

Козин Р.Г.

# **МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ**

Учебное пособие

Москва  
МИФИ, 2008

УДК 519.876.5(075)  
ББК 22.19я7  
К59

**Козин Р.Г. Математическое моделирование: учеб. пособие. – М.: МИФИ, 2008. – 89 с.**

Учебное пособие в конспективной форме знакомит с практическим опытом решения математических задач, которые принадлежат различным предметным областям знаний.

Пособие написано по материалам лекций, прочитанных в МИФИ, и предназначено для студентов и специалистов, занимающихся математическим моделированием.

© (P) МИФИ, 2008

## Содержание

Введение	
1. Моделирование работы трубчатого химического реактора .....	4
2. Модель биосистемы «хищник – жертва» .....	8
3. Оптимизация работы магазина .....	14
4. Моделирование движения точки под действием центральной силы притяжения и силы возмущения .....	21
5. Моделирование виртуального мира .....	28
6. Задача оптимизации параметров балки .....	39
7. Оптимизация параметров точечных источников энергии, обеспечивающих равномерное стационарное температурное поле в плоской прямоугольной области .....	48
8. Построение оптимального температурного поля для плоской усеченной прямоугольной области .....	67
9. Задача оптимизации температурного поля для произвольной области и при наличии нелинейности .....	75
Список литературы	

## Введение в математического моделирования

Математическое моделирование (ММ) – исследование реальных явлений или поведения объектов с помощью математических моделей. К ММ прибегают в двух случаях: во-первых, явление или объект таковы, что их натурное исследование невозможно (нет объекта, не «пощупаешь» явление), во-вторых, реальное исследование слишком дорого и трудоемко.

ММ распадается на следующие этапы:

- формулировка задачи исследования на содержательном уровне
- уточнение задачи, концептуальная постановка задачи – знакомство с терминологией предметной области, разумный выбор ограничений, составление списка параметров  $p_i$ , входных  $x_i$  и выходных  $y_i$  переменных, основополагающих теоретических или экспериментальных зависимостей между ними
- формализация задачи – построение математической модели  $L(y_i, x_i, p_i) = 0$ , описывающей реальное явление или поведение объекта. При этом используются законы сохранения (энергии, вещества или другой субстанции) и экспериментальные зависимости. Как правило, модели представляют собой набор простых или дифференциальных уравнений, функционалов оптимизации и т.д.
- обобщение математической задачи, путем приведения ее к безразмерному виду и сокращению числа параметров модели
- тестирование и проверка адекватности построенной модели реальным прототипам. Для этого привлекаются экспериментальные данные, инварианты, законы сохранения и используются упрощенные варианты модели, имеющие простое решение в виде аналитических выражений
- выбор оптимального метода решения составленной математической задачи. Метод должен обладать устойчивостью, экономичностью и логической простотой алгоритма
- реализация программного продукта, который решает поставленную математическую задачу и выдает результаты в наиболее рациональной и удобоваримой форме
- исследование математической модели с помощью разработанного программного продукта.

В данном пособии особенности реализации указанных этапов обсуждаются на примерах решения конкретных задач.

### 1. Моделирование работы трубчатого химического реактора

#### 1.1. Постановка задачи

Рассматривается трубчатый химический реактор, в котором в результате изотермической реакции первого порядка происходит преобразование входного реагента А в выходной реагент В. Имеет место конвективный перенос вещества вдоль трубы и диффузия. Требуется определить длину реактора  $L$ , обеспечивающую на выходе полное преобразование входного реагента.

Заданы:

$C_0$  - входная концентрация реагента А;

$v$  - скорость перемещения реагента в реакторе;

$K$  - скорость химической реакции;

$E$  - эффективный коэффициент диффузии реагента.

#### 1.2. Построение математической модели

Получим математическую модель, описывающую изменение концентрации  $C(x)$  реагента А при перемещении вдоль реактора. Для этого запишем закон сохранения вещества в элементе трубы реактора длиной  $dx$  (рис. 1) с учетом конвективного и диффузионного потоков реагента (количество вещества переносимого в единицу времени через единичные граничные сечения) и скорости его преобразования в реагент В

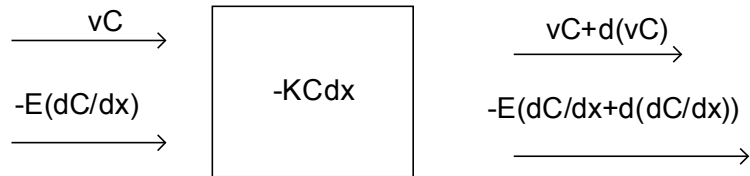


Рис. 1. Потоки реагента А через трубу химического реактора длиной  $dx$

$$vC - E \frac{dC}{dx} = vC + \frac{d(vC)}{dx} dx - E \left( \frac{dC}{dx} + \frac{d^2C}{dx^2} dx \right) + KCdx$$

Отсюда легко выводится дифференциальное уравнение, определяющее изменение концентрации реагента А вдоль координаты  $x$  (принимается, что  $v = const$ )

$$E \frac{d^2C}{dx^2} - v \frac{dC}{dx} - KC = 0 \quad (1)$$

Граничные условия для данной задачи:

- на входе реактора

$$x = 0, \quad vC_0 = vC - E \frac{dC}{dx} \quad (2)$$

- на выходе реактора (напомним, что  $L$  следует определить)

$$x = L, \quad \frac{dC}{dx} = 0 \quad (2)$$

Введем безразмерные величины:

$$y = \frac{C}{C_0}, \quad x \equiv \frac{x}{L}, \quad p1 = \frac{vL}{E}, \quad p2 = \frac{KL}{v} \quad (3)$$

Тогда (1), (2) переписутся в следующем виде

$$\frac{d^2y}{dx^2} - p1 \frac{dy}{dx} - p1p2y = 0 \quad (4)$$

$$x = 0, \quad 1 = y - \frac{1}{p1} \frac{dy}{dx}; \quad x = 1, \quad \frac{dy}{dx} = 0 \quad (5)$$

### 1.3. Исследование модели

Решение уравнения (4) (использована подстановка в уравнение зависимости  $y = \exp(sx)$ , которая приводит к квадратному уравнению для определения неизвестного параметра  $s$ :  $s^2 - p1s - p1p2 = 0$ )

$$y(x) = c_1 \exp(s_1x) + c_2 \exp(s_2x) \quad (6)$$

где

$$s_{1,2} = \frac{p1 \pm \sqrt{p1(p1 + 4p2)}}{2} \quad (7)$$

Постоянные интегрирования в (6) определяются из граничных условий (5)

$$c_1 = \frac{-\frac{s_2}{s_1} \exp(s_2 - s_1)}{\Delta} \quad (8)$$

$$c_2 = \frac{1}{\Delta}$$

где

$$\Delta = \left(1 - \frac{s_2}{p1}\right) - \frac{s_2}{s_1} \left(1 - \frac{s_1}{p1}\right) \exp(s_2 - s_1)$$

Обсудим стратегию поиска длины реактора  $L$ , при которой реагент А полностью переходит в реагент В, т.е.  $C(L) \approx 0$  или в новых обозначениях  $y(1) \approx 0$ .

Введем новый параметр, независящий от  $L$

$$\alpha = \frac{p1}{p2} = \frac{v^2}{KE} \quad (9)$$

Тогда поставленную задачу можно решить следующим образом:

1. Для заданных параметров задачи ( $K, E, v$ ), рассчитываем  $\alpha$ , а затем с помощью формул (6)-(8) (в них  $p1 = \alpha p2$ ) методом перебора («метод стрельб») определяем значение  $p2$ , при котором  $y(1) \approx 0$ . Интерфейс программы, позволяющей реализовать эту процедуру, представлен на рис. 2.

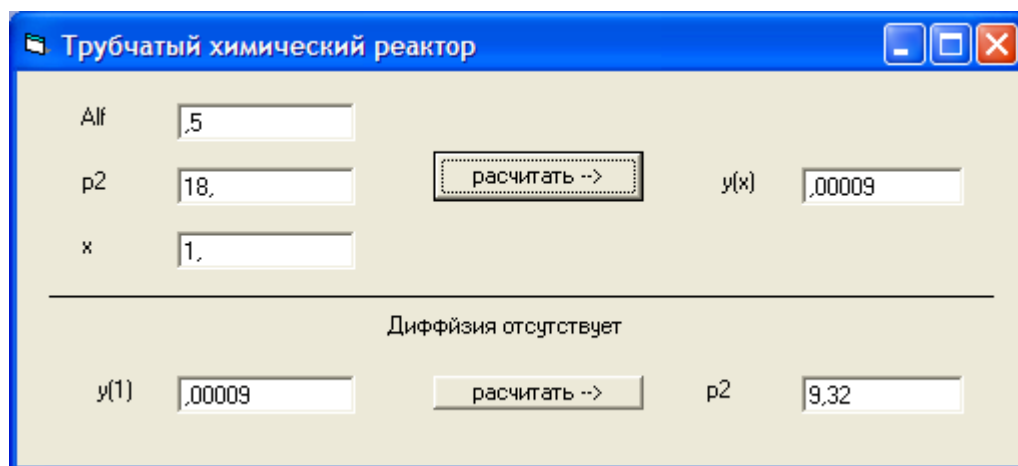


Рис. 2. Форма программы для определения относительной концентрации реагента для заданного  $x$

2. По найденному параметру  $p2$  вычисляем требуемую длину реактора  $L = \frac{v p2}{K}$ .

На рис. 3 построен предельный график (использовался пакет Maple 7) изменения относительной концентрации реагента  $y$  вдоль относительной длина реактора  $x$  для случая  $\alpha = 0.5$ . Здесь  $y(1) = 0.00009$ , когда  $p2 = 18$ .

Перебирая различные значения  $\alpha$ , можно построить номограмму (рис. 4 и 5) для нахождения  $L$  (по параметрам задачи рассчитываем  $\alpha$ , по графику определяем  $p2$ , а затем и  $L$ ). При решении указанной задачи принималось, что допустимое значение реагента на выходе реактора равно  $y(1) = 0.00009$ . Видно, что с ростом  $\alpha$  (постепенное уменьшение вклада диффузии в общий поток реагента А) значение  $p2$  стремится к своему предельному значению  $p2_{\text{предельное}} = 9.32$  (см. следующий параграф).

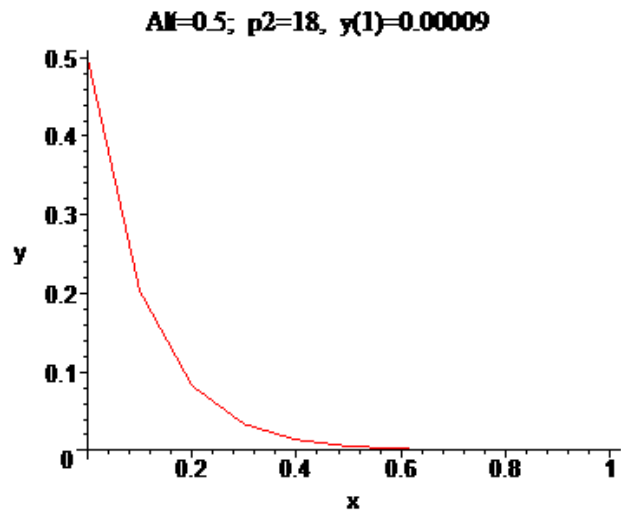


Рис. 3. Изменение относительной концентрации исходного реагента вдоль относительной длины химического реактора для случая  $\alpha = 0.5$  и  $p_2 = 18$

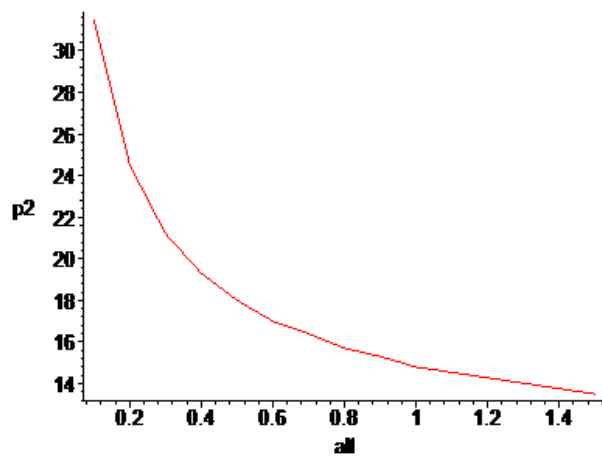


Рис. 4. Номограмма, устанавливающая зависимость  $p_2$  от  $\alpha$

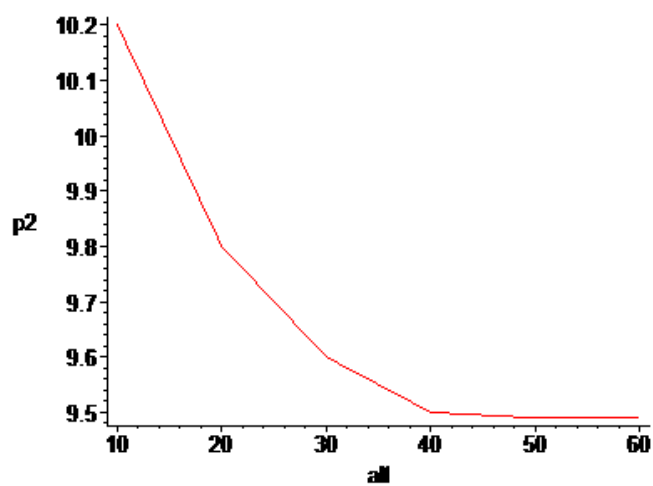


Рис.5. Продолжение графика рис. 4

#### 1.4. Исследование модели при отсутствии диффузии

Рассмотрим ситуацию, когда диффузией реагента А можно пренебречь ( $E \approx 0$ ). В этом случае дифференциальная задача (4), (5) переписывается следующим образом

$$\begin{aligned} \frac{dy}{dx} + p^2 y &= 0 \\ x = 0, \quad y &= 1 \end{aligned} \quad (10)$$

Ее решение имеет вид

$$y(x) = \exp(-p^2 x) \quad (11)$$

В соответствии с (11) условие для нахождения  $p^2$ , а, следовательно, и  $L = \frac{v p^2}{K}$ , записывается в виде

$$p^2 = \ln\left(\frac{1}{y(1)}\right) \quad (12)$$

Отсюда, принимая  $y(1) = 0.00009$ , получим  $p^2_{\text{предельное}} = 9.32$ .

На рис. 6 приведен график изменения относительной концентрации реагента А вдоль относительной длины химического реактора для этого предельного случая.

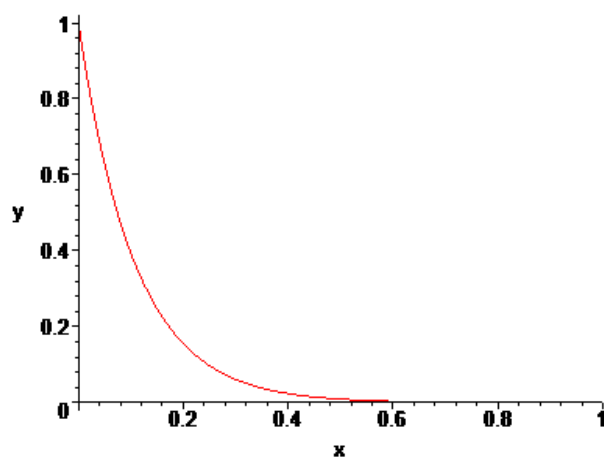


Рис. 6. Изменения относительной концентрации реагента А вдоль относительной длины химического реактора при отсутствии диффузии.

## 2. Модель биосистемы «хищник – жертва»

### 2.1. Формулировка задачи

Построим модель, описывающую изменения численности двух биологических популяций, взаимодействующих по принципу «хищник – жертва», и исследуем характер поведения этой модели в зависимости от ее параметров.

При выводе уравнений математической модели будем руководствоваться следующими соображениями:

- при отсутствии хищников скорость изменения жертв пропорциональна их численности, причем коэффициент пропорциональности определяется только их естественной смертностью и рождаемостью;
- при наличии хищников скорость убыли жертв пропорциональна количеству, как жертв, так и хищников (чем больше жертв, тем легче их поймать хищнику; чем больше хищников, тем больше жертв они съедят);
- скорость естественной убыли изолированных хищников пропорциональна их численности, а скорость прироста за счет съедания жертв – численности обеих видов.



## 2.2. Математическая модель биосистемы

При сделанных предположениях изменение численности популяций в указанной биосистеме можно описать системой дифференциальных уравнений модели Лотки-Вольтерра [1,2])

$$\begin{aligned}\frac{dx}{dt} &= x(a_1 - b_1 y) \\ \frac{dy}{dt} &= -y(a_2 - b_2 x)\end{aligned}\quad (1)$$

Здесь коэффициенты  $a_i, b_i > 0$  и их размерность  $[a_i], [b_i] \approx [1/t]$ .

В данной модели не учитывается внутривидовая конкуренция за ресурсы и допускается, что при отсутствии хищников жертвы, имеющие неограниченное количество пищи, будут размножаться неограниченно.

Приравняв нулю правые части уравнений (1), получим две точки равновесия биосистемы:  $x_p = 0, y_p = 0$ ;  $x_p = \frac{a_2}{b_2}, y_p = \frac{a_1}{b_1}$ . Иногда эти точки называют стационарными или особыми. Первая тривиальная точка нас не интересует, поскольку она не соответствует исходной постановке задачи.

Используя координаты полученной точки равновесия, введем новые безразмерные величины:

$$t \equiv ta_2, \quad x \equiv \frac{x}{x_p}, \quad y \equiv \frac{y}{y_p}, \quad q = \frac{a_1}{a_2}\quad (2)$$

Тогда уравнения (1) переписутся в следующем виде ( $q > 0$ )

$$\begin{aligned}\frac{dx}{dt} &= qx(1 - y) \equiv F_1(x, y) \\ \frac{dy}{dt} &= -y(1 - x) \equiv F_2(x, y)\end{aligned}\quad (3)$$

## 2.3. Исследование математической модели

В новых относительных переменных биосистема имеет следующую не тривиальную точку равновесия  $x_p = 1, y_p = 1$ . Исследуем поведение биосистемы возле этой точки. Для этого введем переменные  $\delta x = x - x_p, \delta y = y - y_p$  и разложим правые части системы (3) в ряды возле точки равновесия, ограничивая их слагаемыми порядка  $O(\delta x, \delta y)$ . При этом для переменных  $\delta x, \delta y$  получим следующую линейную систему дифференциальных уравнений

$$\begin{aligned}\frac{d\delta x}{dt} &= \left(\frac{\partial F_1}{\partial x}\right)_p \delta x + \left(\frac{\partial F_1}{\partial y}\right)_p \delta y \\ \frac{d\delta y}{dt} &= \left(\frac{\partial F_2}{\partial x}\right)_p \delta x + \left(\frac{\partial F_2}{\partial y}\right)_p \delta y\end{aligned}\quad (4)$$

где

$$\begin{aligned} \left(\frac{\partial F_1}{\partial x}\right)_p &= q(1 - y_p) = 0 \\ \left(\frac{\partial F_1}{\partial y}\right)_p &= -qx_p = -q \\ \left(\frac{\partial F_2}{\partial x}\right)_p &= y_p = 1 \\ \left(\frac{\partial F_2}{\partial y}\right)_p &= -(1 - x_p) = 0 \end{aligned}$$

Характеристическое уравнение системы (4)

$$\begin{vmatrix} -\lambda & -q \\ 1 & -\lambda \end{vmatrix} = 0$$

имеет два чисто мнимых корня

$$\lambda = i\sqrt{q}, \quad \lambda = -i\sqrt{q}$$

Это означает, что точка равновесия биосистемы  $x_p, y_p$  является центром, вокруг которого на фазовой плоскости  $x, y$  располагаются замкнутые фазовые траектории, и изменение численности популяций носит колебательный характер. Сказанное подтверждается графиками (рис.1 и 2), построенными для значения параметра  $q = 1,5$  и начальных условий  $x_0 = 0,5, y_0 = 0,9$

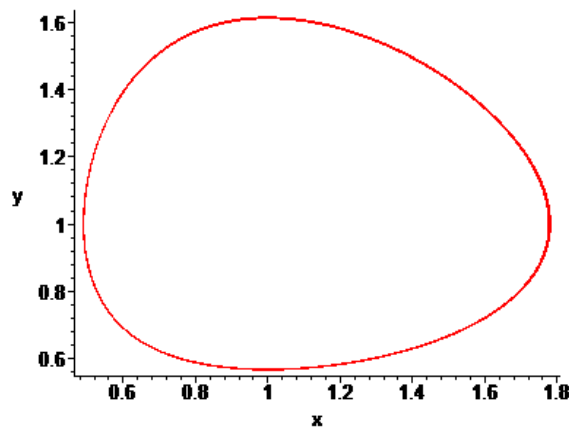


Рис. 1. Фазовая траектория для случая  $q = 1,5, x_0 = 0,5, y_0 = 0,9$

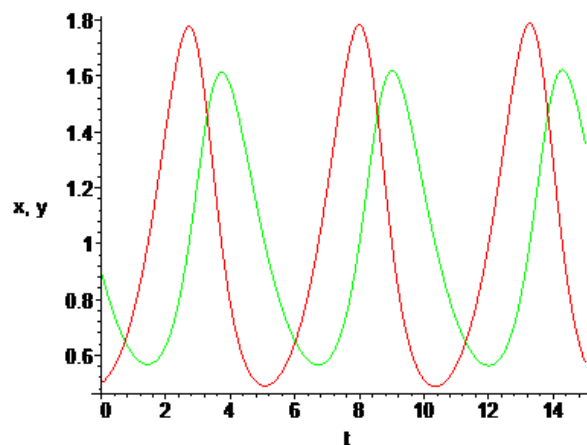


Рис. 2. Изменение при  $q = 1,5$  относительной численности жертв ( $x_0 = 0,5$ ) и хищников ( $y_0 = 0,9$ )

## 2.4. Расширение исходной модели биосистемы

Учтем в исходной модели внутривидовую конкуренцию за ограниченные ресурсы у обеих популяций. Можно считать, что столкновения представителей одной популяции ведут к снижению скорости роста этой популяции пропорционально квадрату ее численности. В соответствие с этим в уравнениях (1) появляются дополнительные слагаемые ( $c_1 \geq 0, c_2 \geq 0$ )

$$\begin{aligned} \frac{dx}{dt} &= x(a_1 - b_1 y - c_1 x) \\ \frac{dy}{dt} &= -y(a_2 - b_2 x + c_2 y) \end{aligned} \quad (5)$$

Данная система с помощью прежних безразмерных переменных (2) приводится к виду

$$\begin{aligned} \frac{dx}{dt} &= qx(1 - y - p_1 x) \equiv F_1(x, y) \\ \frac{dy}{dt} &= -y(1 - x + p_2 y) \equiv F_2(x, y) \end{aligned} \quad (6)$$

где

$$p_1 = \frac{c_1 a_2}{a_1 b_2}, \quad p_2 = \frac{c_2 a_1}{a_2 b_1}, \quad q = \frac{a_1}{a_2} \quad (7)$$

## 2.5. Исследование расширенной математической модели

Для новой модели (6) не тривиальная точка равновесия, определяемая из условия  $F_1(x, y) = 0, F_2(x, y) = 0$ , задается координатами

$$x_p = \frac{1 + p_2}{1 + p_1 p_2}, \quad y_p = \frac{1 - p_1}{1 + p_1 p_2} \quad (8)$$

Остальные точки равновесия, в которых одна из популяций вырождается, нас не интересуют. По этой же причине (требование  $y_p > 0$ ) из (8) следует ограничение на один из параметров  $0 \leq p_1 < 1$ .

Расписывая систему (6) возле этой точки равновесия, получим систему вида (4), в которой производные определяются формулами

$$\begin{aligned} \left(\frac{\partial F_1}{\partial x}\right)_p &= q(1 - y_p - p_1 x_p) - qp_1 x_p = -qp_1 x_p = -qp_1 \frac{1 + p_2}{1 + p_1 p_2} \\ \left(\frac{\partial F_1}{\partial y}\right)_p &= -qx_p = -q \frac{1 + p_2}{1 + p_1 p_2} \\ \left(\frac{\partial F_2}{\partial x}\right)_p &= y_p = \frac{1 - p_1}{1 + p_1 p_2} \\ \left(\frac{\partial F_2}{\partial y}\right)_p &= -(1 - x_p + p_2 y_p) - p_2 y_p = -p_2 y_p = -p_2 \frac{1 - p_1}{1 + p_1 p_2} \end{aligned} \quad (9)$$

Запишем его характеристическое уравнение

$$\begin{vmatrix} -qp_1 \frac{1 + p_2}{1 + p_1 p_2} - \lambda & -q \frac{1 + p_2}{1 + p_1 p_2} \\ \frac{1 - p_1}{1 + p_1 p_2} & -p_2 \frac{1 - p_1}{1 + p_1 p_2} - \lambda \end{vmatrix} = 0$$

или

$$\lambda^2 + \lambda \frac{qp_1(1+p_2) + p_2(1-p_1)}{(1+p_1p_2)} + q \frac{(1-p_1)(1+p_2)}{(1+p_1p_2)} = 0 \quad (10)$$

Корни этого уравнения в зависимости от значений параметров  $q, p_1, p_2$

$$\lambda_{1,2} = \frac{-[qp_1(1+p_2) + p_2(1-p_1)] \pm \sqrt{D}}{2(1+p_1p_2)} \quad (11)$$

где

$$D = [qp_1(1+p_2) + p_2(1-p_1)]^2 - 4q(1+p_1p_2)(1-p_1)(1+p_2)$$

могут быть числами двух типов: комплексно-сопряженными с отрицательной действительной частью (при  $D(q, p_1, p_2) < 0$ ), либо действительными, отрицательными (при  $D(q, p_1, p_2) \geq 0$ ).

В трехмерном пространстве параметров  $q, p_1, p_2$  первая зона с комплексно-сопряженными корнями, отделяется от второй зоны с действительными корнями поверхностью, которая задается уравнением  $D(q, p_1, p_2) = 0$  или в развернутом виде

$$q = 4 \frac{(1+p_1p_2)(1-p_1)(1+p_2)}{[p_1(1+p_2) + p_2(1-p_1)]^2} \quad (12)$$

Эта поверхность и ее срезы для частных случаев отсутствия внутривидовой конкуренции у одной из популяций представлены на рис. 3, 4 и 5.

Если параметры  $q, p_1, p_2$  принадлежат первой зоне (на графике рис. 3 она лежит под поверхностью, где  $D(q, p_1, p_2) < 0$ ), то точка равновесия, заданная координатами (8), является устойчивым фокусом на фазовой плоскости  $x, y$  (рис. 6), а изменение численности популяций имеет характер затухающих колебаний (рис. 7).

Для второй зоны (на графике рис. 3 она располагается над поверхностью, где  $D(q, p_1, p_2) > 0$ ) точка равновесия - устойчивый узел, одна из ветвей, которого представлена на рис. 8, а изменение численности популяций имеет аperiodический характер (рис. 9).

Итого, при наличии внутривидовой конкуренции численность популяций в биосистеме стационарна. При любых возмущениях численности биосистема возвращается в это равновесное состояние. Скорость возврата зависит от того, какой зоне принадлежит точка, координаты которой задаются значениями параметров системы  $q, p_1, p_2$ .

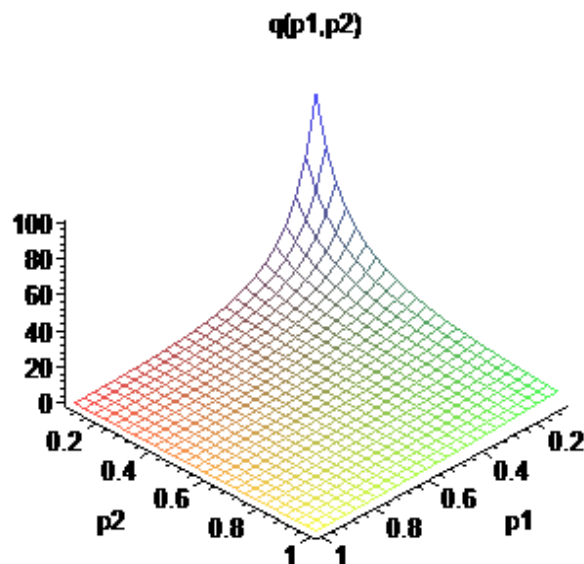


Рис. 3. Поверхность, определяемая уравнением  $D(q, p_1, p_2) = 0$

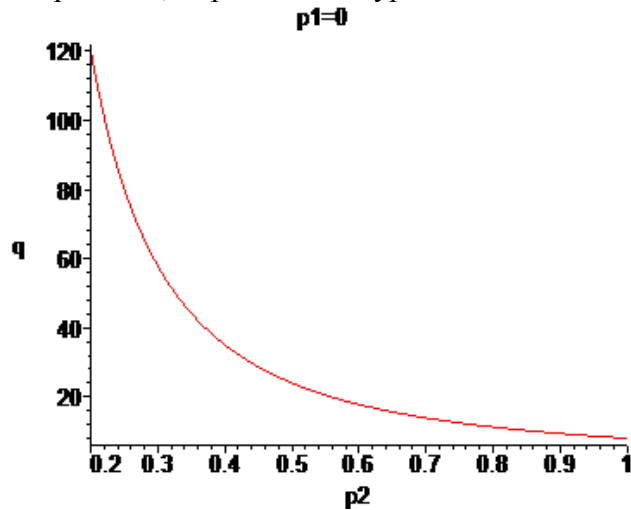


Рис. 4. Сечение поверхности  $D(q, p_1, p_2) = 0$  плоскостью  $p_1 = 0$

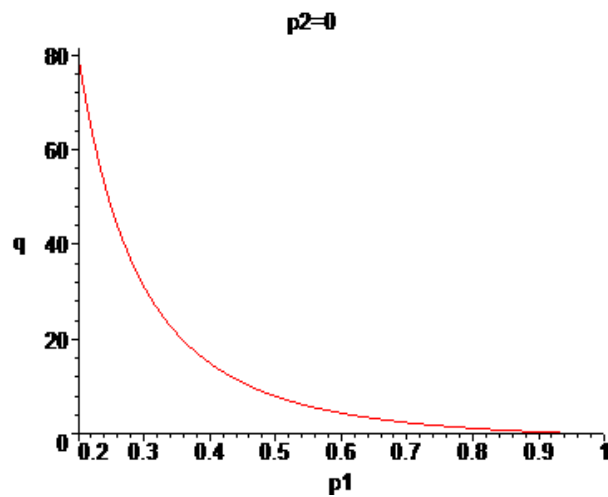


Рис. 5. Сечение поверхности  $D(q, p_1, p_2) = 0$  плоскостью  $p_2 = 0$

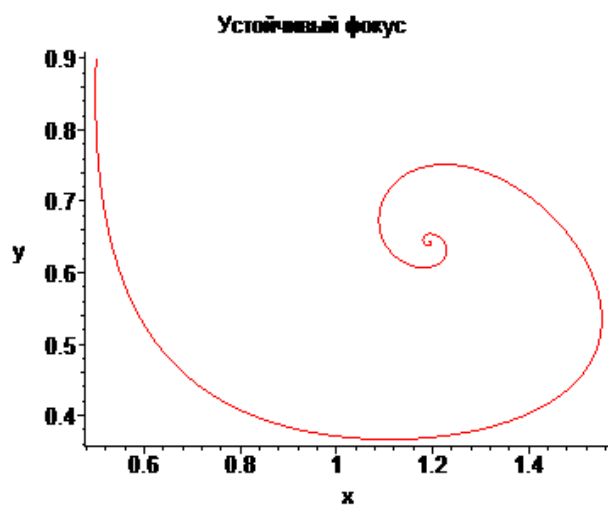


Рис. 6. . Фазовая кривая для случая устойчивого фокуса ( $q = 1.5, p_1 = 0.3, p_2 = 0.3, x(0) = 0.5, y(0) = 0.9, x_p = 1.19, y_p = 0.64$ )

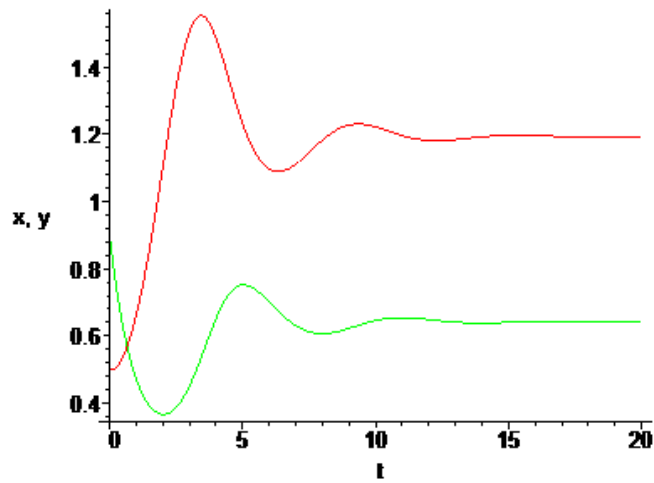


Рис. 7. . Кривые изменения численности популяций для случая устойчивого фокуса (  $q = 1.5, p_1 = 0.3, p_2 = 0.3, x(0) = 0.5, y(0) = 0.9, x_p = 1.19, y_p = 0.64$  )

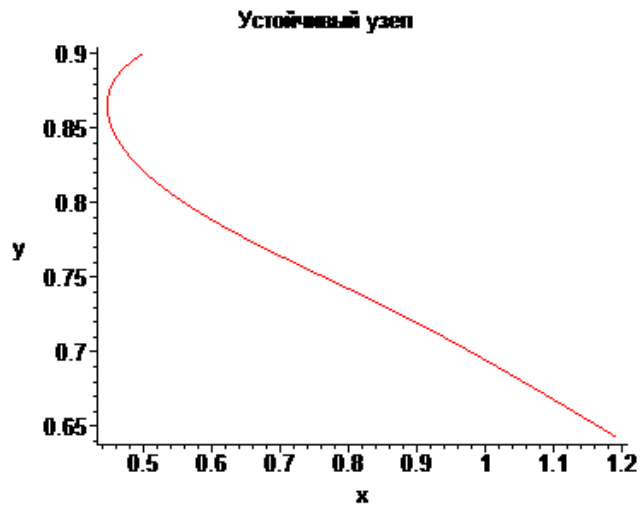


Рис. 8. Фазовая кривая для случая устойчивого узла (  $q = 100, p_1 = 0.3, p_2 = 0.3, x(0) = 0.5, y(0) = 0.9, x_p = 1.19, y_p = 0.64$  )

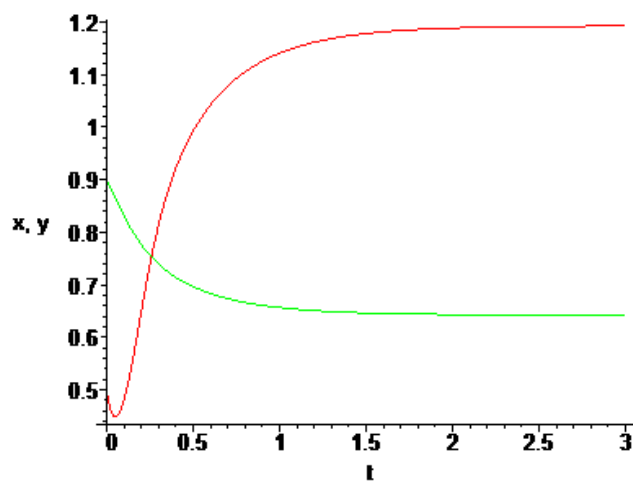


Рис. 9. Кривые изменения численности популяций для случая устойчивого узла (  $q = 100, p_1 = 0.3, p_2 = 0.3, x(0) = 0.5, y(0) = 0.9, x_p = 1.19, y_p = 0.64$  )

### 3. Оптимизация работы магазина

#### 3.1. Постановка задачи

Найти оптимальное число продавцов, обеспечивающих максимальную прибыль магазину.

Используемые исходные параметры:

$\lambda$  - среднее число покупателей, приходящих в магазин в единицу времени ( $\frac{1}{\lambda}$  - средний интервал времени между приходом покупателей);

$\mu$  - среднее число покупателей, обслуживаемых одним продавцом в единицу времени;

$\nu$  - среднее число покупателей, уходящих из общей очереди (нет времени или желания ждать) не обслуженными, в единицу времени;

$s_1$  - средняя сумма прибыли от одного покупателя в единицу времени;

$s_2$  - средняя зарплата продавца в единицу времени,

$m$  - среднее максимальное число покупателей в общей очереди (при виде такой «длинной» очереди покупатель сразу уходит из магазина),

$n$  - число продавцов в магазине.

#### 3.2. Математическая формулировка задачи

Магазин можно рассматривать, как систему массового обслуживания (СМО), имеющую конечное число состояний (такие как: все продавцы свободны, обслуживаются  $1, 2, \dots, n$  покупателей,  $1, 2, \dots, m$  покупателей стоят в очереди), переход между которыми регулируется простейшими потоками случайных событий (приход покупателя, завершение обслуживания покупателя, уход покупателя из очереди) (см. рис 1).

Простейший поток случайных событий характеризуется следующими свойствами [3]:

1. Стационарность – вероятностные характеристики не зависят от времени.
2. Отсутствие последствий – события происходят независимо друг от друга.
3. Ординарность – события происходят поодиночке.

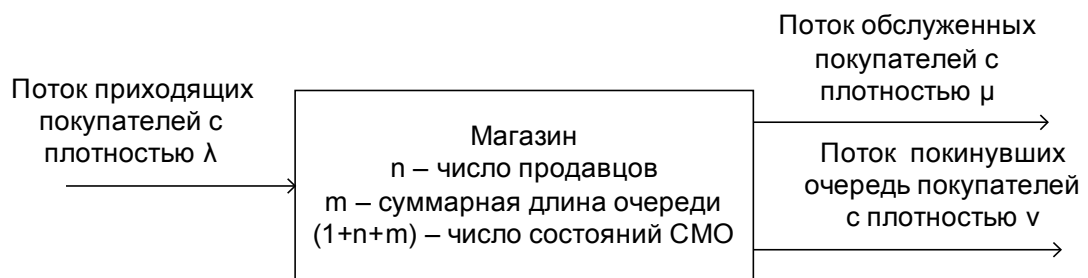


Рис. 1. Схема постановки задачи

Для простейшего потока число событий, попадающих на временной интервал  $dt$ , распределено по закону Пуассона, т.е. вероятность того, что за время  $dt$  произойдет  $k$  событий, равно

$$p_k(dt) = \frac{(\lambda dt)^k}{k!} e^{-\lambda dt}$$

где  $\lambda$  - среднее число событий, происходящих в единицу времени.

Отсюда с учетом ординарности потока при  $dt \rightarrow 0$  имеем:

$p_0(dt) = e^{-\lambda dt} \approx 1 - \lambda dt$  - вероятность, что за время  $dt$  не произошло ни одного события;

$1 - p_0(dt) = 1 - e^{-\lambda dt} \approx \lambda dt$  - вероятность, что за время  $dt$  произошло одно событие.

Составим систему дифференциальных уравнений, которые определяют изменения вероятностей  $p_0(t), p_1(t), \dots, p_n(t), p_{n+1}(t), p_{n+2}(t), \dots, p_{n+m}(t)$  всех возможных состояний магазина, как СМО, в зависимости от времени.

Пусть в момент времени  $t$  СМО находится в состоянии  $k$  ( $0 < k < n$ ) с вероятностью  $p_k(t)$ . Найдем вероятность того, что и в момент времени  $t + dt$  система останется в том же состоянии. Для этого необходимо сложить вероятности следующих событий (см. рис.2):

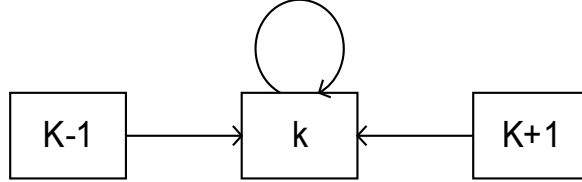


Рис. 2. Схема переходов для  $k$ -го состояния

1. Система находилась в состоянии  $k$  и осталась в нем (за время  $dt$  не пришел ни один покупатель и не освободился ни один из  $k$  продавцов)

$$p_k(t)e^{-\lambda dt}(e^{-\mu dt})^k \approx p_k(t)(1 - \lambda dt)(1 - k\mu dt) \approx p_k(t)[1 - (\lambda + k\mu)dt]$$

2. Система находилась в состоянии  $k - 1$  и перешла в состояние  $k$  (за время  $dt$  пришел один покупатель и не освободился ни один из  $k - 1$  продавцов)

$$p_{k-1}(t)(1 - e^{-\lambda dt})(e^{-\mu dt})^{k-1} \approx p_{k-1}(t)(\lambda dt)[1 - (k - 1)\mu dt] \approx p_{k-1}(t)(\lambda dt)$$

3. Система находилась в состоянии  $k + 1$  и перешла в состояние  $k$  (за время  $dt$  не пришел ни один покупатель и освободился один из  $k + 1$  продавцов)

$$p_{k+1}(t)e^{-\lambda dt}(k + 1)(1 - e^{-\mu dt}) \approx p_{k+1}(t)(1 - \lambda dt)(k + 1)\mu dt \approx p_{k+1}(t)(k + 1)\mu dt$$

Итого имеем

$$p_k(t + dt) \approx p_k(t)[1 - (\lambda + k\mu)dt] + p_{k-1}(t)\lambda dt + p_{k+1}(t)(k + 1)\mu dt$$

Отсюда, используя соотношение  $\frac{p_k(t + dt) - p_k(t)}{dt} \rightarrow \frac{dp_k(t)}{dt}$  при  $dt \rightarrow 0$ , получим

дифференциальные уравнения для вероятностей  $p_k(t)$ ,  $0 \leq k < n$

$$\begin{aligned} \frac{dp_0}{dt} &= -\lambda p_0 + \mu p_1 \\ \frac{dp_k}{dt} &= \lambda p_{k-1} - (\lambda + k\mu)p_k + (k + 1)\mu p_{k+1} \quad k = 1, 2, \dots, (n - 1) \end{aligned} \quad (1)$$

Теперь запишем пункты 1-3 для случая  $k \equiv n + s$ , где  $0 \leq s < m$ .

1. Система находилась в состоянии  $n + s$  и осталась в нем (за время  $dt$  не пришел ни один покупатель, не освободился ни один из  $n$  продавцов и ни один из  $s$  покупателей не ушел из общей очереди)

$$p_{n+s}(t)e^{-\lambda dt}(e^{-\mu dt})^n(e^{-\nu dt})^s \approx p_{n+s}(t)(1 - \lambda dt)(1 - n\mu dt)(1 - s\nu dt) \approx p_{n+s}(t)[1 - (\lambda + n\mu + s\nu)dt]$$

2. Система находилась в состоянии  $n + s - 1$  и перешла в состояние  $n + s$  (за время  $dt$  пришел один покупатель, не освободился ни один из  $n$  продавцов и ни один из  $s - 1$  покупателей не ушел из общей очереди)

$$p_{n+s-1}(t)(1 - e^{-\lambda dt})(e^{-\mu dt})^n(e^{-\nu dt})^{s-1} \approx p_{n+s-1}(t)(\lambda dt)[1 - n\mu dt - (s - 1)\nu dt] \approx p_{n+s-1}(t)(\lambda dt)$$

3. Система находилась в состоянии  $n + s + 1$  и перешла в состояние  $n + s$  (за время  $dt$  не пришел ни один покупатель и освободился один из  $n$  продавцов или один из  $s + 1$  покупателей покинул очередь)

$$\begin{aligned} p_{n+s+1}(t)e^{-\lambda dt}[n(1 - e^{-\mu dt}) + (s + 1)(1 - e^{-\nu dt})] &\approx p_{n+s+1}(t)(1 - \lambda dt)[n\mu dt + (s + 1)\nu dt] \approx \\ p_{n+s+1}(t)[n\mu + (s + 1)\nu]dt & \end{aligned}$$

Отсюда окончательно получим



$$\frac{dp_{n+s}}{dt} = \lambda p_{n+s-1} - (\lambda + n\mu + s\nu) p_{n+s} + [n\mu + (s+1)\nu] p_{n+s+1} \quad s = 0, 1, 2, \dots, (m-1) \quad (2)$$

Осталось вывести уравнение для вероятности состояния  $n+m$ . Для этого необходимо рассмотреть только два вышеперечисленных пункта (третий пункт отбрасывается, т.к. при наличии полной очереди покупатель сразу покидает магазин).

1. Система находилась в состоянии  $n+m$  и осталась в нем (за время  $dt$  не освободился ни один из  $n$  продавцов и ни один из  $m$  покупателей не ушел из общей очереди или пришел один покупатель и освободился один из продавцов или один из покупателей ушел из очереди)

$$\frac{p_{n+m}(t) \{ (e^{-\mu dt})^n (e^{-\nu dt})^m + (1 - e^{-\lambda dt}) [n(1 - e^{-\mu dt}) + m(1 - e^{-\nu dt})] \}}{p_{n+m}(t) [(1 - n\mu dt)(1 - m\nu dt) + \lambda dt(n\mu dt + m\nu dt)]} \approx p_{n+m}(t) [1 - (n\mu + m\nu) dt]$$

2. Система находилась в состоянии  $n+m-1$  и перешла в состояние  $n+m$  (за время  $dt$  пришел один покупатель, не освободился ни один из  $n$  продавцов и ни один из  $m-1$  покупателей не ушел из общей очереди)

$$p_{n+m-1}(t)(1 - e^{-\lambda dt})(e^{-\mu dt})^n (e^{-\nu dt})^{m-1} \approx p_{n+m-1}(t)(\lambda dt)[1 - n\mu dt - (m-1)\nu dt] \approx p_{n+m-1}(t)(\lambda dt)$$

С помощью этих соотношений получаем требуемое уравнение

$$\frac{dp_{n+m}}{dt} = \lambda p_{n+m-1} - (n\mu + m\nu) p_{n+m} \quad (3)$$

Ведem безразмерные параметры:  $\tau = \lambda t$  (перешли от времени к среднему числу покупателей, пришедших к данному моменту времени),  $\alpha = \frac{\mu}{\lambda}$ ,  $\beta = \frac{\nu}{\lambda}$ , где  $\alpha$  - число покупателей обслуженных продавцом за средний интервал времени между приходом покупателей,  $\beta$  - число покупателей покинувших очередь за тот же интервал времени. Тогда система (1)-(3) перепишется в виде

$$\begin{aligned} \frac{dp_0}{d\tau} &= -p_0 + \alpha p_1 \\ \frac{dp_k}{d\tau} &= p_{k-1} - (1 + k\alpha) p_k + (k+1)\alpha p_{k+1} \quad k = 1, 2, \dots, (n-1) \\ \frac{dp_{n+s}}{d\tau} &= p_{n+s-1} - (1 + n\alpha + s\beta) p_{n+s} + [n\alpha + (s+1)\beta] p_{n+s+1} \quad s = 0, 1, 2, \dots, (m-1) \\ \frac{dp_{n+m}}{d\tau} &= p_{n+m-1} - (n\alpha + m\beta) p_{n+m} \end{aligned} \quad (4)$$

Систему уравнений (4) следует дополнить начальными условиями

$$p_0(0) = 1, p_1(0) = p_2(0) = \dots, p_{n+m}(0) = 0 \quad (5)$$

Система (4) и условия (5) представляют собой математическую задачу Коши для системы обыкновенных дифференциальных уравнений. Ее численное интегрирование можно выполнить методом Рунге-Кутты 2-го порядка, известным также, как метод «предиктор-корректор» или более точно методом Рунге-Кутты 4-го порядка (использована векторная запись задачи:  $\frac{d\bar{p}}{dt} = \bar{F}(t, \bar{p}); t = 0, \bar{p}(0) = \bar{p}_0; j$  - номер шага интегрирования;  $h$

- шаг интегрирования)

1. Метод Рунге-Кутты 2-го порядка

$$\bar{p}_{j+1} = \bar{p}_j + \frac{h}{2} (\bar{F}(t_j, \bar{p}_j) + \bar{F}(t_j + h, \bar{p}_{j+1}^*)), \quad j = 0, 1, \dots$$

где

$$\bar{p}_{j+1}^* = \bar{p}_j + h\bar{F}(t_j, \bar{p}_j)$$

2. Метод Рунге-Кутты 4-го порядка

$$\bar{p}_{j+1} = \bar{p}_j + \frac{h}{6}(\bar{k}_1 + 2\bar{k}_2 + 2\bar{k}_3 + \bar{k}_4), \quad j = 0, 1, \dots$$

где

$$\begin{aligned} \bar{k}_1 &= \bar{F}(t_j, \bar{p}_j), \quad \bar{k}_2 = \bar{F}\left(t_j + \frac{h}{2}, \bar{p}_j + \frac{h}{2}\bar{k}_1\right) \\ \bar{k}_3 &= \bar{F}\left(t_j + \frac{h}{2}, \bar{p}_j + \frac{h}{2}\bar{k}_2\right), \quad \bar{k}_4 = \bar{F}(t_j + h, \bar{p}_j + h\bar{k}_3) \end{aligned}$$

Из уравнений (4) легко получить системы уравнений, описывающих изменения вероятностей для различных вариантов функционирования магазина.

1. Магазин без очереди - при отсутствии свободного продавца покупатель сразу покидает магазин ( $m = 0$ ).

$$\begin{aligned} \frac{dp_0}{dt} &= -p_0 + \alpha p_1 \\ \frac{dp_k}{dt} &= p_{k-1} - (1 + k\alpha)p_k + (k+1)\alpha p_{k+1} \quad k = 1, 2, \dots, (n-1) \\ \frac{dp_n}{dt} &= p_{n-1} - n\alpha p_n \end{aligned} \quad (6)$$

2. Магазин с неограниченной очередью – описывается уравнениями (4) (без последнего уравнения), в которых  $m = \infty$ .

3. Магазин с неограниченным временем ожидания покупателей в очереди – описывается уравнениями (4), в которых  $\beta = 0$ .

### 3.3. Стационарный случай

Если в системе (4) все производные приравнять нулю, то получим систему алгебраических уравнений для определения вероятностей состояний для установившегося режима

$$\begin{aligned} -p_0 + \alpha p_1 &= 0 \\ p_{k-1} - (1 + k\alpha)p_k + (k+1)\alpha p_{k+1} &= 0 \quad k = 1, 2, \dots, (n-1) \\ p_{n+s-1} - (1 + n\alpha + s\beta)p_{n+s} + [n\alpha + (s+1)\beta]p_{n+s+1} &= 0 \quad s = 0, 1, 2, \dots, (m-1) \\ p_{n+m-1} - (n\alpha + m\beta)p_{n+m} &= 0 \end{aligned} \quad (7)$$

Сюда необходимо добавить условие

$$\sum_{i=0}^{n+m} p_i = 1 \quad (8)$$

Из системы (7) можно последовательно, начиная с первого уравнения, выразить все вероятности  $p_i$  через  $p_0$

$$\begin{aligned}
p_1 &= \frac{1}{\alpha} p_0 \\
p_2 &= \frac{1}{2\alpha^2} p_0 \\
p_3 &= \frac{1}{3!\alpha^3} p_0 \\
&\dots\dots\dots \\
p_k &= \frac{1}{k!\alpha^k} p_0, \quad k = 4..n \\
p_{n+1} &= \frac{1}{n!\alpha^n(n\alpha + \beta)} p_0 \\
p_{n+2} &= \frac{1}{n!\alpha^n(n\alpha + \beta)(n\alpha + 2\beta)} p_0 \\
&\dots\dots\dots \\
p_{n+s} &= \frac{1}{n!\alpha^n \prod_{i=1}^s (n\alpha + i\beta)} p_0, \quad s = 3..m
\end{aligned}$$

Подставляя эти формулы в условие (8), находим и само  $p_0$

$$p_0 = \frac{1}{\sum_{k=0}^n \frac{1}{k!\alpha^k} + \sum_{s=1}^m \frac{1}{n!\alpha^n \prod_{i=1}^s (n\alpha + i\beta)}}$$

### 3.4. Статистические характеристики

С помощью рассчитанных вероятностей  $p_k(t)$  можно определить различные статистические характеристики, связанные с работой магазина:

1. Среднее число занятых продавцов

$$N_{\text{продавцов}} = \sum_{k=1}^n k p_k + n \sum_{s=1}^m p_{n+s}$$

2. Вероятность того, что произвольный продавец занят

$$p_{\text{занят}} = \frac{N_{\text{продавцов}}}{n}$$

3. Среднее время простоя продавца (следует из очевидного соотношения  $\frac{t_{\text{простоя}}}{t_{\text{занят}}} = \frac{P_{\text{простоя}}}{P_{\text{занят}}}$ )

$$t_{\text{простоя}} = \frac{1}{\mu} \frac{(1 - p_{\text{занят}})}{p_{\text{занят}}}$$

Здесь  $\frac{1}{\mu}$  - среднее время обслуживания одного покупателя,  $(1 - p_{\text{занят}})$  - вероятность простоя продавца.

4. Среднее число покупателей в очереди

$$N_{\text{очереди}} = \sum_{s=1}^m s p_{n+s}$$

5. Вероятность простоя всех продавцов

$$P_{\text{простоя}} = p_0$$

6. Вероятность отсутствия очереди

$$P_{\text{нет очереди}} = \sum_{k=0}^n P_k$$

7. Вероятность наличия очереди

$$P_{\text{очереди}} = 1 - \sum_{k=0}^n P_k$$

8. Среднее время пребывания покупателя в очереди (среднее число покупателей в очереди умножается на средний интервал между приходом покупателей  $\frac{1}{\lambda}$ )

$$t_{\text{очередь}} = \frac{N_{\text{очереди}}}{\lambda}$$

9. Вероятность ухода любого покупателя из очереди не обслуженным (вероятность ухода произвольного покупателя равна отношению среднего числа уходящих из очереди покупателей к среднему числу приходящих в магазин покупателей, умноженному на среднее число покупателей в очереди, т.к. имеет место сложение вероятностей)

$$P_{\text{отказа}} = \frac{\nu}{\lambda} N_{\text{очереди}} = \beta N_{\text{очереди}}$$

Отметим, что при  $m = 0$  (магазин без очереди)  $P_{\text{отказа}} = P_n$ .

10. Среднее число покупателей, обслуживаемых магазином в единицу времени (пропускная способность магазина)

$$N_{\text{обс. покупателей}} = \lambda (1 - P_{\text{отказа}})$$

Здесь  $(1 - P_{\text{отказа}})$  – вероятность отказа в обслуживании произвольного покупателя.

### 3.4. Оптимизация прибыли магазина

Прибыль магазина в единицу времени рассчитывается следующим образом

$$\phi(n) = s_1 N_{\text{обс. покупателей}} - n s_2 \quad (9)$$

Задача ее оптимизации решается для стационарного случая. При этом используется метод целочисленного перебора, когда значения прибыли последовательно рассчитываются для  $n = 1, 2, \dots$  и определяется  $n_{\text{opt}}$ , при котором прибыль имеет максимальное значение.

### 3.5. Программа для исследования статистических характеристик работы магазина

Изложенные результаты были использованы при реализации программы для исследования статистических характеристик работы магазина.

Приведем несколько экранов этой программы с полученными результатами.

1. Вход в программу (рис. 3). Справа показаны значения статистических параметров, рассчитанные для установившегося режима работы магазина.

2. Графики изменения статистических параметров, которые получены после численного интегрирования системы дифференциальных уравнений (4) методом Рунге-Кутты (рис. 4).

3. Результаты решения задачи поиска оптимального числа продавцов, обеспечивающего оптимальную прибыль магазину в единицу времени для введенных входных параметров (рис. 5). Видно, что для заданных параметров при оптимальном выборе числа продавцов можно повысить прибыль почти в три раза (см. рис. 3 и 5).

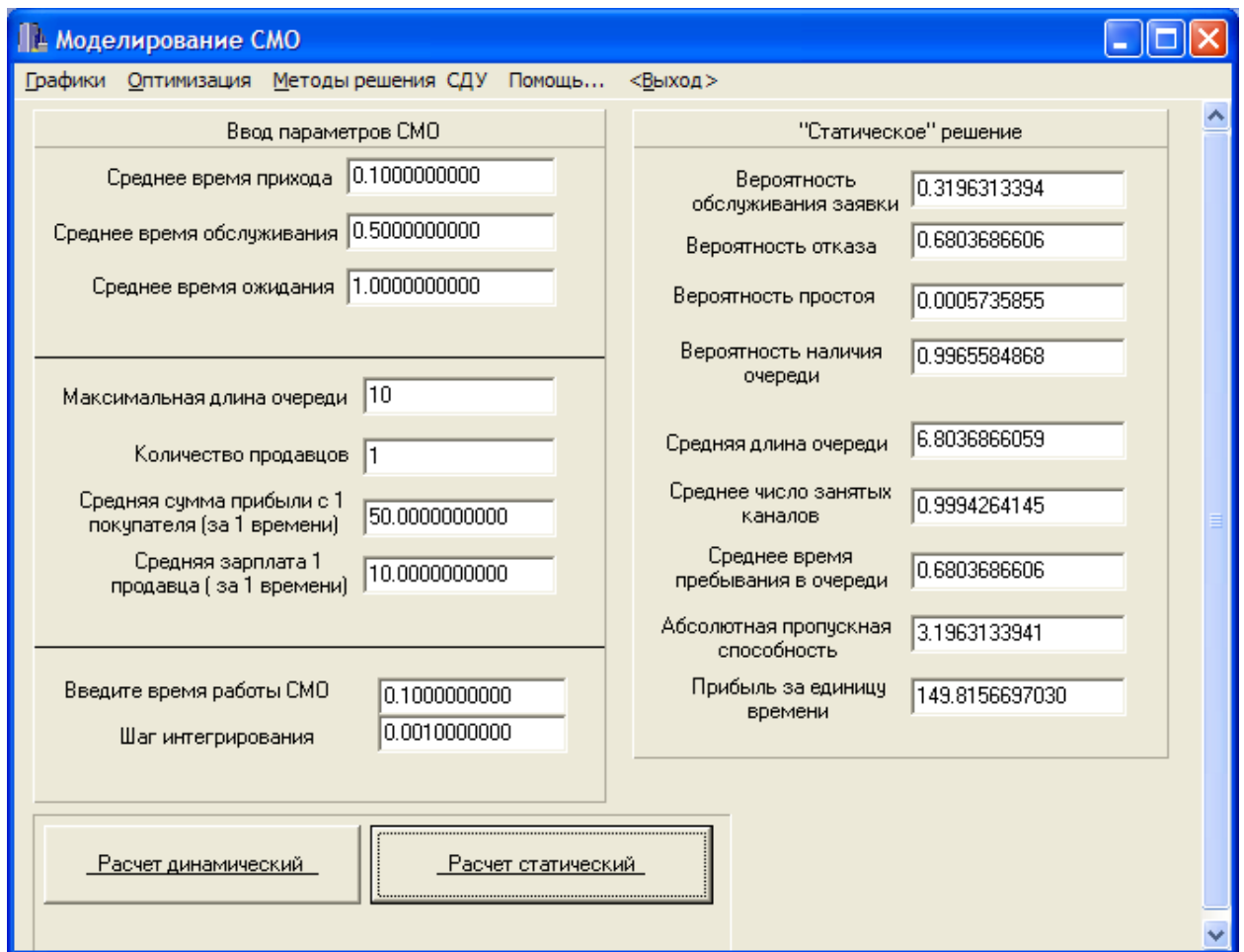


Рис. 3. Главная форма программы моделирования СМО

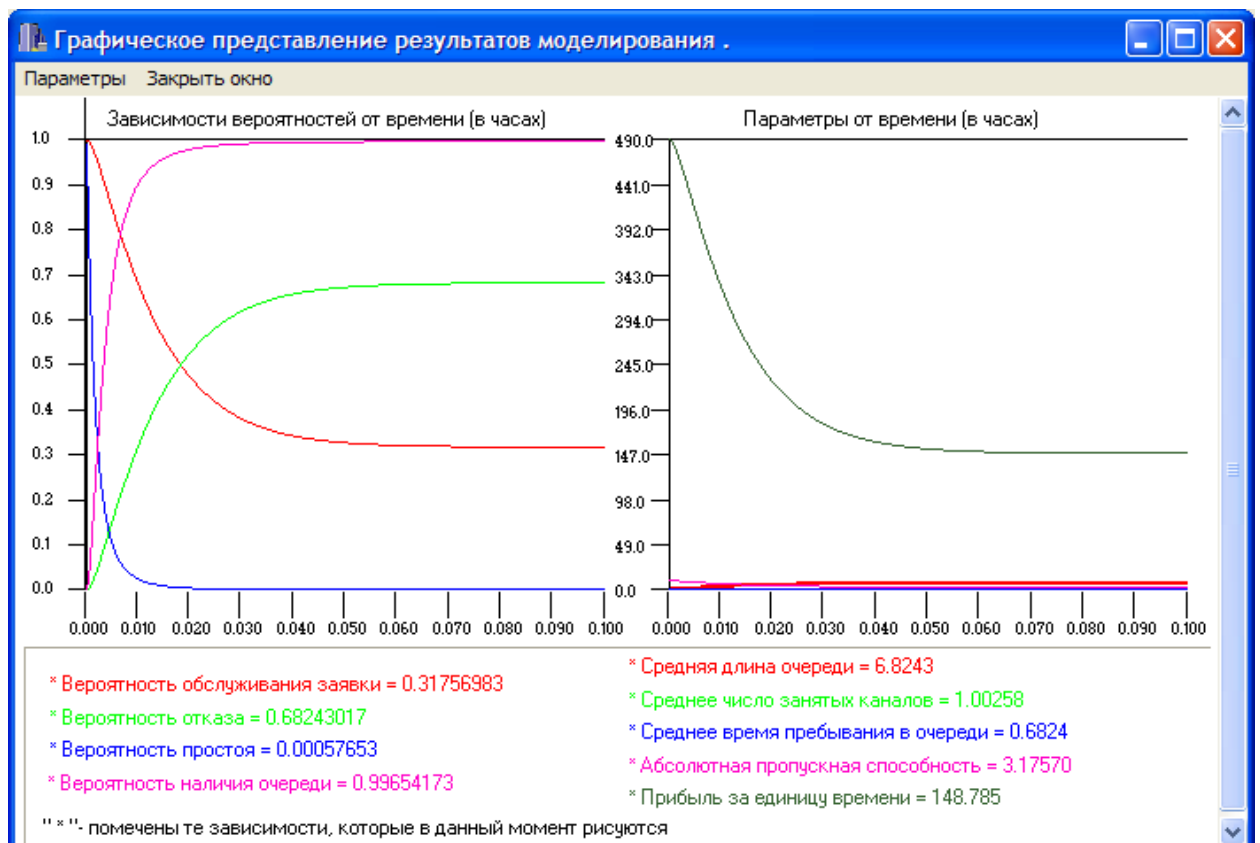


Рис. 4. Графическое представление динамических результатов моделирования

Вероятность обслуживания заявки	0.9638436374
Вероятность отказа	0.0361563626
Вероятность простоя	0.0065169134
Вероятность наличия очереди	0.1618008021
Средняя длина очереди	0.3615636261
Среднее число занятых каналов	4.8185084880
Среднее время пребывания в очереди	0.0361563626
Абсолютная пропускная способность	9.6384363739
Прибыль за единицу времени	411.9218186937

Рис. 5. Форма для задачи оптимизации

#### 4. Моделирование движения точки под действием центральной силы притяжения и силы возмущения

##### 4.1. Постановка задачи

Качественно исследовать форму плоской траектории движения точки, под действием центральной силы притяжения и постоянной силы возмущения, направленной вдоль оси  $X$  (например, силы, обусловленной «солнечным ветром» или силы притяжения удаленной планеты). Точка стартует с одной из четырех точек круговой, плоской орбиты радиуса  $r_0$  с начальной скоростью  $v_0$ , направленной под углом  $\varphi$  относительно касательной к окружности в точке старта (рис. 1).

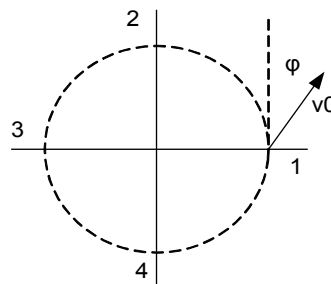


Рис. 1. Схема постановки задачи

##### 4.2. Математическая формулировка задачи

Основные уравнения движения точки единичной массы, на которую действуют центральная сила притяжения и постоянной сила возмущения, направленная вдоль оси  $X$ , записывается следующим образом

$$\begin{aligned} \frac{d^2 x}{dt^2} &= -\frac{kx}{r^3} + \alpha \left(\frac{k}{r_0^2}\right) \\ \frac{d^2 y}{dt^2} &= -\frac{ky}{r^3} \end{aligned} \quad (1)$$

начальные условия

$$t = 0, \quad r(0) = r_0, \quad |v(0)| = v_0 \equiv v_1 k_\phi \quad (2)$$

Здесь

$k$  - коэффициент пропорциональности центральной силы,

$\alpha$  - весовой коэффициент силы возмущения, которая выражается в долях центральной силы, действующей на расстоянии  $r_0$ ,

$r = \sqrt{x^2 + y^2}$  - расстояние точки от центра,

$v_1 = \sqrt{\frac{k}{r_0}}$  - вычисляется как скорость движения точки под действием только центральной

силы по круговой орбите радиуса  $r_0$  с периодом обращения  $T = 2\pi \sqrt{\frac{r_0^3}{k}}$

(получены из следующих соотношений:

$\frac{k}{r^2} = \frac{v^2}{r}$  - равенство силы притяжения и центростремительной силы при равномерном движении по окружности;  $vT = 2\pi r$  - за период точка совершает полный оборот по окружности радиуса  $r$ );

$k_\phi$  - коэффициент формы орбиты невозмущенного движения точки ( $\alpha = 0$ ) под действием центральной силы, позволяющий корректировать модуль скорости

Причем:

- при  $0 < k_\phi < \sqrt{2}$  - эллиптическая орбита невозмущенного движения,
- в частности, при  $k_\phi = 1$  - круговая орбита,
- при  $k_\phi = \sqrt{2}$  - параболическая орбита
- при  $k_\phi > \sqrt{2}$  - гиперболическая орбита.

Систему (1) при начальных условиях (2) целесообразно численно интегрировать методом Рунге-Кутты 4-го порядка, поскольку он может обеспечить хорошую точность для больших интервалов интегрирования.

#### 4.3. Программная реализация математической модели

Для исследования формы траектории точки разработана программа, основной экран которой представлен на рис. 2. Программа позволяет исследовать форму траектории в зависимости от следующих параметров:

1. величины возмущения (коэффициент  $\alpha$ ) (сравните рис. 2 и 3);
2. величины центральной силы притяжения (коэффициент  $k$ ) (сравните рис. 3 и 4);

3. начального радиуса стартовой окружности  $r_0$  и соответствующей ему начальной скорости (сравните рис. 4 и 5);
4. угла наклона начальной скорости к стартовой круговой орбите (сравните рис. 5 и 6);
5. коэффициента формы орбиты  $k_\phi$ , корректирующего величину модуля стартовой скорости (сравните рис. 2, 7 и 10);
6. положения стартовой точки на стартовой круговой орбите (одна из четырех) (сравните рис. 7 и 8);
7. шага интегрирования (сравните рис. 2 и 10);

Дополнительно можно регулировать интервал действия возмущающей силы и масштаб выводимого изображения траектории. Интегрирование ведется от начальных условий до конца заданного интервала интегрирования (кнопка «построить траекторию»). Далее интегрирование можно продолжить от последней конечной точки до конца нового интервала интегрирования (кнопка «продолжить»). При этом возможно изменения всех параметров. Пересчет составляющих текущей начальной скорости при текущем изменении ее наклона к траектории осуществляется с помощью известной матрицы вращения

$$\begin{aligned} v_x &= v_{x0} \cos(\varphi) - v_{y0} \sin(\varphi) \\ v_y &= v_{x0} \sin(\varphi) + v_{y0} \cos(\varphi) \end{aligned} \quad (3)$$

На рис. 11 показаны две корректировки траектории путем изменения угла между текущей скоростью и касательной к траектории в текущей начальной точке ( $\varphi = -30^\circ$ ).

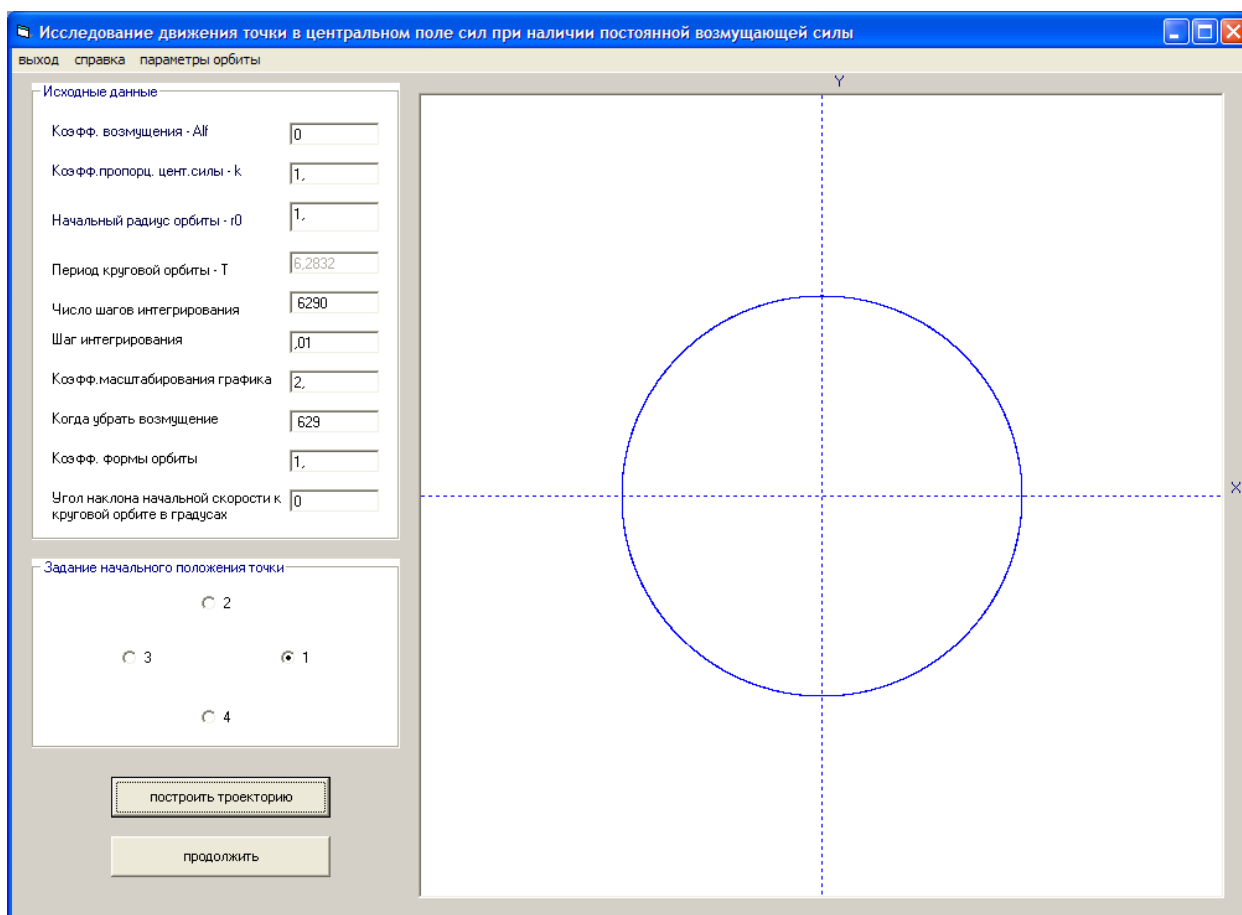


Рис. 2. Траектория точки при отсутствии возмущения  $\alpha = 0$  и при  $k_\phi = 1$ .



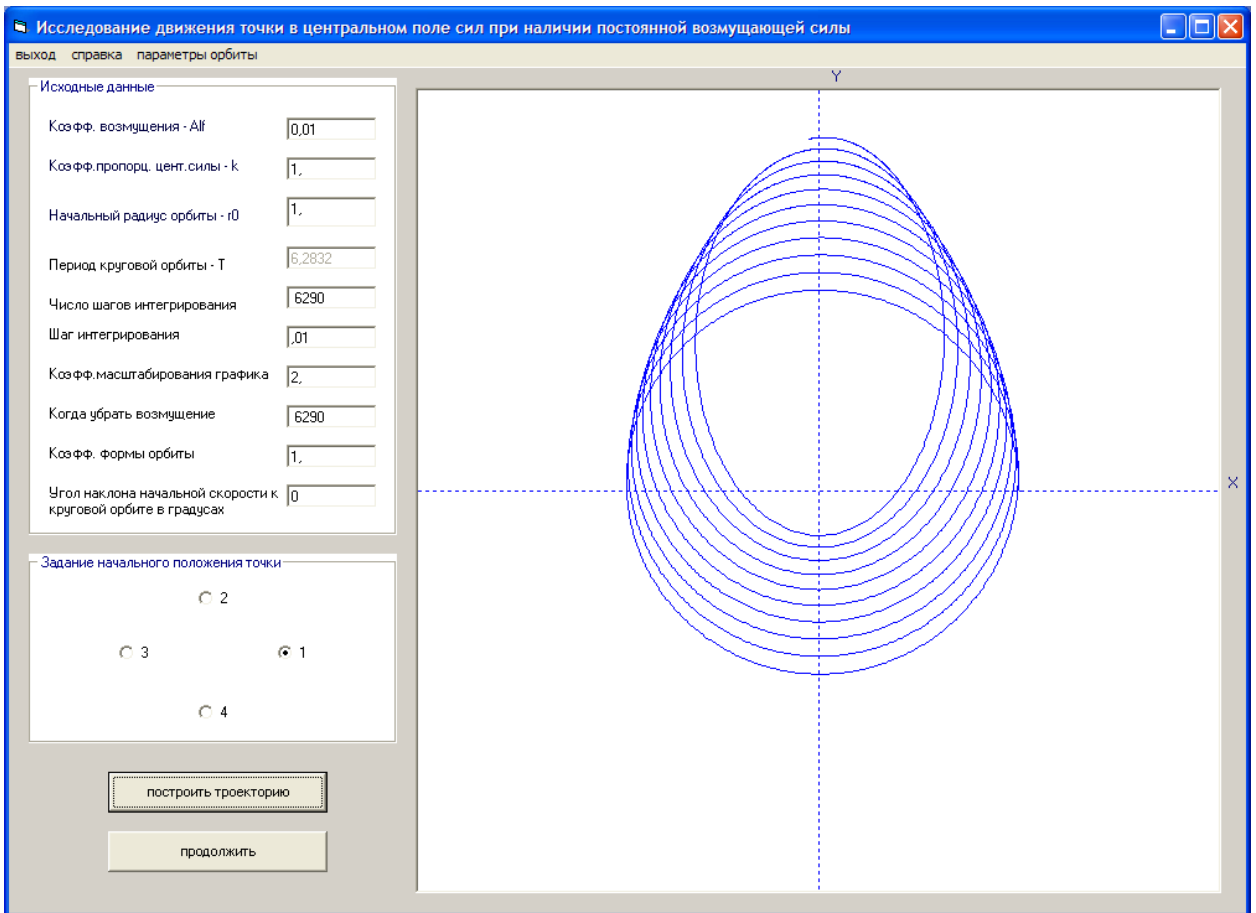


Рис. 3. Траектория точки при наличии возмущения  $\alpha = 0.01$  и при  $k_\phi = 1$ .

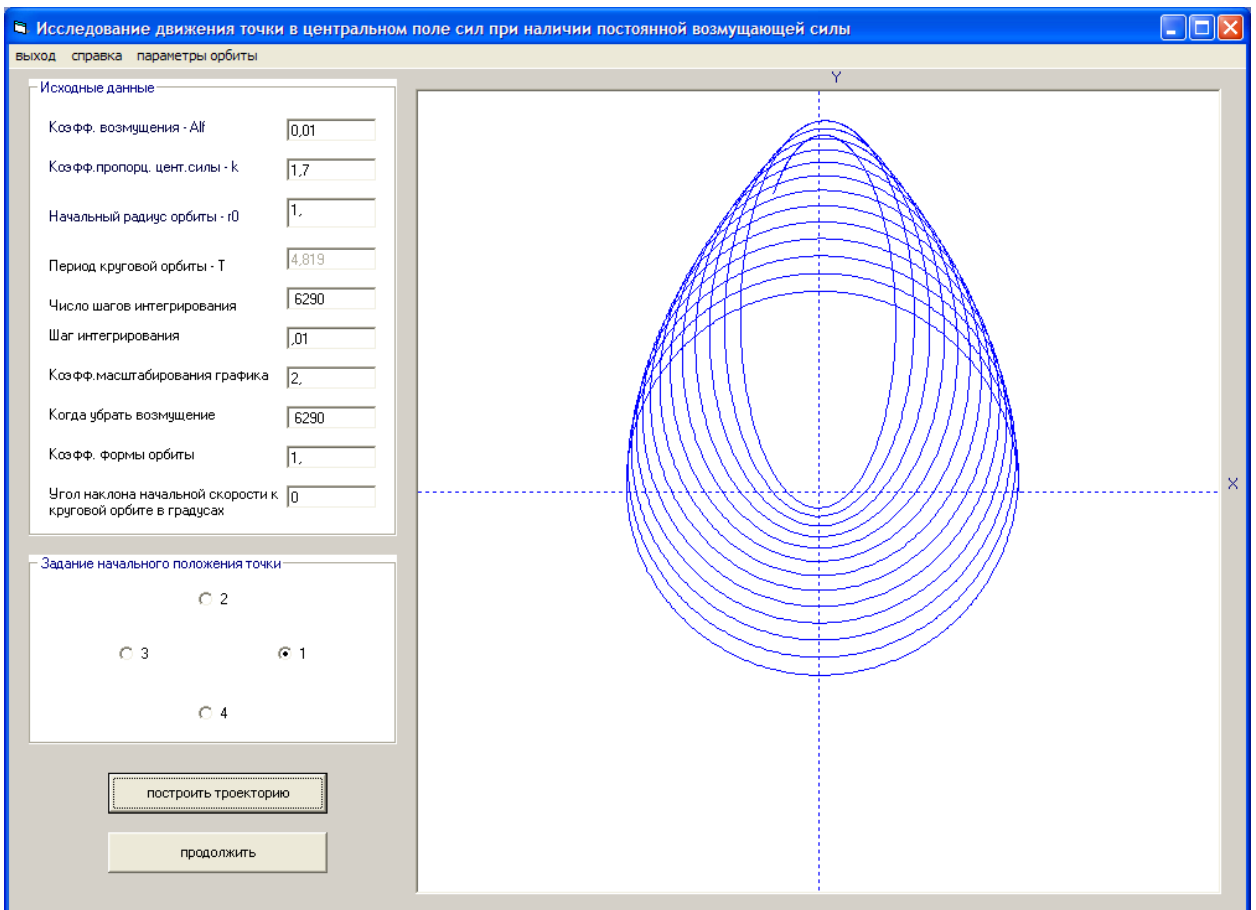


Рис. 4. Траектория точки при наличии возмущения  $\alpha = 0.01$ , при  $k_\phi = 1$  и для возросшей силы притяжения - коэффициент  $k = 1.7$  (сравните рис. 3, где  $k = 1$ )

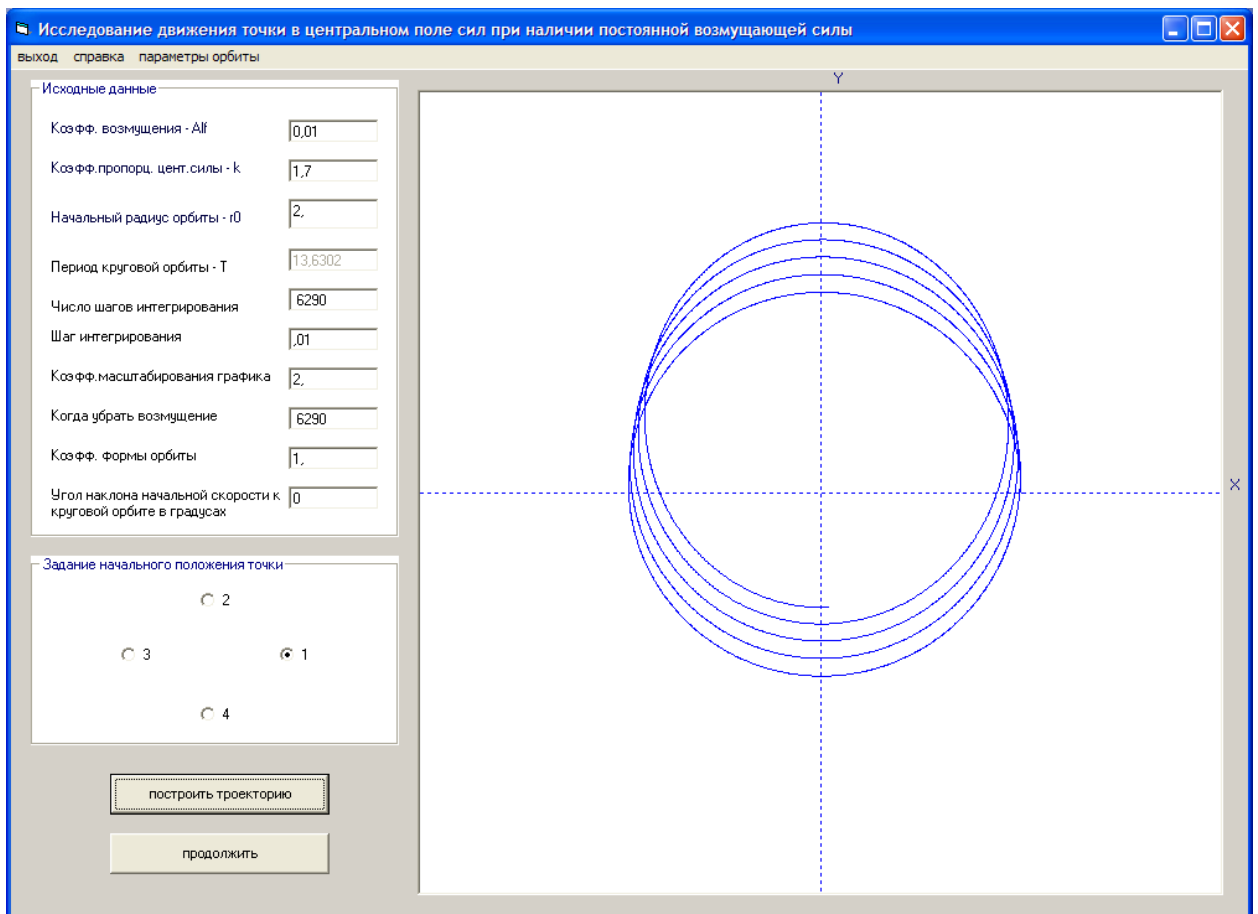


Рис. 5. Траектория точки при наличии возмущения  $\alpha = 0.01$ , при  $k_\phi = 1$ ,  $k = 1.7$  и увеличения начального радиуса стартовой орбиты -  $r_0 = 2$  (сравните рис. 4, где  $r_0 = 1$ )

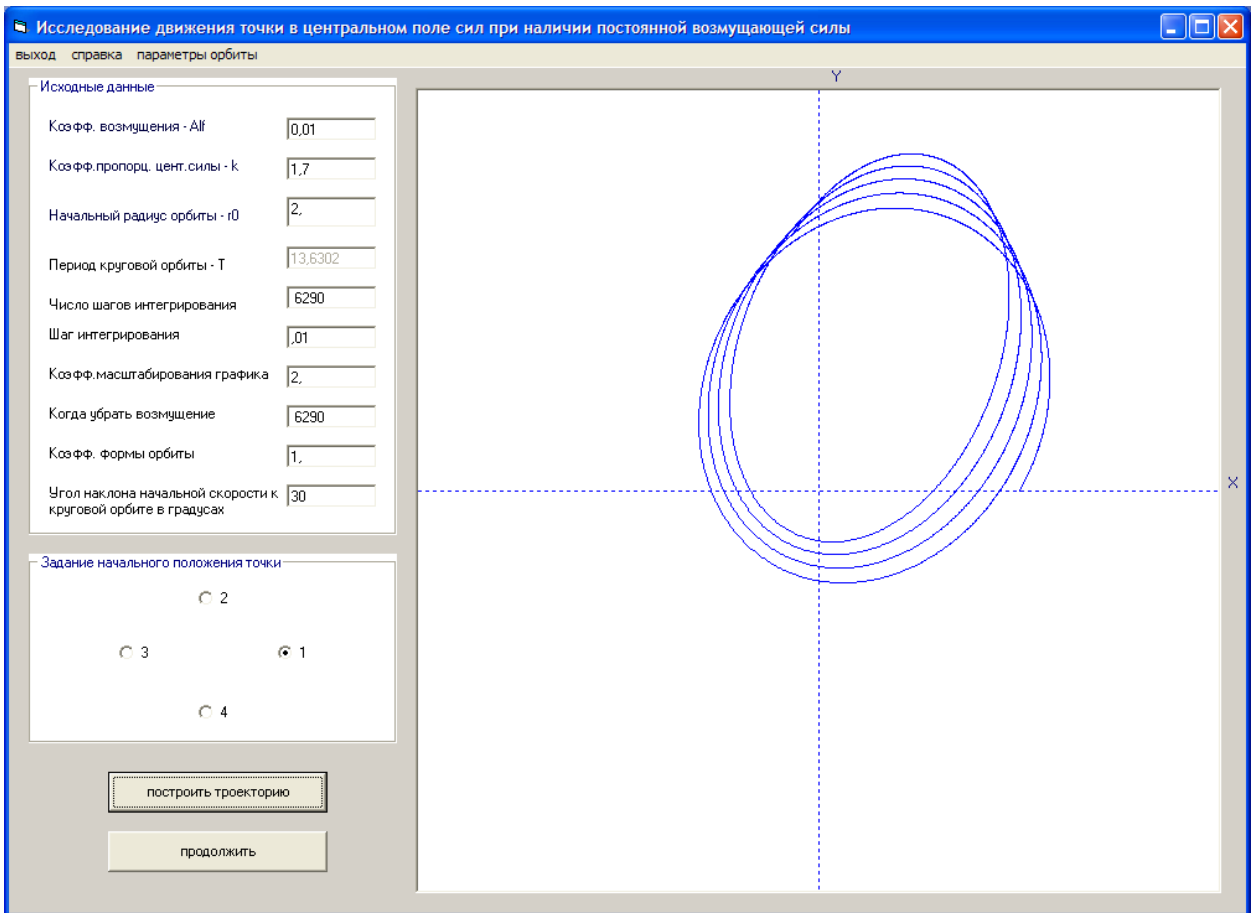


Рис. 6. Изменение расположения орбиты при изменении угла наклона начальной скорости к стартовой круговой орбите (сравните с рис. 5)

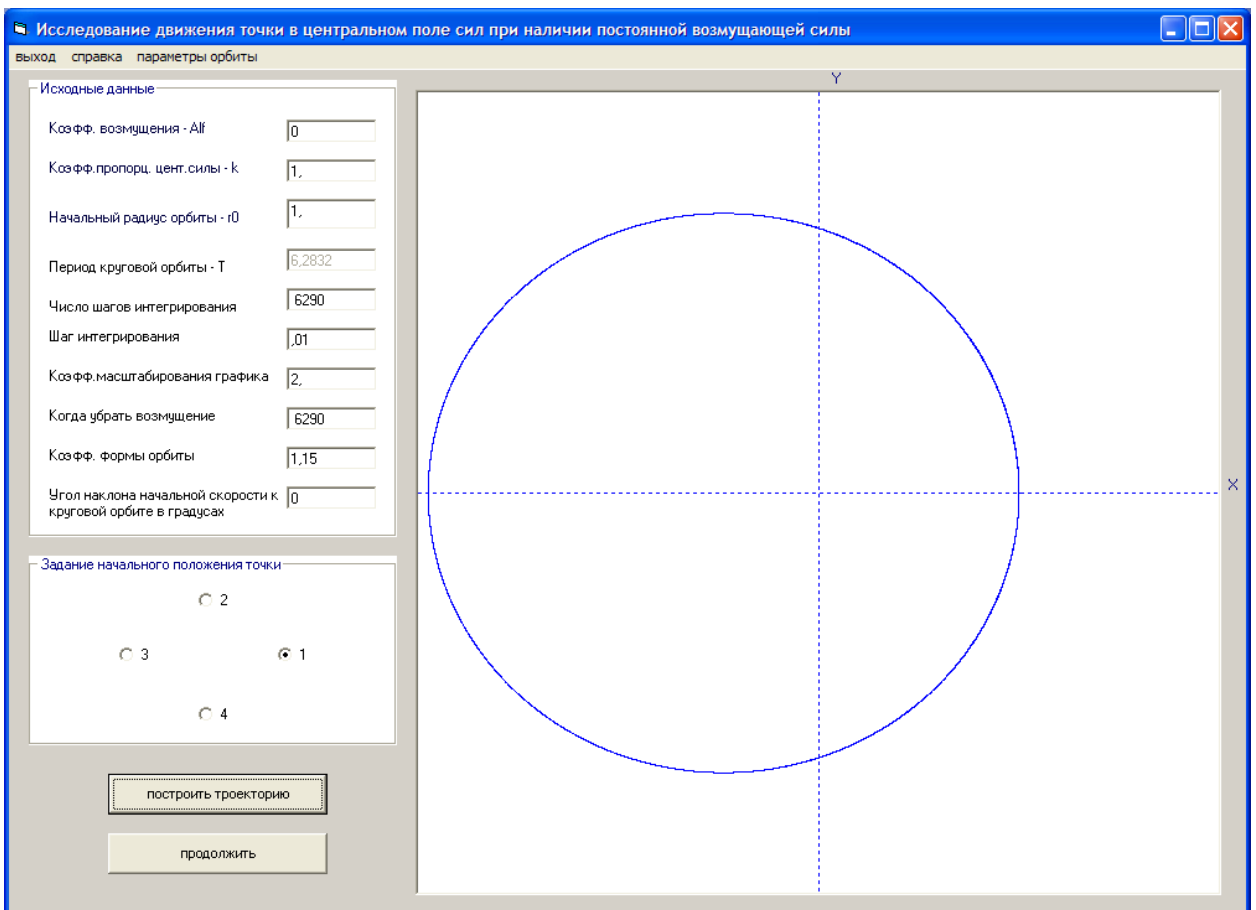


Рис. 7. Изменение формы орбиты при изменении коэффициента формы -  $1 < k_\phi = 1.15 < \sqrt{2}$  (сравните с рис. 2, 10)

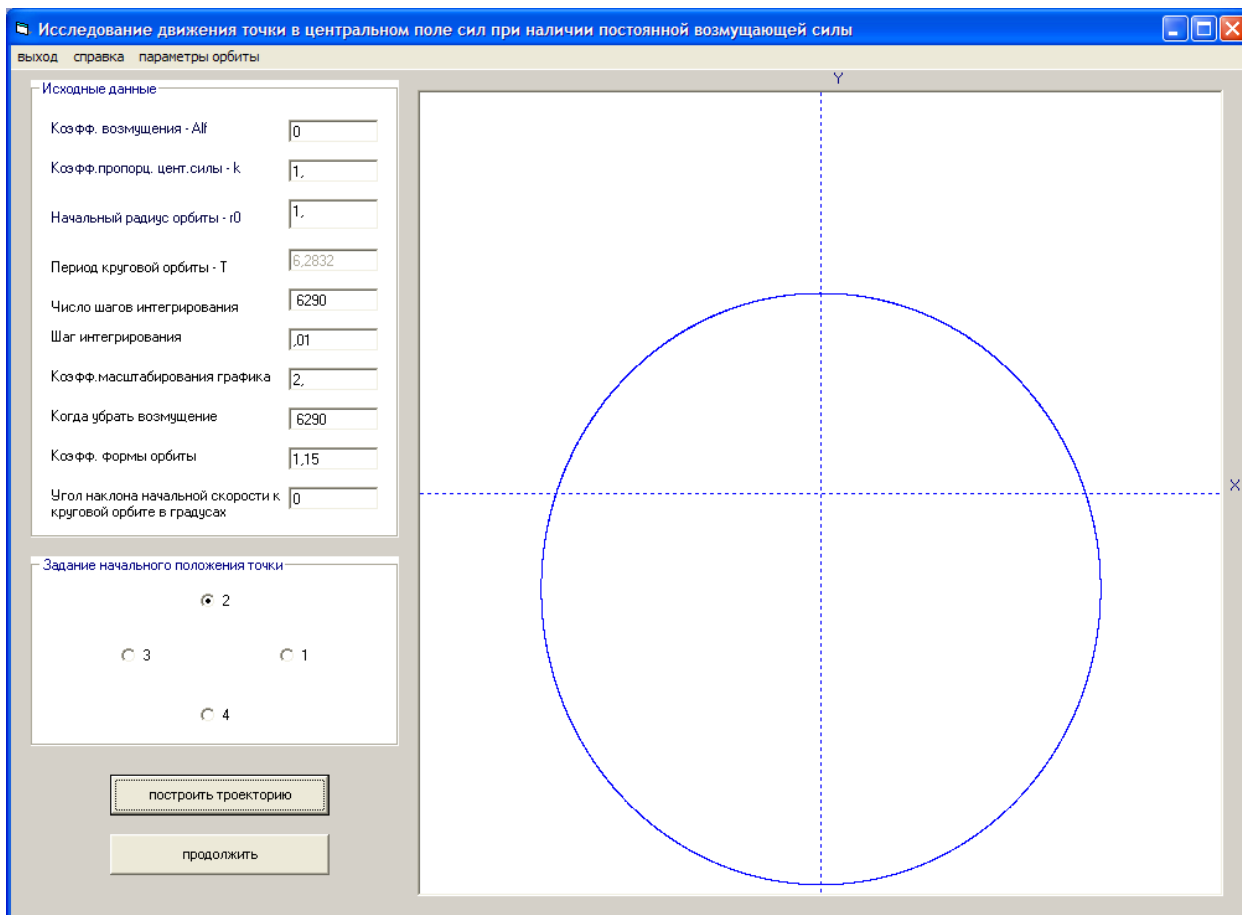


Рис. 8. . Изменение местоположения орбиты при смене положения стартовой точки (сравните рис. 7)

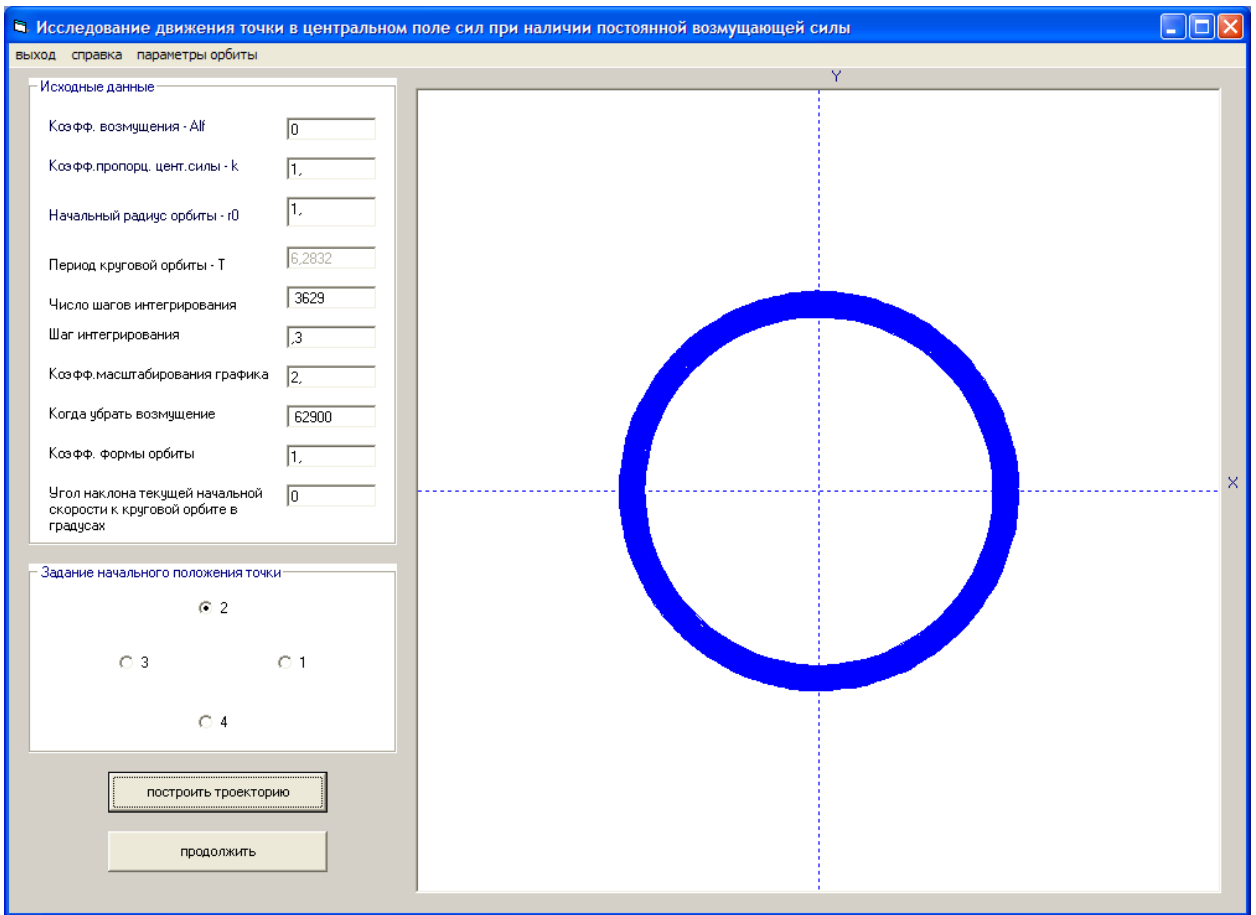


Рис. 9. «Расфокусирование» траектории орбиты при увеличении шага интегрирования ( $h = 0,3$ ), обусловленное возрастанием ошибок метода численного интегрирования (сравните с рис. 2)

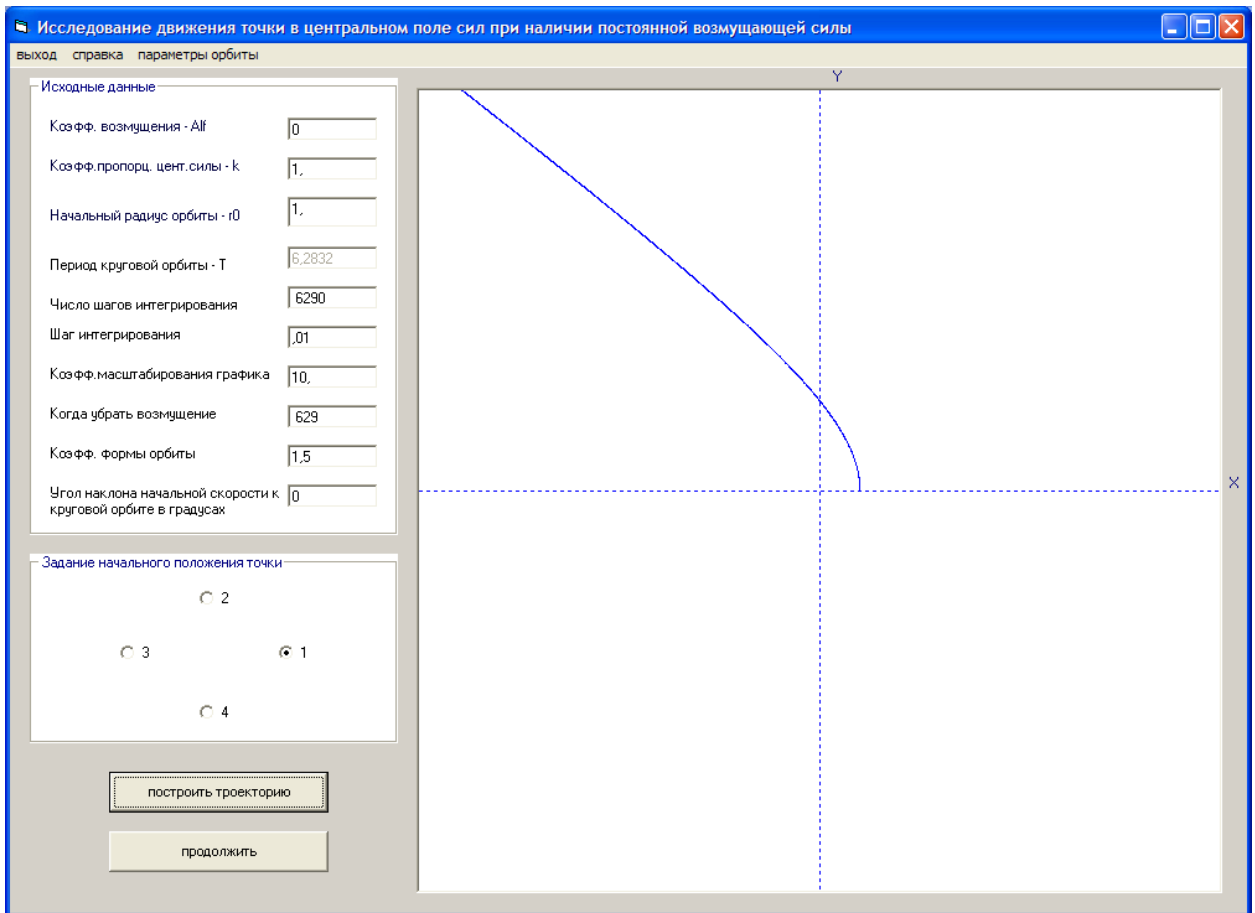


Рис. 10. Изменение формы орбиты при изменении значения коэффициента формы -  $k_\phi = 1.5 > \sqrt{2}$  (сравните с рис. 2, 7)

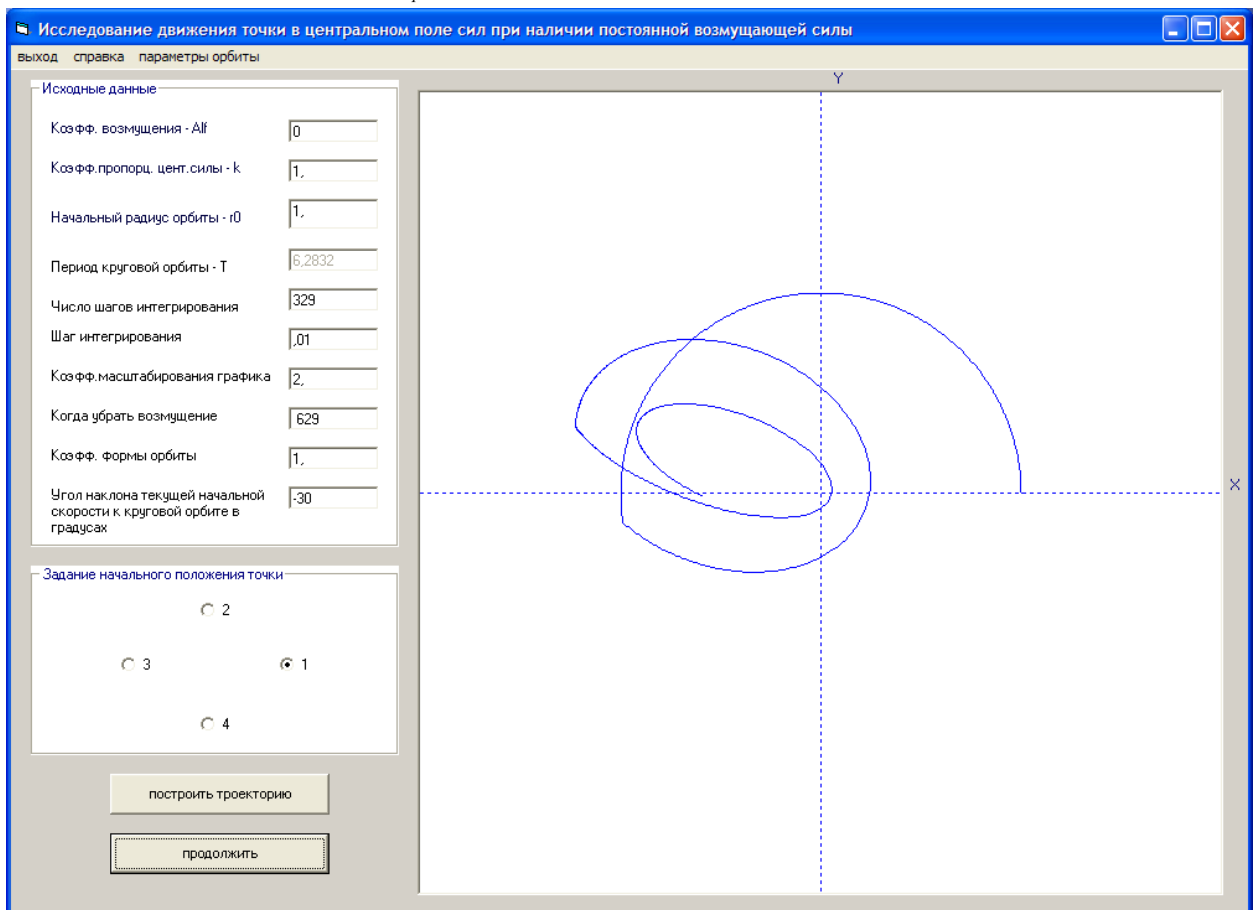


Рис. 12. Произведены две корректировки траектории точки путем изменения угла между скоростью и касательной к траектории в текущей стартовой точке ( $\varphi = -30^0$ )

## 5. Моделирование виртуального мира

### 5.1. Постановка задачи

Моделируется виртуальный мир, состоящий из растений, травоядных и хищных животных. При постановке задачи использованы идеи, изложенные в работе [4].

В этом мире травоядные поедают растения, а хищники - травоядных. Животные имеют начальную энергию, которая расходуется при их перемещении, поворотах направо или налево. Съедание объекта увеличивает энергию животного. Животное умирает, когда его энергия обнуляется (смерть от истощения) или возраст достигает предельного значения (смерть от старости, от износа). Если энергия животного превышает величину, равную произведению начальной энергии на энергетический коэффициент размножения, то оно рождает новое животное, делясь с ним своей текущей энергией поровну. Общее количество животных поддерживается в интервале, левая граница которого задается произведением начального числа животных на предельный коэффициент добавления, а правая - предельным максимальным числом животных. Пополнение животных осуществляется путем искусственного добавления нового животного (когда после удаления очередного съеденного или умершего животного их число становится меньше нижнего предела, указанного выше) или благодаря описанной выше процедуре размножения. Число растений при поедании пополняется в соответствие с формулой - произведение числа съеденных растений на коэффициент воспроизводства растений.

Первоначально тип генерируемых животных задается случайным образом либо устанавливается принудительно. При искусственном добавлении - только случайным образом. Также случайным образом задается местоположение всех объектов на плоской квадратной решетке (одно животное в ячейке), представляющей пространство, в котором протекает жизнь. Решетка склеена своими противоположными сторонами. Т.е. животное, пересекая край решетки, переходит на ее противоположную сторону и продолжает движение в том же направлении.

Каждое животное снабжено сенсорной системой, позволяющей ему оценивать состояние переднего плана мира (при генерации животного направление его движения: вверх, вниз, налево, направо - задается случайным образом): наличие объекта впереди, налево, направо и рядом. Схема переднего плана, анализируемого сенсорами, представлена на рис. 1

ф	ф	ф	ф	ф
л	р	р	р	п
л	р	Ж	р	п

Рис. 1. Схема переднего плана, просматриваемого животным

где ф - передний фронт; л - слева; р - рядом; п - справа; Ж - положение животного, которое на данной схеме смотрит вверх.

На основании показаний сенсоров животное может выбрать одно из возможных действий: повернуться налево, повернуться направо, передвинуться на одну клетку вперед (если она не занята другим животным) или съесть соответствующий объект,

расположенный рядом.

Интеллект животных моделируется трехслойной нейронной сетью (первый слой распределительный). Уровень интеллекта определяется значениями компонент матриц весовых коэффициентов нейронной сети. По умолчанию они задаются случайным образом из интервала  $(-0,5 \div 0,5)$  отдельно для класса травоядных и класса хищников. Кроме того, их можно загрузить из файлов, содержащих веса нейронных сетей для каждой группы животных обученных в дополнительной программе.

## 5.2. Моделирование поведения животного

Нейронная сеть, используемая при моделировании поведения животного, представлена на рис. 2. На ее вход поступают сигналы от сенсорной системы  $x_k, k = 1 \div 12$  ( $x_k = 0, 1, \dots$ ), а на выходе формируются соответствующие числовые значения для четырех возможных реакций животного  $y_j, j = 1 \div 4$  ( $y_j = 0 \div 1$ ).

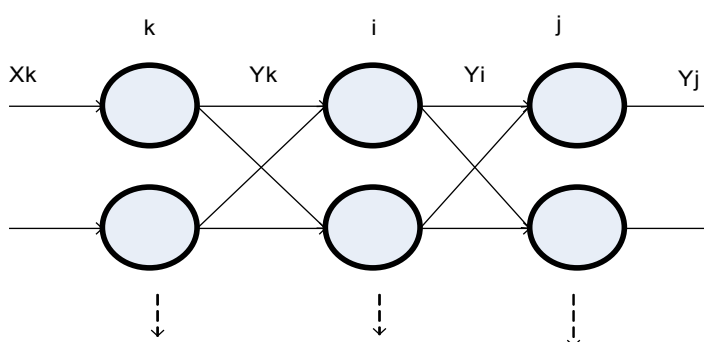


Рис. 2. Используемая нейронная сеть

Здесь  $k$ -ый слой нейронов – входной или распределительный ( $y_k = x_k$ );  $i$ -ый слой нейронов – промежуточный или скрытый, его размерность может варьироваться от трех и выше (в программе моделирования принято четыре);  $j$ -ый слой нейронов – выходной.

Назначение входов и выходов сети и их допустимые числовые значения приведены в таблицах 1 и 2.

Таблица 1 Назначение входов сети и их возможные значения

к	Наименование ситуации	Значения сигнала (нет, да-1,2,..)
1	Травоядное впереди	0, 1, 2,...
2	Хищник впереди	0, 1, 2,...
3	Растение впереди	0, 1, 2,...
4	Травоядное слева	0, 1, 2,...
5	Хищник слева	0, 1, 2,...
6	Растение слева	0, 1, 2,...
7	Травоядное справа	0, 1, 2,...
8	Хищник справа	0, 1, 2,...
9	Растение справа	0, 1, 2,...
10	Травоядное рядом	0, 1, 2,...
11	Хищник рядом	0, 1, 2,...
12	Растение рядом	0, 1, 2,...

Таблица 2. Назначение выходов сети и их возможные значения



j	Наименование действия	Значение сигнала
1	Повернуть налево	0 ÷ 1
2	Повернуть направо	0 ÷ 1
3	Переместиться вперед	0 ÷ 1
4	Съесть	0 ÷ 1

Примято, что после анализа выходных сигналов животное выполняет действие с максимальным числовым значением реакции.

Выходные сигналы нейронов промежуточного и выходного слоев формируются следующим образом:

$$y_i = F(s_i), \quad s_i = \sum_k w_{ki} y_k - T_i$$

$$y_j = F(s_j), \quad s_j = \sum_i w_{ij} y_i - T_j$$

Здесь

$F(s)$  - функция активации нейрона, в качестве которой используется сигмоидная функция

$$F(s) = \frac{1}{1 + e^{-s}} \in [0,1] \text{ (рис. 3);}$$

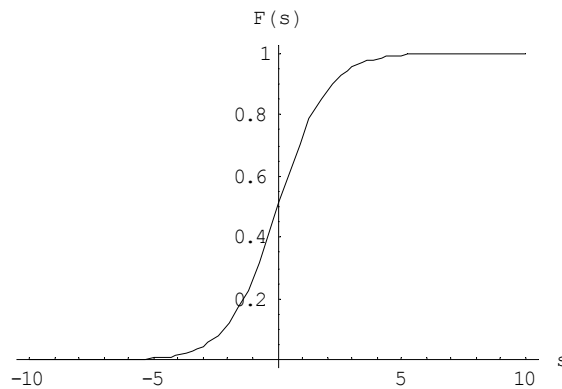


Рис.3. Функция активации - сигмоидная

$w_{ki}, w_{ij}$  - компоненты матриц весовых коэффициентов нейронов соответствующих слоев;  
 $T_i, T_j$  - компоненты векторов смещения (пороги) нейронов соответствующих слоев.

Для правильного выбора значений весов и порогов нейронная сеть должна пройти обучение на обучающей выборке, состоящей из совокупностей эталонных входных ( $x_k$ ) и соответствующих им выходных сигналов ( $t_j$ ). При обучении используется алгоритм обратного распространения ошибки [5], который минимизирует среднеквадратичную ошибку на выходах нейронной сети  $E(w_{ki}, w_{ij}, T_i, T_j) = \frac{1}{2} \sum_j (y_j - t_j)^2 \equiv \frac{1}{2} \sum_j \gamma_j^2$  методом градиентного спуска в пространстве весовых коэффициентов и пороговых значений нейронов ( $E$  - функционал, спуск в сторону противоположную градиенту от  $E$ ). Согласно методу параметры оптимизации на  $n$ -ой итерации вычисляются по формулам (первоначально их значения задаются случайным образом из интервала  $(-0,5 \div 0,5)$ )

$$w_{ij}^{(n)} = w_{ij}^{(n-1)} - \alpha \left( \frac{\partial E}{\partial w_{ij}} \right)^{(n-1)}$$

$$T_j^{(n)} = T_j^{(n-1)} - \alpha \left( \frac{\partial E}{\partial T_j} \right)^{(n-1)} \quad (1)$$

где

$\alpha$  - параметр спуска (коэффициент обучения), выбирается произвольно из интервала  $(0 \div$

1) и может изменяться в процессе спуска;  $i$  и  $j$  номера предыдущего и последующего слоев.

Производные в формулах (1) расшифровываются следующим образом:

- для  $j$ -ого **выходного** слоя нейронной сети

$$\frac{\partial E}{\partial w_{ij}} = \gamma_j y_j (1 - y_j) y_i, \quad \frac{\partial E}{\partial T_j} = -\gamma_j y_j (1 - y_j) \quad (2)$$

Здесь при выводе использованы следующие соотношения:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial s_j} \frac{\partial s_j}{\partial w_{ij}}, \quad \frac{\partial E}{\partial T_j} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial s_j} \frac{\partial s_j}{\partial T_j}$$

$$\frac{\partial E}{\partial y_j} = \gamma_j, \quad \frac{\partial y_j}{\partial s_j} = \frac{\partial F(s_j)}{\partial s_j} = \frac{e^{-s_j}}{(1 + e^{-s_j})^2} = \frac{1}{(1 + e^{-s_j})} \left(1 - \frac{1}{(1 + e^{-s_j})}\right) = y_j (1 - y_j),$$

$$\frac{\partial s_j}{\partial w_{ij}} = y_i, \quad \frac{\partial s_j}{\partial T_j} = -1$$

- для  $i$ -ого **промежуточного** слоя нейронной сети (суммирование ведется по всем  $j$ -ым нейронам выходного слоя, на которые влияет  $i$ -ый нейрон промежуточного слоя)

$$\frac{\partial E}{\partial w_{ki}} = \gamma_i y_i (1 - y_i) y_k, \quad \frac{\partial E}{\partial T_i} = -\gamma_i y_i (1 - y_i), \quad (3)$$

$$\gamma_i \equiv \frac{\partial E}{\partial y_i} = \sum_j \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial s_j} \frac{\partial s_j}{\partial y_i} = \sum_j \gamma_j y_j (1 - y_j) w_{ij}$$

Здесь при выводе использованы следующие соотношения:

$$\frac{\partial E}{\partial w_{ki}} = \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial s_i} \frac{\partial s_i}{\partial w_{ki}} = \left( \sum_j \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial s_j} \frac{\partial s_j}{\partial y_i} \right) \frac{\partial y_i}{\partial s_i} \frac{\partial s_i}{\partial w_{ki}} \equiv \gamma_i \frac{\partial y_i}{\partial s_i} \frac{\partial s_i}{\partial w_{ki}}$$

$$\frac{\partial E}{\partial T_i} = \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial s_i} \frac{\partial s_i}{\partial T_i} = \left( \sum_j \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial s_j} \frac{\partial s_j}{\partial y_i} \right) \frac{\partial y_i}{\partial s_i} \frac{\partial s_i}{\partial T_i} \equiv \gamma_i \frac{\partial y_i}{\partial s_i} \frac{\partial s_i}{\partial T_i}$$

$$\frac{\partial y_i}{\partial s_i} = \frac{\partial F(s_i)}{\partial s_i} = y_i (1 - y_i), \quad \frac{\partial s_i}{\partial w_{ki}} = y_k, \quad \frac{\partial s_j}{\partial y_i} = w_{ij}, \quad \frac{\partial s_i}{\partial T_i} = -1$$

Приведем последовательность операций, выполняемых при реализации метода обратного распространения ошибки:

1. Задаются коэффициент обучения  $\alpha$  ( $0 \div 1$ ), предельное число итераций спуска и желательная среднеквадратичная ошибка на выходе нейронной сети  $E_m$ .

2. Иницируются весовые коэффициенты и пороговые значения нейронной сети случайными числами с интервалом изменения  $(-0,5 \div 0,5)$ . Псевдослучайные равномерно распределенные на отрезке  $(0 \div 1)$  числа  $\theta$ , генерируемые стандартным датчиком, приводятся к требуемому интервалу по формуле  $\theta \equiv \theta - 0,5$ . Весовые коэффициенты и пороговые значения нейронной сети целесообразно хранить в виде двух расширенных матриц (матриц весов) отдельно для промежуточного и выходного слоев (рис. 4).

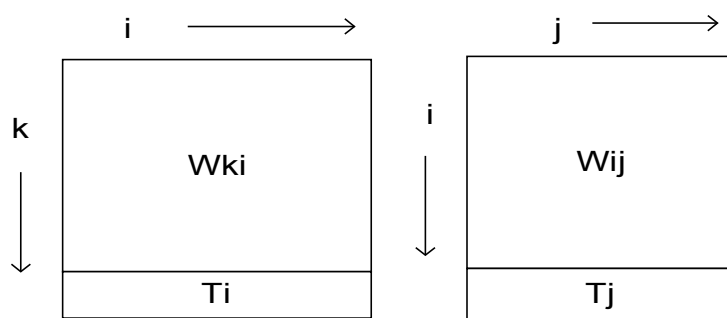


Рис. 4, Структура используемых матриц

3. Задается обучающая выборка в виде набора векторов, 16 компонент которых являются входными и соответствующими им выходными эталонными значениями нейронной сети (см. таблицы 1 и 2).

4. Начало цикла по итерациям спуска

4.1. Начало цикла по векторам выборки

4.1.1. Проводятся вычисления прямого распространения входного сигнала

$$y_i = F\left(\sum_k w_{ki} y_k - T_i\right)$$

$$y_j = F\left(\sum_i w_{ij} y_i - T_j\right)$$

4.1.2. Проводятся вычисления обратного распространения ошибок

$$\gamma_j = y_j - t_j, \quad \gamma_i = \sum_j \gamma_j y_j (1 - y_j) w_{ij}$$

4.1.3. Проводится пересчет весовых коэффициентов и пороговых значений по формулам (1).

4.1. Конец цикла по выборке

4.2. Вычисляется средняя среднеквадратичная ошибка для всей выборки ( $L$  - размерность выборки)

$$E = \frac{1}{2L} \sum_{l=1}^L \left( \sum_j (y_j^l - t_j^l)^2 \right)$$

4.3. Если  $E \leq E_m$ , то выходим из цикла по итерациям

4. Конец цикла по итерациям

5. Если желаемая точность не достигнута, то можно повторить цикл по итерациям.

**Замечания:**

1. В случае плохой сходимости метода можно изменить число нейронов в промежуточном слое сети.

2. При выборе значения  $\alpha$  следует учитывать, что при малых значениях  $\alpha$  увеличивается время обучения и возможно зависание в локальном минимуме функционала, а при больших значениях  $\alpha$  - возможно проскакивание глобального минимума и метод может расходиться.

3. Для практического применения неважно, какой минимум достигнут, главное, чтобы выполнялось условие  $E \leq E_m$ .

5.3. Программа обучения животных

Главный экран программы, реализующей обучение животных (настройка весов и пороговых значений нейронной сети животного), приведен на рис. 5.

Программа позволяет вводить обучающую выборку вручную (для этого используются поля ввода и редактирования) и сохранять ее в таблице (кнопка «Сохранить в таблице»),

либо загрузить выборку из файла (кнопка «Загрузить из файла»). Введенную или отредактированную выборку (выбор строки в таблице переносит ее в поля ввода и редактирования, а кнопка «Вернуть в таблицу» возвращает ее в таблицу на первоначальное место) можно сохранить в файле (кнопка «Сохранить в файле») для последующего использования.

Для введенных или выбранных из таблицы входных сигналов и используемых в данный момент матриц весов можно рассчитать выходные значения нейронной сети (кнопка «Рассчитать»).

**Входные значения сети**

1. Травоядное впереди - 0,1...	0
2. Хищник впереди - 0,1...	0
3. Растение впереди - 0,1...	0
4. Травоядное слева - 0,1...	0
5. Хищник слева - 0,1...	1
6. Растение слева - 0,1...	0
7. Травоядное справа - 0,1...	0
8. Хищник справа - 0,1...	1
9. Растение справа - 0,1...	0
10. Травоядное рядом - 0,1...	0
11. Хищник рядом - 0,1...	0
12. Растение рядом - 0,1...	0

**Выходные эталонные и реальные значения сети**

1. Повернуть налево - 0,1	0	,010293	-,010293
2. Повернуть направо - 0,1	0	,023665	-,023665
3. Переместиться вперед - 0,1	1	,96509	,03491
4. Съесть-0,1	0	,029686	-,029686

Среднеквадратичная ошибка: ,001383

Число нейронов в промежуточном слое: 4

Измнить

Сохранить в таблице | Вернуть в таблицу | **Рассчитать** | Веса слов

**Массив входных и эталонных выходных значений для уточнения весовых коэффициентов нейронной сети**

N	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	
1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
2	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1
3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
4	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1
5	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0
6	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	1
7	0	1	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0
8	0	1	0	0	1	0	1	1	1	0	0	0	0	1	0	0	0

Сохранить в файле | Загрузить из файла

Очистить таблицу

Число итераций <32000: 10000

Коэффициент обучения: ,2

Уточнить веса

Рис. 5. Главный экран программы обучения нейронной сети

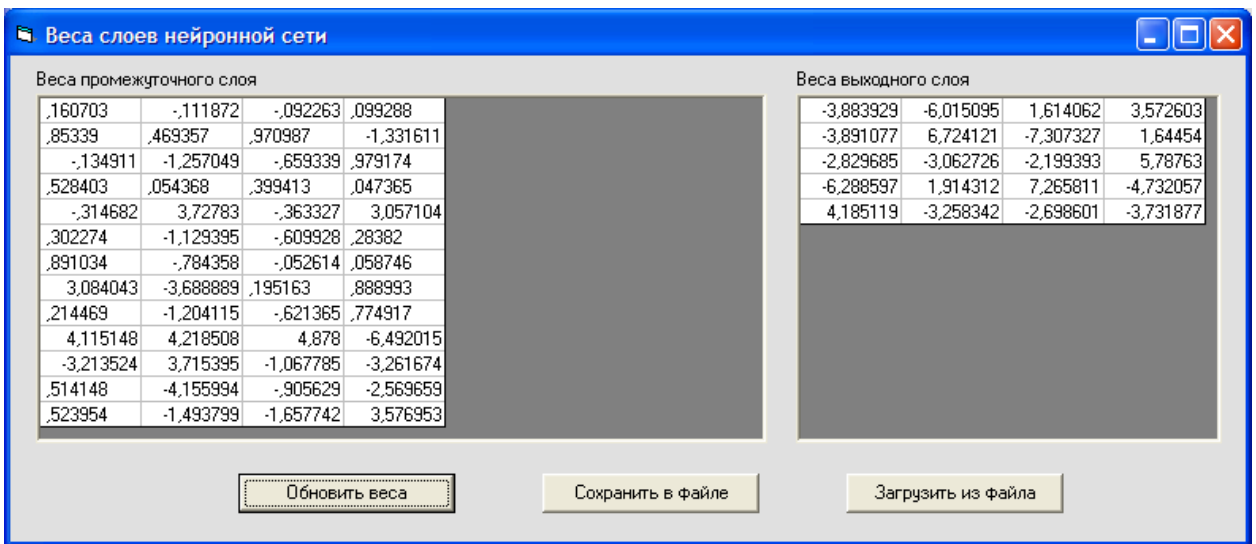


Рис. 6. Экран для просмотра весов нейронной сети

Веса и пороговые значения, первоначально заданные случайным образом, можно уточнить с помощью обучающей выборки (кнопка «Уточнить веса»). Исходные или уточненные веса и пороговые значения можно посмотреть (кнопка «Веса слоев») в отдельном окне (рис. 6, промежуточный слой состоит из 4-х нейронов). В этом окне матрицы слоев можно вновь случайным образом обновить (кнопка «Обновить веса»), а также сохранить (кнопка «Сохранить в файле») или загрузить их из файла (кнопка «Загрузить из файла»).

#### 5.4. Программа моделирования виртуального мира

Основное окно программы моделирования виртуального мира приведено на рис. 7.

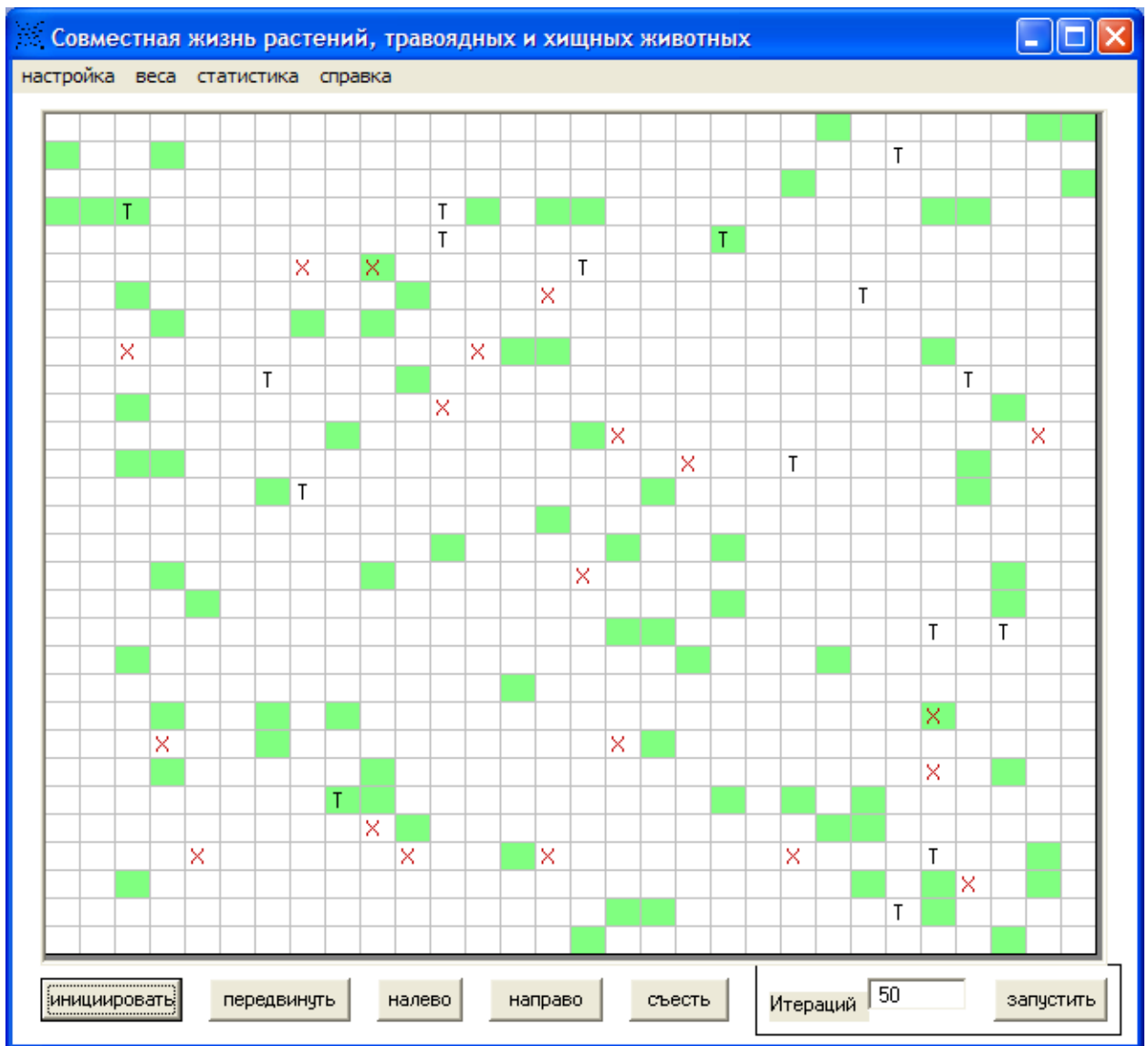


Рис. 7. Экран программы моделирования виртуального мира

Здесь виртуальное пространство описывается таблицей (30x30) с замкнутыми друг на друга противоположными краями, а объекты в ячейках таблицы обозначаются следующим образом: т или х – травоядное или хищное животное в данной ячейки таблицы, а закрашенная ячейка указывает на наличие в ней растения. О каждом животном можно получить полную информацию, выбрав его на сетке щелчком мыши (см. рис. 8). С помощью управляющих кнопок: передвинуть, налево, направо, съесть - можно для выбранного животного выполнить соответствующие действия вручную. Кнопка «инициировать» - инициируют начальную картину виртуального мира, а «запустить» - запускает жизненный цикл на заданном временном интервале (итерация – единица виртуального времени).

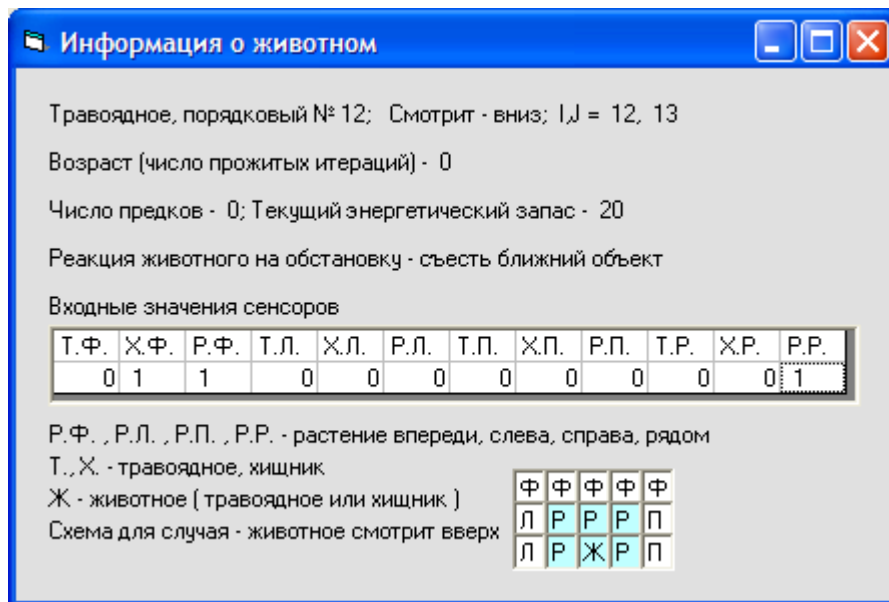


Рис. 8. Экран с информацией о выбранном животном

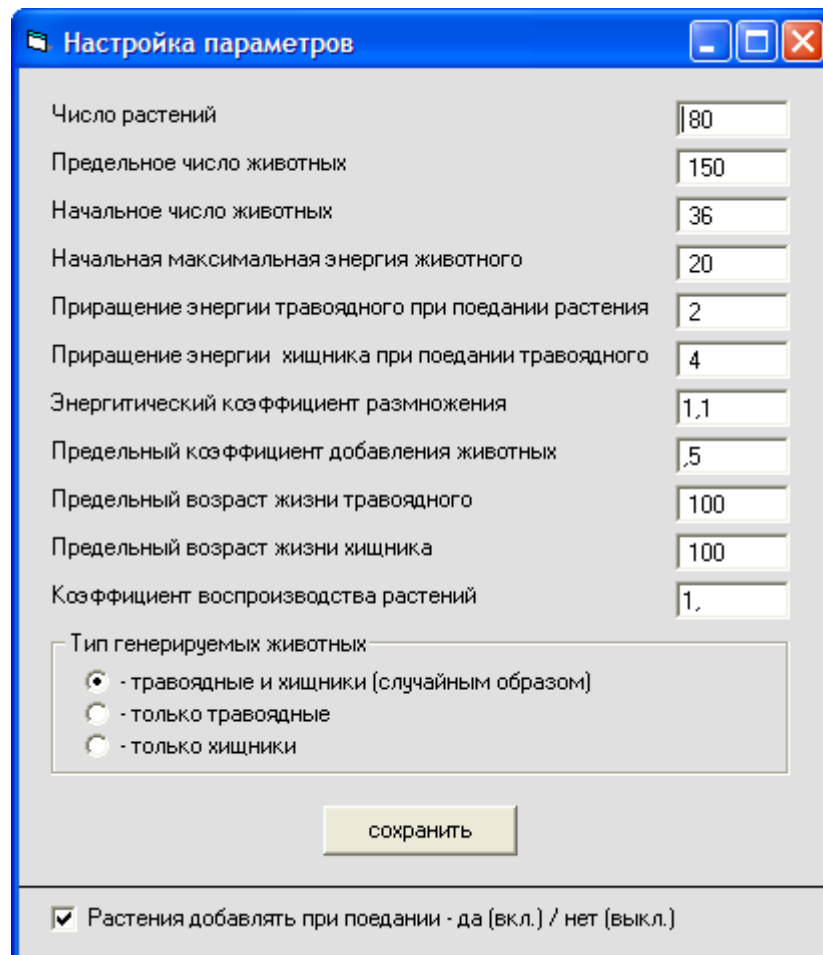


Рис. 9. Экран для настройки параметров моделирования

Пункты меню: "настройка" позволяет задать параметры моделирования (см. рис 9); "веса" – посмотреть, обновить (случайным образом из интервала  $-0,5 \div 0,5$ ) или загрузить из файла матрицы весов промежуточного и выходного слоев нейронных сетей,

моделирующих поведение травоядных и хищных животных (см. рис 10).

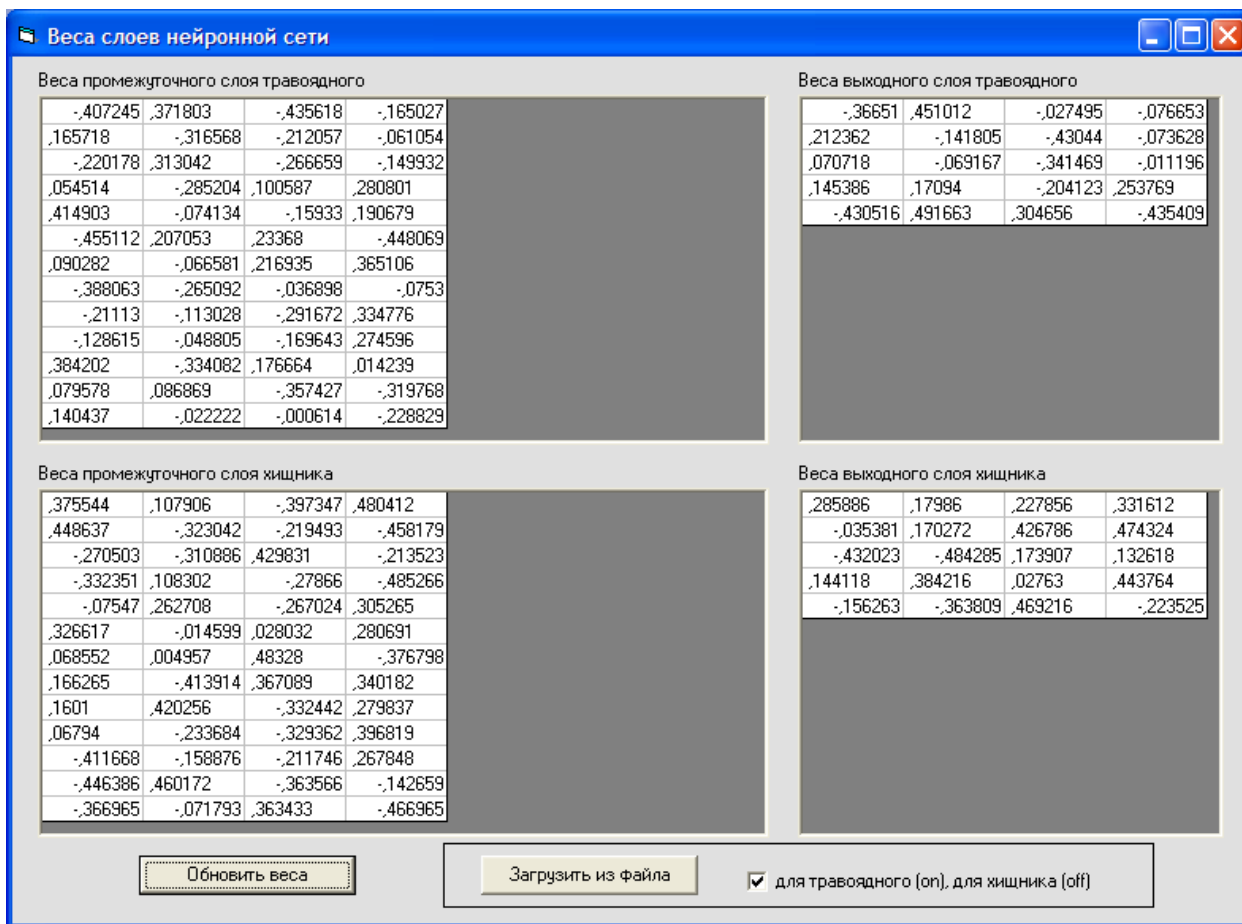


Рис. 10. Экран для отображения значений весов нейронных сетей травоядных и хищников

В процессе функционирования системы для каждой начальной реализации мира собираются статистические данные, которые можно посмотреть (см. рис. 11) на любом конечном промежуточном этапе моделирования (пункт меню «статистика»).



Статистическая информация

	Число прожитых итераций	Средние значения для одной итерации	
Начальное и текущее число растений	50	80	80,
Начальное и текущее число травоядных	23	10	13,44
Начальное и текущее число хищников	13	10	11,36
Число съеденных растений			
	165		3,3
Число появившихся растений			
	165		3,3
Число съеденных травоядных			
	33		,66
Число умерших травоядных			
	9		,18
Число родившихся травоядных			
	16		,32
Число добавленных травоядных			
	13		,26
Максимальное число предков			
	2		
Максимальный возраст			
	50		
Число умерших хищников			
	23		,46
Число родившихся хищников			
	6		,12
Число добавленных хищников			
	14		,28
Максимальное число предков			
	2		
Максимальный возраст			
	34		

Рис. 11. Экран для отображения статистических результатов моделирования

### 5.5. Особенности реализации программы моделирования виртуального мира.

Для хранения информации об объектах в программе используются структуры .

#### Тип агент – животное

- тип: 0 – травоядное, 1 – хищник, -1 – мертвое животное (компонента массива и ячейка графической сетки свободны )
- положение на сетке: координаты ячейки сетки -  $i, j$
- направление движения: 0,1,2,3 – вверх, направо, вниз, налево
- возраст – общее число «прожитых» итераций
- число предков
- текущая энергия
- значения сенсорных входов (12 штук)
- значения выходов нейронной сети (4 штуки: действия – налево, направо, вперед, съесть)

#### Тип растение

- состояние: 1 – живет, -1 – съедено – мертво
- положение на сетке: координаты ячейки сетки -  $i, j$

Весы и пороговые значения слоев нейронных сетей животных хранятся в двух массивах (матрицах весов):

$w\_hid(t, i, j)$  - матрица весов промежуточного слоя

$w\_out(t, i, j)$  - матрица весов выходного слоя

Здесь  $t$  - тип животного (0 – травоядное, 1 – хищник),  $i, j$  - номер компоненты матрицы весов.

Для фиксации положения объектов на графической сетке в программе используются два массива отдельно для растений и животных.

Приведем некоторые алгоритмы, реализующие определенные действия в программе.

### Инициализация мира

1. Инициализация растений
2. Инициализация животных

### Инициализация животных

1. Задаем случайным образом положение животного – координаты свободной ячейки. Если ячейка уже занята животным, то разыгрывается новое случайное положение.
2. Задается случайным образом тип животного (если выбран этот режим инициализации).
3. Задается случайным образом направление движения животного.
4. Присваивается заданное начальное значение энергии.
5. Сбрасываются входы сенсорной системы животного и связанные с ним счетчики: возраст (число прожитых итераций), число предков.

**Моделирование жизненного цикла** в виртуальном мире на временном отрезке, заданном числом итераций

1. Цикл по итерациям
  - 1.1. Цикл по животным
 

Если животное живет, то выполняем для него операции, включенные в итерацию

    - 1.1. Конец цикла по животным
    - 1.2. Проверяем возможность отложенного рождения животного
    - 1.3. Убираем стариков (смерть по старости) и, если выполняются соответствующие условия, добавляем новых животных взамен умерших
    - 1.4. Сбор статистики

1. Конец цикла по итерациям

### Одна итерация для животного

1. Оценка состояния среды.
2. Вычисление ответной реакции, вырабатываемой нейронной сетью в ответ на сигналы сенсорной системы.
3. Выполнение животным соответствующего действия: повернуть направо или налево, переместиться вперед, съесть объект.

**Оценка животным состояния окружающей среды.** Схема анализируемых при этом ячеек сетки приведена на рис. 12, на примере случая, когда животное смотрит вверх. При выходе схемы на край области индексы соответствующим образом перестраиваются, чтобы адресовать ячейки на другом крае области.

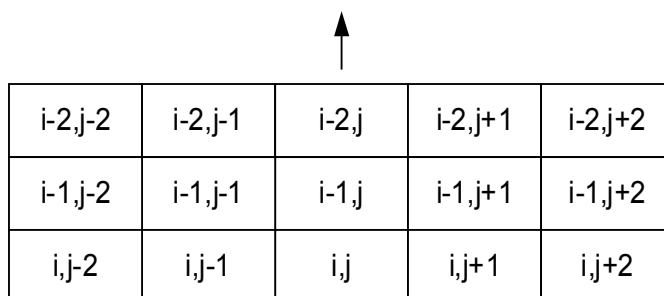


Рис. 12. Схема ячеек, анализируемых животным

1. Проверка наличия рядом растения, хищника и травоядного
2. Проверка наличия впереди (фронт) растения, хищника и травоядного
3. Проверка наличия слева растения, хищника и травоядного

4. Проверка наличия справа растения, хищника и травоядного

**Выполнение животным действия: переместиться вперед, повернуть направо или налево**

1. Выполнить действие
2. Уменьшить энергию
3. Увеличить число прожитых итераций

**Съедание травоядного (аналогично растения)**

1. Анализируем соседние ячейки, находящиеся «рядом», в поисках травоядного
2. Если травоядное найдено, то его съедаем с добавлением энергии хищнику и удалением съеденное травоядное на сетке и в общем массиве с информацией о животных
3. Если энергии животного достаточно для рождения нового животного и есть свободные места в массиве животных (общее максимальное число животных задается при настройке), то создаем новое животное, передавая ему половину энергии родителя
4. Присваиваем числу предков нового животного = число предков родителя + 1
5. Корректируем общий счетчик рожденных животных

**Сбор статистики**

1. Находим животное с максимальным числом предков
2. Находим животное с максимальным возрастом

## 6. Задача оптимизации параметров балки.

### 6.1. Формулировка задачи.

Найти оптимальное продольное распределение момента инерции сечений балки, обеспечивающее для заданной нагрузки минимум ее максимального прогиба.

Рассмотреть два варианта крепления концов балки: два конца жестко закреплены; один конец свободен, а другой жестко закреплен.

### 6.2. Построение математической модели.

Получим дифференциальное уравнение, описывающее прогиб средней линии балки в зависимости от приложенной поперечной нагрузки с плотностью распределения  $q(x)$ .

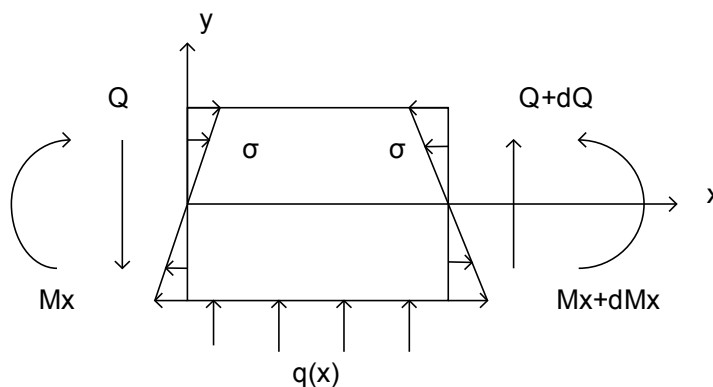


Рис.1. Элемент балки с приложенными нагрузками

Для этого первоначально запишем условия равновесие элементарного участка балки  $dx$  (рис.1) с учетом внешней нагрузки и сил, обусловленных внутренними касательными и нормальными напряжениями в плоских сечениях деформированной балки [6]

$$\sum Y = -Q + (Q + dQ) + qdx = 0$$
$$\sum M_z = -M_z + (M_z + dM_z) + qdx^2 / 2 + (Q + dQ)dx = 0$$

Отсюда, отбрасывая слагаемыми второго порядка малости относительно  $dx$ , получим

$$\frac{dQ}{dx} = -q, \quad \frac{dM_z}{dx} = -Q \Rightarrow \frac{d^2M_z}{dx^2} = q \quad (1)$$

С другой стороны, учитывая известную экспериментальную зависимость (закон Гука)  $\sigma_x = E\varepsilon_x$ , где  $\sigma_x, \varepsilon_x, E$  - нормальное напряжение (действуют в плоскости сечений балки), относительная деформация и модуль Юнга, можно записать

$$M_z = \int_{y'} \sigma_x(y') y' l(y') dy' = E \int_{y'} \varepsilon_x(y') y' l(y') dy' \quad (2)$$

Здесь  $y'$  - соответствующая локальная координата,  $l(y')$  - текущая ширина сечения.

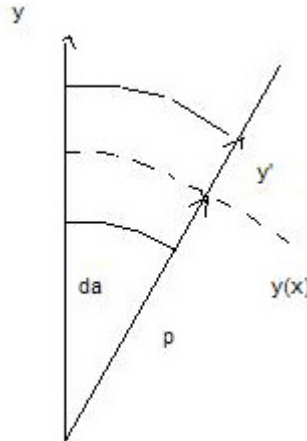


Рис.2. Изгиб элементарного элемента балки

При малых прогибах можно считать, что сечения балки остаются плоскими, а ее нейтральный слой не деформируется. Поэтому относительную деформацию  $\varepsilon_x(y')$  можно связать с радиусом кривизны нейтральной линии  $\rho$  следующим образом (рис.2)

$$\varepsilon_x(y') = \frac{(\rho + y')d\alpha - \rho d\alpha}{\rho d\alpha} = \frac{y'}{\rho} \quad (3)$$

Подставляя (3) в выражение (2), получим

$$M_z = \frac{E}{\rho} J, \quad (4)$$

где  $J(x) = \int_{y(x)} (y')^2 l(y') dy'$  - момент инерции сечения балки.

Согласно дифференциальной геометрии радиус кривизны нейтральной линии можно связать с ее уравнением  $y(x)$  (принято во внимание, что  $(\frac{dy}{dx})^2 \ll 1$ )

$$\frac{1}{\rho} = \frac{\frac{d^2 y}{dx^2}}{\sqrt{\left[1 + \left(\frac{dy}{dx}\right)^2\right]^{3/2}}} \approx \frac{d^2 y}{dx^2} \quad (5)$$

С учетом формул (4), (5) выражение (1) приводится к требуемому дифференциальному уравнению, описывающему прогиб балки под действием нагрузки,

$$E \frac{d^2}{dx^2} (J(x) \frac{d^2 y}{dx^2}) = q(x) \quad (6)$$

К данному уравнению необходимо добавить граничные условия (здесь  $L$  - длина балки):

- оба конца закреплены

$$x=0, x=L \quad y = \frac{dy}{dx} = 0 \quad (7)$$

- один конец ( $x=0$ ) закреплен, другой ( $x=L$ ) - свободен ( $M_z \approx \frac{d^2y}{dx^2} = 0, Q \approx \frac{d^3y}{dx^3} = 0$ )

$$x=0 \quad y = \frac{dy}{dx} = 0, \quad x=L \quad \frac{d^2y}{dx^2} = \frac{d^3y}{dx^3} = 0 \quad (8)$$

### 6.3. Унификация задачи

Чтобы сделать задачу независимой от исходных параметров, введем следующие безразмерные величины

$$x \equiv x/L, \quad q(x) \equiv q(x)/q_0, \quad J(x) \equiv J(x)/J_0, \quad u(x) = y(x)/F \quad (9)$$

$$\text{где } q_0 = \frac{1}{L} \int_0^L |q(x)| dx, \quad J_0 = \frac{1}{L} \int_0^L J(x) dx, \quad F = \frac{q_0 L^4}{EJ_0}$$

Для новых безразмерных параметров справедливы следующие интегральные соотношения

$$\int_0^1 |q(x)| dx = 1, \quad \int_0^1 J(x) dx = 1 \quad (10)$$

Теперь для двух вариантов граничных условий (7), (8) получаем две безразмерные краевые задачи (здесь и в дальнейшем будем считать нагрузку, направленную вниз, положительной):

*Задача 1*

$$\begin{aligned} \frac{d^2}{dx^2} \left( J(x) \frac{d^2 u}{dx^2} \right) &= -q(x), \quad x \in [0,1] \\ x=0, x=1 \quad u &= \frac{du}{dx} = 0 \end{aligned} \quad (11)$$

*Задача 2*

$$\begin{aligned} \frac{d^2}{dx^2} \left( J(x) \frac{d^2 u}{dx^2} \right) &= -q(x), \quad x \in [0,1] \\ x=0 \quad u &= \frac{du}{dx} = 0 \\ x=1 \quad \frac{d^2 u}{dx^2} &= \frac{d^3 u}{dx^3} = 0 \end{aligned} \quad (12)$$

### 6.4. Проверка адекватности полученной математической модели

Для случая постоянного по длине балки момента инерции сечения  $J(x)=1$  и равномерно распределенной нагрузки  $q(x)=1$  задачи 1 и 2 имеют простые аналитические решения

$$u_1(x) = -(x^2 - 2x^3 + x^4)/24$$

$$u_2(x) = -(6x^2 - 4x^3 + x^4)/24$$

На рис.3,4 приведены графики этих функций. Характер кривых соответствует реально наблюдаемым прогибам балки для подобных случаев. Это говорит о том, что составленная математическая модель адекватно описывает поведение балки под нагрузкой.

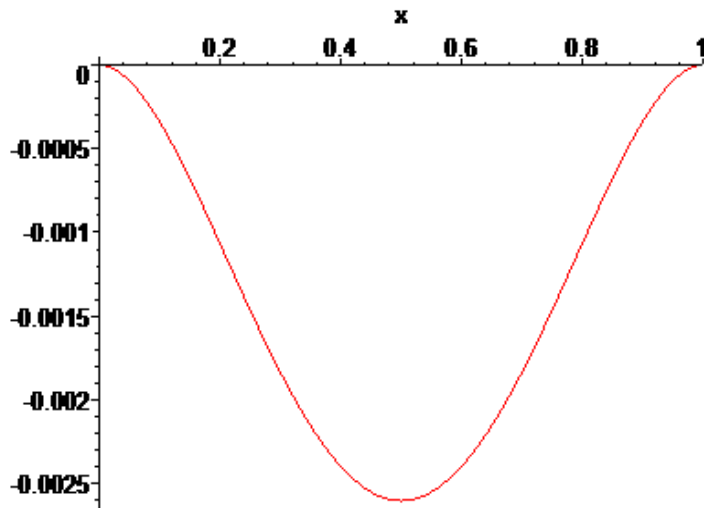


Рис.3. Прогиб балки постоянного сечения при закрепленных концах

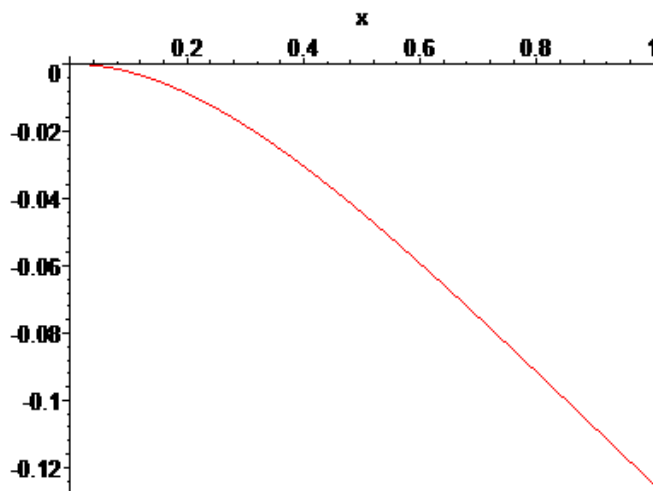


Рис.4. Прогиб балки постоянного сечения при левом закрепленном конце

#### 6.5. Выбор метода решения краевой задачи для общего случая.

Для решения краевых задач (11), (12) целесообразно использовать разностный метод, поскольку в этом случае задача сводится к системе линейных уравнений (СЛУ) с пяти диагональной матрицей, для решения которой разработан экономичный и устойчивый метод прогонки.

Схему вывода СЛУ подробно рассмотрим на примере задачи (11).

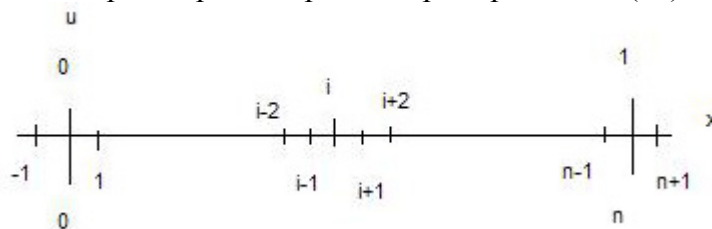


Рис.5. Дискретная модель балка

Заменим отрезок  $x \in (0,1)$  расширенной системой точек  $i \in (-1, n+1)$  (рис.5) с величиной шага  $h$ . Для точки  $i \in (1, n-1)$  дискретный аналог дифференциального оператора уравнения задачи можно записать

$$\frac{d^2}{dx^2} (J(x) \frac{d^2 u}{dx^2})_i \approx \frac{(J \frac{d^2 u}{dx^2})_{i+1} - 2(J \frac{d^2 u}{dx^2})_i + (J \frac{d^2 u}{dx^2})_{i-1}}{h^2} \approx \frac{J_{i+1}(u_{i+2} - 2u_{i+1} + u_i) - 2J_i(u_{i+1} - 2u_i + u_{i-1}) + J_{i-1}(u_i - 2u_{i-1} + u_{i-2})}{h^4} \quad (13)$$

Итого с учетом (13), дискретный аналог задачи имеет следующий вид: граничные условия при  $x = 0 (i = 0)$

$$u_0 = 0, \quad \frac{u_1 - u_{-1}}{2h} = 0 \Rightarrow u_0 = 0, \quad u_{-1} = u_1 \quad (14)$$

уравнения  $i \in (1, n-1)$

$$a_i u_{i-2} + b_i u_{i-1} + c_i u_i + d_i u_{i+1} + e_i u_{i+2} = f_i \quad (15)$$

граничные условия при  $x = 1 (i = n)$

$$u_n = 0, \quad \frac{u_{n+1} - u_{n-1}}{2h} = 0 \Rightarrow u_n = 0, \quad u_{n+1} = u_{n-1} \quad (16)$$

где

$$a_i = J_{i-1}, \quad b_i = -2(J_i + J_{i-1}), \quad c_i = J_{i+1} + 4J_i + J_{i-1}, \quad d_i = -2(J_{i+1} + J_i), \quad e_i = J_{i+1}, \quad f_i = -q_i h^4$$

Подставляя граничные условия (14), (16) в уравнения (15) для  $i = 1, 2, n-2, n-1$ , окончательно получим СЛУ с пяти диагональной матрицей для определения значений безразмерного прогиба балки  $u_i$  в заданной системе точек

$$\begin{aligned} (a_1 + c_1)u_1 + d_1 u_2 + e_1 u_3 &= f_1 & i = 1 \\ b_2 u_1 + c_2 u_2 + d_2 u_3 + e_2 u_4 &= f_2 & i = 2 \\ a_i u_{i-2} + b_i u_{i-1} + c_i u_i + d_i u_{i+1} + e_i u_{i+2} &= f_i & i \in (3, n-3) \\ a_{n-2} u_{n-4} + b_{n-2} u_{n-3} + c_{n-2} u_{n-2} + d_{n-2} u_{n-1} &= f_{n-2} & i = n-2 \\ a_{n-1} u_{n-3} + b_{n-1} u_{n-2} + (c_{n-1} + e_{n-1}) u_{n-1} &= f_{n-1} & i = n-1 \end{aligned} \quad (17)$$

Решение системы (17) можно найти методом прогонки. Для этого введем следующее рекуррентное соотношение

$$u_i = \alpha_i u_{i+1} + \beta_i u_{i+2} + \gamma_i \quad i \in (1, n-3) \quad (18)$$

Сравнивая первое уравнение системы (17) с (18) сразу получаем

$$\alpha_1 = -\frac{d_1}{a_1 + c_1}, \quad \beta_1 = -\frac{e_1}{a_1 + c_1}, \quad \gamma_1 = \frac{f_1}{a_1 + c_1} \quad (19)$$

Далее, подставляя соотношение (18) с  $i = 1$  во второе уравнение системы, аналогично выводим

$$\alpha_2 = -\frac{d_2 + b_2 \beta_1}{\tau}, \quad \beta_2 = -\frac{e_2}{\tau}, \quad \gamma_2 = \frac{f_2 - b_2 \gamma_1}{\tau}, \quad \tau = c_2 + b_2 \alpha_1 \quad (20)$$

Теперь заменяя в  $i$ -ом уравнение системы (17) переменные  $u_{i-2}, u_{i-1}$  согласно (18), получим следующие формулы для определения остальных неизвестных параметров соотношения (18)

$$\alpha_i = -\frac{d_i + \eta \beta_{i-1}}{\tau}, \quad \beta_i = -\frac{e_i}{\tau}, \quad \gamma_i = \frac{f_i - \eta \gamma_{i-1} - a_i \gamma_{i-2}}{\tau} \quad i \in (3, n-3) \quad (21)$$

Здесь  $\eta = b_i + a_i \alpha_{i-2}$ ,  $\tau = c_i + \eta \alpha_{i-1} + a_i \beta_{i-2}$

Расчет по формулам (19)-(21) называется «прямых» ходом прогонки.

Для запуска «обратного» хода прогонки необходимо определить  $u_{n-2}, u_{n-1}$ , после чего остальные неизвестные  $u_i \quad i = (n-3, \dots, 1)$  последовательно рассчитываются по формуле (18).

Начальные неизвестные  $u_{n-2}, u_{n-1}$  находятся из системы, состоящей из двух последних уравнений исходной системы (17) и двух рекуррентных соотношений (18)

$$u_{n-3} = \alpha_{n-3}u_{n-2} + \beta_{n-3}u_{n-1} + \gamma_{n-3}$$

$$u_{n-4} = \alpha_{n-4}u_{n-3} + \beta_{n-4}u_{n-2} + \gamma_{n-4} = \alpha_{n-4}(\alpha_{n-3}u_{n-2} + \beta_{n-3}u_{n-1} + \gamma_{n-3}) + \beta_{n-4}u_{n-2} + \gamma_{n-4}$$

Решая эту систему, получим

$$u_{n-2} = \frac{F_1 A_{22} - F_2 A_{12}}{\Delta}, \quad u_{n-1} = \frac{F_2 A_{11} - F_1 A_{21}}{\Delta} \quad (22)$$

где

$$\Delta = A_{11}A_{22} - A_{12}A_{21}$$

$$A_{11} = b_{n-1} + a_{n-1}\alpha_{n-3}, \quad A_{12} = c_{n-1} + a_{n-1}\beta_{n-3}, \quad F_1 = f_{n-1} - a_{n-1}\gamma_{n-3}$$

$$A_{21} = c_{n-2} + b_{n-2}\alpha_{n-3} + a_{n-2}(\alpha_{n-3}\alpha_{n-4} + \beta_{n-4})$$

$$A_{22} = d_{n-2} + b_{n-2}\beta_{n-3} + a_{n-2}\beta_{n-3}\alpha_{n-4}$$

$$F_2 = f_{n-2} - a_{n-2}(\gamma_{n-3}\alpha_{n-4} + \gamma_{n-4}) - b_{n-2}\gamma_{n-3}$$

Прогиб балки со свободным правым концом (задача (12)) определяется аналогичным образом. При этом для данного случая СЛУ имеет вид

$$\begin{aligned} (a_1 + c_1)u_1 + d_1u_2 + e_1u_3 &= f_1 & i = 1 \\ b_2u_1 + c_2u_2 + d_2u_3 + e_2u_4 &= f_2 & i = 2 \\ a_iu_{i-2} + b_iu_{i-1} + c_iu_i + d_iu_{i+1} + e_iu_{i+2} &= f_i & i \in (3, n-2) \\ a_{n-1}u_{n-3} + b_{n-1}u_{n-2} + (c_{n-1} - e_{n-1})u_{n-1} + (d_{n-1} + 2e_{n-1})u_n &= f_{n-1} & i = n-1 \\ u_{n-2} - 2u_{n-1} + u_n &= 0 & i = n \end{aligned} \quad (23)$$

Здесь были использованы следующие дискретные аналоги граничных условий при  $x = 1$

$$(M_Z)_n = \left(\frac{d^2u}{dx^2}\right)_n = u_{n-1} - 2u_n + u_{n+1} = 0 \Rightarrow u_{n+1} = 2u_n - u_{n-1}$$

$$Q_n = \left(\frac{dM_Z}{dx}\right)_n = (M_Z)_n - (M_Z)_{n-1} = -(M_Z)_{n-1} = -(u_{n-2} - 2u_{n-1} + u_n) = 0$$

## 6.6. Решение задачи минимизации максимального прогиба балки

Задача оптимизации состоит в нахождении оптимального распределении момента инерции сечений балки по ее длине в соответствии со следующим критерием

$$\min_{J(x)} \max_x u(x, J(x)) \quad (24)$$

при ограничениях  $J(x) > 0$  и  $\int_0^1 J(x)dx = 1$ .

Для упрощения решения поставленной задачи оптимизации будем искать распределение функции  $J(x)$  в виде квадратичного полинома (полиномы 3-го порядка и выше за счет точек перегиба дают участки, где  $J(x) < 0$ )

$$J(x) = a_1 + a_2x + a_3x^2 \quad (25)$$



Его коэффициенты в соответствии с условием нормировки  $\int_0^1 J(x)dx = 1$  должны

подчиняться следующему равенству  $a_1 + \frac{a_2}{2} + \frac{a_3}{3} = 1$ .

Минимизация функционала  $F(a_1, a_2, a_3) = \max_x u(x, J(x))$  целесообразно выполнить методом наискорейшего спуска (МНС), поскольку малое число параметров оптимизации (=3) делает процедуру расчета градиента функционала мало затратной. Приведем основные составляющие алгоритм МНС для одной итерации.

1. Расчет направляющих ортов градиента (здесь можно взять  $\partial a \approx 0,001$ )

$$\begin{aligned}
 &F_0 = F(a_1, a_2, a_3), \quad a_{10} = a_1, \quad a_{20} = a_2, \quad a_{30} = a_3, \quad \Delta = 0 \\
 &\text{do for } i = 1, 3 \\
 &\quad a_i = a_{i0} + \partial a \\
 &\quad \Sigma = a_1 + a_2/2 + a_3/3 \\
 &\quad a_1 = a_1/\Sigma, \quad a_2 = a_2/\Sigma, \quad a_3 = a_3/\Sigma \\
 &\quad F_i = F(a_1, a_2, a_3) \\
 &\quad \frac{\partial F}{\partial a_i} = \frac{F_i - F_0}{\partial a_i} \\
 &\quad \Delta = \Delta + \left(\frac{\partial F}{\partial a_i}\right)^2 \\
 &\text{end } i \\
 &\Delta = \sqrt{\Delta} \\
 &\quad \frac{\partial F}{\partial a_1} = \frac{\partial F}{\Delta}, \quad \frac{\partial F}{\partial a_2} = \frac{\partial F}{\Delta}, \quad \frac{\partial F}{\partial a_3} = \frac{\partial F}{\Delta}
 \end{aligned}$$

2. Спуск вдоль градиента – один из простейших вариантов алгоритма ( $h$  - шаг спуска)

$$\begin{aligned}
 &h = 0,05, \quad a_{10} = a_1, \quad a_{20} = a_2, \quad a_{30} = a_3, \quad F_0 = F(a_{10}, a_{20}, a_{30}) \\
 &\text{do while } h > 0,001 \\
 &\quad a_1 = a_{10} - h\partial a_1, \quad a_2 = a_{20} - h\partial a_2, \quad a_3 = a_{30} - h\partial a_3 \\
 &\quad \Sigma = a_1 + a_2/2 + a_3/3 \\
 &\quad a_1 = a_1/\Sigma, \quad a_2 = a_2/\Sigma, \quad a_3 = a_3/\Sigma \\
 &\quad F_i = F(a_1, a_2, a_3) \\
 &\quad \text{if } F_i > F_0 \text{ then} \\
 &\quad \quad h = h/2 \\
 &\quad \text{else} \\
 &\quad \quad a_{10} = a_1, \quad a_{20} = a_2, \quad a_{30} = a_3, \quad F_0 = F_i \\
 &\quad \text{end if} \\
 &\text{end while}
 \end{aligned}$$

Используются два критерия окончания итерационного процесса:

1. разность функционалов двух соседних итераций меньше заданного  $eps$  ;
2. использовано максимального числа итераций.

## 6.7 Программная реализация модели

Интерфейс программы для исследования поведения балки под нагрузкой (распределение нагрузки задается коэффициентами квадратичного полинома) и нахождения оптимального распределения по длине балки момента инерции сечений  $J(x)$  приведен на рис. 6 и 7. На последнем рисунке представлено дополнительное окно, задающее параметры оптимизации.

Программа позволяет:

- изучать задачи с двумя вариантами граничных условий;
- задавать распределения нагрузки и начального момента инерции сечений от постоянного до квадратичного;
- для случая  $J(x) = const$  исследовать влияние частоты сетки на точность численного решения (этот случай допускает точное аналитическое решение, которое «зашито» в программе);
- находить оптимальное распределение  $J(x)$  для заданной нагрузки и качественно сравнивать графики прогибов балки для случаев оптимального и постоянного распределений момента инерции сечений.

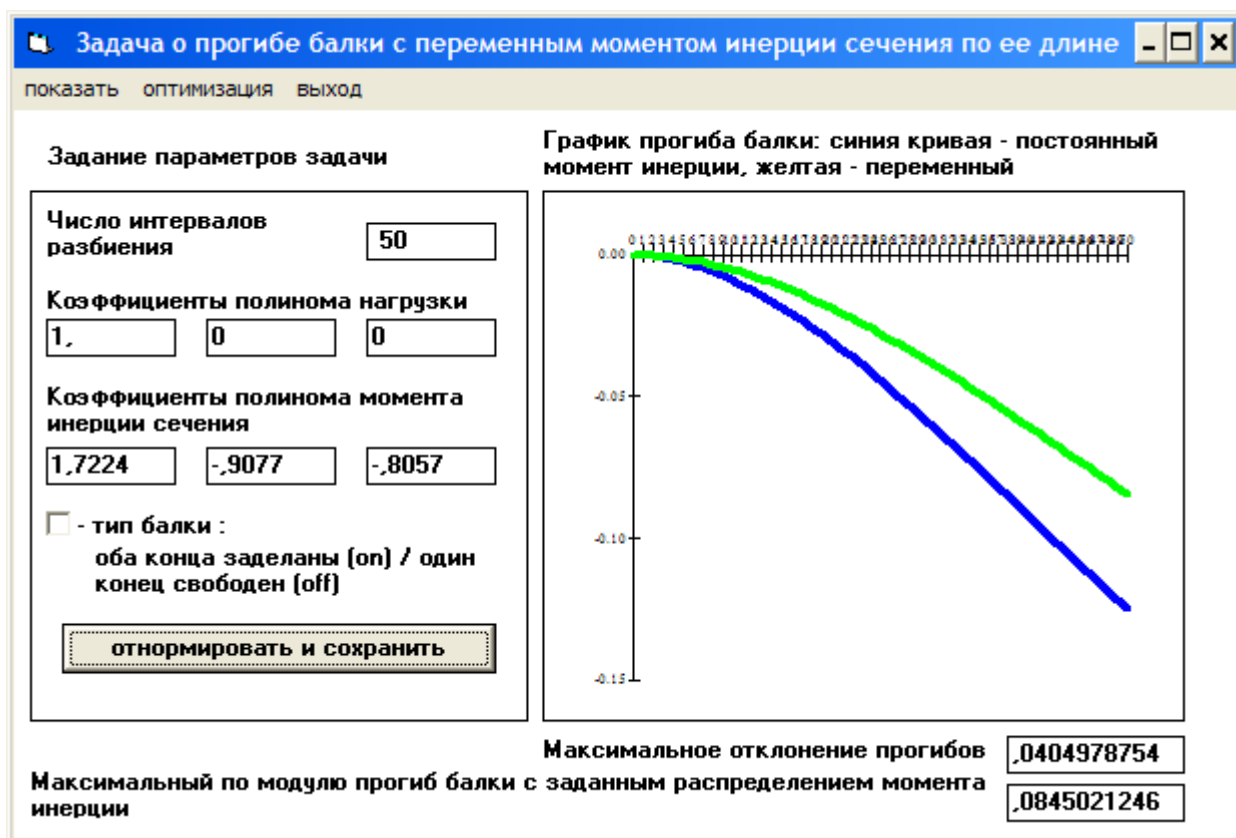


Рис.6. Основной экран программы, на котором представлено распределение прогиба балки с одним закрепленным концом

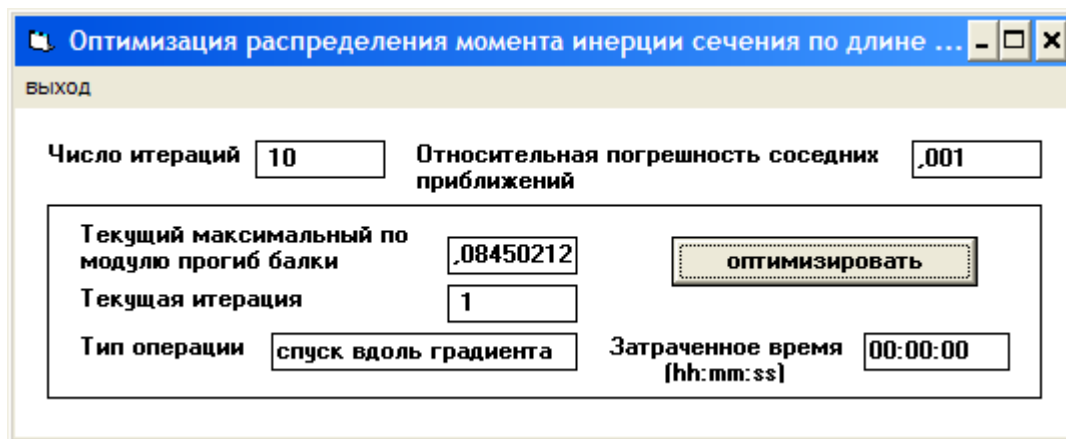


Рис.7. Вспомогательный экран для задачи оптимизации

### 6.8. Результаты исследований модели

На рис.6 и 8 (второй и первый варианты граничных условий) видны совместные графики прогибов балки для случаев оптимального (расположены сверху) и постоянного распределений момента инерции сечений. Нагрузка принималась постоянной. За счет выбора оптимального распределения момента инерции удалось уменьшить максимальный прогиб балки, соответственно, на 30% и 10% по сравнению с постоянным распределением. Подобные числа говорят в пользу использования на практике балок с оптимальными продольным распределением сечений по длине (что обычно и осуществляется на практике). Графики оптимальных распределений моментов инерции сечений для обсуждаемых случаев представлены на рис.9 (сплошная кривая – для балки с двумя закрепленными концами, точечная – правый конец балки свободен).

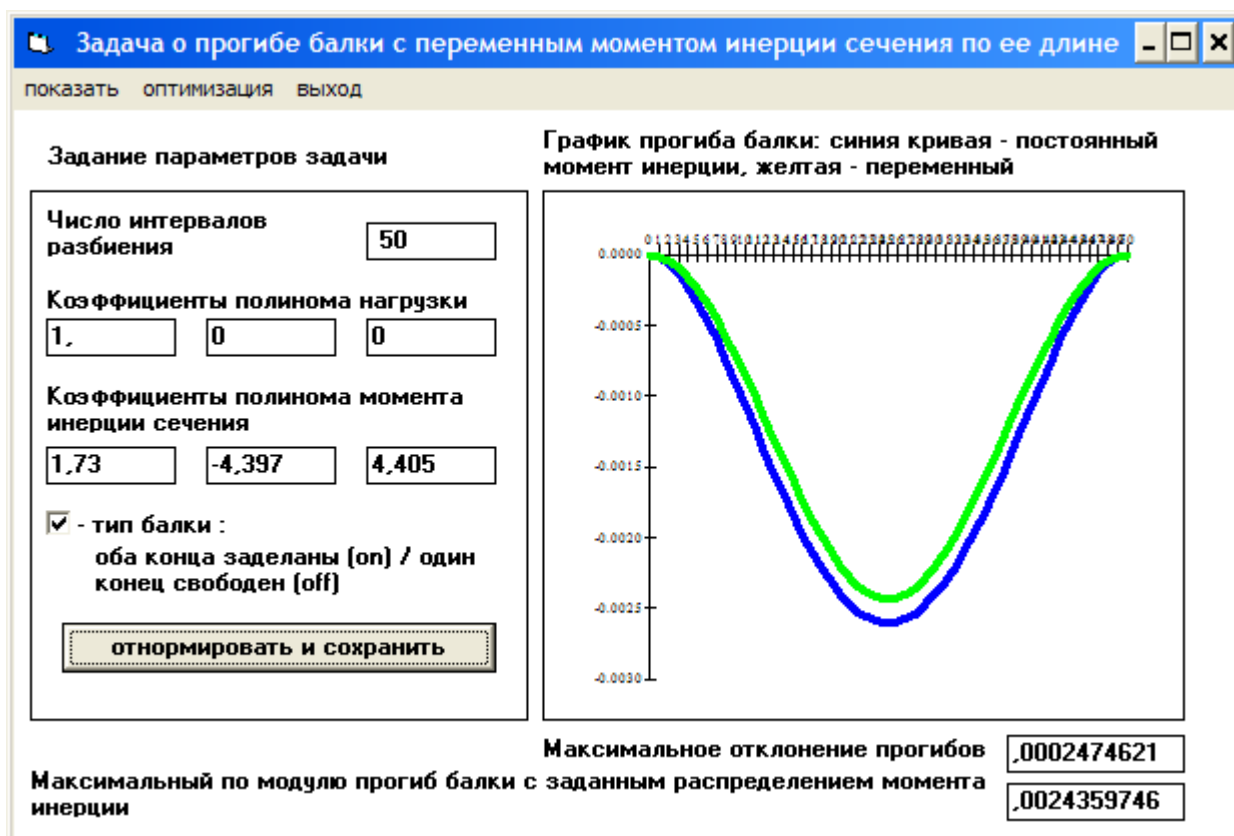


Рис.8. Распределение прогиба балки с закрепленными концами

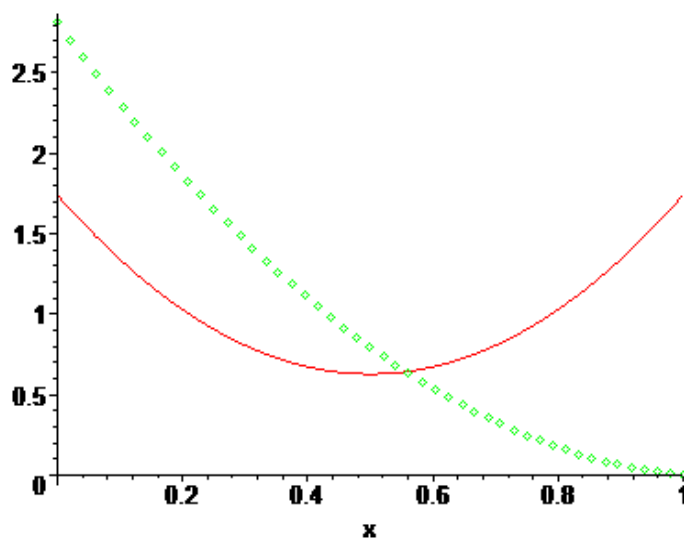


Рис. 9. Графики оптимальных распределений моментов инерции сечений балки для обоих случаев закрепления ее концов

## 7. Оптимизация параметров точечных источников энергии, обеспечивающих равномерное стационарное температурное поле в плоской прямоугольной области

### 7.1 Формулировка задачи

Найти мощности и положение заданного количества  $I$  точечных источников энергии, которые в плоской прямоугольной области  $x, y \in (-l_x, l_x; -l_y, l_y)$  поддерживают равномерное стационарное температурное поле заданной величины  $T_f$ . Имеет место отток энергии через наружную границу области.

### 7.2 Математическая постановка задачи

Известно, что согласно закону Фурье плотность потока энергии через единичную поверхность задается соотношением  $-k \frac{dT}{dn}$ , где  $k$  - коэффициент теплопроводности,  $n$  - единичная нормаль к данной поверхности. Записав уравнение баланса энергии в элементе  $dxdy$  (разность потока вытекающей энергии и потока втекающей равна количеству энергии, выделяемой источниками в данном элементе) (рис. 1), легко вывести дифференциальное уравнение, связывающее плотность мощности источников энергии  $q(x)$  и стационарное распределение температуры  $T(x, y)$  в плоскости  $x, y$

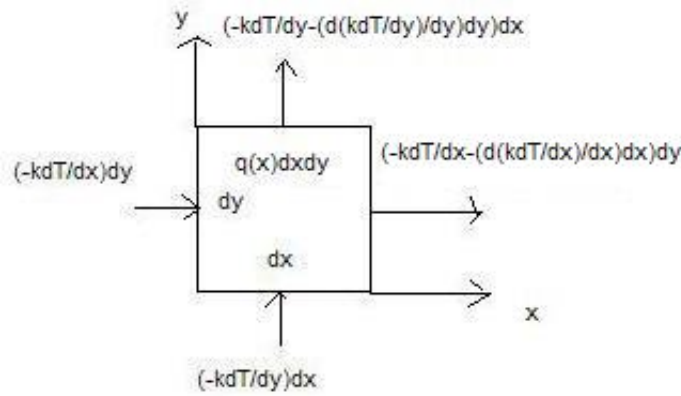


Рис.1. Схема потоков энергии в элементе  $dx dy$

$$\frac{\partial}{\partial x} (k(x, y) \frac{\partial T}{\partial x}) + \frac{\partial}{\partial y} (k(x, y) \frac{\partial T}{\partial y}) = -q(x, y)$$

или для случая постоянного коэффициента теплопроводности  $k(x, y) \equiv k$

$$\Delta T = -q(x, y)/k,$$

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

В предположении симметричности температурного поля относительно осей координат  $x, y$  исходная задача математически записывается следующим образом:

а. Уравнение стационарной теплопроводности

$$\Delta T = -\frac{1}{k} \sum_{i=1}^{I/4} q_i \delta(x - x_i) \delta(y - y_i), \quad x, y \in (0, l_x; 0, l_y)$$

б. Граничные условия

$$\frac{\partial T}{\partial x} = 0, x = 0; \quad \frac{\partial T}{\partial y} = 0, y = 0 \quad - \text{теплоизолированные границы}$$

$$-k \frac{\partial T}{\partial x} = h(T - T_0), x = l_x; \quad -k \frac{\partial T}{\partial y} = h(T - T_0), y = l_y \quad - \text{поток энергии через границу}$$

$T_0$  - температура окружающей среды,  $h$  - коэффициент поверхностной теплопроводности,  $\delta$  - дельта-функция,  $x_i, y_i, q_i$  - координаты и мощность  $i$ -го источника энергии.

в. Функционал, минимизация которого обеспечивает оптимальную среднюю и локальную гладкость температурного поля и его близость к заданному значению ( $T(x, y) \equiv T_f$ )

$$\Phi(q_i, x_i, y_i) = \alpha_1 \frac{1}{S} \int_S |T_f - T(x, y)| ds + \alpha_2 \max_S |T_f - T(x, y)| + \alpha_3 \frac{1}{S} \int_S |\text{grad} T(x, y)| ds +$$

$$\alpha_4 \max_S |\text{grad} T(x, y)| + \alpha_5 \sum_{i=1}^{I/4} q_i$$

Здесь  $\alpha_i$  - весовые коэффициенты,  $S$  - площадь области  $(l_x l_y)$ . Последнее слагаемое добавлено для обеспечения корректности задачи оптимизации.

г. Ограничения на параметры оптимизации

$$x_i, y_i \in (0, l_x; 0, l_y), \quad q_i > 0$$

### 7.3. Унификация задачи

Для уменьшения зависимости задачи от исходных параметров введем следующие безразмерные величины

$$\begin{aligned}
y &\equiv \frac{y}{l_y}, \quad x \equiv \frac{x}{l_x} \Rightarrow y \in (0,1), x \in (0, l_x \equiv \frac{l_x}{l_y}) \\
T &\equiv \frac{T - T_0}{T_f - T_0} \Rightarrow T_f = 1 \\
q_i &\equiv \frac{q_i l_y^2}{k(T_f - T_0)}, \quad h \equiv \frac{h l_y}{k}
\end{aligned} \tag{1}$$

С учетом этих изменений задача записывается в следующем виде ( $I \equiv I/4$ ,  $s = l_x * 1$ )

а.

$$\Delta T = - \sum_{i=1}^I q_i \delta(x - x_i) \delta(y - y_i), \quad x, y \in (0, l_x; 0,1) \tag{2}$$

б.

$$\begin{aligned}
\frac{\partial T}{\partial x} = 0, x = 0; \quad \frac{\partial T}{\partial y} = 0, y = 0 \\
\frac{\partial T}{\partial x} = -hT, x = l_x, \quad \frac{\partial T}{\partial y} = -hT, y = 1
\end{aligned} \tag{3}$$

в.

$$\begin{aligned}
\Phi(q_i, x_i, y_i) = \alpha_1 \frac{1}{l_x s} \int |1 - T(x, y)| dx dy + \alpha_2 \max_S |1 - T(x, y)| + \alpha_3 \frac{1}{l_x s} \int |\text{grad} T(x, y)| dx dy + \\
\alpha_4 \max_S |\text{grad} T(x, y)| + \alpha_5 \sum_{i=1}^I q_i
\end{aligned} \tag{4}$$

г.

$$x_i, y_i \in (0, l_x; 0,1), \quad q_i > 0 \tag{5}$$

7.4. Управляемость задачи и критерий выбора начального приближения для параметров оптимизации.

Для проверки возможности управления температурным полем в заданной постановке рассмотрим одномерный безразмерный случай данной задачи с одним источником

$$\begin{aligned}
\frac{d^2 T}{dx^2} = -q \delta(x - x_q), \quad x, x_q \in (0,1) \\
\frac{dT}{dx} = 0, x = 0; \quad \frac{dT}{dx} = -hT, x = 1
\end{aligned} \tag{6}$$

Решение этой задачи записывается следующим образом

$$T = \begin{cases} C, & x \leq x_q \\ C - q(x - x_q), & x > x_q \end{cases}$$

где  $C = q[\frac{1}{h} + (1 - x_q)]$ .

С помощью этого выражения задача оптимизации решается элементарно:

- источник следует разместить на границе ( $x_q = 1$ )
- безразмерную мощность точечного источника взять равной  $h$ .

Выведем еще одно полезное ограничение на суммарную безразмерную мощность точечных источников.

Согласно второму граничному условию (3) приведенный поток энергии через всю границу области при идеальной оптимизации ( $T(x, y) = 1$ ) составляет  $h(l_x + 1)$  и при

стационарном режиме должен совпадать с суммарной безразмерной мощностью всех точечных источников, расположенных внутри этой области  $\sum_{i=1}^I q_i$ . Поскольку построение идеально постоянного температурного поля не реально, то всегда будет выполняться следующее неравенство

$$\sum_{i=1}^I q_i \leq h(l_x + 1) \quad (7)$$

Это неравенство следует использовать как на стадии задания начальных мощностей точечных источников, так и для проверки «правдоподобия» полученных после оптимизации результатов.

### 7.5. Решение температурной задачи

При построении решения задачи (2), (3) нам потребуется полная линейно-независимая система собственных функций оператора Лапласа, удовлетворяющих граничным условиям (3). Рассмотрим задачу нахождения этих функций

$$\begin{aligned} \Delta u &= -\lambda u, \quad x, y \in (0, l_x; 0, 1) \\ \frac{\partial u}{\partial x} &= 0, \quad x = 0; \quad \frac{\partial u}{\partial y} = 0, \quad y = 0 \\ \frac{\partial u}{\partial x} &= -hu, \quad x = l_x; \quad \frac{\partial u}{\partial y} = -hu, \quad x = 1 \end{aligned} \quad (8)$$

В соответствии с методом разделения переменных будем искать ее решение в виде  $u(x, y) = X(x)Y(y)$ . В этом случае уравнение (8) переписывается следующим образом

$$\frac{d^2 X}{dx^2} Y + \frac{d^2 Y}{dy^2} X = -\left(\frac{\lambda_x^2}{l_x^2} + \lambda_y^2\right) XY$$

Здесь собственное значение представлено в эквивалентном виде  $\lambda = \left(\frac{\lambda_x^2}{l_x^2} + \lambda_y^2\right)$ .

Задача (2.8) распадется на две самостоятельные задачи на собственные значения

$$\begin{aligned} \frac{d^2 X}{dx^2} + \frac{\lambda_x^2}{l_x^2} X &= 0, \quad \frac{dX}{dx} = 0, \quad x = 0, \quad \frac{dX}{dx} = -hX, \quad x = l_x \\ \frac{d^2 Y}{dy^2} + \lambda_y^2 Y &= 0, \quad \frac{dY}{dy} = 0, \quad y = 0, \quad \frac{dY}{dy} = -hY, \quad y = 1 \end{aligned} \quad (9)$$

Их собственные функции

$$X_k(x) = \cos\left(\frac{\lambda_{xk}}{l_x} x\right), \quad Y_n(y) = \cos(\lambda_{yn} y), \quad k, n = 1, 2, 3, \dots$$

удовлетворяют первым граничным условиям при  $x = 0, y = 0$  автоматически, а вторым граничным условиям – только в том случае, если в качестве собственных значений взять корни соответствующих нелинейных уравнений

$$\operatorname{tg} \lambda_{xk} = \frac{hl_x}{\lambda_{xk}}, \quad \operatorname{tg} \lambda_{yn} = \frac{h}{\lambda_{yn}} \quad (10)$$

Для поиска корней этих уравнений можно предложить следующий алгоритм.

Графически (рис. 2,  $y = \operatorname{tg}(x)$  – сплошная линия;  $y = k/x$  – точечная,  $k = 2$ ) легко устанавливается, что корни этих уравнений располагаются на интервалах  $[i\pi, (i\pi + \frac{\pi}{2})]$ ,  $i = 0, 1, 2, \dots$ . Видно что, чем больше номер корня, тем он ближе смещается к левой границы своего интервала.

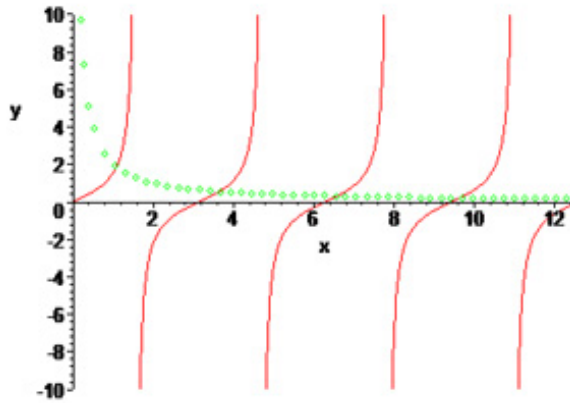


Рис. 2. Локализация корней нелинейных уравнений

На каждом интервале вначале производится локализация корня с постоянным шагом 0,01 (отрезок содержит корень функции, если ее значения на концах отрезка имеют разные знаки), а затем положение этого корня уточняется методом дихотомии – отрезок, содержащий корень, последовательно делится пополам до тех пор, пока значение функции  $f(x) = tg(x) - k/x$  в средней точке не станет меньше заданной погрешности, например  $\varepsilon = 0,0001$ . Приведем алгоритм дихотомии для интервала  $[x_b, x_e]$

```

 $f_b = f(x_b), f_e = f(x_e)$ 
for  $i = 1, 100$  - цикл по числу делений отрезка
 $x_t = (x_b + x_e) / 2, f_t = f(x_t)$ 
if  $|f_t| < \varepsilon$  then exit for
if  $f_b f_t < 0$  then
 $x_e = x_t, f_e = f_t$ 
else
 $x_b = x_t, f_b = f_t$ 
end if
end for

```

Для дальнейшего использования все найденные корни сохраняются в двух соответствующих двоичных файлах kx.\$\$\$, ky.\$\$\$.

Теперь вернемся к исходной задаче (2), (3). Ее решение будем искать методом Бубнова-Галеркина [8]. Для этого запишем искомое решение в виде конечного разложения по найденным собственным функциям оператора Лапласа

$$u_{kn}(x, y) = \cos\left(\frac{\lambda_{xk}}{l_x} x\right) \cos(\lambda_{yn} y)$$

$$u(x, y) \approx \sum_{k=1}^K \sum_{n=1}^N B_{kn} u_{kn}(x, y) \quad (11)$$

Это разложение удовлетворяет всем граничным условиям (2.3) при любых значениях неизвестных коэффициентов  $B_{kn}$ .

Неизвестные коэффициенты  $B_{kn}$  в количестве  $K * N$  определим из условий ортогональности невязки  $\delta(x, y)$ , которая получается при подстановке в уравнение (2.2) разложения (11), конечному числу полной линейно-независимой системе тех же собственных функций

$$u_{lm}(x, y) = \cos\left(\frac{\lambda_{xl}}{l_x} x\right) \cos(\lambda_{ym} y)$$



$$(\delta * u_{lm}) \equiv \int_0^{l_x} \int_0^1 \left[ \sum_{k=1}^K \sum_{n=1}^N B_{kn} \Delta u_{kn}(x, y) + \sum_{i=1}^I q_i \delta(x - x_i) \delta(y - y_i) \right] u_{lm}(x, y) dx dy = 0, \quad l = 1..K, m = 1..N$$

Расписав эти условия ортогональности более детально, получим следующую систему линейных уравнений для определения неизвестных коэффициентов  $B_{kn}$

$$\sum_{k=1}^K \sum_{n=1}^N A_{lmkn} B_{kn} = F_{lm}, \quad l = 1..K, m = 1..N \quad (12)$$

где

$$A_{lmkn} = \left[ \left( \frac{\lambda_{xk}}{l_x} \right)^2 + \lambda_{yn}^2 \right] \int_0^{l_x} \cos\left(\frac{\lambda_{xl}x}{l_x}\right) \cos\left(\frac{\lambda_{xk}x}{l_x}\right) dx \int_0^1 \cos(\lambda_{ym}y) \cos(\lambda_{yn}y) dy$$

$$F_{lm} = - \sum_{i=1}^I q_i \int_0^{l_x} \int_0^1 \delta(x - x_i) \delta(y - y_i) \cos\left(\frac{\lambda_{xl}x}{l_x}\right) \cos(\lambda_{ym}y) dx dy = - \sum_{i=1}^I q_i \cos\left(\frac{\lambda_{xl}x_i}{l_x}\right) \cos(\lambda_{ym}y_i)$$

Интегралы в выражении  $A_{lmkn}$  легко вычисляются по формулам (на примере

$$f(x) = \cos\left(\frac{\lambda_{xl}x}{l_x}\right) \cos\left(\frac{\lambda_{xk}x}{l_x}\right)$$

$$\int_0^{l_x} f(x) dx = \begin{cases} \frac{l_x}{2} \left[ \frac{\sin(\lambda_{xl} + \lambda_{xk})}{(\lambda_{xl} + \lambda_{xk})} + \frac{\sin(\lambda_{xl} - \lambda_{xk})}{(\lambda_{xl} - \lambda_{xk})} \right], & l \neq k \\ \frac{l_x}{2} \left[ 1 + \frac{\sin(2\lambda_{xl})}{2\lambda_{xl}} \right], & l = k \end{cases}$$

При программной реализации задачи нахождения температурного поля логичнее перейти к новым массивам (использованы вложенные циклы):

$$B_m = B_{kn}, \quad m = (k = 1..K, (n = 1..N))$$

$$A_{lm} = A_{lmkn}, \quad l = (l = 1..K, (m = 1..N)); m = (k = 1..K, (n = 1..N))$$

$$F_l = F_{lm}, \quad l = (l = 1..K, (m = 1..N))$$

В этом случае систему линейных уравнений (12) можно переписать в привычном виде

$$\sum_{j=1}^{K*N} A_{ij} B_j = F_i, \quad i = 1..(K * N)$$

Заметим, что поскольку на каждой итерации задачи оптимизации параметров источников энергии в этой системе уравнений будет изменяться только правая часть, то неизвестный вектор коэффициентов разложения  $B_j$  рассчитывались через обратную матрицу системы  $A^{-1}$ . Эта обратная матрица определяется только один раз при вводе начальных параметров задачи. При таком подходе на каждой итерации оптимизации при расчете коэффициентов разложения температурного поля потребуется в  $(K * N)$  раз меньше операций по сравнению с методом исключения Гаусса.

Обратную матрицу можно получить из единичной с помощью тех же преобразований, которые приводят исходную матрицу к единичной. Обе преобразуемые матрицы можно хранить на месте исходной и обрабатывать по следующим формулам (для  $k$ -го преобразования,  $k = 1..n$ ,  $n$  - размерность матрицы)

$$A_{ij} = A_{ij} - \frac{A_{ik} A_{kj}}{A_{kk}}, \quad i, j = 1..n, i, j \neq k$$

$$A_{ik} = - \frac{A_{ik}}{A_{kk}}, \quad i = 1..n, i \neq k$$

$$A_{kj} = \frac{A_{kj}}{A_{kk}}, \quad j = 1..n, j \neq k$$

$$A_{kk} = \frac{1}{A_{kk}}$$

Для устойчивости алгоритма перед выполнением  $k$ -го преобразования следует производить выбор максимального по модулю элемента матрицы в правой  $k$ -ой подстроке и, если необходимо (такой элемент находится в  $j(>k)$ -ом столбце), переставлять  $j$  и  $k$  столбцы матрицы. Номера переставляемых столбцов необходимо сохранять, поскольку после окончания преобразований в полученной матрице придется выполнить последовательную перестановку строк соответствующих номеров, но в обратном порядке.

#### 7.6. Решение задачи оптимизации.

Сделаем несколько пояснений о порядке расчета составляющих функционала (4):

- входящие в функционал интегралы над прямоугольной областью вычисляются обобщенным методом трапеций с использованием прямоугольной сетки по формуле

$$\frac{1}{S} \int_S f(x, y) dx dy = \frac{1}{4N_x N_y} (\sum f_{угл} + 2\sum f_{гран} + 4\sum f_{внутр})$$

где  $N_x, N_y$  - число узлов сетки вдоль соответствующих осей координат;

$f_{угл}, f_{гран}, f_{внутр}$  - значения функции в угловых, граничных и внутренних точках сетки, построенной над областью

- максимальные значения функций над областью ищутся на той же сетке.
- Для решения задачи минимизации функционала предлагаются три метода:
- метод покоординатного спуска (МПС)
  - метод наискорейшего спуска (МНС)
  - метод деформируемого многогранника (МДМ) [7].

Использование нескольких методов дает определенную свободу в выборе пути поиска оптимальных параметров точечных источников энергии, а также позволяет сравнить эффективность этих методов в решении поставленной задачи.

#### Алгоритм МПС

В МПС одномерный спуск выполняется последовательно вдоль каждого параметра оптимизации. При этом можно выбрать один из четырех вариантов перебора параметров:

- по источникам -  $q_i, x_i, y_i$
- по источникам -  $x_i, y_i, q_i$
- по координатам всех источников, а затем по их мощностям -  $x_1, y_1, \dots, x_I, y_I; q_1, \dots, q_I$
- по мощностям всех источников, а затем по их координатам -  $q_1, \dots, q_I; x_1, y_1, \dots, x_I, y_I$

Одномерный спуск состоит из двух этапов: определение направления спуска вдоль данного параметра  $x$  (направление уменьшения значения функционала) и спуск в сторону

выбранного направления (аналог затухающих колебаний шарика при скатывании на дно «лунки»). Приведем их алгоритмы:

1.

```

fl = 1,  $x_t = x$ ,  $f_t = \phi(\dots, x_t, \dots)$ ,  $h = 0,001$  - шаг
A:  $x = x_t + h * fl$ 
if ( $x \leq x_s$  - граничное значение) then
     $f = \phi(\dots, x, \dots)$ 
    if ( $f < f_t$ ) then "выход" end if
end if
if ( $fl = -1$ ) then "спуск не возможен" else ( $fl = -1$ , goto A) end if

```

2.

```

h = 0,05 * fl - первоначальный шаг спуска
 $x_{\min} = x$ ,  $f_{\min} = \phi(\dots, x_{\min}, \dots)$ 
do while  $|h| > 0,001$ 
     $x = x + h$ 
    if ( $x > x_s$ ) then ( $h = x_s - (x - h)$ ,  $x = x_s$ ) end if
     $f = \phi(\dots, x, \dots)$ 
    if ( $f < f_{\min}$ ) then ( $f_{\min} = f$ ,  $x_{\min} = x$ ) else  $h = -h/3$  end if
end while

```

Здесь  $x_s$  - граничное значение параметра.

### Алгоритм МНС

В данном методе одномерный спуск из фиксированной точки  $\bar{x}$  пространства параметров в сторону уменьшения значения функционала осуществляется по направлению  $-\frac{\text{grad}(\phi(\bar{x}))}{|\text{grad}(\phi(\bar{x}))|}$ ,

где (с учетом ограничений выбирается  $\pm \partial x_i$ )

$$|\text{grad}(\phi(\bar{x}))| = \sqrt{\sum_i \left[ \frac{\partial \phi}{\partial x_i} \right]^2}, \quad \frac{\partial \phi}{\partial x_i} = \frac{\phi(\dots, x_i \pm \partial x_i, \dots) - \phi(\dots, x_i, \dots)}{\pm \partial x_i}$$

Спуск вдоль указанного направления выполняется по следующему алгоритму

```

h = 0,05 - первоначальный шаг спуска
 $\bar{x}_{\min} = \bar{x}$ ,  $f_{\min} = \phi(\bar{x}_{\min})$ 
do while  $h > 0,001$ 
    for i (цикл по всем параметрам)
        
$$x_i = x_{\min i} - h \frac{\frac{\partial \phi}{\partial x_i}}{|\text{grad}(\phi(\bar{x}))|}$$

        if ( $x_i > x_{si}$ ) then  $x_i = x_{si}$  end if
    end for i
     $f = \phi(\bar{x})$ 
    if ( $f < f_{\min}$ ) then ( $f_{\min} = f$ ,  $\bar{x}_{\min} = \bar{x}$ ) else  $h = h/3$  end if
end while

```

В данном алгоритме спуск осуществляется с постоянным шагом. При каждом переходе через минимальное значение функционала делается шаг назад, величина шага уменьшается в три раза и спуск продолжается в том же направлении. При нарушении

ограничения значение параметра приравнивается соответствующему граничному значению. При этом, конечно, искажается направление наискорейшего спуска.

В МПС и МНС выход из итерационного процесса оптимизации осуществляется согласно критерию  $|\phi_n - \phi_{n-1}| < eps$ , где  $n, eps$  - номер итерации и заданная точность (например, 0,001).

### Алгоритм МДМ

Первоначально в  $n$ - мерном пространстве параметров оптимизации возле исходной точки  $\bar{p}_0$  строится регулярный симплекс – правильный многогранник с вершинами ( $\bar{p}_0, \bar{p}_1, \dots, \bar{p}_n$ ), координаты которых представлены столбцами матрицы размерности  $n*(n+1)$

$$\begin{matrix} p_{01} & p_{01} \pm d_1 & p_{01} \pm d_2 & \dots \\ p_{02} & p_{02} \pm d_2 & p_{02} \pm d_1 & \dots \\ p_{03} & p_{03} \pm d_2 & p_{03} \pm d_2 & \dots \\ \dots & \dots & \dots & \dots \end{matrix}$$

Здесь  $d_1 = \frac{t}{n\sqrt{2}}(\sqrt{n+1} + n - 1)$ ,  $d_2 = \frac{t}{n\sqrt{2}}(\sqrt{n+1} - 1)$ ,  $t$  - задаваемый характерный размер

симплекса. В каждой строке матрицы используются знаки «+» или «-» в зависимости от условия выполнения ограничений, наложенных на параметры. На рис. 3 подобные симплексы построены для случая  $n = 2$ . Здесь в области допустимых значений параметров полностью расположен только закрашенный симплекс с координатами вершин

$$\begin{matrix} p_{01} & p_{01} - d_1 & p_{01} - d_2 \\ p_{02} & p_{02} - d_2 & p_{02} - d_1 \end{matrix}$$

Для этих же целей можно варьировать величину характерного размера симплекса.

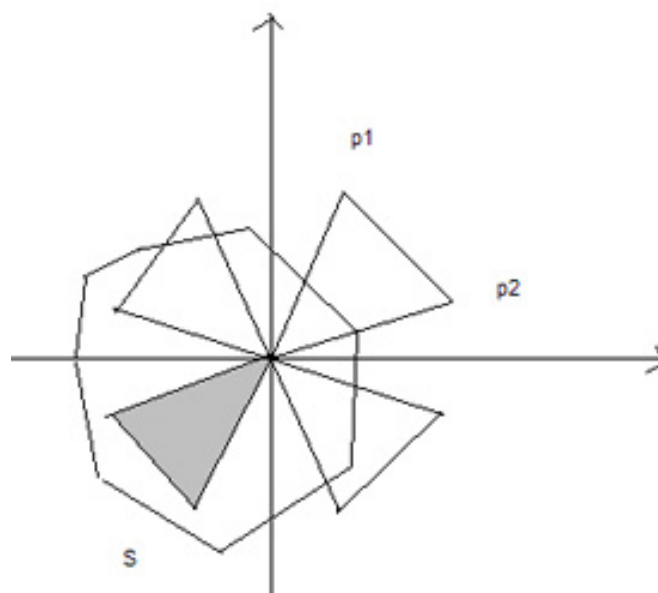


Рис. 3. Положение симплексов для двумерного случая

После построения исходного симплекса запускается итерационный процесс, во время которого на каждой итерации производится перестройка многогранника по следующему алгоритму.

1. Рассчитываем функционал во всех вершинах симплекса и определяем вершины с минимальным и максимальным значениями функционала -  $\bar{p}_l, \bar{p}_h$ .

2. Находим центр «тяжести» всех вершин многогранника за исключением точки  $\bar{p}_h$

$$\bar{p}_c = \frac{1}{n} \left( \sum_{i=1}^{n+1} \bar{p}_i - \bar{p}_h \right)$$

3. Строим новую вершину симплекса  $\bar{p}_{new}$  отражением точки  $\bar{p}_h$  относительно точки  $\bar{p}_c$  (см. рис. 4 – пример двумерного случая)

$$\bar{p}_{new} = \bar{p}_c + \alpha (\bar{p}_c - \bar{p}_h) \quad (13)$$

Здесь первоначально принимается  $\alpha = 1$ . Если новая вершина выходит за пределы ограничений, то начинаем постепенно уменьшать значение  $\alpha$  с шагом 0,1 до значения 0,5, после чего переходим на пункт 5 (т.к. дальнейшее уменьшение  $\alpha$  может привести к вырождению симплекса).

4. Рассчитываем функционал в новой точке и если  $\phi_{new} > \phi_h$ , то выполняем сжатие симплекса, при котором все вершины, кроме  $\bar{p}_l$ , пересчитываются по следующей формуле (см. рис. 4)

$$\bar{p}_i = \bar{p}_l + 0,5(\bar{p}_i - \bar{p}_l)$$

5. Если  $\alpha < 0,5$  или  $\phi_{new} > \phi_i$  для всех  $i \neq h$ , то строим отраженную точку внутри многогранника по формуле (13) с  $\alpha = -0,5$  (см. рис. 4 - отражение внутрь), отбрасываем точку  $\bar{p}_h$  и возвращаемся к пункту 1.

6. Если  $\alpha = 1$  и  $\phi_{new} < \phi_l$ , то выполняем дополнительное внешнее растяжение по формуле (2.13) с  $\alpha = \alpha + \delta$ . При этом первоначально берем  $\delta = 0,5$ , а в случае нарушения ограничений последовательно уменьшаем его наполовину и еще раз наполовину, а затем отказываемся от дополнительного растяжения.

7. Заменяем точку  $\bar{p}_h$  отраженной  $\bar{p}_{new}$  и возвращаемся к пункту 1.

Перед началом новой итерации проверяем выполнение критерия окончания итерационного процесса  $|\phi_h - \phi_l| < eps$ .

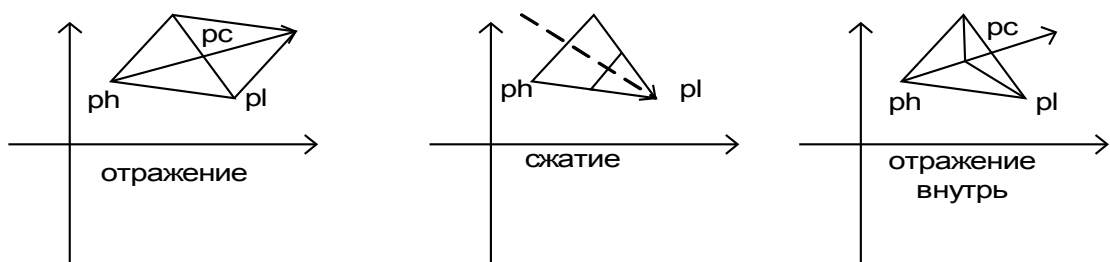


Рис. 4. Варианты изменения многогранника для двумерного случая

## 7.7. Графическое представление результатов расчета температурного поля

Для качественной оценки формы поверхности полученного температурного поля используется его графическое представление, как в виде трехмерной поверхности, так и в виде линий уровня.

Трехмерная поверхность формируется кусочно-непрерывными линиями среза  $T(x, y_i)$ ;  $x \in (0, l_x)$ ,  $y_i \in (0, 1)$ ;  $y_i = 0,05i, i = 0 \div 20$ . При построении кусочков линий шаг по оси  $x$  принимается равным одному пикселю. Линии строятся последовательно от переднего среза к дальнему с учетом их видимости - при отображении очередного дальнего среза отсекаются его отрезки, перекрываемые уже построенной поверхностью. Для этого используются два массива  $y_{\max}(j)$ ,  $y_{\min}(j)$ ,  $j = 0 \div m$ ,  $m$  - максимальное число пикселей по оси  $x$ , в которых хранятся и постоянно корректируются граничные верхние и нижние значения экранной координаты  $y$  формируемой поверхности. На рис. 5 показан порядок проведения линии между двумя пикселями ( $i$  и  $i+1$ ) с учетом их видимости

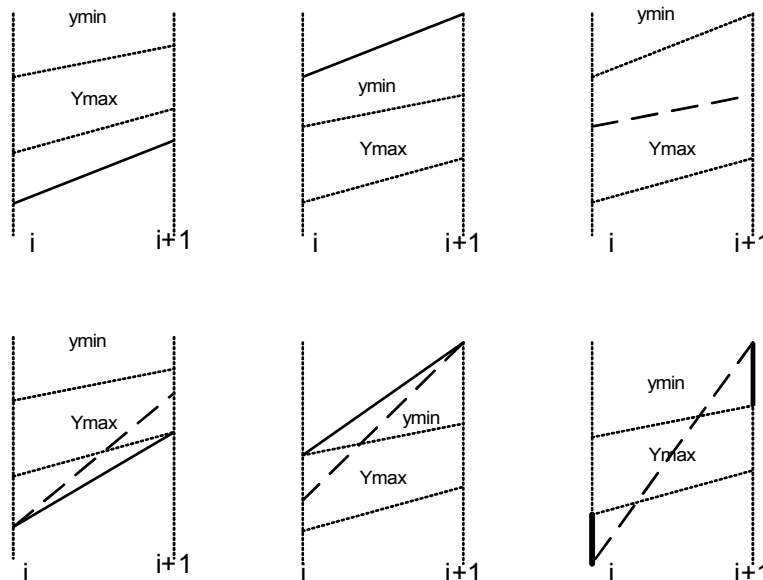


Рис. 5. Правила построения линии между двумя пикселями с учетом их видимости



Рис. 6. Положение координат трехмерного пространства на плоском экране

При построении линий пересчет координат трехмерного пространства в координаты экрана выполняется по следующим формулам (см. рис. 6)

$$x_{\text{экр}} = X_{\text{экр}} - xK_x + yK_y, \quad y_{\text{экр}} = Y_{\text{экр}} + yK_y - T(x, y)K_t$$

где  $K_x, K_y, K_t$  - коэффициенты масштабирования,  $X_{\text{экр}}, Y_{\text{экр}}$  - координаты начала изометрической системы координат на экране.

Линии уровня могут формироваться двумя способами: точками и линейными отрезками.

В первом случае плоскость  $x, y$  покрывается прямоугольной сеткой и определяются точки пересечения плоскости уровня  $T(x, y) = const$  с каждой стороной ячейки сетки. При этом над каждой стороной сетки функция  $T(x, y)$  заменяется линейным интерполяционным полиномом с использованием ее значений в соответствующих узлах сетки (см. рис. 7, где демонстрируется, как вычисляется координата  $x$  точки пересечения).

Во втором случае каждая ячейка сетки дополнительно делится диагоналями на четыре треугольника и над каждым из них функция  $T(x, y)$  интерполируется плоскостью  $T(x, y) = a + bx + cy$  с использованием значений функции в вершинах треугольника.

Далее находятся точки пересечения сторон треугольника с плоскостью уровня (см. рис. 8). Затем эти точки соединяются линиями.

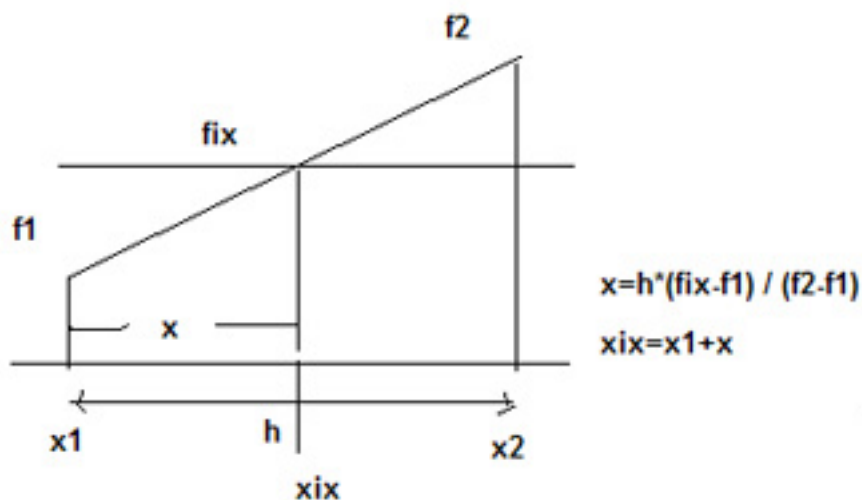


Рис. 7. Порядок вычисления координаты точки пересечения  $x$

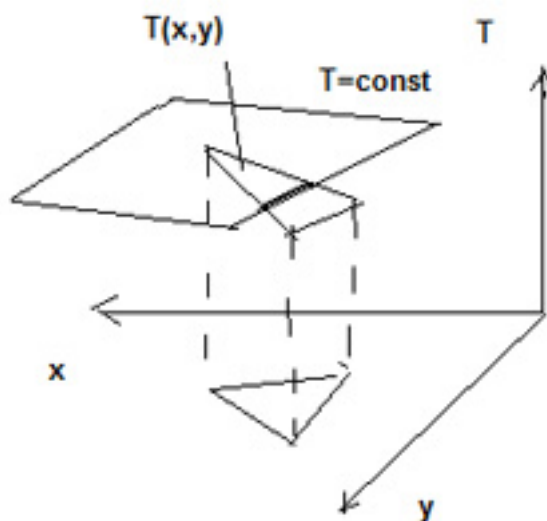


Рис. 8. Построения линии уровня  $T(x, y) = const$  с использованием треугольников

При построении уровней специально обрабатываются «сингулярные» варианты, когда при первом способе линия уровня совпадает со звеном сетки, а при втором способе – со стороной треугольника, со всеми сторонами или с одной из его вершин. Следует обратить внимание, что при таком подходе плоские участки уровня покрываются прямоугольниками (первый способ) или треугольниками (второй способ).

## 7.8. Программная реализация моделирующего комплекса

Программа для решения поставленной задачи реализована в виде нескольких, связанных между собой задач, каждая со своим интерфейсным окном. На рис. 9 изображено главное окно программного комплекса.

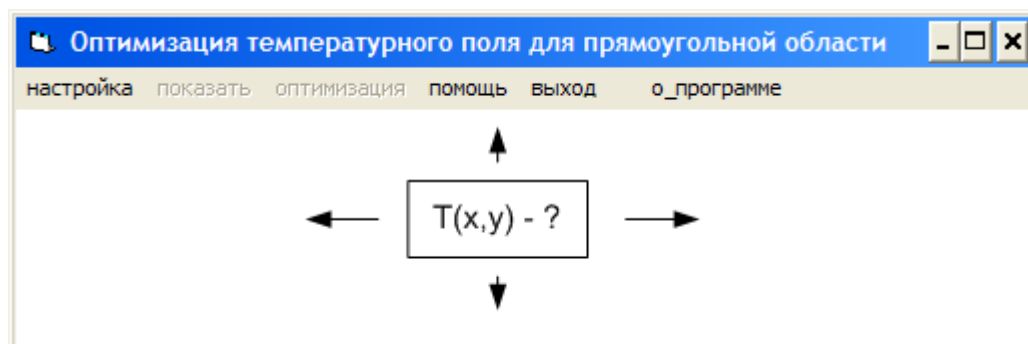


Рис. 9. Главное окно программного комплекса

Работа с программой начинается с «настройки» параметров модели и предварительного расчета температурного поля для заданных параметров источников энергии (рис. 10).

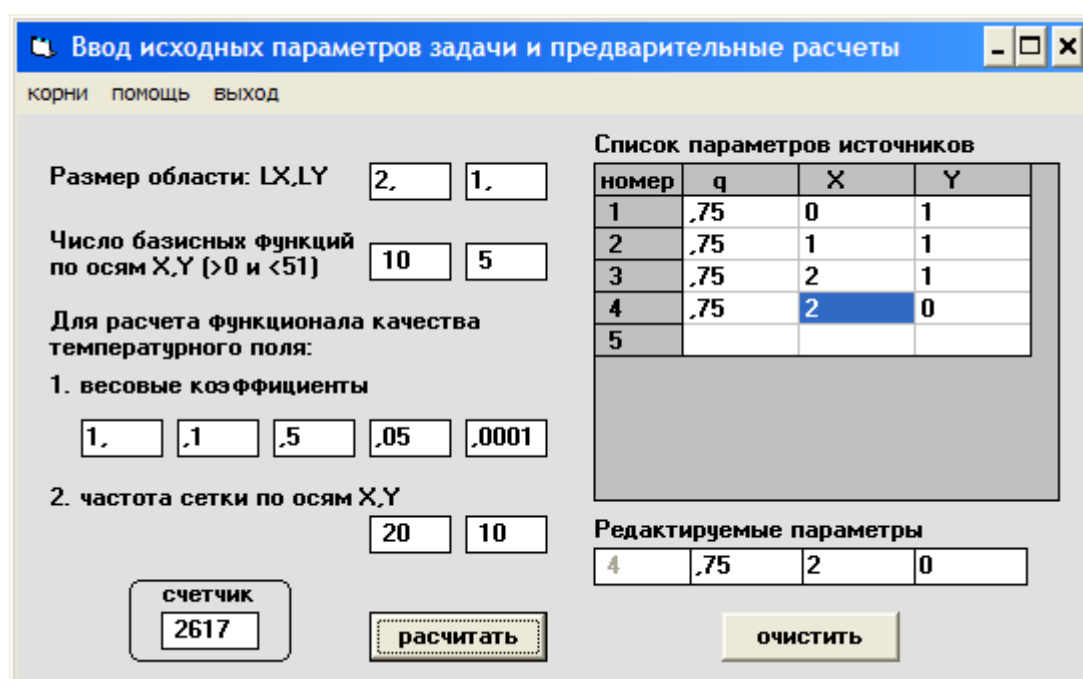


Рис. 10. Окно «настройки» параметров модели и предварительного расчета температурного поля

Из этого окна с помощью пункта меню «корни» можно загрузить программу нахождения собственных чисел  $\lambda_{xi}$ ,  $\lambda_{yi}$ , фигурирующих в разложении температурного поля (рис.11). Эти числа можно сохранить в соответствующих файлах для последующего использования или ввести из файлов для просмотра. Всюду ниже принималось  $h = 1$ .



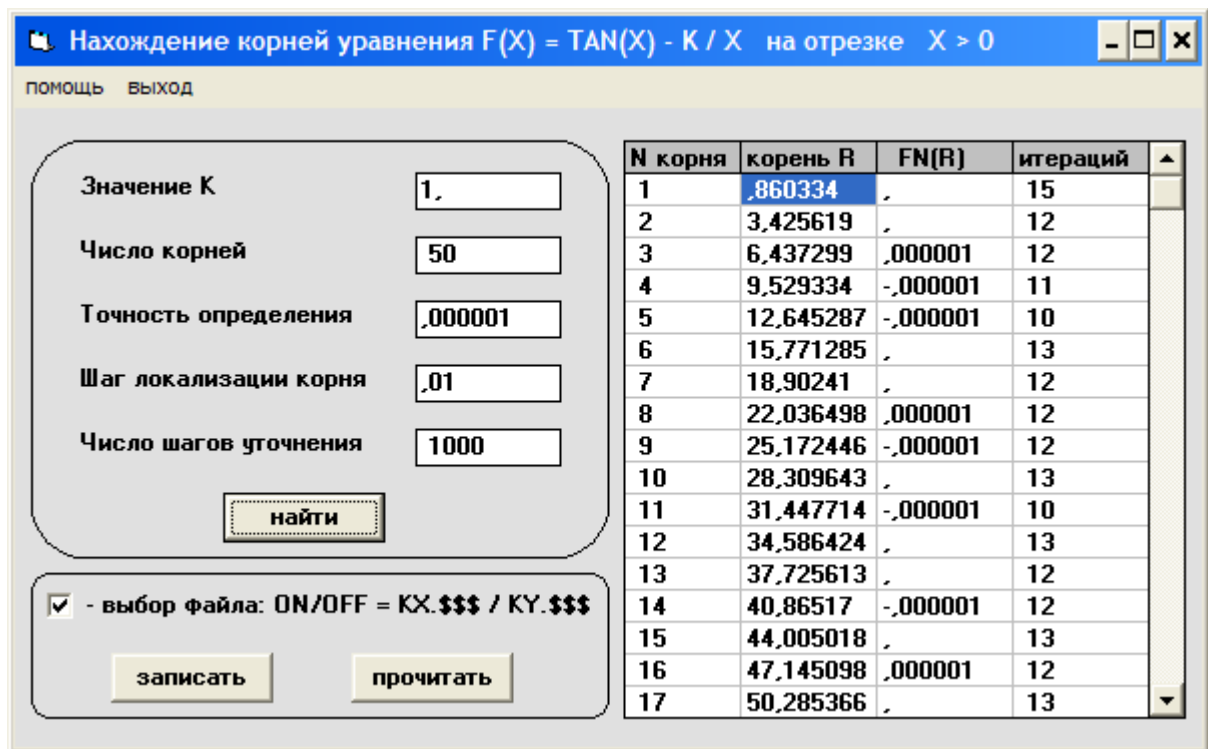


Рис. 11. Окно программы нахождения корней нелинейных уравнений

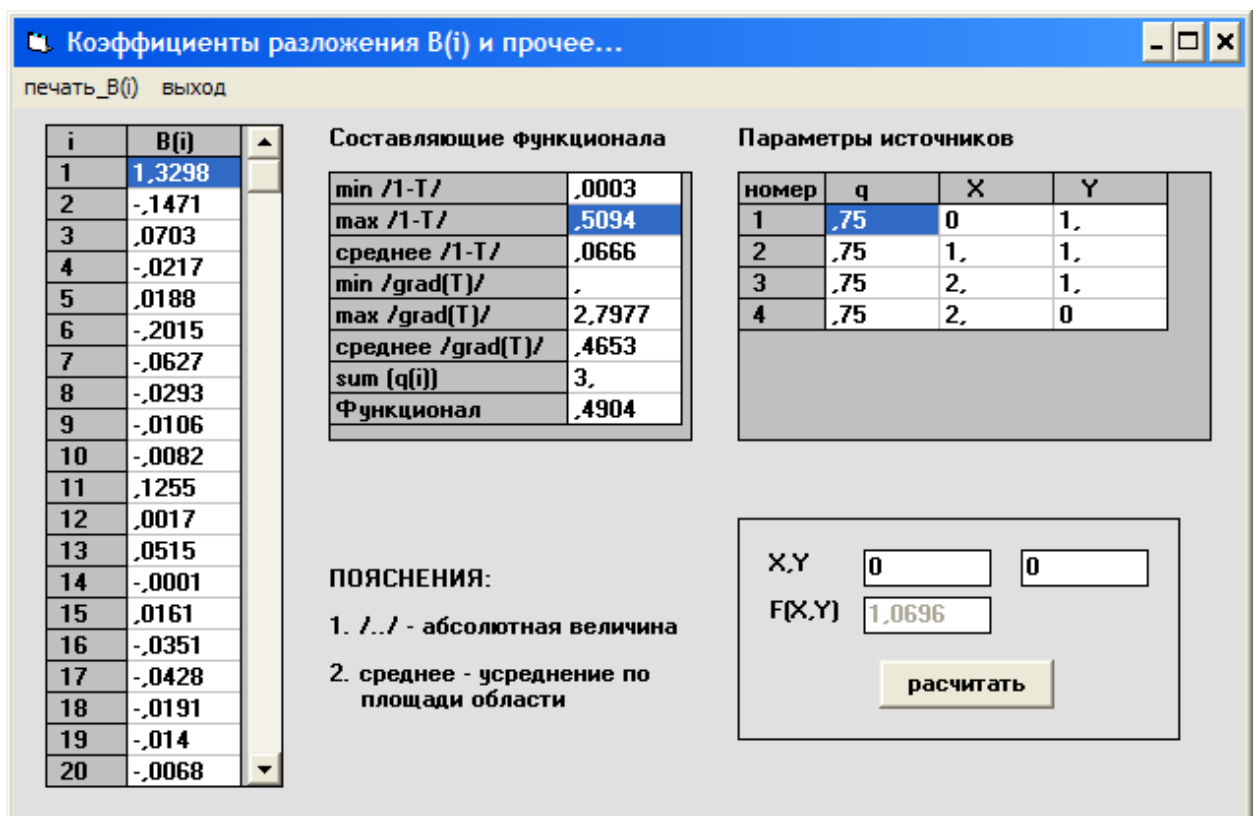


Рис. 12. Окно для просмотра результатов расчета

Подменю пункта меню «показать» основного окна (рис. 9) позволяют: просмотреть значения коэффициентов разложения и оценить скорость сходимости ряда, получить составляющие функционала и рассчитать температуру в любой точке области (рис. 12);

построить трехмерную поверхность для  $T(x, y)$  и других, связанных с этим поле функций (рис. 13);

сформировать линии уровня для рассчитанного температурного поля (рис. 14).

Задача оптимизации температурного поля вызывается пунктом меню «оптимизация». Ее окно представлено на рис. 15. Результат оптимизации первоначально заданного поля (см. рис. 10, 12, 13) приведены на рис. 16, 17, 18.

На рис. 19 видно как изменится гладкость и увеличится значение функционала (за счет градиентных составляющих) оптимизированного температурного поле, если добавить в его разложение дополнительное число высокочастотных слагаемых (было  $10 \times 5$ , стало  $20 \times 10$ ).

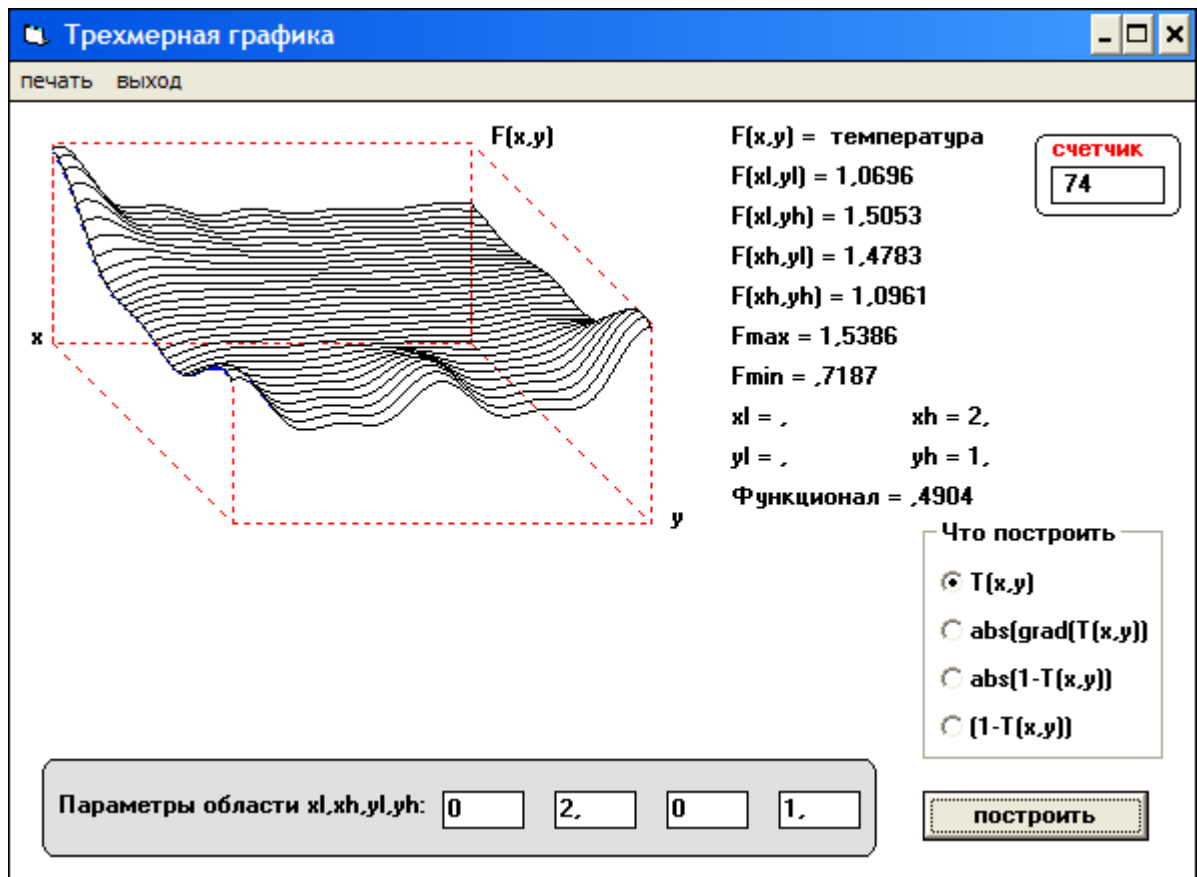


Рис. 13. Пример построения трехмерной поверхности для  $T(x, y)$

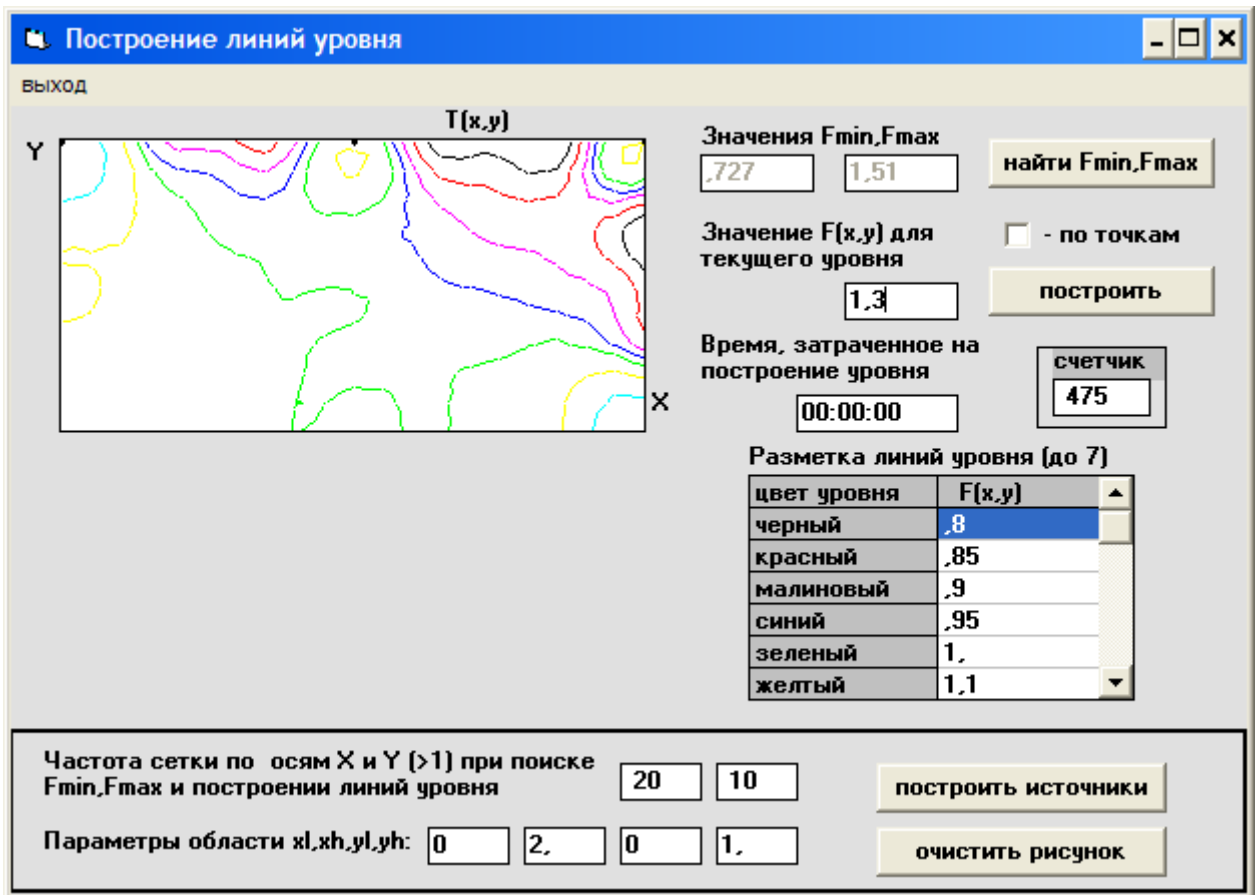


Рис. 14. Пример построения линий уровня для поверхности  $T(x, y)$

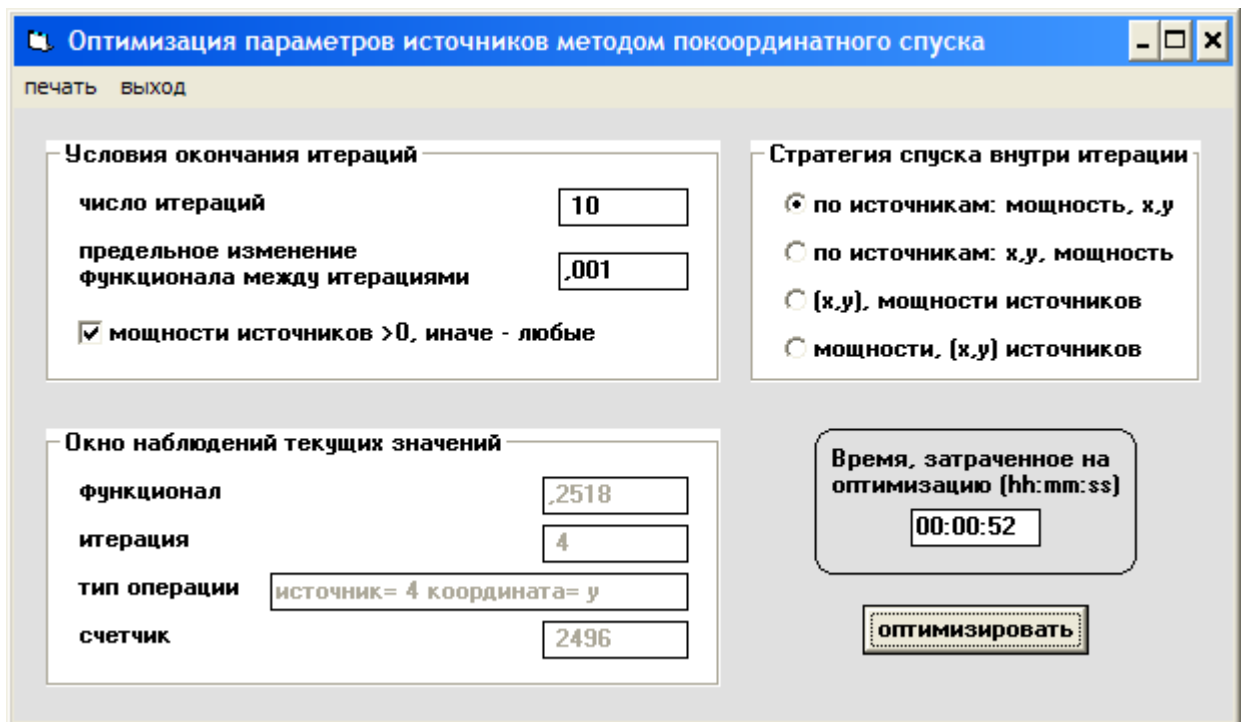


Рис. 15. Окно для решения задачи оптимизации

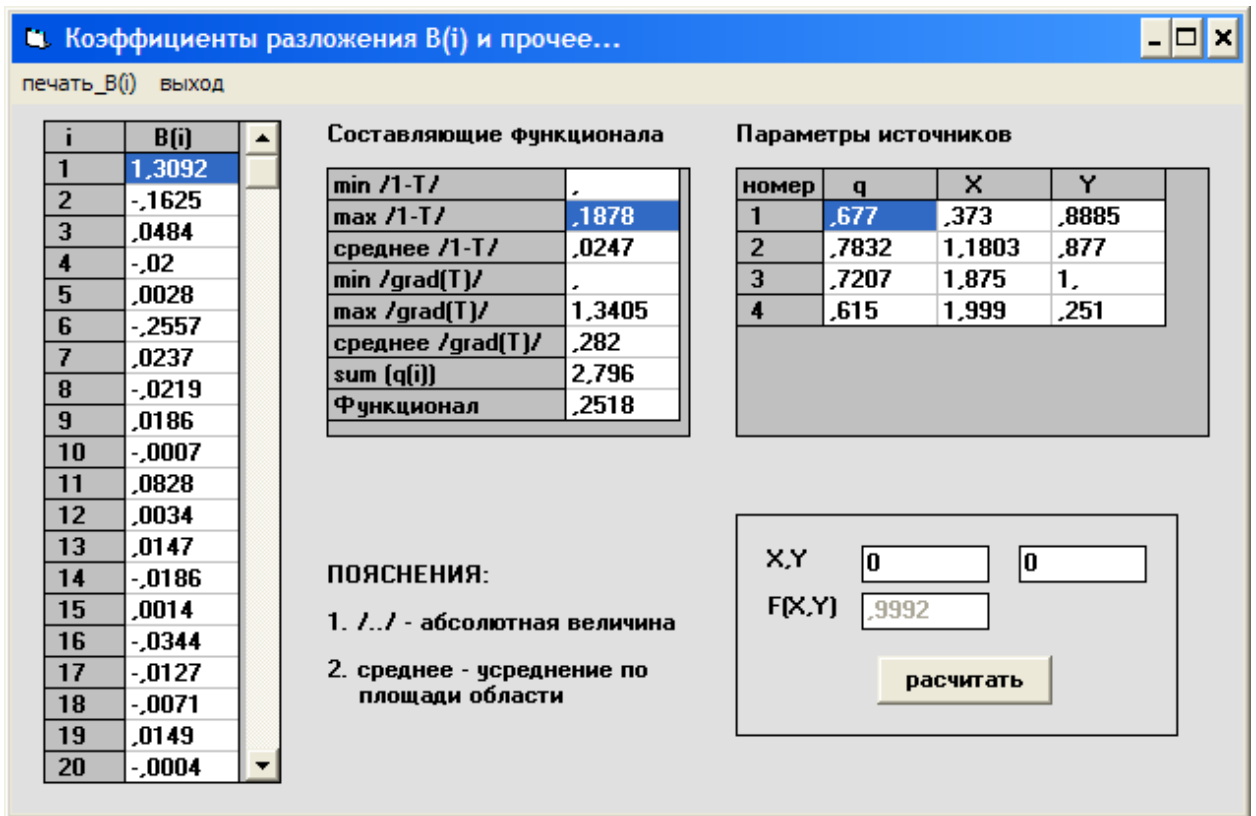


Рис. 16. Просмотр результатов оптимизации

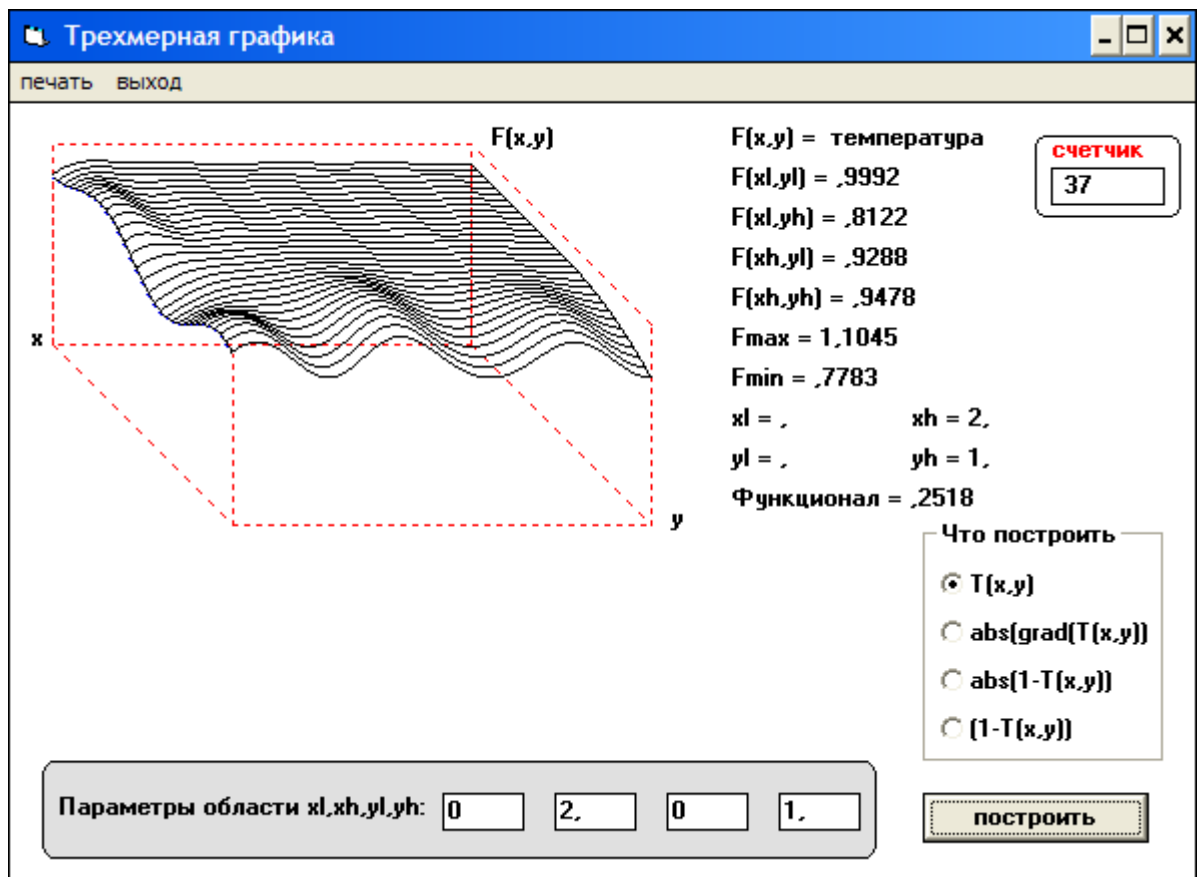


Рис. 17. Поверхность температурного поля после оптимизации

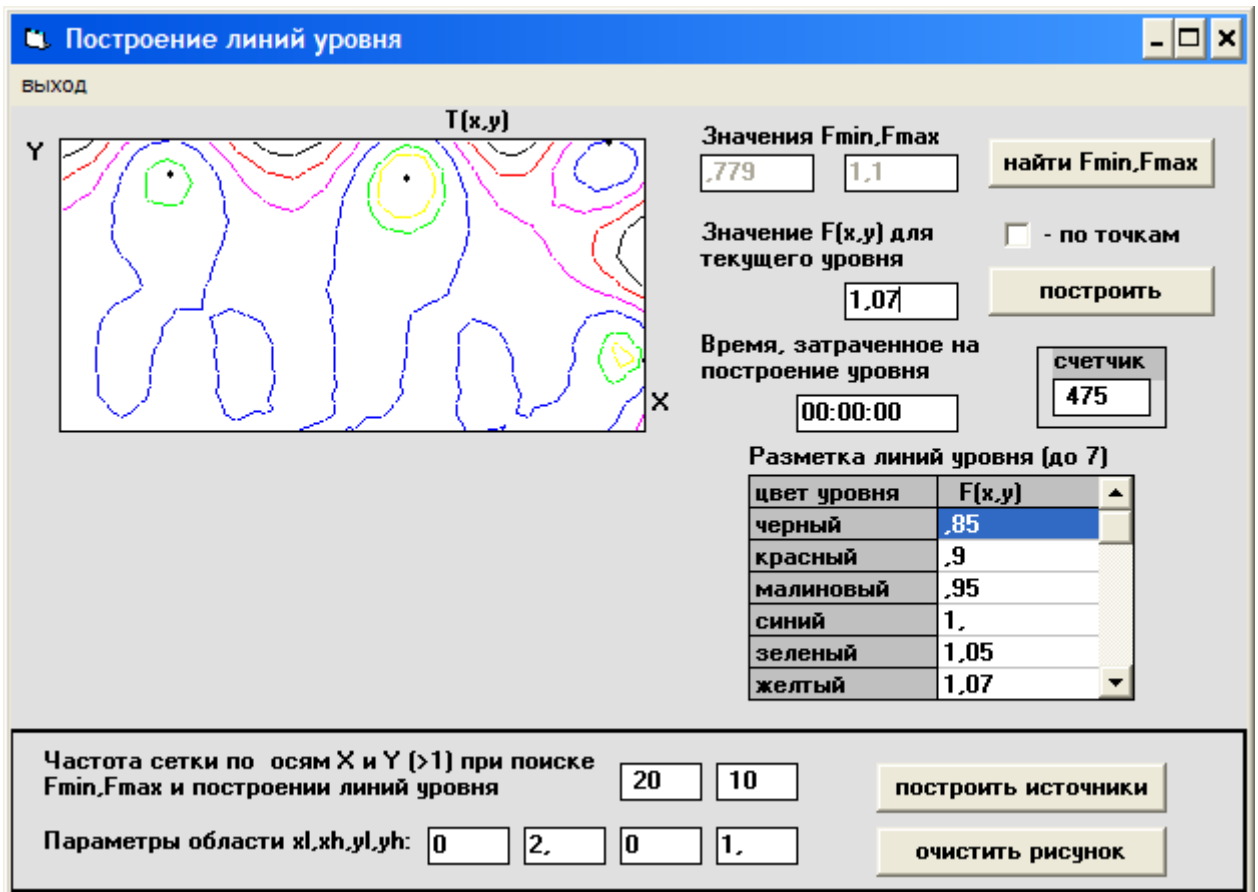


Рис. 18. Линии уровня для температурного поля после оптимизации

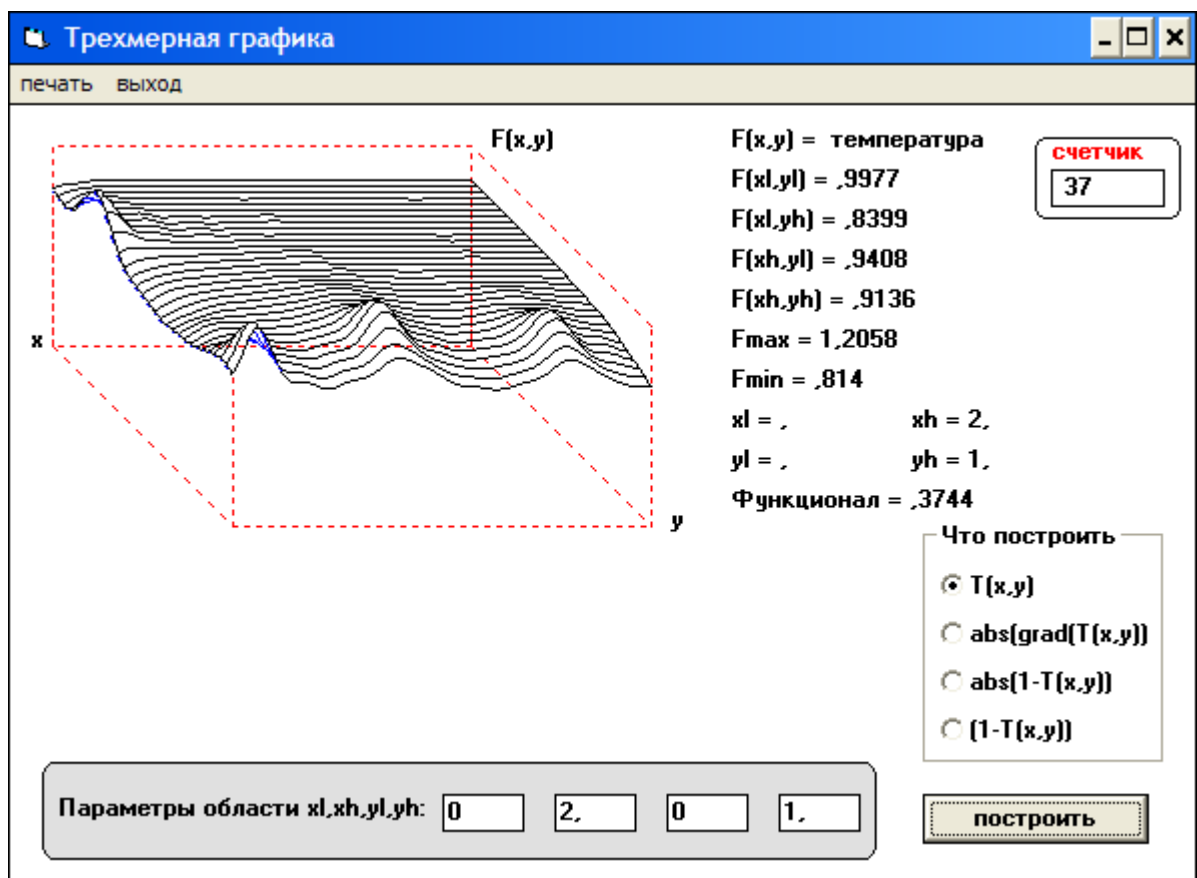


Рис. 19. Вид поверхности температурного поля после увеличения числа слагаемых в разложении

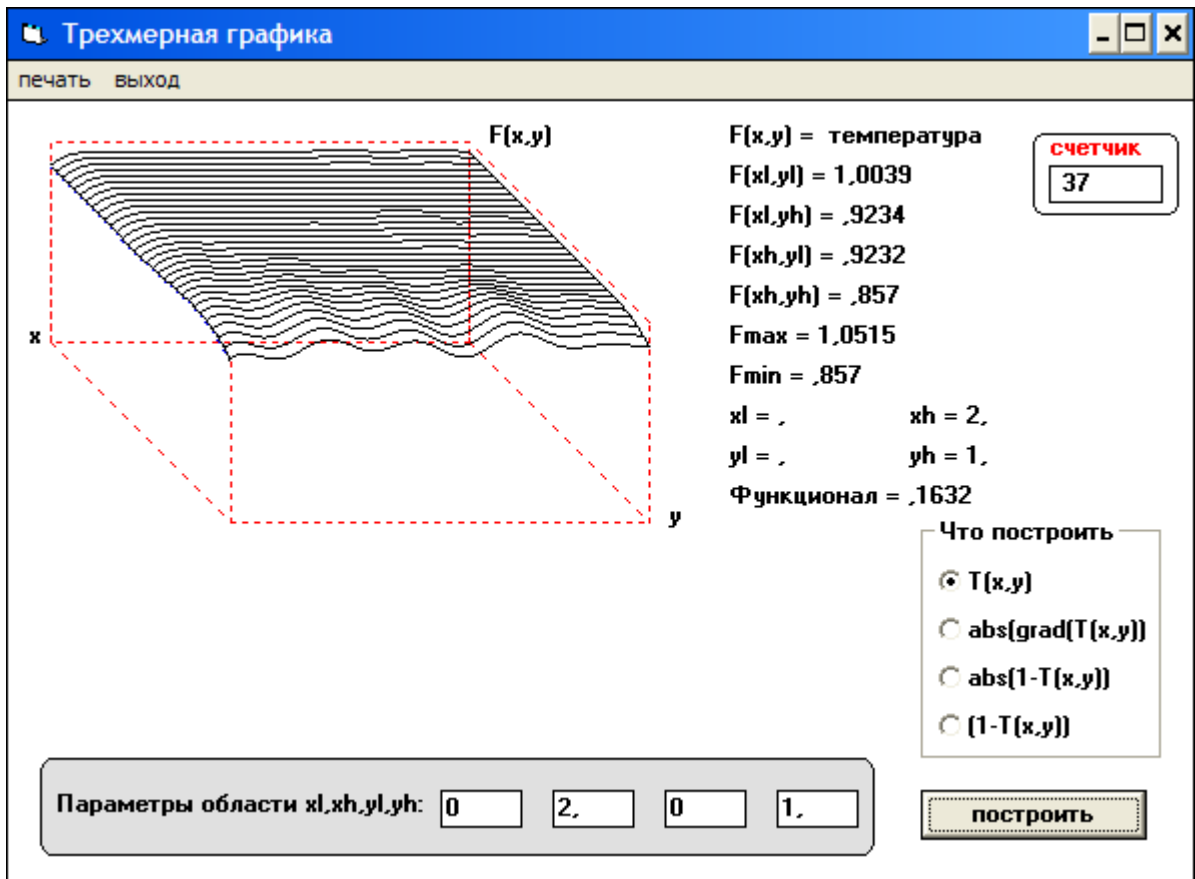


Рис. 20. Поверхность  $T(x,y)$  после оптимизации для семи источников

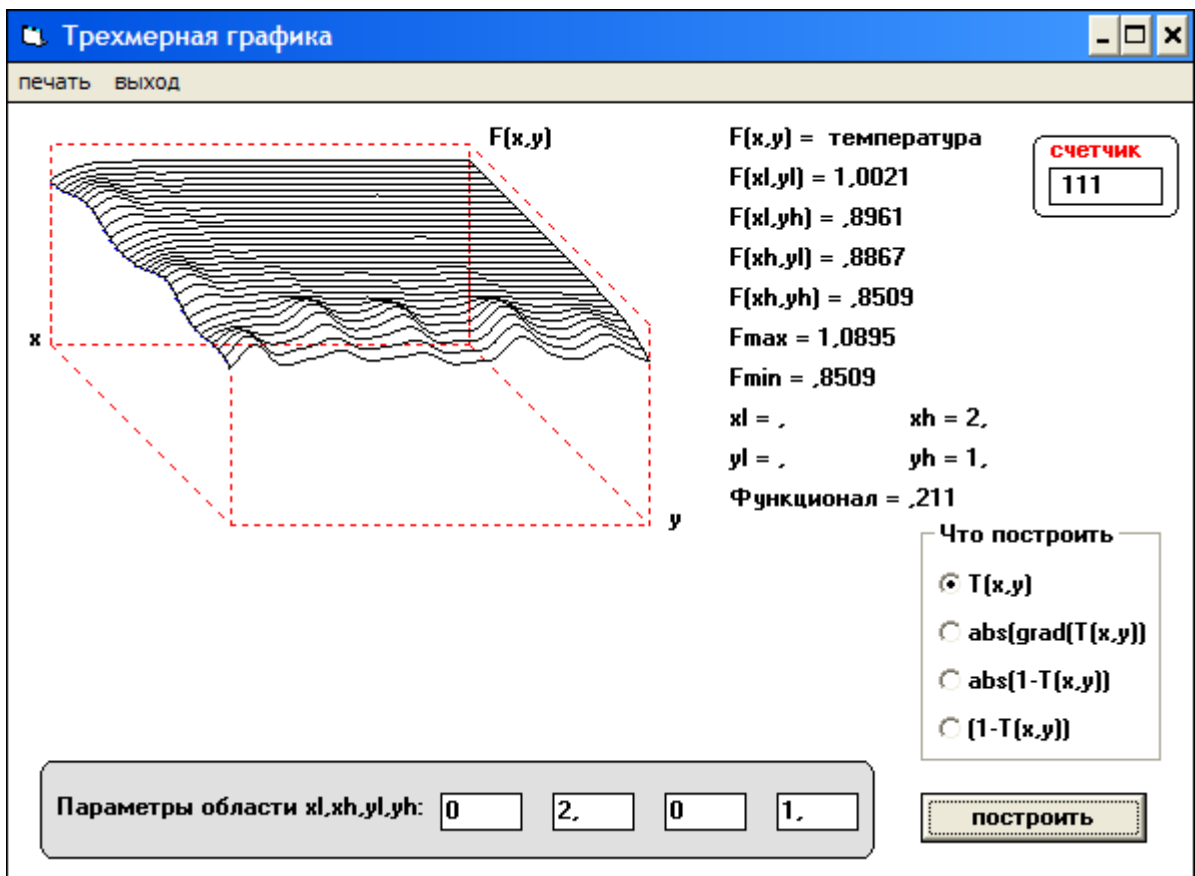


Рис. 21 Вид поверхности ранее оптимизированного поля  $T(x, y)$  для семи источников после увеличения числа слагаемых в разложении

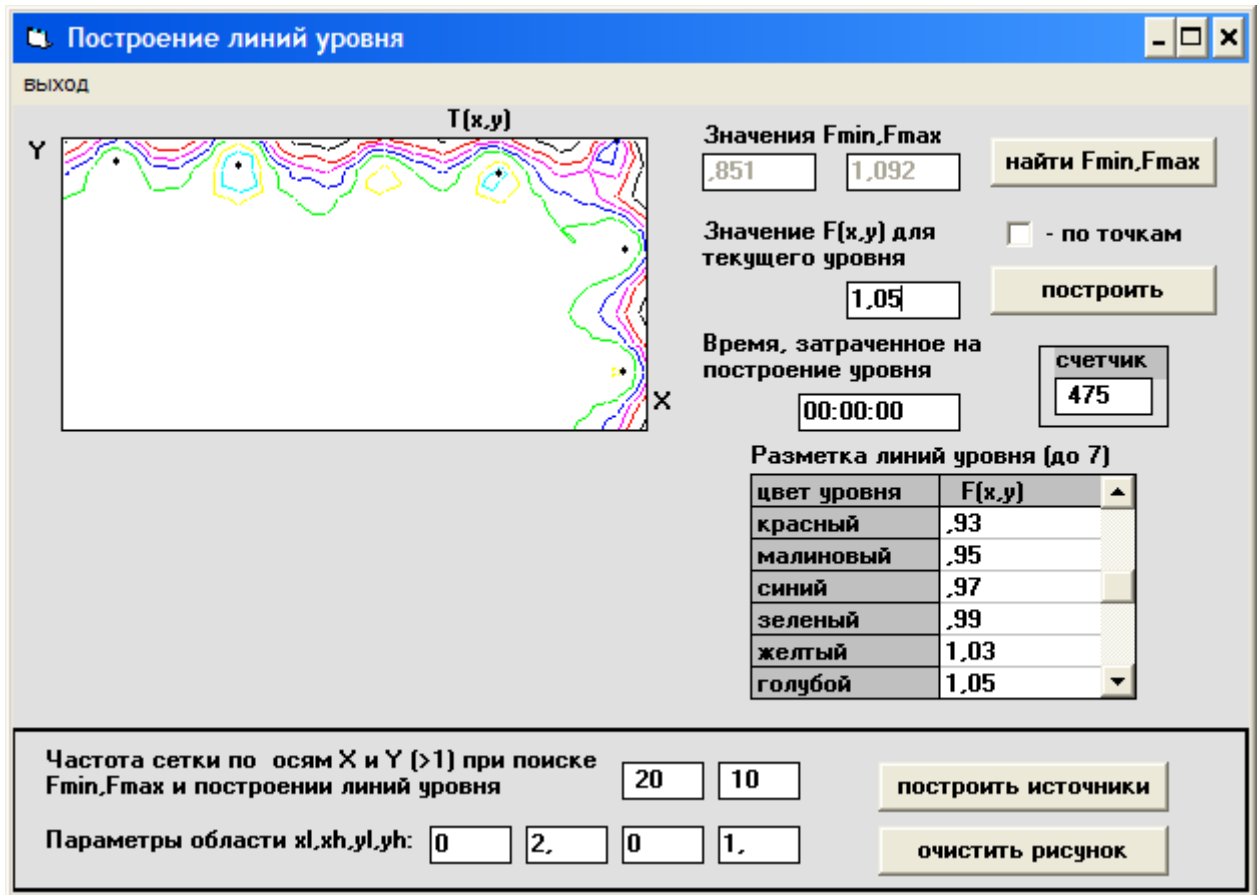


Рис.22. Показано оптимальное положение семи источников

## 7.9. Результаты исследования модели

Исследования, проведенные с помощью построенной модели и ее программной реализации, показали:

оптимизация температурного поля в предложенной постановке возможна (сравните рис. 13 и рис. 17);

методы оптимизации МНС и МДМ более трудоемкие по сравнению с МПС, а в качестве оптимизации не дают ни каких выгод;

чем больше слагаемых используется в разложении для аппроксимации поля, тем выше значения градиентных составляющих в функционале качества оптимизации и более четко обозначаются на поверхности поля местоположение источников (см. рис. 17 и 19 – четыре источника; рис. 19 и 20 – семь источников; в обоих случаях первоначально было  $10 \times 5$  слагаемых, а затем -  $20 \times 10$ );

увеличение числа источников позволяет улучшить качество оптимизированного температурного поля (функционал для случая четырех источников (рис. 19) равен 0,3744, для семи (рис. 21) – 0,211);

после оптимизации всегда выполняется инвариант (7) относительно суммарной мощности всех источников, что позволяет доверять полученным результатам;

если при первоначальном задании параметров источников руководствоваться рекомендациями параграфа 7.4, то итерационный процесс оптимизации сходиться достаточно быстро (за 6 итераций). Рассчитанное оптимальное положение четырех и семи источников показано, соответственно, на рис. 18 и рис. 22.

## 8. Построение оптимального температурного поля для плоской усеченной прямоугольной области.

### 8.1. Формулировка задачи.

Рассматривается задача, сформулированная в разделе 7, но для усеченной прямоугольной области (рис. 1).

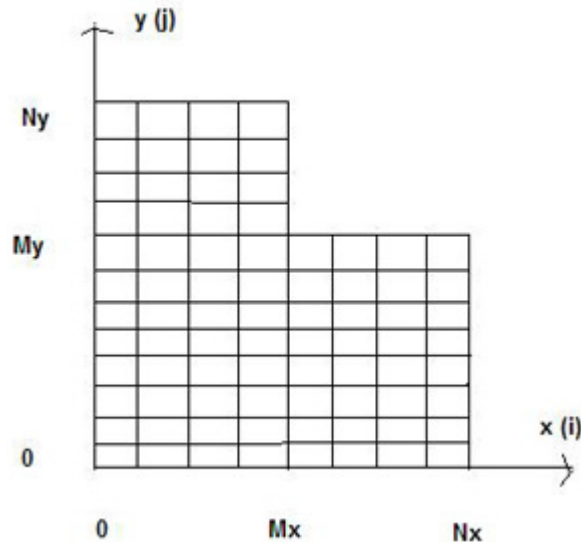


Рис.1. Форма усеченной прямоугольной области

### 8.2. Математическая постановка задачи.

В данном случае разложение (11 из раздела 7) не удовлетворяет граничным условиям на усеченных участках правой и верхней сторонах области. Поэтому решение температурной задачи целесообразно искать разностным методом [9]. Для этого область покрывается прямоугольной сеткой, а уравнение стационарной теплопроводности и граничные условия (8 из раздела 7) записываются через значения температуры и мощности источников в узлах сетки (используются безразмерные величины из раздела 7) следующим образом:

а. Уравнение теплопроводности (применяется только для внутренних узлов сетки)

$$\frac{T_{i+1j} - 2T_{ij} + T_{i-1j}}{h_x^2} + \frac{T_{ij+1} - 2T_{ij} + T_{ij-1}}{h_y^2} = -q_{ij} \quad (1)$$

При переходе от непрерывной модели к дискретной предполагается, что источники размещаются только в узлах сетки, а их мощности в непрерывной и дискретной моделях связаны между собой следующим образом (точечный источник равномерно «размазывается» вокруг узла по прямоугольной области размера  $s = h_x h_y$ )

$$q_{ij}^{непр} = \int_s q_{ij}^{непр} \delta(x - x_i) \delta(y - y_j) dx dy = \int_s q_{ij}^{дискр} dx dy = q_{ij}^{дискр} h_x h_y$$

где  $h_x = \frac{l_x}{N_x}$ ,  $h_y = \frac{1}{N_y}$  - шаги сетки по соответствующим осям.

б. Граничные условия



Границы теплоизолированные ( $\frac{\partial T}{\partial n} = 0$ ,  $n$  - внешняя нормаль):

- на левой стороне области:  $T_{0j} = T_{1j} \quad j = 1..N_y$
- на нижней стороне области:  $T_{i0} = T_{i1} \quad i = 1..N_x$

Границы с теплопотерями ( $\frac{\partial T}{\partial n} = -hT$ ):

на нижней части правой стороны области:

$$\frac{T_{N_x j} - T_{N_x - 1 j}}{h_x} = -hT_{N_x j} \Rightarrow T_{N_x j} = T_{N_x - 1 j} \frac{1}{(1 + hh_x)}, \quad j = 1..M_y - 1$$

$$\frac{T_{N_x M_y} - T_{N_x - 1 M_y - 1}}{\sqrt{h_x^2 + h_y^2}} = -hT_{N_x M_y} \Rightarrow T_{N_x M_y} = T_{N_x - 1 M_y - 1} \frac{1}{(1 + h\sqrt{h_x^2 + h_y^2})}$$

на левой части верхней стороны области:

$$T_{i N_y} = T_{i N_y - 1} \frac{1}{(1 + hh_y)}, \quad i = 1..M_x - 1$$

$$T_{M_x N_y} = T_{M_x - 1 N_y - 1} \frac{1}{(1 + h\sqrt{h_x^2 + h_y^2})}$$

на правой части верхней стороны области:

$$T_{i M_y} = T_{i M_y - 1} \frac{1}{(1 + hh_y)}, \quad i = M_x + 1..N_x - 1$$

$$T_{M_x M_y} = T_{M_x - 1 M_y - 1} \frac{1}{(1 + h\sqrt{h_x^2 + h_y^2})}$$

на верхней части правой стороны области:

$$T_{M_x j} = T_{M_x - 1 j} \frac{1}{(1 + hh_x)}, \quad j = M_y + 1..N_y - 1$$

#### в. Критерий оптимальности температурного поля

Используется тот же критерий в форме 4 из раздела 7. В нем все интеграла по области заменяются суммированием по всем внутренним узлам сетки с последующим делением результата суммирования на число указанных точек. Максимальные значения функций ищутся на тех же узлах, а градиент температуры в узле  $i, j$  вычисляется по следующей формуле

$$|\text{grad}(T_{ij})| = \sqrt{\left(\frac{T_{i+1j} - T_{i-1j}}{2h_x}\right)^2 + \left(\frac{T_{ij+1} - T_{ij-1}}{2h_y}\right)^2}$$

г. Ограничения на параметры источников:  $q_{ij} > 0$ ;  $i, j$  - внутренние узлы сетки и в узле может размещаться только один источник.

#### 8.3. Решение температурной задачи

Преобразовав с помощью граничных условий разностные уравнения (1) для приграничных узлов и разрешив все имеющиеся линейные уравнения системы относительно диагональных элементов разреженной матрицы, получим систему уравнений, которую будем решать итерационным методом Зейделя. Ниже приведен алгоритм ее решения. Принимается следующий порядок обхода узлов сетки: слой - за

слоем, снизу – вверх; при этом учитывается возможное вырождение усеченной области в чисто прямоугольную, когда  $M_x = N_x, M_y = N_y$ .

В пределах одной итерации выполняются следующие действия:

1. Пересчет температуры во внутренних точках

$$\begin{aligned}
 & T_{11} = g_1 T_{21} + g_2 T_{12} + g_3 q_{11} \\
 & \text{for } i = 2, N_x - 2 \quad \{ T_{i1} = b_1 (T_{i+11} + T_{i-11}) + b_2 T_{i2} + b_3 q_{i1} \} \\
 & T_{N_x-11} = d_1 T_{N_x-21} + d_2 T_{N_x-12} + d_3 q_{N_x-11} \\
 & \text{for } j = 2, M_y - 2 \\
 & \{ \\
 & \quad T_{1j} = c_1 T_{2j} + c_2 (T_{1j+1} + T_{1j-1}) + c_3 q_{1j} \\
 & \quad \text{for } i = 2, N_x - 2 \quad \{ T_{ij} = a_1 (T_{i+1j} + T_{i-1j}) + a_2 (T_{ij+1} + T_{ij-1}) + a_3 q_{ij} \} \\
 & \quad T_{N_x-1j} = e_1 T_{N_x-2j} + e_2 (T_{N_x-1j+1} + T_{N_x-1j-1}) + e_3 q_{N_x-1j} \\
 & \} \\
 & \text{if } M_x \neq N_x \text{ then} \\
 & \{ \\
 & \quad T_{1M_y-1} = c_1 T_{2M_y-1} + c_2 (T_{1M_y} + T_{1M_y-2}) + c_3 q_{1M_y-1} \\
 & \quad \text{for } i = 2, M_x - 1 \quad \{ T_{iM_y-1} = a_1 (T_{i+1M_y-1} + T_{i-1M_y-1}) + a_2 (T_{iM_y} + T_{iM_y-2}) + a_3 q_{iM_y-1} \} \\
 & \quad \text{for } i = M_x, N_x - 2 \quad \{ T_{iM_y-1} = k_1 (T_{i+1M_y-1} + T_{i-1M_y-1}) + k_2 T_{iM_y-2} + k_3 q_{iM_y-1} \} \\
 & \quad T_{N_x-1M_y-1} = z_1 T_{N_x-2M_y-1} + z_2 T_{N_x-1M_y-2} + z_3 q_{N_x-1M_y-1} \\
 & \quad \text{for } j = M_y, N_y - 2 \\
 & \quad \{ \\
 & \quad \quad T_{1j} = c_1 T_{2j} + c_2 (T_{1j+1} + T_{1j-1}) + c_3 q_{1j} \\
 & \quad \quad \text{for } i = 2, M_x - 2 \quad \{ T_{ij} = a_1 (T_{i+1j} + T_{i-1j}) + a_2 (T_{ij+1} + T_{ij-1}) + a_3 q_{ij} \} \\
 & \quad \quad T_{M_x-1j} = e_1 T_{M_x-2j} + e_2 (T_{M_x-1j+1} + T_{M_x-1j-1}) + e_3 q_{M_x-1j} \\
 & \quad \} \\
 & \} \\
 & \quad T_{1N_y-1} = u_1 T_{2N_y-1} + u_2 T_{1N_y-2} + u_3 q_{1N_y-1} \\
 & \quad \text{for } i = 2, M_x - 2 \quad \{ T_{iN_y-1} = k_1 (T_{i+1N_y-1} + T_{i-1N_y-1}) + k_2 T_{iN_y-2} + k_3 q_{iN_y-1} \} \\
 & \quad T_{M_x-1N_y-1} = z_1 T_{M_x-2N_y-1} + z_2 T_{M_x-1N_y-2} + z_3 q_{M_x-1N_y-1}
 \end{aligned}$$

Здесь

$$\begin{aligned}
a_1 &= \frac{h_y^2}{2(h_x^2 + h_y^2)}, & a_2 &= \frac{h_x^2}{2(h_x^2 + h_y^2)}, & a_3 &= a_1 h_x^2 \\
b_1 &= \frac{h_y^2}{(h_x^2 + 2h_y^2)}, & b_2 &= \frac{h_x^2}{(h_x^2 + 2h_y^2)}, & b_3 &= b_1 h_x^2 \\
c_1 &= \frac{h_y^2}{(2h_x^2 + h_y^2)}, & c_2 &= \frac{h_x^2}{(2h_x^2 + h_y^2)}, & c_3 &= c_1 h_x^2 \\
d_3 &= \frac{1}{\left(\frac{2}{h_x^2} + \frac{1}{h_y^2} - \frac{1}{h_x^2(1+h_x h)}\right)}, & d_1 &= \frac{d_3}{h_x^2}, & d_2 &= \frac{d_3}{h_y^2} \\
e_3 &= \frac{1}{\left(\frac{2}{h_x^2} + \frac{2}{h_y^2} - \frac{1}{h_x^2(1+h_x h)}\right)}, & e_1 &= \frac{e_3}{h_x^2}, & e_2 &= \frac{e_3}{h_y^2} \\
g_1 &= \frac{h_y^2}{(h_x^2 + h_y^2)}, & g_2 &= \frac{h_x^2}{(h_x^2 + h_y^2)}, & g_3 &= g_1 h_x^2 \\
z_3 &= \frac{1}{\left(\frac{2}{h_x^2} + \frac{2}{h_y^2} - \frac{1}{h_x^2(1+h_x h)} - \frac{1}{h_y^2(1+h_y h)}\right)}, & z_1 &= \frac{z_3}{h_x^2}, & z_2 &= \frac{z_3}{h_y^2} \\
u_3 &= \frac{1}{\left(\frac{1}{h_x^2} + \frac{2}{h_y^2} - \frac{1}{h_y^2(1+h_y h)}\right)}, & u_1 &= \frac{u_3}{h_x^2}, & u_2 &= \frac{u_3}{h_y^2} \\
k_3 &= \frac{1}{\left(\frac{2}{h_x^2} + \frac{2}{h_y^2} - \frac{1}{h_y^2(1+h_y h)}\right)}, & k_1 &= \frac{k_3}{h_x^2}, & k_2 &= \frac{k_3}{h_y^2}
\end{aligned}$$

2. Пересчет температуры в граничных точках через значения температуры во внутренних узлах сетки по формулам, приведенным в предыдущем параграфе 8.2.

Критерии выхода из итерационного процесса:

$$\max_{i,j} |T_{ij}^k - T_{ij}^{k-1}| < eps$$

$it > it\_max$

где  $eps$  - требуемая точность (0.00001);  $it, it\_max$  - номер текущей итерации и заданное максимальное число итераций (1000).

Для рассматриваемой системы линейных уравнений метод Зейделя всегда сходится, поскольку в каждом уравнении сумма недиагональных коэффициентов при неизвестных не превосходит диагональный коэффициент.

При программной реализации для температуры и параметров источников (мощность, координаты) использовались следующие массивы:  $T(N_x, N_y), q(k_q), ij(k_q, 2)$ . Это приводит к «перерасходу» памяти, но существенно упрощает логику программы.

#### 8.4. Решение задачи оптимизации.

При оптимизации в пределах одной итерации дискретные (положение источников) и непрерывные (мощности источников) параметры оптимизации обрабатываются раздельно. При этом для дискретных параметров применяется МПС, а для непрерывных –

возможен выбор между МПС и МНС. Спуск осуществляется в соответствии с алгоритмом, описанным параграфе (6) предыдущего раздела (с отступлением и дроблением шага при переходе через минимум функционала или при нарушении наложенных на параметры ограничений). В реализованной программе допускается альтернативный выбор начального шага спуска как для дискретных (2 или 4 узла), так и непрерывных (5 или 10) параметров. Таким образом, имеется возможность исследовать влияние на скорость оптимизации следующих факторов: большой шаг – быстрый спуск, но при этом увеличивается число итераций при расчете температуры методом Зейделя и выполняется большее число шагов отступления.

### 8.5. Графическое представление результатов.

Построение трехмерной поверхности рассчитанного температурного поля и его линий уровня выполняется методами и программами, которые были описаны в предыдущем разделе 7. Для этого распределение температуры в пределах каждой ячейки сетки представляется следующим интерполяционным полиномом (обеспечивается непрерывность температуры на гранях сетки и, следовательно, непрерывность температуры над всей областью)

$$T(x, y) = a + b\delta x + c\delta y + d\delta x\delta y$$

где

$$a = T_{ij}, b = T_{i+1j} - T_{ij}, c = T_{ij+1} - T_{ij}, d = T_{i+1j+1} - T_{i+1j} - T_{ij+1} - T_{ij}$$

$$i = \text{int}\left(\frac{x}{h_x}\right), j = \text{int}\left(\frac{y}{h_y}\right), \delta x = \frac{x \bmod h_x}{h_x}, \delta y = \frac{y \bmod h_y}{h_y}$$

### 8.6. Программная реализация задачи оптимизации температурного поля.

Приведем несколько последовательных экранов программы, решающей поставленную задачу. Рассматривается конкретный случай для усеченной прямоугольной области  $M_x = 10, N_x = 20, M_y = 5, N_y = 10$ . Первоначально берется пять источников с суммарной

безразмерной мощностью, равной  $\frac{3}{0,1 * 0,1} = 300$  (рис. 2). Источники равномерно

размещаются по границе области во внутренних приграничных точках сетки. Трехмерный график и линии уровня температурного поля, порождаемого этими источниками, показаны на рис. 3, 4 (видно, как в силу специфики построения линий уровня, уровень -  $T(x, y) = 0$ , представляющий собой плоскость, изображается треугольной сеткой). Затем, для данных начальных параметров источников решается задача оптимизации (рис. 5). При этом значение функционала качества температурного поля уменьшается с 0,591 до 0,2482. Трехмерный график температурного поля для оптимальных значений параметров источников приведен на рис. 6.

Ввод исходных параметров задачи и предварительные расчеты

ПОМОЩЬ ВЫХОД

Шаг сетки по осям:  $N_x, N_y$

Размерность сетки по осям X,Y -  $M_x, N_x, M_y, N_y$  (>0 и <101)

Весовые коэффициенты функционала качества температурного поля

Коэффициент поверхностной теплопередачи  $H$

Относительная точность и число итераций для метода Зейделя

номер	q	i (ось x)	j (ось y)
1	60	1	9
2	60	9	9
3	60	9	4
4	60	19	4
5	60	19	1
6			

Редактируемые параметры

Итерация  Точность  Затраченное время

Функционал

Рис. 2. Ввод исходных параметров источников

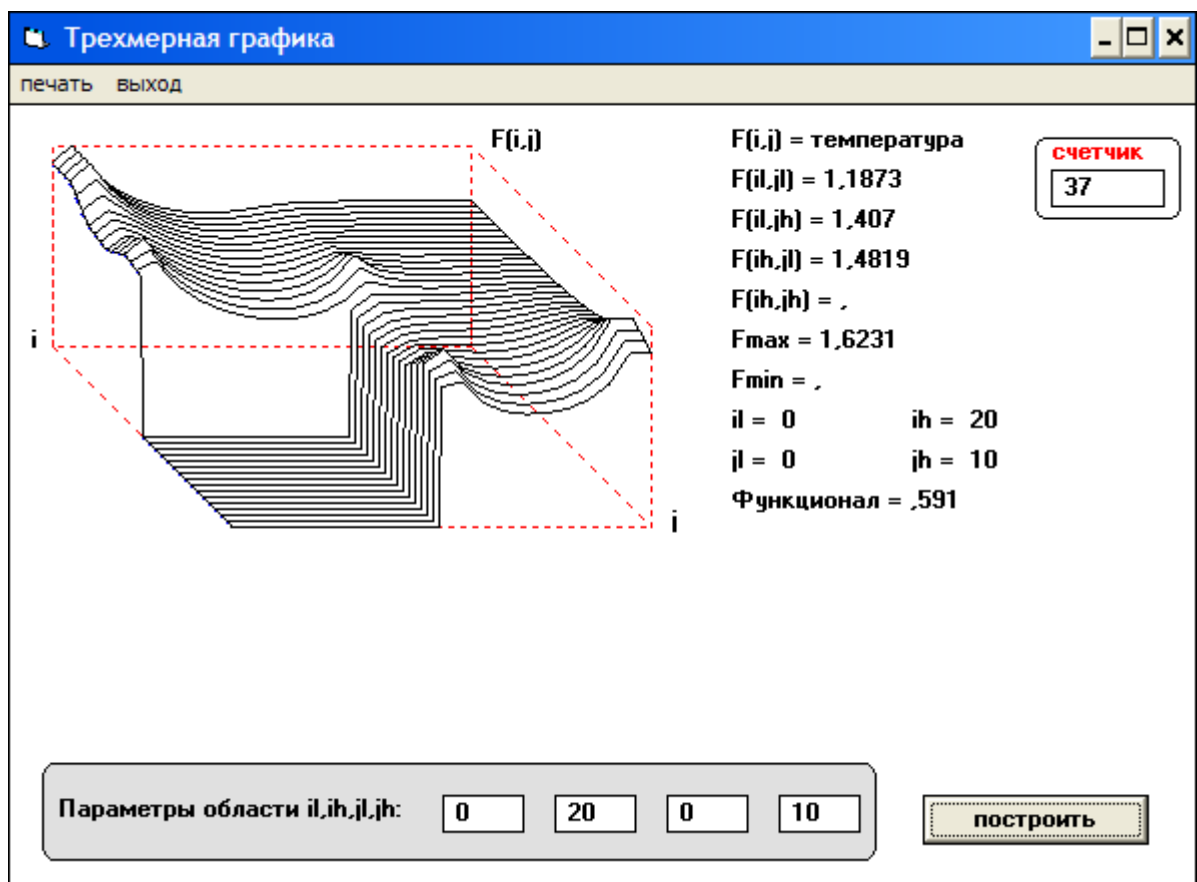


Рис. 3. Распределение температурного поля для исходных параметров источников

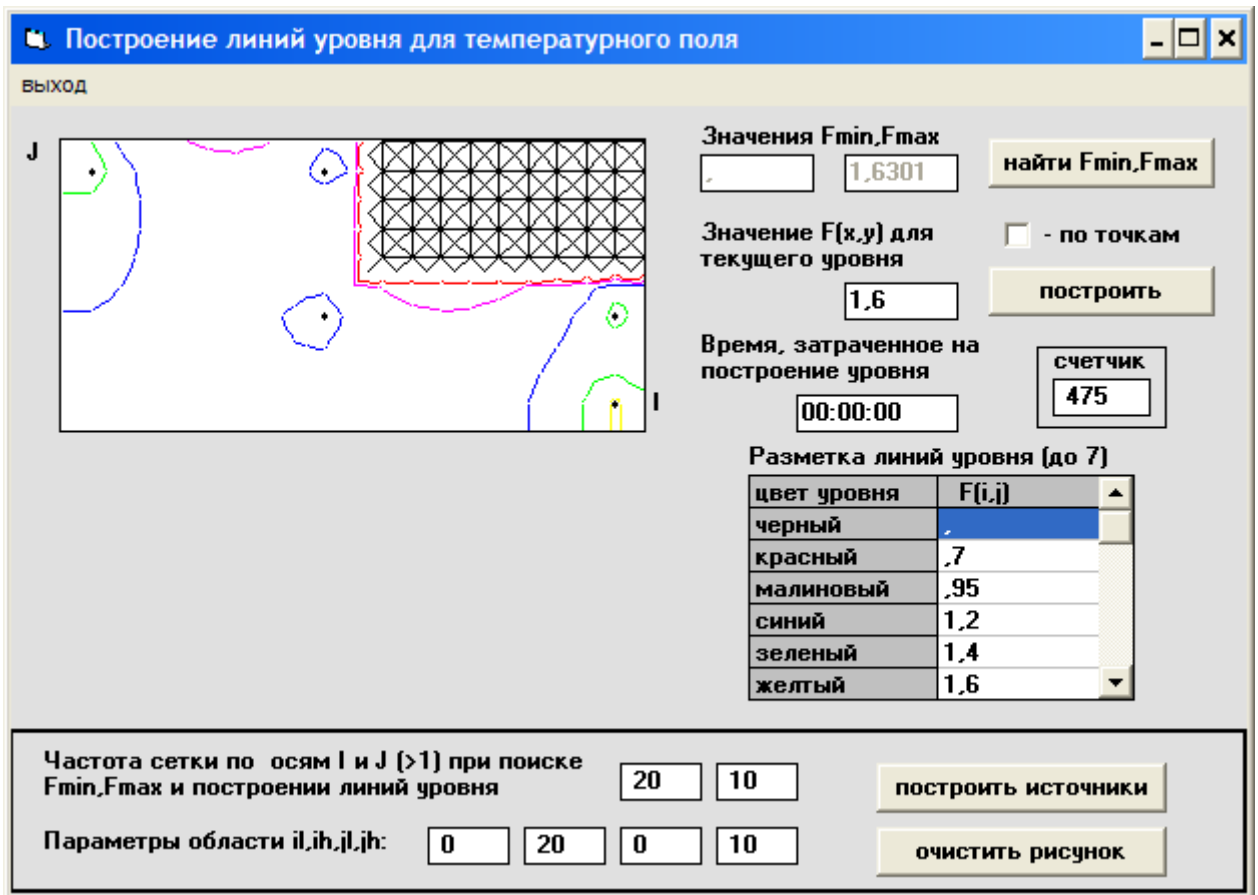


Рис. 4. Линии уровня температурного поля для исходных параметров источников

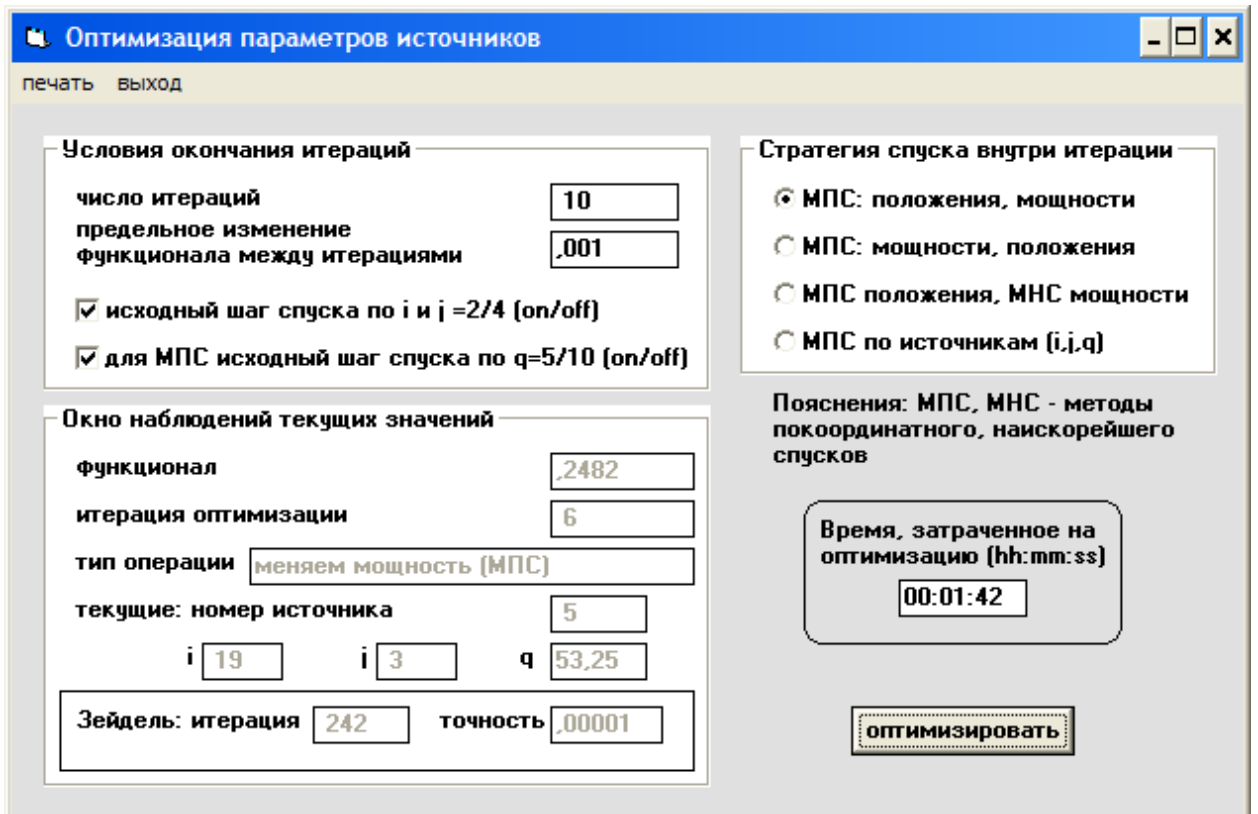


Рис. 5. Состояние панели по окончанию оптимизации

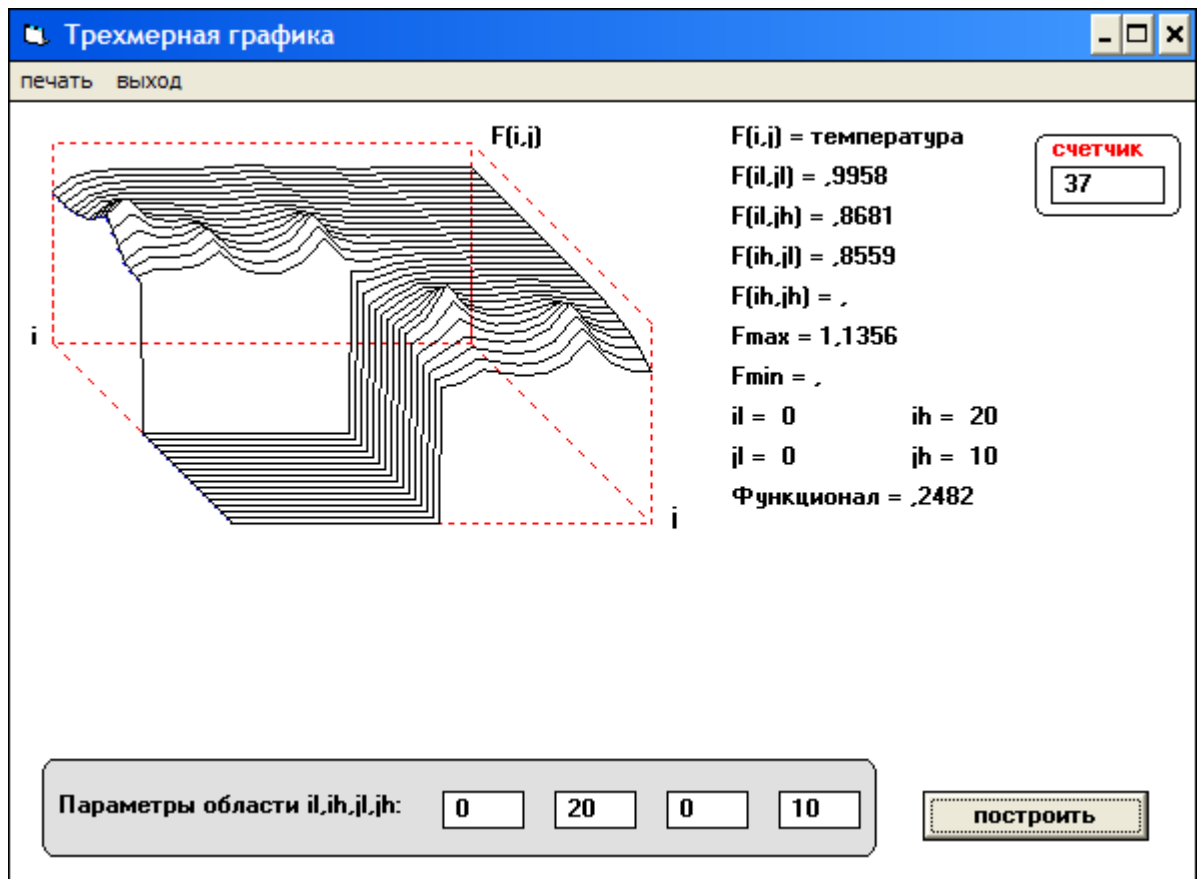


Рис. 6. Распределение температурного поля после оптимизации параметров источников

## 9. Задача оптимизации температурного поля для произвольной области и при наличии нелинейности.

### 9.1. Формулировка задачи.

Требуется оптимизировать характеристики точечных источников энергии, обеспечивающих равномерное стационарное температурное поле величины  $T_f$  в произвольной плоской области  $S$  с различными граничными условиями на отдельных участках ее границы  $\Gamma$ . Коэффициент теплопроводности материала области зависит от температуры следующим образом  $k = k_0(1 + k_1T + k_2T^2)$ . Это усложненный вариант задачи, описанной в разделе 7.

### 9.2. Математическая постановка задачи.

#### а. Уравнение стационарной теплопроводности

$$\nabla (k(T)\nabla T) = - \sum_{i=0}^{I_q-1} q_i \delta(x - x_i) \delta(y - y_i) \equiv -Q(x, y), \quad (x, y) \in S \quad (1)$$

#### б. Граничные условия

$$\begin{aligned} T &= 0, \quad (x, y) \in \Gamma_1 \\ \frac{\partial T}{\partial n} &= -h(x, y)T, \quad (x, y) \in \Gamma_2 \end{aligned} \quad (2)$$

Здесь

$$x \equiv \frac{x}{l_y}, y \equiv \frac{y}{l_y}, T \equiv \frac{T - T_0}{T_f - T_0}, q_i \equiv \frac{q_i l_y^2}{k_0 (T_f - T_0)}, h \equiv \frac{h l_y}{k_0} - \text{безразмерные величины,}$$

$l_y$  - характерный размер области  $S$  вдоль оси  $y$ ,  $T_0$  - температура окружающей среды,

$n$  - внешняя нормаль к границе области,  $I_q$  - общее число источников,

$\nabla u = \overline{\text{grad}(u)} = \vec{i} \frac{\partial u}{\partial x} + \vec{j} \frac{\partial u}{\partial y}$ ,  $h(x, y)$  - кусочно-непрерывная функция, принимающая

постоянные значения на фиксированных участках границы  $\Gamma_2$

$h(x, y) = h_i (\geq 0)$ ,  $x_{n_i}, y_{n_i} \leq x, y \leq x_{k_i}, y_{k_i}$ ,  $(x, y) \in \Gamma_2$  - для  $i$ -го участка границы.

в. Функционал качества оптимизации температурного поля имеет вид 4 из раздела 7.

г. Ограничения на параметры оптимизации: координаты источников энергии  $(x_i, y_i) \in S$ , а их мощности  $q_i > 0$ .

### 9.3. Решение температурной задачи методом конечного элемента (МКЭ).

Для произвольной области разностный метод, примененный в разделе 8, может давать большие погрешности в аппроксимации граничных условий при переходе от исходной непрерывной задачи к дискретной с областью, задаваемой прямоугольной сеткой. Поэтому в таких случаях используется МКЭ [11], который позволяет более точно аппроксимировать границы.

Покроем область  $S$  произвольной сеткой треугольных элементов (рис. 1), в общем случае гуще на тех участках, где предполагаемые изменения поля более значительны.

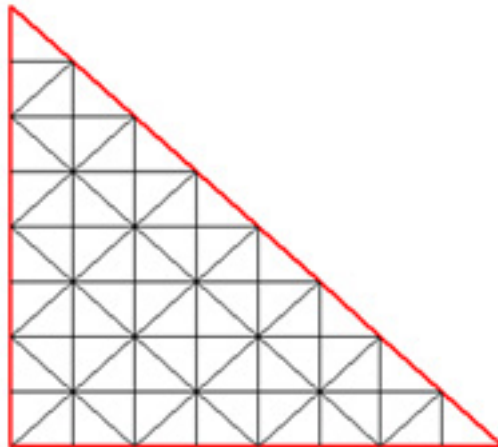


Рис. 1. Пример задания треугольных элементов над заданной областью

Над каждым треугольным элементом  $e$  заменим искомую функцию  $T(x, y)$  плоскостью, построенной по неизвестным значениям температуры в вершинах этого элемента. Уравнение указанной плоскости аналитически можно записать следующим образом

$$T^e(x, y) = \sum_{i=0}^2 T_{n^e(i)} u_i^e(x, y) \quad (3)$$

где  $i$  - локальный номер вершины элемента,  $n^e(i)$  - номер  $i$ -ой вершины  $e$ -го элемента в общей нумерации глобальной треугольной сетке узлов, а  $u_i^e(x, y)$  - многочлены вида

$$u_i^e(x, y) = A_i^e + B_i^e x + C_i^e y \quad (4)$$

Многочлены определены только над плоскостью треугольника (вне треугольника  $u_i^e(x, y) \equiv 0$ ) и удовлетворяют условиям ( $i, j = 0, 1, 2$ )



$$u_i^e(x_j, y_j) = \begin{cases} 0, & j \neq i \\ 1, & j = i \end{cases}$$

В соответствии с этими условиями коэффициенты многочленов вычисляются через координаты вершин треугольника по формулам (с циклической перестановкой всех индексов согласно последовательности 012012...)

$$A_0^e = \frac{x_1^e y_2^e - x_2^e y_1^e}{s^e} \equiv \frac{a_0^e}{s^e}$$

$$B_0^e = \frac{y_1^e - y_2^e}{s^e}$$

$$C_0^e = -\frac{x_1^e - x_2^e}{s^e}$$

где  $s^e = a_0^e + a_1^e + a_2^e$  - удвоенная площадь треугольника.

После этого приближенное кусочно-линейное решение температурной задачи над всей областью  $S$  представляется в виде ( $E_e$  - общее число треугольных элементов аппроксимирующих исходную область)

$$T(x, y) \approx \sum_{e=0}^{E_e-1} \sum_{i=0}^2 T_{n^e(i)} u_i^e(x, y) \quad (5)$$

В этом выражении значения температуры во всех узлах глобальной треугольной сетки, не принадлежащих участкам границы  $\Gamma_1$ , не известны и могут быть определены (согласно методу взвешенных невязок) из условий ортогональности невязки (получается после подстановки выражения (5) в уравнение (1)) конечному числу линейно-независимых функций вида (рис. 2)

$$w_j = \sum_e u_{i^{(e-j)}}^e(x, y) \quad (6)$$

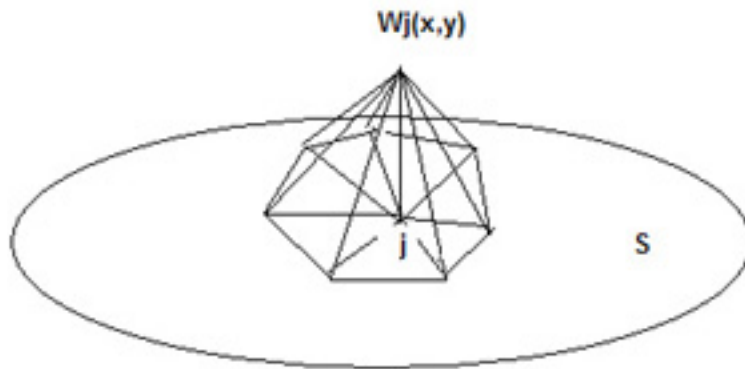


Рис. 2. Вид используемых линейно-независимых функций  $w_j$

Здесь суммирование выполняется по всем треугольникам, у которых одна из вершин совпадает с  $j$ -ым узлом в общей глобальной нумерации треугольной сетки. Вне этих элементов функция  $w_j$  тождественно равна нулю. Индекс  $j$  пробегает номера всех узлов глобальной сетки, которые не лежат на участках границы  $\Gamma_1$ , где температура задана и равна нулю.

Упомянутые выше условия ортогональности имеют следующий вид

$$\int_S w_j (\nabla(k(T)\nabla T) + Q) ds = 0, \quad j = 0, 1, 2, \dots \quad (7)$$

С помощью первой формулы Грина [10]

$$\int_S w \nabla (k \nabla T) ds = \int_{\Gamma} w k \frac{\partial T}{\partial n} d\Gamma - \int_S k \nabla w \nabla T ds$$

понизим в уравнениях (7) порядок производных (учтено, что согласно способу построения на участках границы  $\Gamma_1$   $w_j(x, y) = 0$ )

$$\int_S w_j Q ds + \int_{\Gamma_2} w_j k \frac{\partial T}{\partial n} d\Gamma - \int_S k \nabla w_j \nabla T ds = 0, \quad j = 0, 1, 2, \dots$$

Подставляя в эти соотношения граничные условия, заданные на границе  $\Gamma_2$ , окончательно получим нелинейную систему уравнений для определения неизвестных значений температуры в узлах глобальной треугольной сетки

$$\int_S w_j Q ds - \int_{\Gamma_2} w_j k h T d\Gamma - \int_S k \nabla w_j \nabla T ds = 0, \quad j = 0, 1, 2, \dots \quad (8)$$

Эту систему можно записать в матричной форме (поменяли местами немые индексы)

$$\sum_j A_{ij}(T) T_j = R_i \quad (9)$$

Здесь  $i, j = 0..(\max\_p1 - 1)$ , остальные старшие глобальные номера  $(\max\_p1..(\max\_p - 1))$  принадлежат узлам, лежащим на границе  $\Gamma_1$ , где температура равна нулю. Такой порядок глобальной нумерации узлов используется и при дальнейшем изложении.

Каждый треугольный элемент вносит в компоненты матрицы  $A$  и вектора  $R$  свои локальные составляющие (получены при подстановке в уравнение (8) соотношений (4) - (6) и последующим вычленением слагаемых, относящихся к конкретному треугольнику)

$$A_{ij}^e = (B_{n(i)}^e B_{m(j)}^e + C_{n(i)}^e C_{m(j)}^e) \int_{S^e} k(T(x, y)) ds + \int_{\Gamma^e} k(T(x, y)) h(x, y) u_{n(i)}^e(x, y) u_{m(j)}^e(x, y) d\Gamma$$

$$R_i^e = \int_{S^e} Q(x, y) u_{n(i)}^e(x, y) ds \quad (10)$$

где

$S^e, \Gamma^e$  - площадь  $e$ -го элемента и его сторона/ы, принадлежащая/ие границе  $\Gamma_2$ ,

$n(i), m(j)$  - локальные номера вершин этого элемента, совпадающие с глобальными номерами  $i, j$ .

В первом выражении (10) интегралы вычисляются приближенно по формулам

$$\int_{S^e} f(x, y) ds = \frac{S^e}{3} \sum_{i=0}^2 f(x_{m_i}^e, y_{m_i}^e)$$

$$\int_{\Gamma_{i+1}^e} f(x, y) d\Gamma = \frac{l_i^e}{6} (f(x_i^e, y_i^e) + 4f(x_{m_i}^e, y_{m_i}^e) + f(x_{i+1}^e, y_{i+1}^e)) \quad (11)$$

где

$x_{m_i}^e = \frac{x_i^e + x_{i+1}^e}{2}, y_{m_i}^e = \frac{y_i^e + y_{i+1}^e}{2}$  и т.д. - координаты середин сторон треугольника,

$l_i^e = \sqrt{(x_i^e - x_{i+1}^e)^2 + (y_i^e - y_{i+1}^e)^2}$  - длина соответствующей стороны.

Интеграл в  $R_i^e$  с учетом вида функции  $Q(x, y)$  вычисляется точно: он равен нулю, если ни один источник не лежит внутри данного треугольного элемента, или определяется по формуле (суммирование ведется по источникам, расположенным в  $e$ -ом элементе)

$$R_i^e = \sum_{iq} q_{iq} u_{n(i)}^e(x_{iq}, y_{iq})$$

Критерием принадлежности точки  $(x_{iq}, y_{iq})$  треугольному элементу является то, что для каждой стороны треугольника его противоположная вершина и проверяемая точка

лежат по одну сторону от этой стороны. Т.е., если уравнение стороны треугольника, проходящей через вершины с локальными номерами 0 и 1, записать в виде

$$f(x, y) = y - y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) = 0$$

то для этой стороны условие одностороннего расположения точки  $(x_{iq}, y_{iq})$  и вершина с локальным номером 2 выглядит следующим образом (функция  $f(x, y)$  имеет одинаковый знак в указанных точках)

$$f(x_{iq}, y_{iq}) * f(x_2, y_2) \geq 0$$

При этом источник, лежащий на стороне или в вершине треугольника (соответствует знак равенства) учитывается только один раз для первого встретившегося треугольника.

Система нелинейных уравнений (9) решается методом последовательной линеаризации. Линеаризация выполняется путем предварительного расчета  $k(T)$  по предыдущим значениям  $T(x, y)$ . Внутри итерации линеаризации получившаяся система линейных уравнений решается итерационным методом Зейделя. Алгоритм определения  $T(x, y)$  выглядит следующим образом

*Цикл по итерациям линеаризации (l)*

*Вычисление для всех элементов локальных составляющих глобальной системы (9) по формулам (10)*

*Цикл по итерациям метода Зейделя (k)*

*Процедура определения температуры методом Зейделя для одной итерации*

*Если  $\max_{i,j} |T_{ij}^k - T_{ij}^{k-1}| \leq eps_{\text{Зейдел}}$ , то выход из цикла*

*конец цикла метода Зейделя*

*если  $\max_{i,j} |T_{ij}^l - T_{ij}^{l-1}| \leq eps_{\text{линеаризации}}$ , то выход из цикла*

*конец цикла линеаризации*

При программной реализации приведенного алгоритма принималось, что (всюду нумерация начинается с нуля):

узлы глобальной сетки нумеруются таким образом, что последними оказываются узлы, принадлежащие участкам границы  $\Gamma_1$ ;

локальная нумерация вершин треугольных элементов производится против часовой стрелки, начиная с вершин, лежащих на сторонах (если они есть), совпадающих с участками границы  $\Gamma_2$ ;

с каждым элементом связан массив, в котором хранятся глобальные номера его вершин и число, указывающее на количество сторон треугольника, принадлежащих границе  $\Gamma_2$ ;

имеется дополнительный массив, в котором для всех узлов глобальной сетки с неизвестной температурой хранятся номера треугольников, включающих этот узел в качестве одной из вершин (массив одномерный, но для каждого узла порция номеров связанных с ним элементов закрывается числом «-1»).

#### 9.4. Решение задачи оптимизации параметров источников.

При решении задачи оптимизации расчет функционала качества температурного поля выполнялся следующим образом

$$\Phi(q_i, x_i, y_i; i = 0 \div (I_q - 1)) = \alpha_1 \frac{1}{\max\_p1} \sum_{p=0}^{\max\_p1-1} |1 - T(x_p, y_p)| + \alpha_2 \max_p |1 - T(x_p, y_p)| +$$

$$\alpha_3 \frac{1}{\max\_e} \sum_{e=0}^{\max\_e-1} |\text{grad}(T(x, y))_e| + \alpha_4 \max_e |\text{grad}(T(x, y))_e| + \alpha_5 \sum |q_i|$$

где  $\max\_p1, \max\_e$  - общее число узлов с неизвестной температурой и число треугольных элементов, покрывающих область  $S$ ,

$$|\text{grad}(T(x, y))_e| = \sqrt{(\sum_i B_i^e T_{n(i)})^2 + (\sum_i C_i^e T_{n(i)})^2}$$

В качестве метода оптимизации использовался метод покоординатного спуска в пространстве параметров точечных источников  $(q_i, x_i, y_i)$ . Этот метод с различными стратегиями спуска в пределах одной итерации подробно описан в разделе 8.

### 9.5. Графическое представление результатов.

Качество полученного температурного поля можно оценить, построив над областью  $S$  его трехмерную поверхность или линии уровня. Эти построения выполняются с помощью процедур, подробно описанных в разделе 8.

При этом первоначально определяется треугольник, содержащий интересующую точку  $(x, y)$ , а затем по формуле (3) вычисляется значение температуры в ней.

### 9.6. Программная реализация задачи оптимизации температурного поля и некоторые результаты ее работы.

Опишем этапы работы с программой, которая позволяет подобрать параметры точечных источников, обеспечивающих оптимально равномерное температурное поле над произвольной плоской областью.

1. Вводим параметры разбивки области на конечные треугольные элементы (рис. 3).

Здесь в строках таблиц собирается следующая информация:

Таблица 1 – глобальный номер узла и его координаты на плоскости (глобальные узлы  $\in \Gamma_1$  нумеруются последними);

Таблица 2 – номер треугольного элемента, глобальные номера его вершин и число сторон, принадлежащих границе  $\Gamma_2$  (элементы нумеруются слева - направо вдоль оси  $X$  и снизу - вверх вдоль оси  $Y$ , а узлы обходятся против часовой стрелки, начиная с узлов  $\in \Gamma_2$ );

Таблица 3 – глобальный номер узла и номер элемента, где он присутствует в качестве вершины, а (-1) закрывает список элементов, связанных с этим узлом.

После сохранения (кнопка «сохранить») введенной или загруженной из файла (пункт меню «файл») информации можно посмотреть (пункт меню «показать») размещение элементов на плоскости (рис. 4). На рисунке 4 видно, что исходная треугольная область покрыта 64-мя элементами, глобальная сетка содержит 45 узлов и область не имеет участков границы  $\Gamma_1$  с нулевой граничной температурой.

2. Задаем:

- - начальные характеристики точечных источников – в нашем случае трех (правая верхняя таблица)
- - безразмерный коэффициент поверхностного теплообмена (левая верхняя таблица)
- - коэффициенты в разложении теплопроводности (в наше случае равны нулю – нелинейность отсутствует)
- - прочие расчетные параметры задачи.

Затем для этих исходных данных вычисляем температурное поле и значение функционала качества (рис. 5). В данном окне пункт меню «тест» позволяет получить на

стадии отладки элементы полиномов (4) и локальные составляющие (10) для любого треугольного элемента (рис. 6).

3. Строим график, рассчитанного температурного поля (рис. 7)

Значения данного температурного поля лежат в интервале от 2,85 до 8,5, а функционал качества равен 9,3.

4. Выполняем оптимизацию параметров источников методом покоординатного спуска (рис. 8)

5. Просматриваем значения полученных оптимальных параметров источников и составляющие функционала (рис. 9)

6. Строим поверхность температурного поля, полученного после оптимизации (рис. 10)

Видно, что оптимизация позволила улучшить характеристики температурного поля (в пределах области оно меняется от 0,46 до 1,11) и существенно понизить значение функционала качества температурного поля (до 0,874).

Рассмотренный пример подтверждает работоспособность программного комплекса.

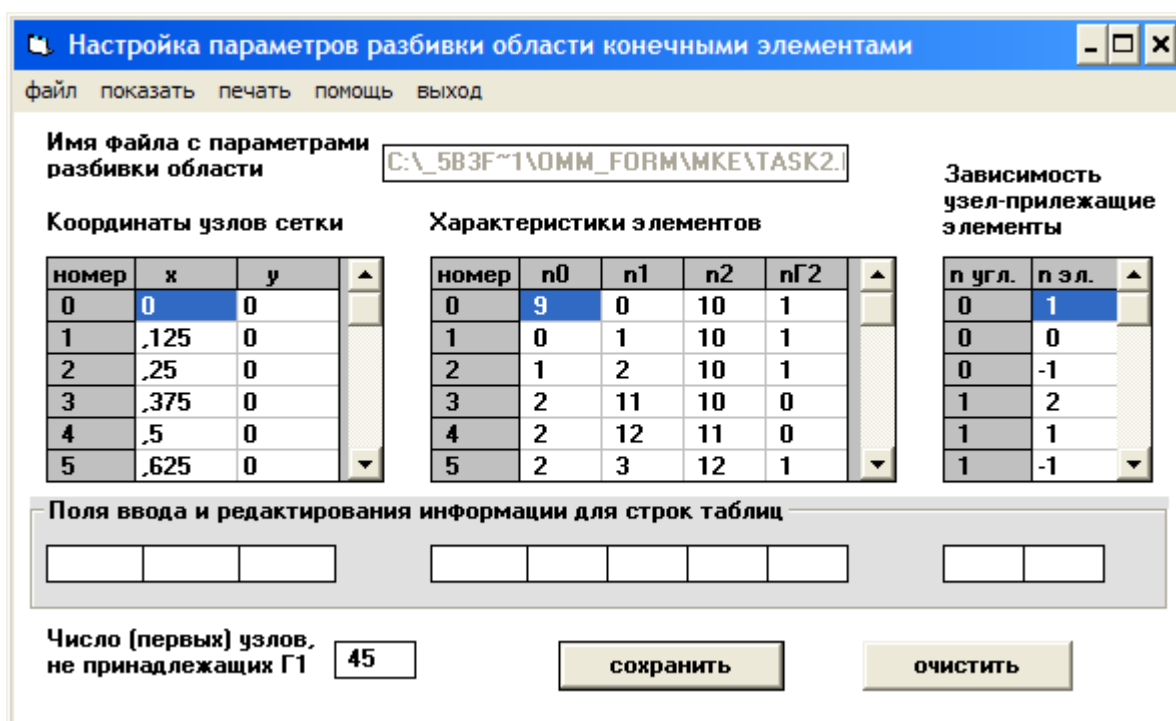


Рис. 3. Панель для задания параметров системы треугольных элементов

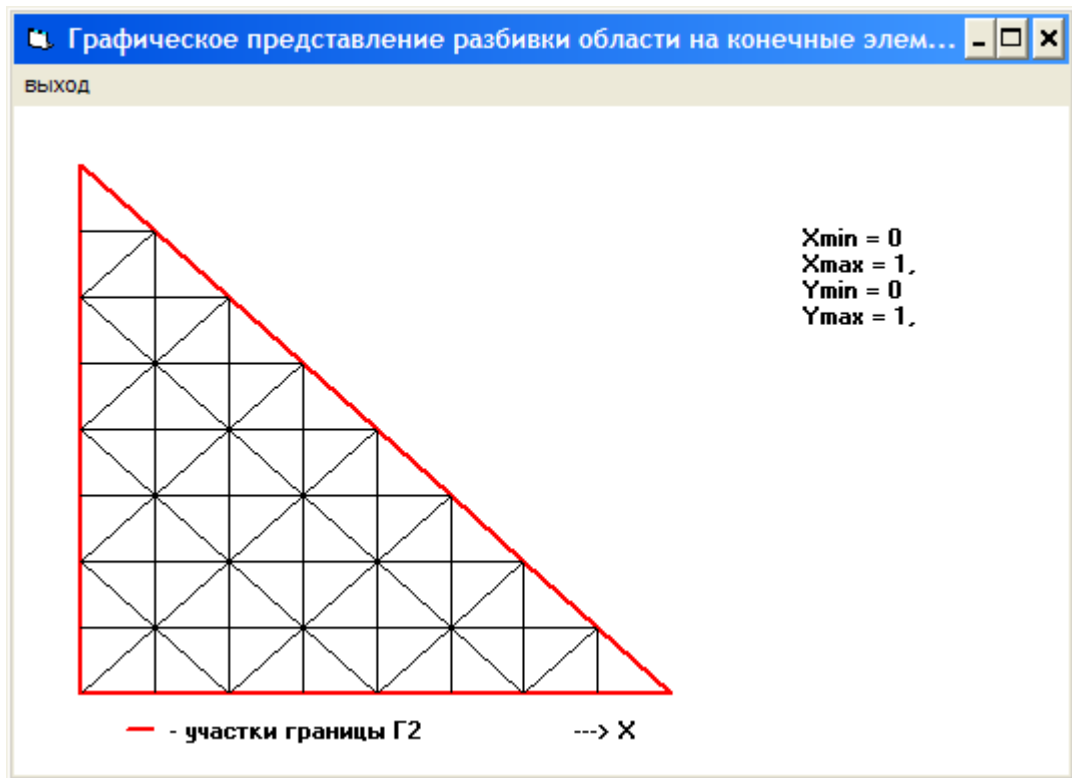


Рис. 4. Отображение введенной разбивки области на конечные треугольные элементы

Настройка параметров задачи и функций - □ ×

ПОМОЩЬ   ТЕСТ   ВЫХОД

Характеристики $h(x,y)$ : координаты участка, значение					
номер	$x_b$	$y_b$	$x_e$	$y_e$	$h_i$
1	0	0	1	0	1
2	1	0	0	1	1
3	0	0	0	1	1
3	0	0	0	1	1

Список параметров источников			
номер	q	X	Y
1	1	.125	.125
2	1	.75	.125
3	1	.125	.75
4			
3	1	.125	.75

Значения  $h(x,y)$  для остальных участков

Коэффициенты разложения для K: K1,K2        

Весовые коэффициенты функционала качества              

Относительная точность и число итераций для метода Зейделя и линеаризации     

Нелинейность и Зейдель

Итерация         Функционал     

Точность         Затраченное время

Рис. 5. Задание исходных параметров задачи

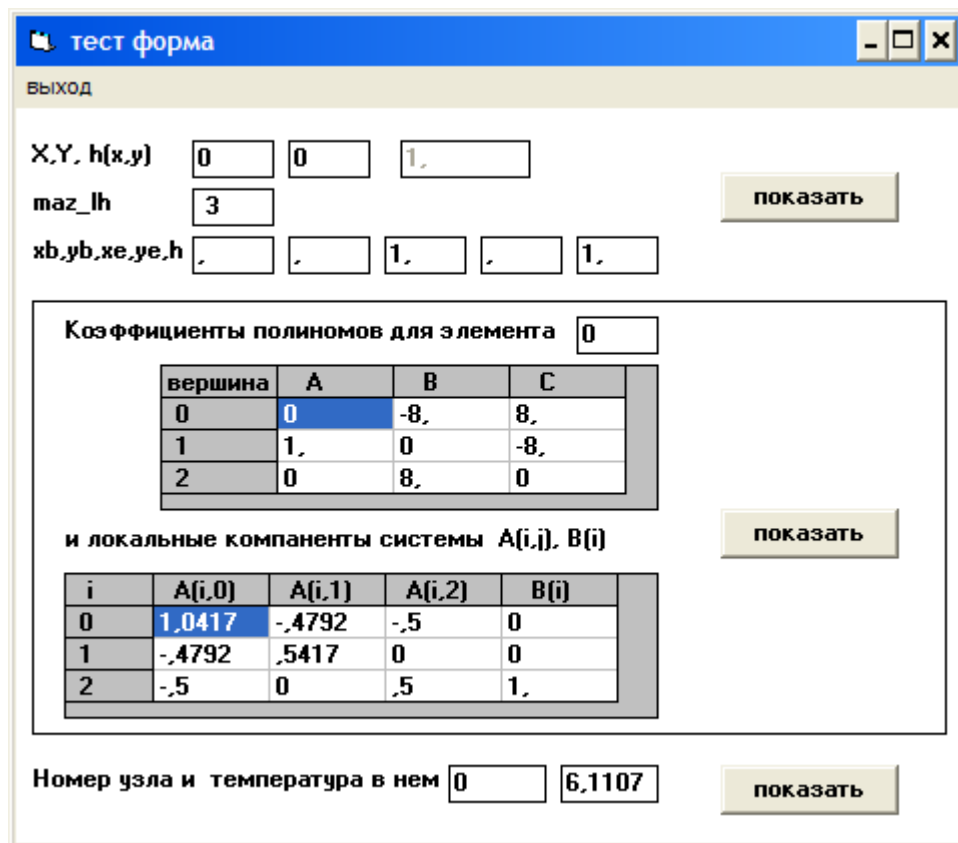


Рис. 6. Панель для просмотра параметров конкретного элемента

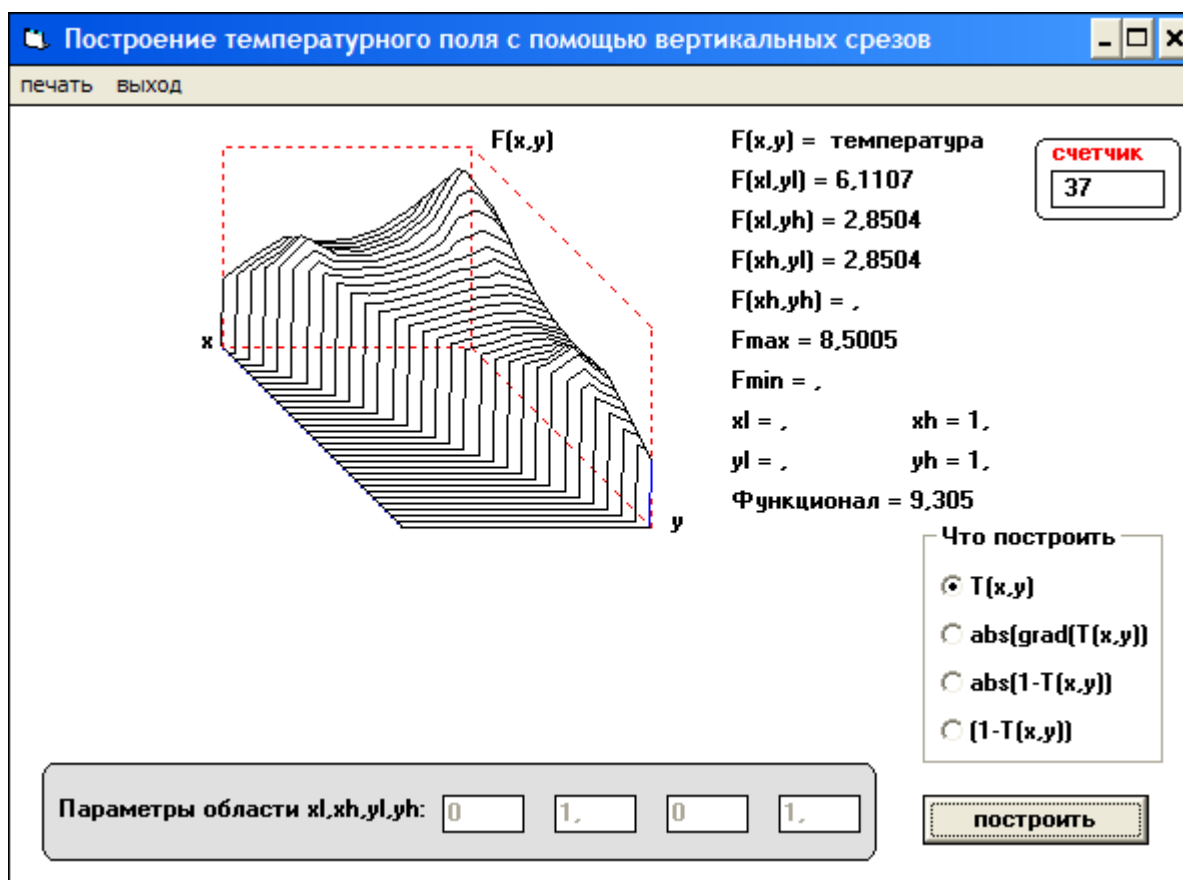


Рис. 7. Распределение температурного поля для заданных параметров источников.

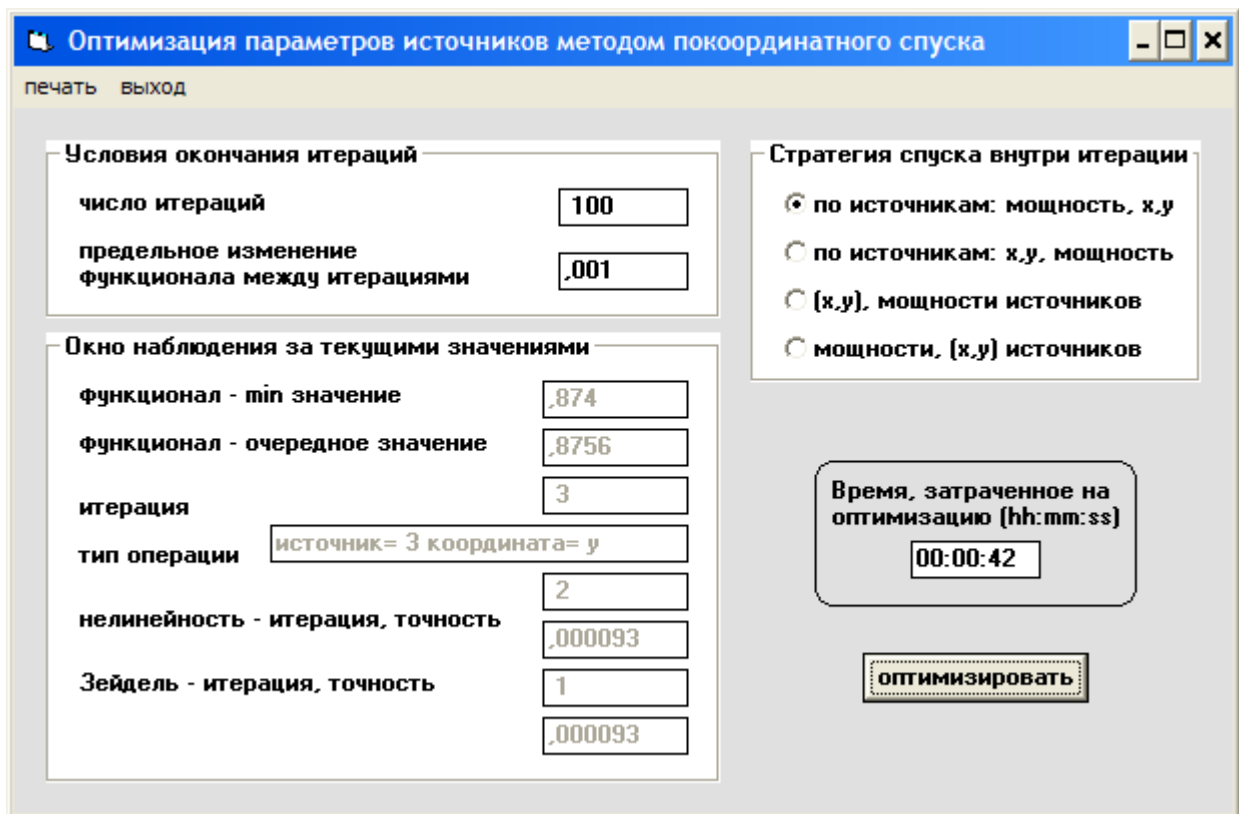


Рис.8. Состояние панели по окончанию оптимизации

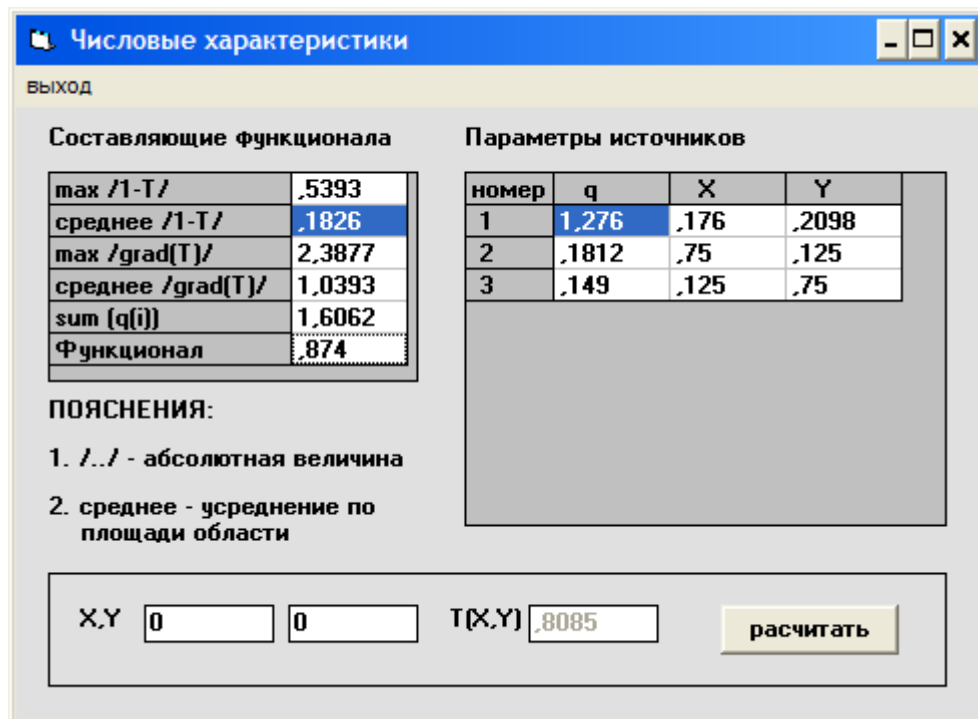


Рис. 9. Значения параметров источников после оптимизации



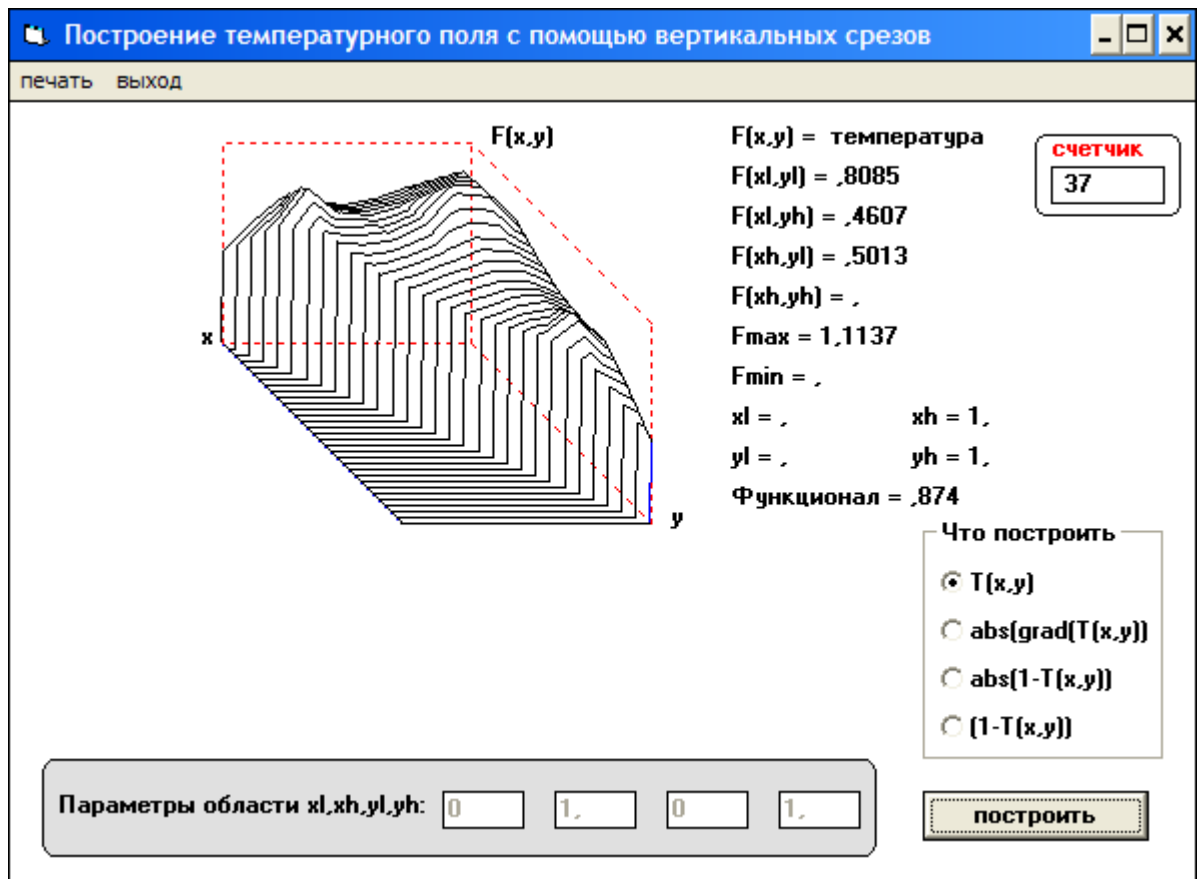


Рис. 10. Распределение температурного поля после оптимизации параметров источников

### Список литературы

1. Дж. Мэрди. Модели популяций. М., Мир, 1979
2. Математическое моделирование. Под ред. Дж. Эндрюса. М., Мир, 1979
3. Вентцель Т.С. Теория вероятностей. М., Наука, 1964
4. М. Тим Джонс. Программирование искусственного интеллекта в приложениях. М., ДМК, 2004
5. Головкин В.А. Нейронные сети: обучение, организация и применение. М., ИПРЖР, 2001
6. Терегулов И.Г. Сопротивление материалов и основы теории упругости и пластичности. М., Высшая школа, 1984
7. Д. Химмельблау. Прикладное нелинейное программирование. М., Мир, 1975
8. Флетчер. Численные методы на основе метода Галеркина. М., Мир, 1988
9. Годунов С.К., Рябенский В.С. Разностные схемы. М., Наука, 1977
10. Тихонов А.Н., Самарский А.А. Уравнения математической физики. М., ГИТТЛ, 1953, стр.291
11. Д. Норри, Ж. де Фриз. Введение в метод конечного элемента. М., Мир, 1981
12. Самарский А.А., Михайлов А.П. Математическое моделирование. М.: Физматлит, 2005
13. Введение в математическое моделирование. Под ред. В.П. Трусова. М.: Логос, 2005
14. Тарасевич Ю.Ю. Математическое и компьютерное моделирование. Вводный курс. М.: Едиториал УРСС, 2004
15. Мышкис А.Д. Элементы теории математических моделей. М.: КомКнига УРСС, 2007