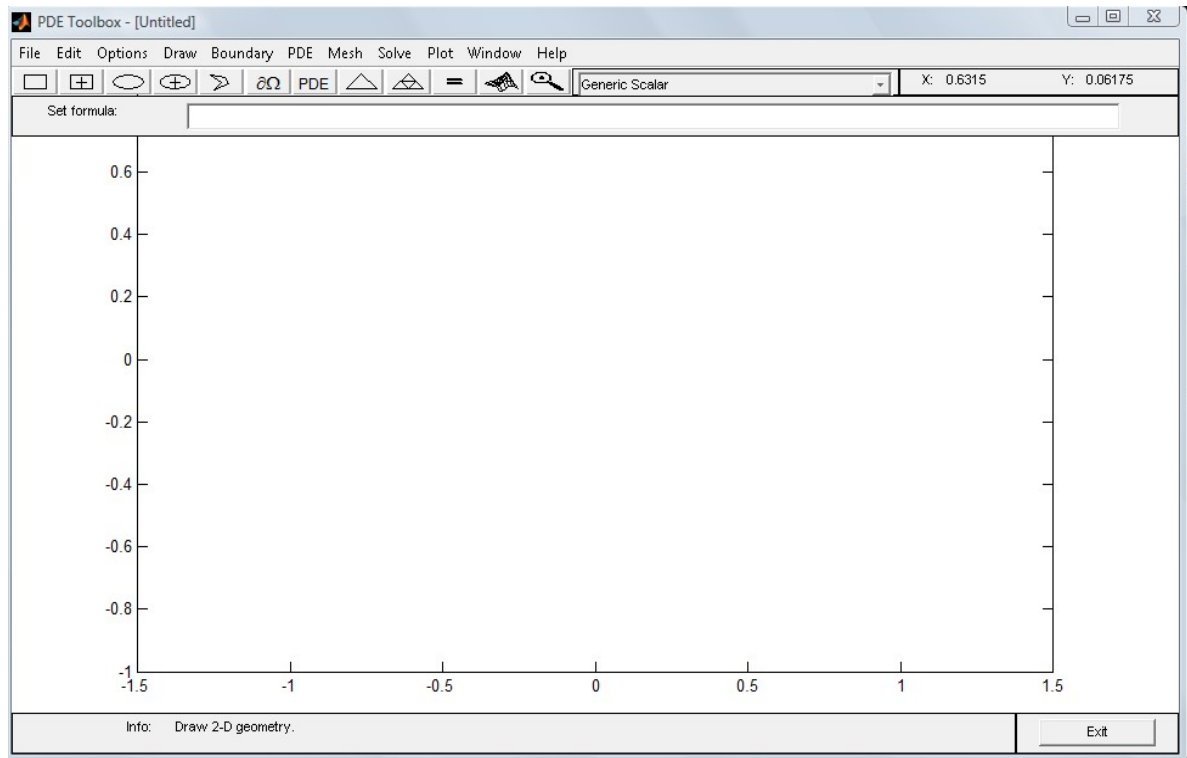


Рис. 1. Основная форма

Начнем с кнопки **Options**. Открывается меню, в котором выберем четыре пункта: **Grid**(координатная сетка), **Grid Spacing** (шаг сетки), **Axes Limits** (границы для переменных x и y), **Axes Equal** (одинаковый масштаб для x и y). В результате на экране появится координатная сетка с заданными шагом и диапазоном для x и y . Далее действуем в следующем порядке.

Задание области Ω

Простейшие области – прямоугольник, эллипс или многоугольник – можно задавать с помощью кнопок с соответствующими иконками или кнопкой **Draw** с последующим выбором области. Области более сложной формы можно получить объединением или пересечением элементарных фигур. Рассмотрим, например, задачу (19).

Сначала установим границы для x и y (**Options**=>**Axes Limits**):

$-2 \leq x \leq 2$; $-2 \leq y \leq 2$. Сектор Ω представим как пересечение квадрата R_1 со

стороной 1 и единичного круга E1 с центром в точке (0; 0). Нажимаем на кнопку с эллипсом и рисуем мышью в рабочем окне эллипс любого размера. Чтобы превратить его в единичный круг с центром в (0; 0), делаем двойной щелчок по эллипсу и в открывшуюся форму вводим координаты центра: 0, 0 и полуоси эллипса: $A = B = 1$. Отметим, что в окне **set formula** появилось E1 – имя круга. Аналогично, нажимаем на кнопку с прямоугольником и рисуем произвольный прямоугольник. Затем двойным щелчком по нему открываем форму и вводим (0; 0) – координаты левой нижней вершины прямоугольника и 1 – его высоту и ширину. Получим квадрат R1, пересекающийся с кругом E1. Отметим, что в окне **set formula** появилось выражение E1+R1. Это означает, что **pdetool** рассматривает полученную фигуру как объединение круга и квадрата. Так как нам нужно пересечение, а не объединение, заменяем в окне **set formula** символ + на символ *. В рабочем окне ничего не меняется, но теперь **pdetool** рассматривает полученную фигуру как пересечение круга и квадрата. Чтобы убедиться в этом, нажмем на кнопку $\partial\Omega$. На рис.2 изображена граница требуемой области.

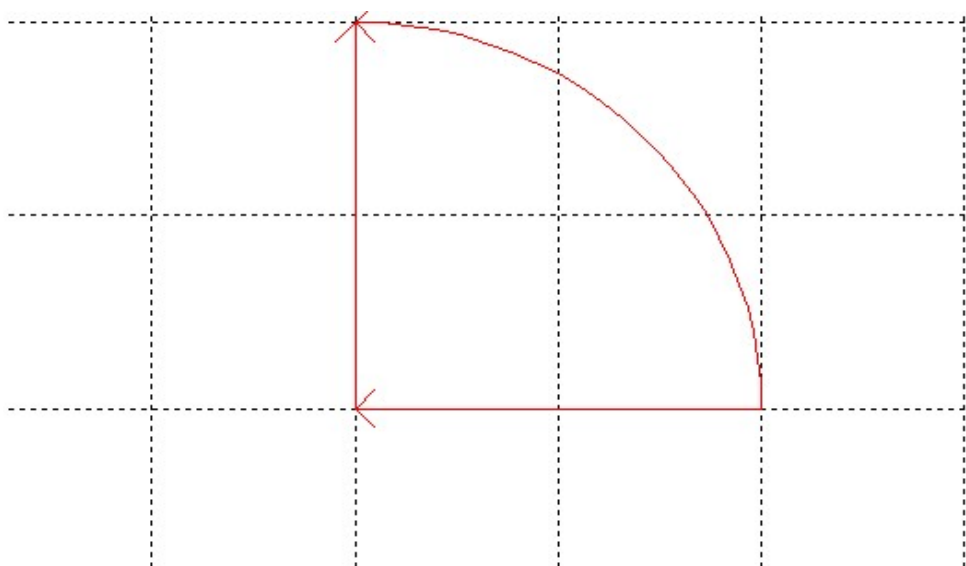


Рис. 2. $E1 \cdot R1$ – сектор представлен как пересечение квадрата и круга

В общем случае применяется следующий порядок построения области. Сначала вводим в рабочее окно необходимые простейшие фигуры и корректируем их размеры и расположение. Затем редактируем выражение в

окне **set formula** с помощью символов $+$, $-$, $*$. Так можно построить область весьма сложной конфигурации. На рис. 3 и 4 показаны области, определяемые формулами $E1+R1$ и $E1-R1$ соответственно.

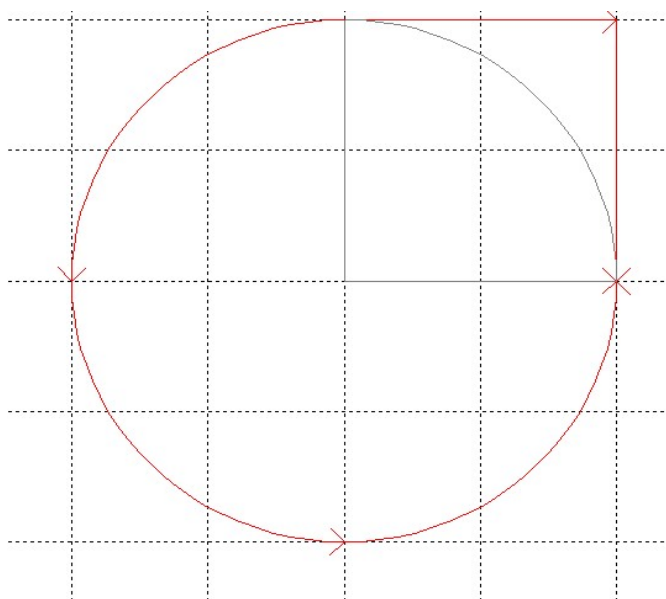


Рис. 3. $E1+R1$ - объединение квадрата и круга

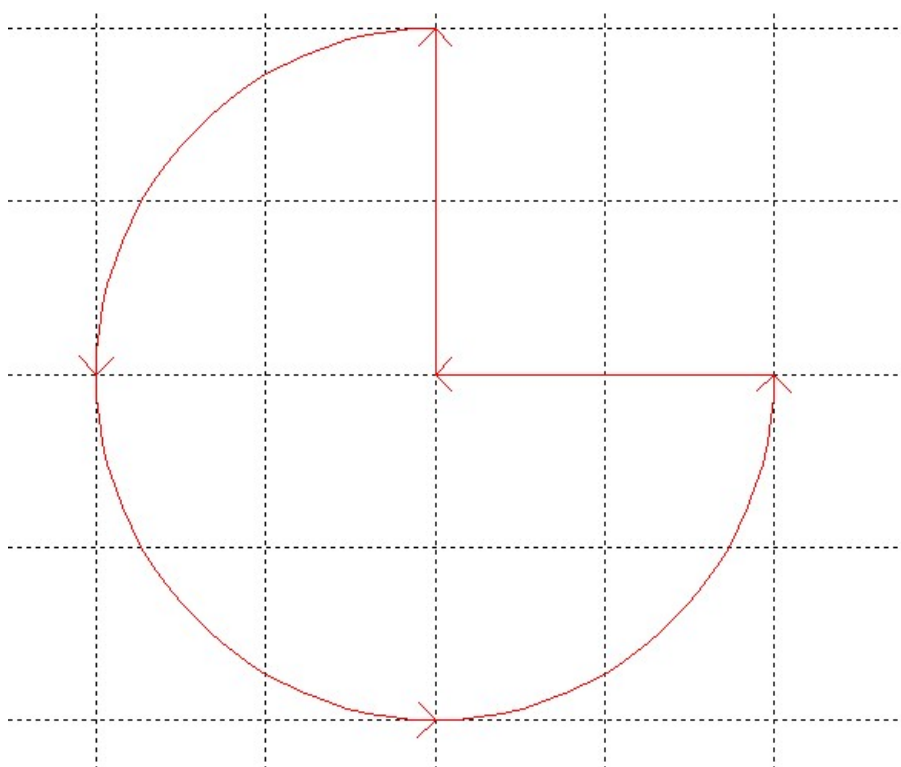


Рис. 4 . $E1-R1$ - из круга удалено его пересечение с квадратом

Задание дифференциального уравнения

Кнопка **PDE** (**P**artial **D**ifferential **E**quations) открывает форму, в которой нужно выбрать, к какому типу относится решаемая задача – эллиптическому, параболическому, гиперболическому или к задачам на собственные числа. Каждому типу соответствует свой набор коэффициентов (соответствующие текстовые окна находятся в правой половине формы). В нашем случае уравнение эллиптическое, поэтому вводим значения коэффициентов c , a , f для задачи (19) (см. рис.5).

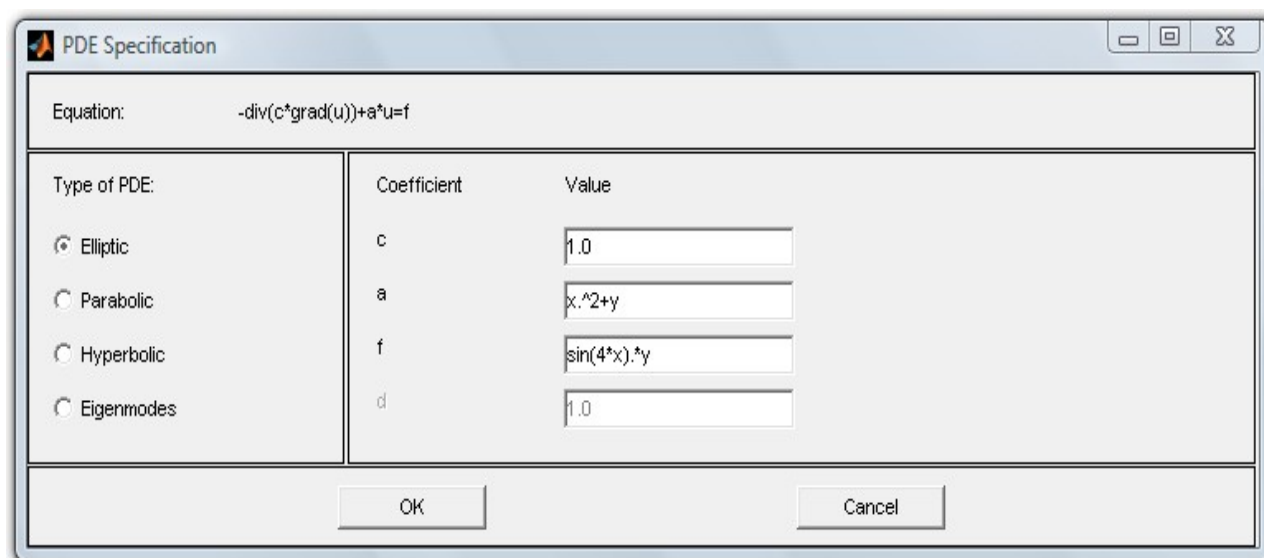


Рис. 5. Задание коэффициентов

Выражения для коэффициентов записываются на языке Matlab'a по стандартным правилам, похожим на правила других языков программирования. Отметим только одну особенность: перед символами $\wedge$ и $*$ (операции возведения в степень и умножения) нужно ставить символ «точка», иначе в ходе решения задачи будет выдано сообщение об ошибке. Поэтому коэффициент $a(x,y)$ введен в виде $x.^2 + y$, а не $x^2 + y$, а коэффициент $f(x,y)$ – в виде $\sin(4 * x) .* y$, а не $\sin(4 * x) * y$ (произведение $4 * x$ не содержит точки, так как при умножении на константу точка не обязательна).

Задание краевых условий

Далее требуется задать краевые условия. Граница области автоматически разбита на участки, отмеченные стрелками (рис.2). В нашем случае на каждом

участке свои краевые условия, однако и в том случае, когда на всей границе одинаковые условия, их все равно пришлось бы задавать по участкам. Входим в режим **Boundary Mode** кнопкой $\partial\Omega$ и выбираем один из отрезков границы. Двойной щелчок по нему открывает форму, позволяющую выбрать тип краевого условия на данном участке и ввести требуемые функции (рис.6).

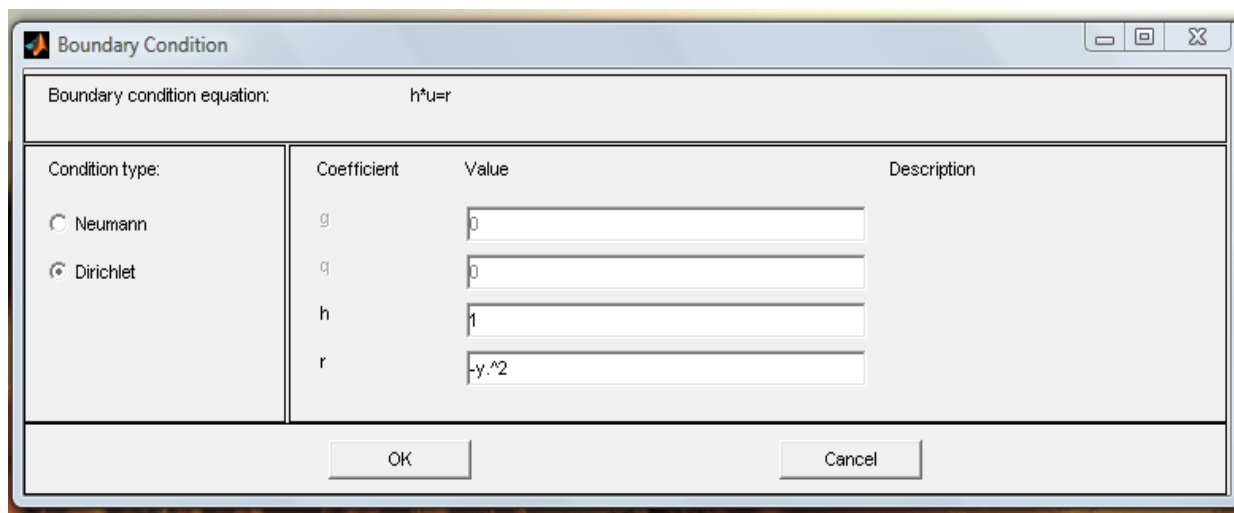


Рис. 6. Задание краевых условий

Аналогично задаются краевые условия на остальных участках границы.

Триангуляция области

Нажимаем на кнопку с треугольником и получаем в рабочем окне исходную область, разбитую на некоторое число треугольников. Чтобы сделать сетку более частой, нужно нажать на кнопку с треугольником со знаком + внутри (рис.7). Тот же результат получится при нажатии кнопки **mesh** (сетка) и последующем выборе **refine mesh**(измельчение сетки) в открывшемся меню. Команда **Solve=>Parameters** открывает форму (см. рис. 8) для ввода параметров вычислительного процесса. Рассмотрим некоторые из них, на остальные при первом знакомстве с системой **pdetool** можно не обращать внимания. В режиме Adaptive mode (адаптивный режим) программа сама определяет на сколько треугольников нужно разбить область, чтобы достичь требуемой точности. При этом по умолчанию определено максимальное число треугольников (maximum number of triangles), на которые разбивается область, и наибольшее число последовательных триангуляций (maximum number of

triangulations).

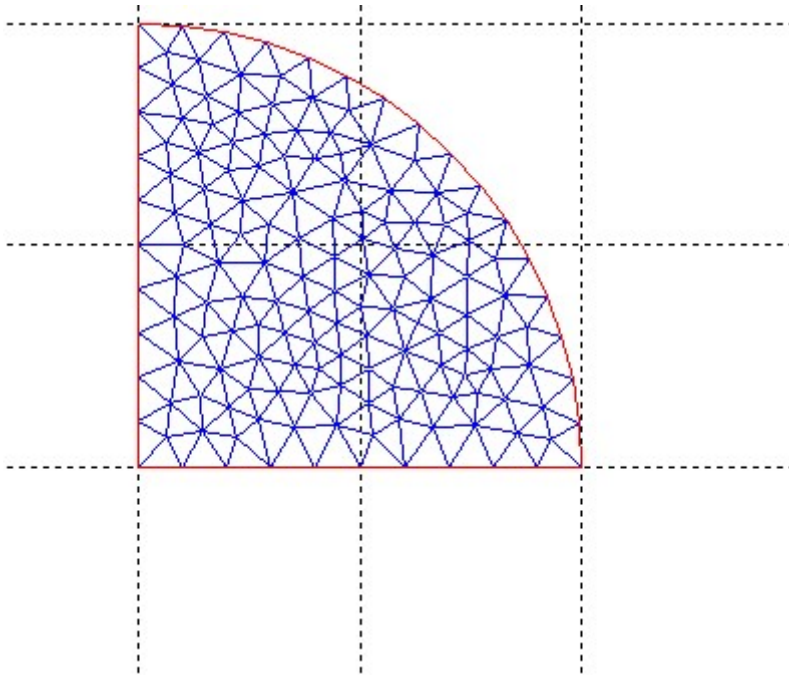


Рис. 7. Триангуляция области

Решение краевой задачи

На самом деле самому заниматься триангуляцией, как это было описано выше, не обязательно. После того как уравнение, область и краевые условия полностью описаны, процесс решения можно запустить кнопками **Solve => Solve PDE**. Программа сама выберет нужную триангуляцию и решит задачу. Если результат почему-либо неудовлетворителен, применяем **Solve=>Parameters**. Открывается форма (рис. 8), позволяющая задать наибольшее число треугольников при триангуляции, наибольшее число шагов по измельчению сетки, относительную погрешность и ряд других параметров. После корректировки параметров нужно снова применить команду **Solve => Solve PDE**.

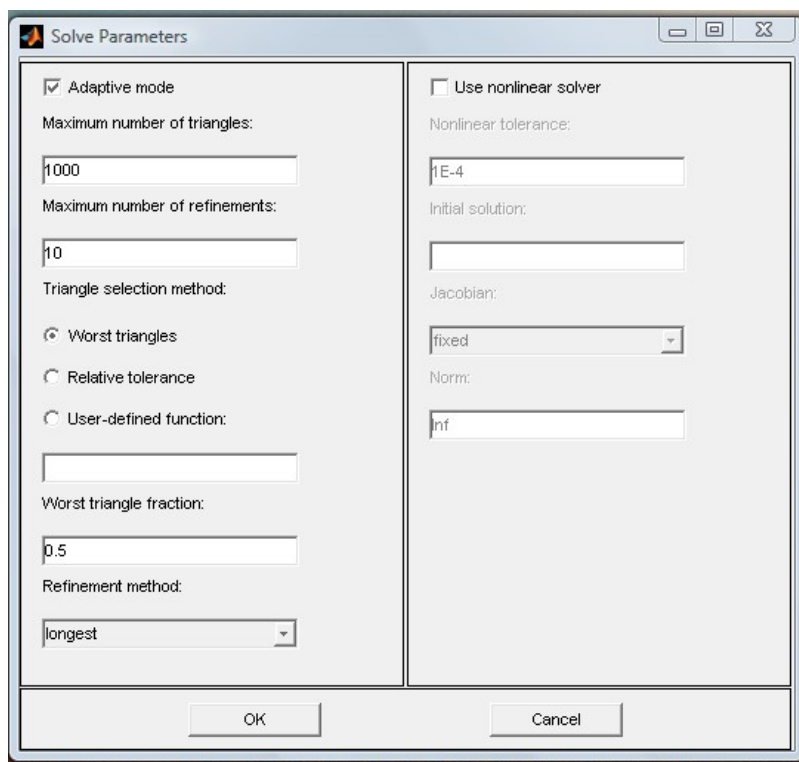


Рис. 8. Параметры вычислительного процесса

6. Визуализация решения

Если не предпринимать никаких дополнительных действий, кроме описанных выше, то **pdetool** выдаст решение в виде области Ω , закрашенной разным цветом в разных точках области, в зависимости от значения полученного приближенного решения $u_n(x, y)$ в точке (x, y) (рис. 9). Цвет каждого пикселя на рис. 9, вернее его числовой код, является линейной функцией от значения $u_n(x, y)$ в точке (x, y) . Соответствие между цветом и значением $u_n(x, y)$ определяется шкалой в правой части рис. 9. На черно-белом рисунке разные значения $u_n(x, y)$ отличаются различной интенсивностью серого цвета.

Такое наглядное представление решения позволяет быстро оценить решение на качественном уровне. На рис. 9 видно, примерно в каких точках функция $u_n(x, y)$ принимает наибольшее и наименьшее значения.

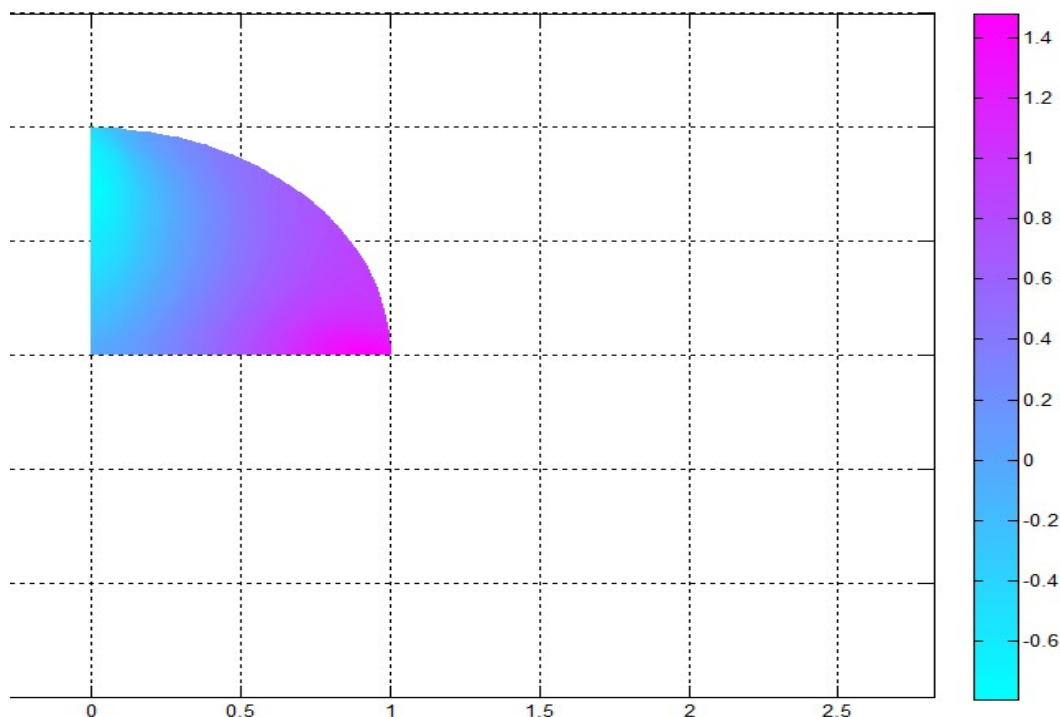


Рис. 9. Визуализация решения

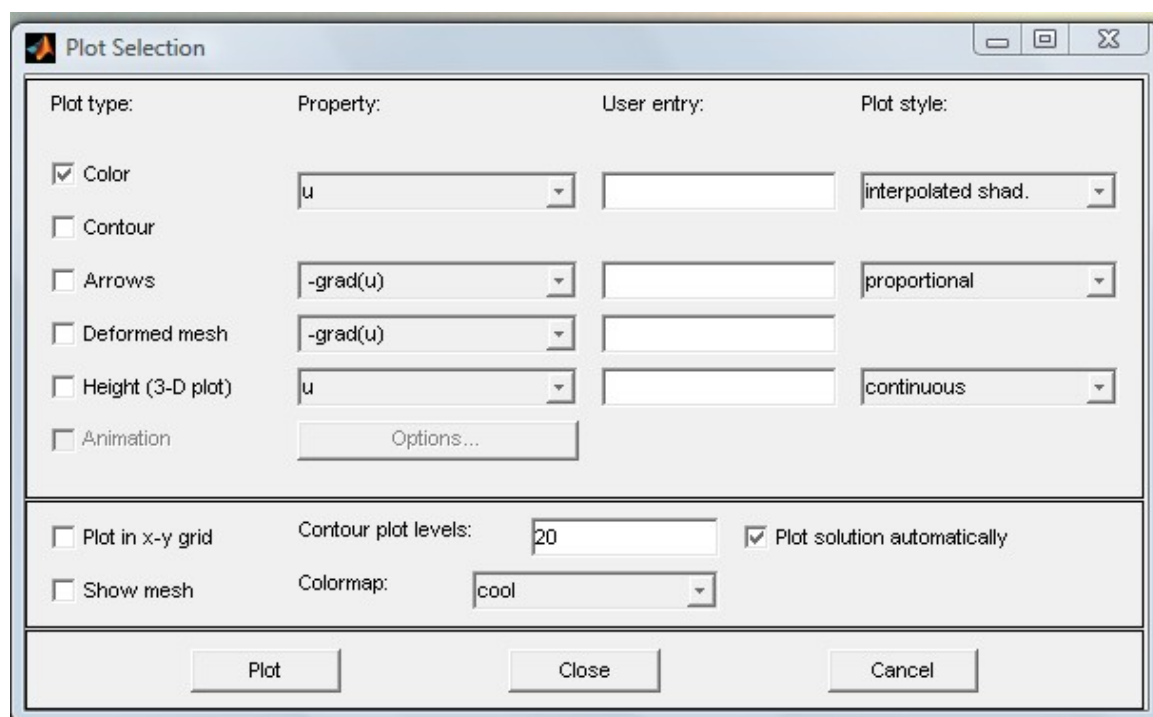


Рис. 10. Параметры графика

Еще более наглядным является представление решения в виде поверхности $u = u_n(x, y)$. Для этого нужно использовать кнопку с изображением поверхности (справа от кнопки со знаком $=$) или последовательность команд **Plot**=>**Parameters**. В обоих случаях открывается форма (рис.10). В ней, в частности, можно задавать следующие параметры:

- цвет;
- линии уровня решения (Contour);
- стрелки, указывающие направление, противоположное градиенту решения (Arrows);
- изображение поверхности (Height (3-D plot));
- координатную сетку (Plot in x-y grid);
- показывать триангуляцию (Show mesh);
- число линий уровня (Contour plot levels).

7. О возможностях **pdetool**

С помощью **pdetool** можно решать довольно широкий круг краевых задач для уравнений с частными производными, однако этот инструмент имеет ряд ограничений. Укажем некоторые из них.

а) только двумерные задачи:

визуальные средства **pdetool** позволяют решать краевые задачи только на плоскости;

б) линейные уравнения только 2-го порядка;

в) области, допустимые в **pdetool**, могут быть только результатом объединений и пересечений нескольких многоугольников и эллипсов;

г) представление результатов.

Действительно, решение с помощью **pdetool** представлено в визуальной форме. Это позволяет быстро оценить решение на качественном уровне. Кроме того, можно найти значение приближенного решения в любой заданной точке. Для этого нужно поместить курсор мыши в заданную точку и нажать левую кнопку мыши. В результате под рабочим окном появится номер треугольника триангуляции, в котором находится заданная точка, и значение приближенного решения в этой точке. Например, так можно найти точку, в которой решение принимает наибольшее значение.

Во многих случаях требуется иметь полный набор значений решения для дальнейшей работы с ним (например, вычислить интеграл). Это невозможно сделать средствами **pdetool**. В этом случае нужно использовать разнообразные

функции, находящиеся в том же подкаталоге **toolbox\pde** , что и **pdetool**, и позволяющие снять перечисленные выше ограничения. Эти функции написаны на языке **Matlab**, и их можно использовать в программах, написанных на этом языке, наряду с обычными элементарными функциями.

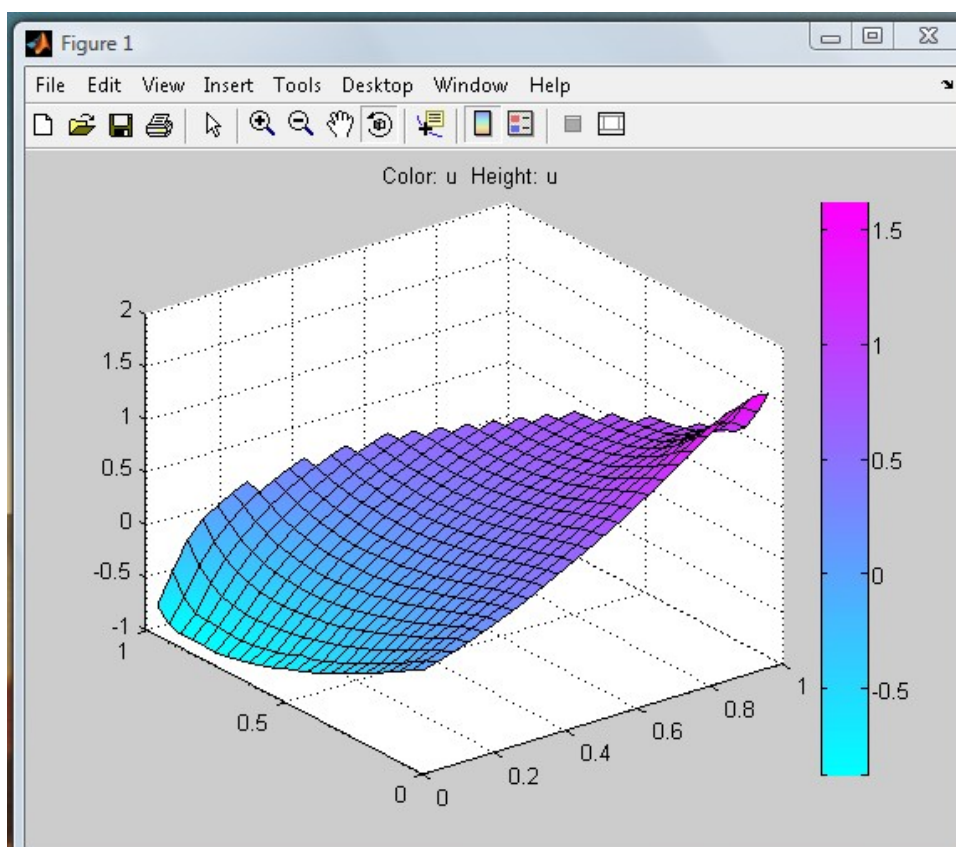


Рис. 11. Поверхность $z = u(x, y)$

Сохранение решения

В процессе работы с **pdetool** каждое наше действие автоматически порождает соответствующий программный код. Этот код накапливается в оперативной памяти. При завершении работы **pdetool** предлагает сохранить итоговый код в виде m-файла. По умолчанию этот файл имеет имя Untitled, но можно задать любое имя (используя латинский алфавит). Пусть, например, сохраняемому файлу присвоено имя Dirichlet. По умолчанию он сохраняется в рабочем каталоге \Matlab\work (можно задать любой другой каталог).

Ознакомиться с содержимым файла можно, зайдя в этот каталог и щелкнув по имени файла. При этом запустится редактор Matlab'a и автоматически откроется файл Dirichlet.m. Он является обычным текстовым файлом, содержащим команды языка Matlab. Открыть этот файл можно и в любом простом текстовом редакторе, даже в Notepad. Можно открывать и редактировать этот файл также в текстовом процессоре Word, однако сохранять его нужно как простой текстовый файл.

Пусть файл Dirichlet.m находится в каталоге \Matlab\work. Введем команду Dirichlet в командную строку Matlab, после чего Matlab загрузит этот файл в оперативную память, исполнит все команды файла Dirichlet и тем самым воспроизведет ход вашего решения. Заметим, что сохранять рабочий файл можно на любой стадии решения. Если почему-либо вы должны прервать работу над данной задачей, нужно сохранить файл, не выходя из **pdetool**, и затем закрыть **pdetool**. Впоследствии можно открыть этот файл в **pdetool** и продолжить работу с прерванного места.

Библиографический список

1. Михлин С.Г. Курс математической физики. – М.: Наука, 1968.
2. Тихонов А.Н., Самарский А.А. Уравнения математической физики. – М.: Наука, 1968.

Зураб Леванович Абжандадзе,
Ольга Евгеньевна Куляхтина,
Марк Эльявич Юдовин

Краевые задачи в системе Matlab

Методическое пособие
для студентов направления
01.03.02 «Прикладная математика и информатика»

Редактор и корректор В.А.Басова
Техн. редактор Л.Я.Титова

Темплан 2015., поз., 42

Подп. к печати 12.05.15 . Формат 60x84/16. Бумага тип №1.
Объем 1,75 печ.л.; 1,75 уч.-изд.л. Тираж 50 экз.
Печать офсетная. Изд. № 42. Цена «С». Заказ

Ризограф Санкт-Петербургского государственного технологического
университета растительных полимеров, 198095, Санкт-Петербург, ул. Ивана
Черных, 4