

Глава 3

НЕПРЕРЫВНО-ДЕТЕРМИНИРОВАННЫЕ МОДЕЛИ ДИНАМИЧЕСКИХ СИСТЕМ С БЕСКОНЕЧНЫМ ЧИСЛОМ СТЕПЕНЕЙ СВОБОДЫ

Методы, рассмотренные в предыдущей главе, позволяют исследовать поведение систем с конечным числом степеней свободы путем решения обыкновенных дифференциальных уравнений. В случае когда число степеней свободы бесконечно велико (теплопроводность, распространение волны, течение жидкости и т. д.), построение компьютерной модели предполагает рассмотрение соответствующего дифференциального уравнения с частными производными (ДУЧП) или решения вариационной задачи. В настоящей главе анализируется проблема компьютерного моделирования систем с бесконечно большим числом степеней свободы.

3.1. Дискретизация пространственно-временной области

Процесс выполнения программы цифровой ЭВМ является дискретным. Поэтому моделирование объектов и процессов, характеризующихся бесконечно большим числом независимых величин, требует дискретизации области изменения аргументов, перехода от непрерывных функций к функциям дискретного аргумента. При этом используется **метод сеток**: исследуемая пространственно-временная область Ω заменяется сеткой $\Omega_{\Delta x, \Delta \tau}$. Допустим, необходимо промоделировать магнитное поле витка с током. Это стационарная задача: моделируемый объект с течением времени не изменяется, индукция магнитного поля зависит только от координат: $\vec{B}(x, y, z)$. Создают трехмерную сетку: непрерывную пространственную область Ω заменяют сеткой $\Omega_{\Delta x, \Delta y, \Delta z}$, узлы которой имеют ко-

ординаты $x_i = x_0 + i\Delta x$, $y_j = y_0 + j\Delta y$, $z_k = z_0 + k\Delta z$, где $i = 0, 1, 2, \dots, N$, $j = 0, 1, 2, \dots, M$, $k = 0, 1, 2, \dots, L$, а шаг $\Delta x = (x_1 - x_0)/N$, $\Delta y = (y_1 - y_0)/M$, $\Delta z = (z_1 - z_0)/L$. Программа должна содержать три вложенных цикла по i, j, k , в которых перебираются узлы пространственной сетки и для каждого узла с помощью уравнения Био-Савара-Лапласа находятся проекции индукции магнитного поля \vec{B} . Результаты расчетов могут выводиться на экран в числовом или графическом виде, сохраняться в массивах, записываться в файл на жесткий диск.

Метод сеток используется при анализе различных стационарных и нестационарных задач. На рис. 3.1.1 показано, как дискретизируется двумерная пространственная область в задаче о теплопроводности пластины. Для расчета распределения температуры вдоль стержня создается пространственно-временная сетка, изображенная на рис. 3.1.2. В качестве примера рассмотрим две задачи о расчете потенциала и напряженности электростатического поля нескольких точечных зарядов.

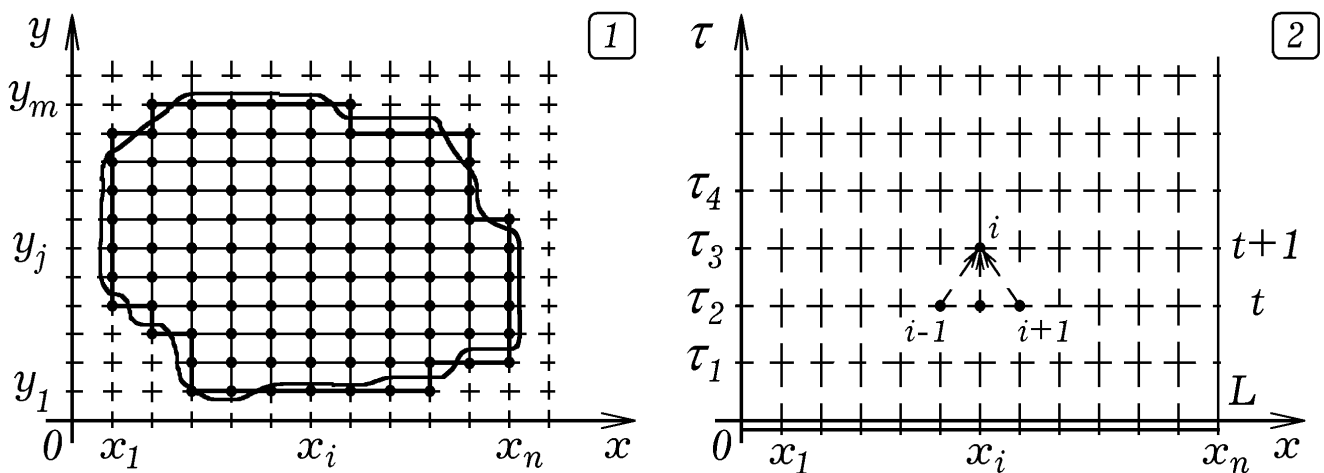


Рис. 3.1. Дискретизация пространственно-временной области

Задача 1. Три точечных заряда q_1, q_2, q_3 имеют координаты $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. Рассчитайте потенциал электростатического поля во всех точках плоскости, содержащей эти заряды.

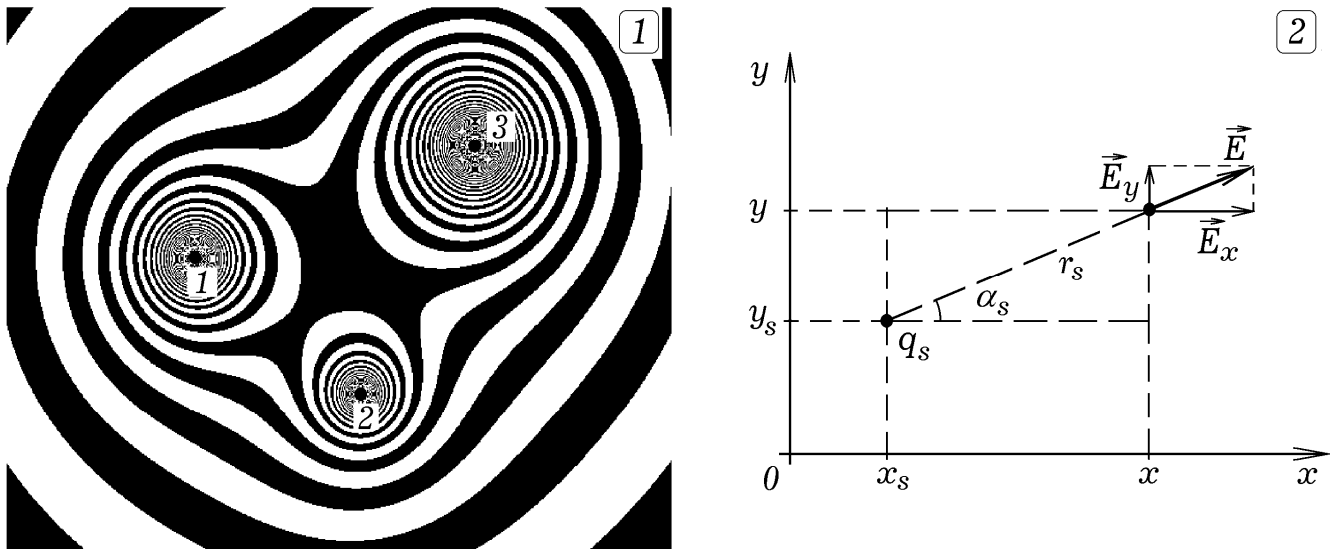


Рис. 3.2. Расчет электростатического поля на компьютере

Используется программа ПР-1 (Приложение). В цикле перебираются все точки плоскости (пиксели экрана), задаваемые целочисленными параметрами i и j . При этом вычисляются расстояния от данной точки до каждого заряда и находится алгебраическая сумма потенциалов:

$$\varphi_{i,j} = \sum_{s=1}^3 k \frac{q_s}{r_{s,i,j}}, \quad r_{s,i,j} = \sqrt{(x_s - x_{i,j})^2 + (y_s - y_{i,j})^2}.$$

В зависимости от $\varphi_{i,j}$ ставится точка соответствующего цвета (рис. 3.2.1).

Задача 2. Три точечных заряда q_1, q_2, q_3 имеют координаты $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. Рассчитайте напряженность электростатического поля во всех точках плоскости, в которой лежат эти заряды.

На плоскости, содержащей заряды, построим сетку с некоторым шагом h по вертикали и горизонтали. Напряженность поля, создаваемого зарядом q_s в точке с координатами x, y (рис. 3.2.2), вычисляется по формулам: $\cos \alpha_s = (x - x_s) / r_s, \sin \alpha_s = (y - y_s) / r_s, s = 1, 2, 3$.

$$r_s = \sqrt{(x - x_s)^2 + (y - y_s)^2}, \quad E_{sx} = kq_s \cos \alpha_s / r_s^2, \quad E_{sy} = kq_s \sin \alpha_s / r_s^2,$$

$$E_x = E_{1x} + E_{2x} + E_{3x}, \quad E_y = E_{1y} + E_{2y} + E_{3y}.$$

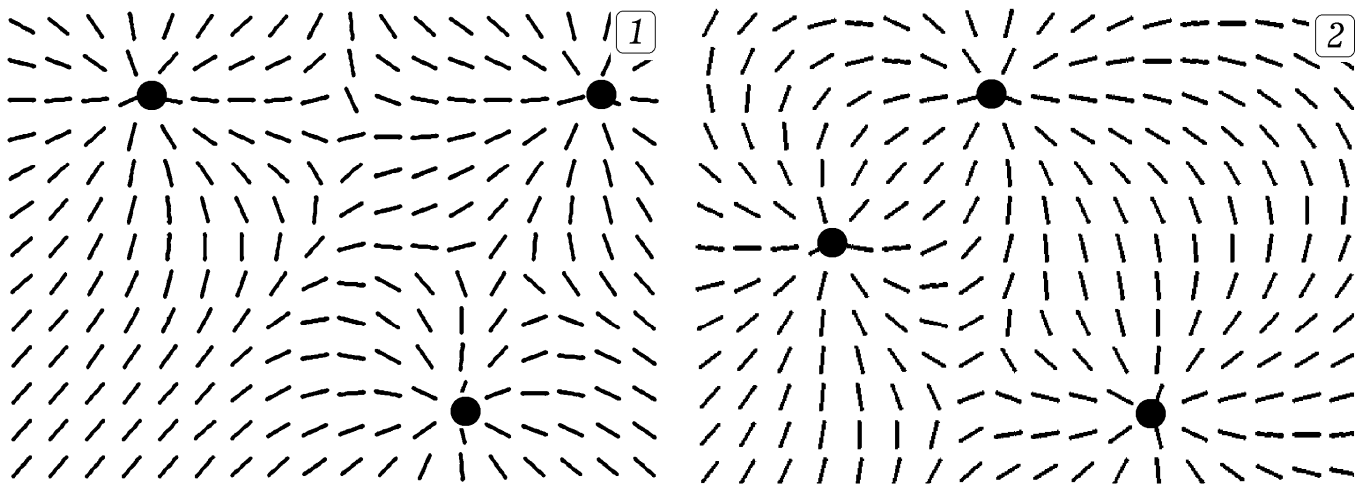


Рис. 3.3. Моделирование электростатического поля трех зарядов

Перебирая все узлы сетки, определяют проекции напряженности E_x , E_y и записывают их в два массива (программа ПР-2). После этого, исходя из значений E_x , E_y , в узлах сетки строятся отрезки, визуализирующие направление \vec{E} . Результаты представлены на рис. 3.3.

Многие физические ситуации сводятся к задаче Дирихле: требуется решить дифференциальное уравнение в частных производных в замкнутой области Ω при известных граничных условиях. При этом граничные условия могут быть трех видов [12]: 1) задано распределение искомой функции на границе области G : $\varphi_G = f_1(x, y, z)$; 2) задана скорость изменения искомой функции на границе G : $\partial\varphi_G / \partial\tau = f_2(x, y, z, \tau)$; 3) задано соотношение $\partial\varphi_G / \partial\tau = a(\varphi_G - f_3(x, y, z, \tau))$, связывающее скорость изменения искомой функции на границе G с некоторой функцией времени $f_3(\tau)$. Различают первую, вторую и третью краевые задачи.

При изучении нестационарных процессов, которые описываются функциями вида $f(x, y, z, \tau)$, содержащих время τ , рассматривают пространственно-временную область Ω . Создается пространственно-временная сетка $\Omega_{\Delta x, \Delta y, \Delta z, \Delta \tau}$, выбирается шаг по времени $\Delta \tau$. Исходя из начального состояния системы в момент $\tau = 0$, рассчитывается состояние системы в следующий дискретный момент времени $\tau + \Delta \tau$. Для этого находят значения функции $f(x, y, z, \tau + \Delta \tau)$ во всех узлах пространствен-

ной сетки. Затем выполняется вторая итерация и вычисляются значения функции $f(x, y, z, \tau + 2\Delta\tau)$ в дискретный момент $\tau + 2\Delta\tau$ и т. д. Например, при решении задачи о нестационарной теплопроводности стержня [13] программа должна содержать цикл по времени, в котором перебираются все узлы пространственно-временной сетки и в них рассчитывается температура T_i^{t+1} на следующем временном слое (рис. 3.1.2).

3.2. Классификация дифференциальных уравнений с частными производными

Математическое моделирование систем с бесконечно большим числом степеней свободы предполагает решение **дифференциальных уравнений с частными производными (ДУЧП)** вида:

$$a_{11} \frac{\partial^2 f}{\partial x^2} + a_{22} \frac{\partial^2 f}{\partial y^2} + \dots + b_{12} \frac{\partial^2 f}{\partial x \partial y} + b_{13} \frac{\partial^2 f}{\partial x \partial z} + \dots + c_1 \frac{\partial f}{\partial x} \dots + cf + \varphi(x, y, \dots) = 0.$$

В случае, когда коэффициенты $a_{11}, a_{22}, \dots, b_{12}, b_{23}, \dots, c_1, c_2, \dots, c$ не зависят от значений искомой функции $f(x, y, z, \dots)$, ДУЧП называется **линейным**, а если зависят, - **нелинейным**. Если $\varphi(x, y, z, \dots) = 0$, то уравнение называется **однородным**. Рассмотрим основные типы ДУЧП [12].

1. К уравнениям **гиперболического типа** относятся уравнения колебаний упругой одномерной (струны), двумерной (мембраны) и трехмерной среды:

$$\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} = \frac{1}{v^2} \frac{\partial^2 \xi}{\partial \tau^2}, \quad \frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} + \frac{\partial^2 \xi}{\partial z^2} = \frac{1}{v^2} \frac{\partial^2 \xi}{\partial \tau^2} + \beta \frac{\partial \xi}{\partial \tau} - F(x, \tau),$$

телеграфные уравнения:

$$\frac{\partial i}{\partial x} + C \frac{\partial u}{\partial t} + \frac{1}{R} u = 0, \quad \frac{\partial u}{\partial x} + L \frac{\partial i}{\partial t} + Ri = 0.$$

Граничные условия для струны (мембраны) могут быть такими: 1) задан режим колебания одной или нескольких точек струны: $\xi(0, \tau) = \mu(\tau)$;

2) задана скорость одной или нескольких точек: $d\xi(0, \tau) / d\tau = v(\tau)$; 3) задана сила, действующая на одну или несколько точек $d^2\xi / d\tau^2 = F(\tau)$.

2. Уравнением **параболического типа** является уравнение теплопроводности (диффузии, вязкости):

$$c\rho \frac{\partial T}{\partial \tau} = \frac{\partial}{\partial x} \left(k_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k_y \frac{\partial T}{\partial y} \right) + q(x, y, \tau).$$

3. К уравнениям **эллиптического типа** относятся уравнение для потенциала стационарного электрического поля:

$$\frac{\partial}{\partial x} \left(\varepsilon_x \frac{\partial \varphi}{\partial x} \right) + \frac{\partial}{\partial y} \left(\varepsilon_y \frac{\partial \varphi}{\partial y} \right) = -\rho(x, y),$$

уравнение, описывающее стационарное распределение температуры равномерно нагретой пластины:

$$\frac{\partial}{\partial x} \left(k_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k_y \frac{\partial T}{\partial y} \right) = 0,$$

уравнение для потенциального течения несжимаемой жидкости без источников:

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = 0, \quad \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0,$$

где φ - потенциал скорости, ψ - функция тока.

3.3. Численное решение нестационарных задач

При изучении процессов, характеристики которых изменяются с течением времени, приходится решать ДУЧП, содержащие время. Такие задачи называются **нестационарными**. Для их решения необходимо [6; 12]:

1) дискретизировать пространственно-временную область $\Omega_{x,y,z,\tau}$, создав сетку $\Omega_{\Delta x, \Delta y, \Delta z, \Delta \tau}$; 2) заменить производные в ДУЧП их конечно-разностными аппроксимациями и получить систему алгебраических

уравнений; 3) решить конечно-разностные уравнения с помощью ЭВМ; 4) правильно интерпретировать полученные результаты.

Многие физические ситуации сводятся к задаче Дирихле: требуется решить дифференциальное уравнение в частных производных в замкнутой области Ω при известном распределении искомой функции на границе области $\varphi(\Omega_0)$. Рассмотрим нестационарную задачу, состоящую в решении уравнения теплопроводности для неоднородного стержня:

$$\frac{\partial T}{\partial \tau} = \frac{\partial}{\partial x} \left(k(x) \frac{\partial T}{\partial x} \right) + \frac{q(x)}{c\rho} = \frac{\partial k}{\partial x} \cdot \frac{\partial T}{\partial x} + k \frac{\partial^2 T}{\partial x^2} + \frac{q(x)}{c\rho}.$$

Введем сетку $\Omega_{\Delta x, \Delta \tau}$ и перейдем к конечным разностям:

$$\frac{T_i^{t+1} - T_i^t}{\Delta \tau} = \frac{k_{i+1} - k_{i-1}}{2\Delta x} \cdot \frac{T_{i+1}^t - T_{i-1}^t}{2\Delta x} + k_i \frac{T_{i-1}^t - 2T_i^t + T_{i+1}^t}{\Delta x^2} + \frac{q_i}{c\rho},$$

$$T_{i,j}^{t+1} = T_{i,j}^t + \left(\frac{k_{i+1} - k_{i-1}}{2\Delta x} \cdot \frac{T_{i+1}^t - T_{i-1}^t}{2\Delta x} + k_i \frac{T_{i-1}^t - 2T_i^t + T_{i+1}^t}{\Delta x^2} + \frac{q_i}{c\rho} \right) \Delta \tau.$$

Зная значения функции T_i^t на временном слое t , можно рассчитать ее значения T_i^{t+1} на следующем временном слое $t+1$.

Задача 3. Имеется неоднородный стержень, известна начальная температура различных его точек, координаты и мощность источника тепла (холода). Один конец теплоизолирован, другой поддерживается при постоянной температуре. Необходимо рассчитать распределение температуры вдоль стержня [8].

Для решения задачи может быть применен алгоритм А-1, реализованный в программе ПР-3. В нем используются два массива $T[i]$ и $TT[i]$, в которых сохраняются значения температуры элементов стержня в моменты t и $t+1$ соответственно. Расчет температуры осуществляется в цикле по выведенной выше формуле. Будем считать, что длина стержня L , его коэффициент теплопроводности k при $x > 0,2L$ равен 1,8, а при $x < 0,2L$ равен 1. Температура точек с координатами от $0,18L$ до $0,22L$ равна $T = 5$, а все остальные точки имеют температуру 0,1. Точки, координата x

которых лежит в интервале от $0,81L$ до $0,82L$, нагреваются источником тепла известной мощностью $q = 0,5$. Левый конец поддерживается при постоянной температуре, а правый - теплоизолирован.

Алгоритм А-1

НАЧАЛО ПРОГРАММЫ

```

ДЛЯ i:=1 ДО N ДЕЛАТЬ {-
    ЕСЛИ (i>18)И(i<22) ТО T[i]:=5 ИНАЧЕ T[i]:=0.1; -}
ДЛЯ j:=1 ДО N ДЕЛАТЬ {-
    ЕСЛИ j>20 ТО k[j]:=1.8 ИНАЧЕ k[j]:=1; -}
ПОВТОРЯТЬ ДО НАЖАТИЯ НА КЛАВИШУ{==   kk:=kk+1;
    ДЛЯ i:=2 ДО N-1 ДЕЛАТЬ {
        ЕСЛИ (i>80) И (i<83) ТО q:=0.5 ИНАЧЕ q:=0;
        TT[i]:=T[i]+k[i]*(T[i+1]-2*T[i]+T[i-1])*dt/(h*h)+
            (k[i+1]-k[i-1])*(T[i+1]-T[i-1])*dt/(4*h*h)+q*dt;   }
        ДЛЯ i:=2 ДО N-1 ДЕЛАТЬ { T[i]:=TT[i]; }
        T[1]:=0.1; T[N]:=T[N-1];
        ЕСЛИ kk mod 1000 = 0 ТО {
            ДЛЯ i:=2 ДО N ДЕЛАТЬ {- ПОСТАВИТЬ ТОЧКУ (i, T[i]); -}
        ==}
    КОНЕЦ ПРОГРАММЫ

```

На рис. 3.4.1 представлен результат решения одномерной задачи о теплопроводности при следующих начальных условиях. В центре неоднородного стержня - источник тепла известной мощности $q > 0$, температура левого конца увеличивается по закону $T(0, \tau) = \alpha \tau$, правый конец поддерживается при постоянной температуре: $T(l, \tau) = const$. Видно, что с течением времени температура левого конца монотонно возрастает, температура средней части стержня, содержащей источники тепла, также увеличивается. На рис. 3.4.2 приведен результат вычислений при других начальных условиях и ином расположении источников тепла.

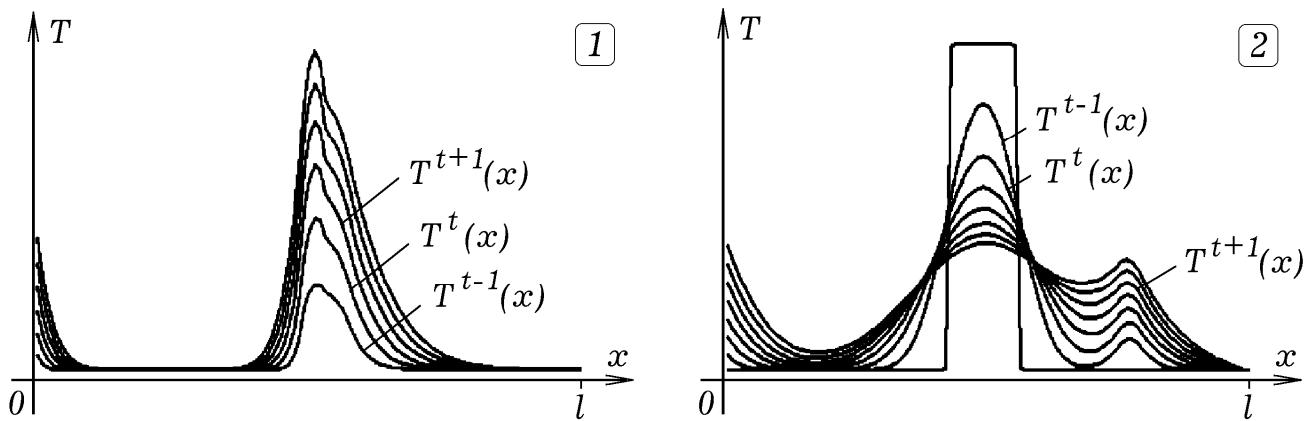


Рис. 3.4. Расчет температуры различных точек стержня

Задача 4. Пластина имеет форму сектора круга, задано начальное распределение температуры, координаты и мощность источника тепла (холода). Используя полярные координаты, рассчитайте температуру различных точек пластины в последующие моменты времени.

Эта задача сводится к уравнению в полярных координатах:

$$\frac{\partial T}{\partial t} = k \left(\frac{1}{r} \frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2 T}{\partial \alpha^2} \right) + \frac{q(x, y)}{\rho c}.$$

В конечных разностях получаем:

$$T_{i,j}^{t+1} = T_{i,j}^t + k \frac{T_{i+1,j}^t - T_{i-1,j}^t}{2r\Delta r} \Delta t + k \frac{T_{i-1,j}^t - 2T_{i,j}^t + T_{i+1,j}^t}{\Delta r^2} \Delta t + k \frac{T_{i,j-1}^t - 2T_{i,j}^t + T_{i,j+1}^t}{\Delta \alpha^2 r^2} \Delta t + \frac{q_{i,j} \Delta t}{\rho c}.$$

Используется программа ПР-4. Она содержит два вложенных цикла по i и j , в которых перебираются все узлы двумерной сетки и пересчитываются значения $T_{i,j}^{t+1}$ на следующем временном слое. При ее запуске на экране появляется цветное изображение, границы одноцветных областей соответствуют изотермам (рис. 3.5).

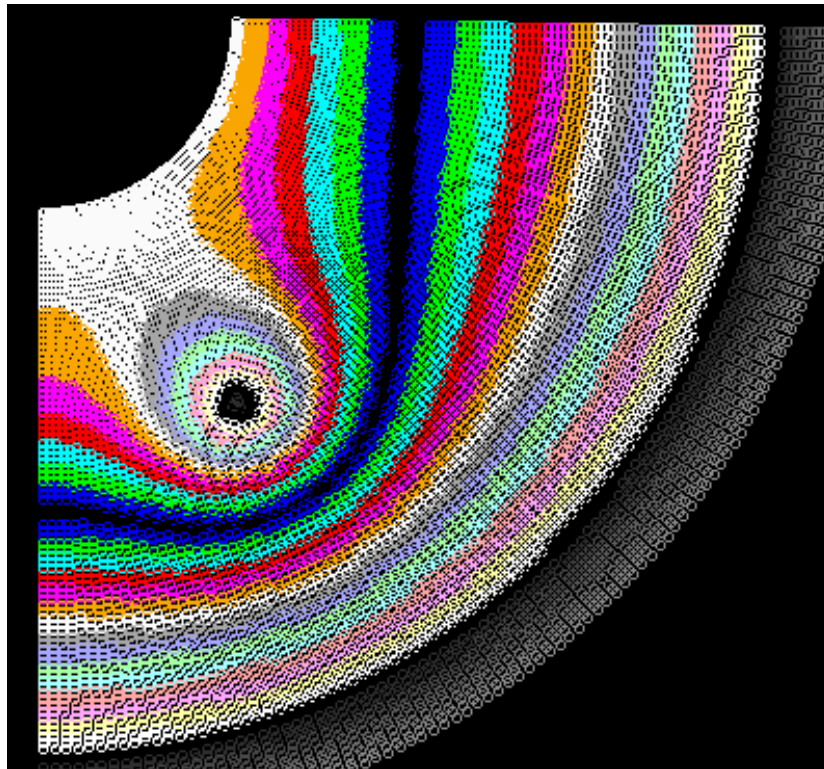


Рис. 3.5. Решение уравнения теплопроводности в полярных координатах

3.4. Численное решение стационарных задач

Если искомая функция зависит только от пространственных координат, но не зависит от времени, то задача называется **стационарной**. К этому типу относится задача о форме упругой (мыльной) пленки, натянутой на каркас, или задача о распределении температуры вдоль стержня, боковая поверхность которого теплоизолирована, а температура концов поддерживается постоянной. Рассмотрим решение стационарной задачи Дирихле для уравнения Пуассона, описывающего распределение потенциала в двумерной области [8; 9]:

$$\Delta\varphi = \frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} = f(x, y), \text{ при граничных условиях } \varphi_G = \varphi_G(x, y, z).$$

Дискретизируем область изменения аргументов x , y и в конечных разностях запишем:

$$\frac{\varphi_{i-1,j} - 2\varphi_{i,j} + \varphi_{i+1,j}}{\Delta x^2} + \frac{\varphi_{i,j-1} - 2\varphi_{i,j} + \varphi_{i,j+1}}{\Delta y^2} = f(x_i, y_j), \quad \Delta x = \Delta y = h.$$

Отсюда получаем явную расчетную схему:

$$\varphi_{i,j} = (\varphi_{i+1,j} + \varphi_{i-1,j} + \varphi_{i,j+1} + \varphi_{i,j-1} - f(x_i, y_j)h^2) / 4.$$

Для решения этой задачи используется **метод последовательных приближений**: во всех внутренних узлах сетки задаются произвольные исходные значения искомой функции $\varphi_0(x_i, y_j)$ (нулевое приближение); значения функции во внешних узлах должны соответствовать граничным условиям. Осуществляется первая итерация, в ходе которой перебираются все внутренние узлы сетки и, исходя из начальных значений, определяют новые уточненные значения функции $\varphi_1(x_i, y_j)$. Затем осуществляется вторая, третья, ... и т. д. итерации, причем результаты k -й итерации используются в качестве исходных для $(k + 1)$ -й итерации.

Иногда при решении стационарной задачи рассматривают соответствующую ей нестационарную задачу, искусственно вводя **псевдовремя** t . Например, при нахождении стационарного распределения температуры $T(x, y)$, удовлетворяющего уравнению Лапласа

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

и граничным условиям $T_G = T_G(x, y)$, переходят к нестационарной задаче

$$\frac{\partial T}{\partial t} = k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right)$$

с теми же граничными условиями. Можно доказать, что при $t \rightarrow \infty$ решение этого нестационарного уравнения эволюционирует к решению исходной стационарной задачи. При этом значения $T_{i,j}^t$ образуют итерационные слои, аналогичные временным слоям в нестационарных задачах.

3.5. Методы численного решения вариационных задач

В некоторых случаях расчет состояния системы с большим числом степеней свободы может быть сведен к решению вариационной задачи, заключающейся в отыскании функции нескольких переменных, удовлетворяющей определенным требованиям. Типичной задачей вариационного исчисления является **задача о брахистохроне**, предложенная Бернулли в 1696 году. Она состоит в нахождении формы линии (профиля горки), соединяющей заданные точки А и В, время скольжения по которой в однородном поле тяжести минимально. Для ее решения необходимо найти функцию $y(x)$, удовлетворяющую условиям $y(x_0) = 0$, $y(x_1) = y_1$, при которой интеграл

$$t = \int_{x_0}^{x_1} \frac{\sqrt{1 + y'^2} dx}{\sqrt{2gy}}$$

принимает минимальное значение. Интеграл, величина которого зависит от выбора подынтегральной функции, называется **функционалом**.

Многие проблемы сводятся к **основной задаче вариационного исчисления**: найти такую функцию (экстремаль), для которой некоторый функционал принимает экстремальное значение, то есть его вариация равна 0. Например, задача о форме свободной поверхности жидкости, капли, мыльной пленки, упругой пленки; задача о форме неоднородной нити или пластины в поле тяжести; задача об оптимальном распределении ресурсов при планировании работы предприятия; задача о распространении света в неоднородной оптической среде и другие.

Эффективным методом численного решения вариационных задач является **метод локальных вариаций** [14]. Рассмотрим одномерную вариационную задачу: необходимо найти такую функцию $u(x)$ на интервале $[a; b]$, удовлетворяющую ограничениям $u^-(x) \leq u(x) \leq u^+(x)$, которая приводит функционал

$$F = \int_a^b f(x, u, u'_x) dx$$

к минимуму. Ограничения включают в себя краевые условия как частный случай. Например, краевые условия $u(a) = A$ и $u(b) = B$ эквивалентны ограничениям $u^-(a) = u^+(a) = A$, $u^-(b) = u^+(b) = B$.

Введем сетку $\Omega_{\Delta x}$, разбив область $[a; b]$ точками $x_i = a + i\Delta x$ с шагом $\Delta x = (b - a) / N$. Тогда функционал приближенно равен:

$$F \approx \sum_{i=0}^{N-1} f\left(x_i + \frac{\Delta x}{2}, \frac{u(x_i) + u(x_{i+1})}{2}, \frac{u(x_i) - u(x_{i+1}))}{\Delta x}\right).$$

Задача сводится к нахождению таких значений $u_i = u(x_i)$ ($i = 0, 1, 2, \dots, N$), удовлетворяющих ограничениям $u^-(x_i) \leq u_i \leq u^+(x_i)$, при которых записанная выше сумма достигает минимума.

Для ее решения используется рекуррентный процесс построения последовательных приближений. Допустим, найдено k -е приближение u_i^k ($i = 0, 1, 2, \dots, N$). Случайным образом выбираем j -й узел и изменяем величину u_j^k на заданный шаг варьирования h (малое положительное число) в сторону увеличения и уменьшения. Получаем три числа: $u'_j = u_j^k - h$, u_j^k , $u''_j = u_j^k + h$. Они должны удовлетворять наложенным ограничениям. Затем для каждой величины пересчитываем функционал F , получая $F(u'_j)$, $F(u_j^k)$, $F(u''_j)$, и сравниваем их друг с другом. Переменная u_j^{k+1} на $(k + 1)$ -й итерации принимает значение:

$$u_j^{k+1} = \begin{cases} u_j^k - h, & \text{если } F(u_j^k - h) < F(u_j^k), \\ u_j^k, & \text{если } F(u_j^k) < F(u_j^k + h), \quad F(u_j^k) < F(u_j^k - h), \\ u_j^k + h, & \text{если } F(u_j^k + h) < F(u_j^k). \end{cases}$$

Затем процесс повторяется. Можно во время одной итерации варьировать не одно, а несколько значений u_j^k , а затем пересчитывать функционал: если его значение уменьшается, то изменения принимаются, а если нет - отвергаются.

Каждый раз вычислять весь функционал может быть неудобно из-за больших затрат времени. Оптимальный алгоритм состоит в том, что при варьировании величины u_j^k на h пересчитывается не весь функционал F , а лишь изменяющаяся сумма:

$$f\left(x_{j-1} + \frac{\Delta x}{2}, \frac{u(x_{j-1}) + u(x_j)}{2}, \frac{u(x_{j-1}) - u(x_j)}{\Delta x}\right) +$$

$$f\left(x_j + \frac{\Delta x}{2}, \frac{u(x_j) + u(x_{j+1})}{2}, \frac{u(x_j) - u(x_{j+1})}{\Delta x}\right)$$

и определяется $u(x_j)$, при котором она минимальна.

Другой подход состоит в представлении функционала в виде функции достаточно большого числа переменных [14]. Для этого искомую функцию $u(x)$ представляют в виде степенного ряда:

$$u(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

или ряда Фурье:

$$u(x) = \frac{a_0}{2} + \sum_{k=1}^n (a_k \cos kx + b_k \sin kx).$$

В общем случае функция $u(x)$ может быть представлена в виде некоторой функции $\varphi(x)$ или суммы функций:

$$u(x) = \sum_{k=1}^n a_k \varphi_k(x).$$

Во всех этих случаях поиск экстремали сводится к нахождению коэффициентов a_k , $k = 1, 2, \dots, n$, при которых функционал достигает минимума.

Для этого используют метод покоординатного спуска, метод градиентного спуска и т. д.

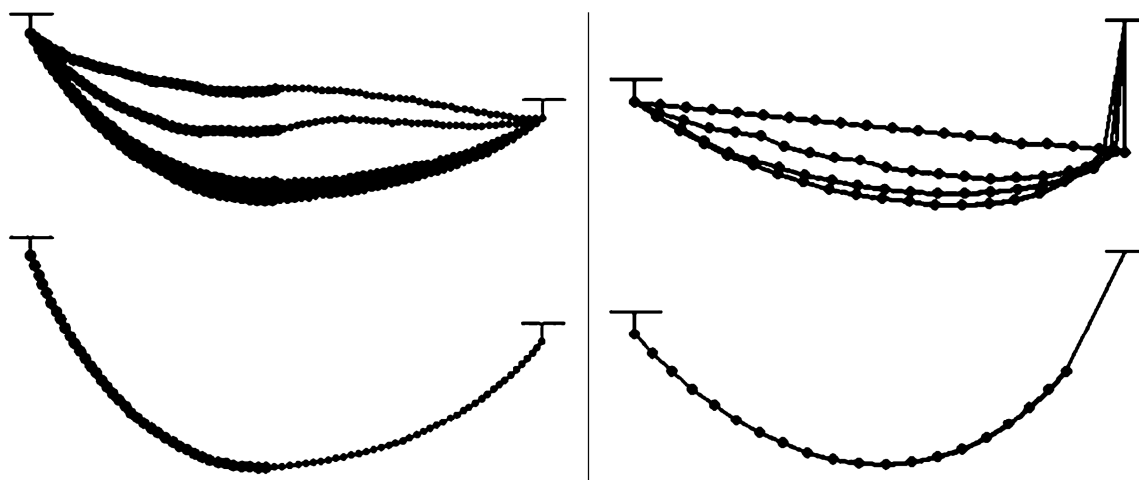


Рис. 3.6. Результаты вычисления формы нити в поле тяжести

Задача 5. Имеется неоднородная нить или цепь. Определите форму, которую она примет в однородном поле тяжести, если ее концы закрепить в фиксированных точках [7].

Нить примет форму, при которой ее потенциальная энергия минимальна. Мысленно заменим нить совокупностью материальных точек с массами m_i , которые связаны друг с другом пружинками жесткостью k и длиной b . Расстояние между соседними частицами и потенциальная энергия всей системы определяются уравнениями:

$$l_i = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}, \quad U = \sum_{i=1}^N \left[k(l_i - b)^2 / 2 + m_i g y_i \right].$$

Допустим, правый конец нити привязан к математическому маятнику длиной L и массой m_N . Заменим нить маятника пружиной с большой жесткостью k , тогда к потенциальной энергии системы следует прибавить величину: $\Delta U = k(l_i - L)^2 / 2$. В программе ТР-5 перебираются материальные точки m_i , случайным образом изменяются их координаты и каждый раз вычисляется соответствующее значение потенциальной энергии

системы. Если при смещении данной частицы потенциальная энергия уменьшилась, то это новое состояние системы принимается, иначе - отвергается. Результат вычислений приведен на рис. 3.6.

3.6. Представление о методе конечных элементов

Если метод конечных разностей применяют для решения краевой задачи для дифференциального уравнения, то перед использованием метода конечных элементов (КЭ) необходимо сформулировать вариационную задачу [1; 3; 5; 11]. **Сущность метода КЭ** состоит в следующем: конструкция (сплошная среда) разбивается на достаточно большое число конечных элементов (областей небольшого размера). Поведение каждого элемента описывается с помощью отдельного набора выбранных функций (механическое напряжение, смещение, температура, потенциал и т. д.). При этом учитывают, что некоторые из них удовлетворяют условиям непрерывности. Составляют функционал F и формулируют **вариационную задачу о нахождении его экстремали**, то есть функции, минимизирующей F . Например, задача о кручении цилиндрического стержня с произвольной формой сечения [11] сводится к уравнению:

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + 2G\theta = 0,$$

где θ - угол закручивания на единицу длины, G - модуль сдвига, φ - функция напряжений. Этому уравнению соответствует функционал:

$$F = \int_s \left[\frac{1}{2} \left(\frac{\partial \varphi}{\partial x} \right)^2 + \frac{1}{2} \left(\frac{\partial \varphi}{\partial y} \right)^2 - 2G\theta\varphi \right] dV.$$

Необходимо найти такую функцию $\varphi(x, y)$, при которой этот функционал достигает минимума.

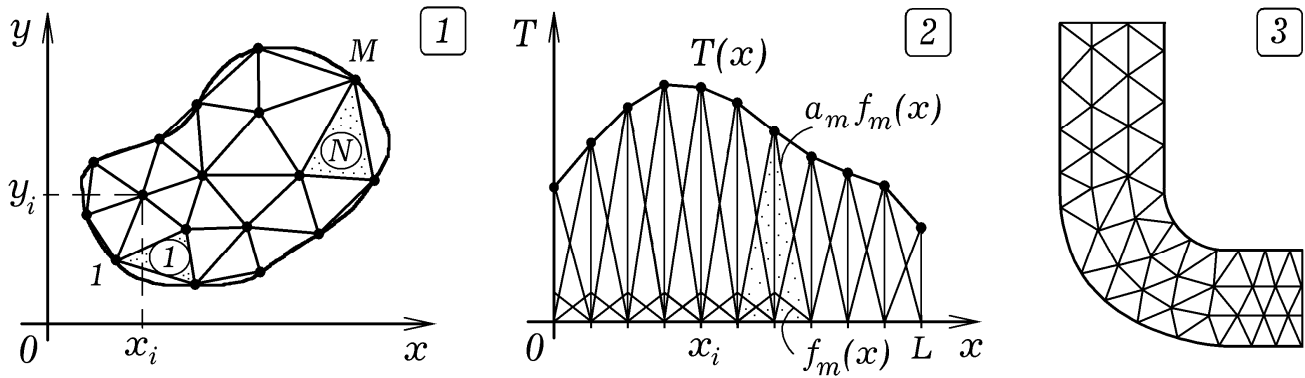


Рис. 3.7. Сущность метода конечных элементов

Рассмотрим задачу о теплопроводности пластины [3]:

$$\frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k \frac{\partial T}{\partial y} \right) + q = 0,$$

с граничными условиями $[k\partial T / \partial N + \alpha T]_L = Q$. Можно показать, что записанному выше ДУЧП соответствует функционал:

$$F[T(x, y)] = \iint_D \left[k \left(\frac{\partial T}{\partial x} \right)^2 + k \left(\frac{\partial T}{\partial y} \right)^2 - 2qT \right] dx dy + \int_L (\alpha T^2 - 2QT) dl.$$

Пусть искомая функция определена в области Ω . Разобьем ее на N элементов (подобластей) и зафиксируем M узловых точек (рис. 3.7.1). В случае когда число узловых точек велико, разбиение на элементы и нумерация узлов осуществляется автоматически. Представим функцию $T(x, y)$ как линейную комбинацию **базисных функций** $f_m(x, y)$:

$$T(x, y) \approx \sum_{m=1}^M a_m f_m(x, y).$$

Базисные функции должны обладать следующими свойствами: 1) в узле t функция $f_m(x, y)$ равна 1, а в остальных узлах - 0; 2) функция $f_m(x, y)$ отлична от нуля только в элементах, содержащих t -ый узел, а в остальной части области Ω она равна 0. Значения коэффициентов a_m примерно равны температуре T_m в t -й узловой точке. В простейшем случае в качестве базисной функции $f_m(x, y)$ используется кусочно-линейная функция

(рис. 3.7.2). Получается, что функционал F зависит только от коэффициентов a_m : $F = F(a_1, a_2, \dots, a_M)$. Он достигает минимума при условии равенства нулю частных производных:

$$\frac{\partial F}{\partial a_1} = 0, \quad \frac{\partial F}{\partial a_2} = 0, \quad \dots, \quad \frac{\partial F}{\partial a_M} = 0.$$

При решении задачи на ЭВМ удобно считать, что функционал F представляют суммой интегралов по всем элементам, на которые разбита область:

$$F = \sum_{n=1}^N F^{(n)}, \quad F^{(n)} = \int_{S^{(n)}} \left[k \left(\frac{\partial T}{\partial x} \right)^2 + k \left(\frac{\partial T}{\partial y} \right)^2 - 2qT \right] dx dy + \int_{L^{(n)}} (\alpha T^2 - 2QT) dl.$$

Последнее слагаемое в выражении для $F^{(n)}$ соответствует граничным элементам. Следует заметить, что от T_m зависят только $F^{(n)}$ тех элементов, которые включают m -й узел. **Условие минимума функционала** можно записать так:

$$\sum_{n=1}^N \frac{\partial F^{(n)}}{\partial a_m} = 0, \quad m = 1, 2, 3, \dots, M.$$

Далее формируют систему алгебраических уравнений путем перебора всех узлов или всех элементов и решают ее методом исключения Гаусса.

Задача отыскания функции, минимизирующей некоторый функционал, решается методом локальных вариаций. Если система сложная, рассматриваемая конструкция неоднородная и состоит из большого числа частей, каждый из которых описывается своим ОДУ, то она решается только методом конечных элементов. Этот метод также используется для анализа задач механики деформируемого тела, теплообмена, гидродинамики и электродинамики. Например, для расчета двумерного течения в изогнутой трубе расчетную область разбивают на КЭ способом, показанным на рис. 3.7.3.

3.7. Примеры использования метода конечных элементов

В качестве примера рассмотрим несколько простых задач на расчет формы упруго деформированного тела. Их решение сводится к нахождению некоторой функции, минимизирующей потенциальную энергию. То есть необходимо решить вариационную задачу, определив координаты различных точек системы, при которых потенциальная энергия минимальна. Алгоритм состоит в следующем. С помощью специальной процедуры перебираются все элементы конструкции и вычисляется суммарная потенциальная энергия U системы. Затем узлы сетки смещают на очень небольшие величины и снова находят потенциальную энергию U_1 . Если она уменьшилась, то эти смещения принимаются, а если нет – отвергаются. После этого все повторяется снова. В результате последовательности итераций определяется искомое состояние системы с наименьшим значением потенциальной энергии. Ограничимся моделированием плоской деформации, при которой точки тела смещаются параллельно плоскости рисунка.

Задача 6. Рассчитайте деформацию фермовой конструкции из 5 звеньев (рис. 3.8.1), к концу которой подвешен груз.

Для решения этой задачи используется программа, в которой перебираются все стержни, вычисляется их удлинение (сжатие) и суммарная потенциальная энергия системы. Рассмотренным выше методом определяется ее состояние, при котором U минимально. Результат решения показан на рис. 3.8.1.

Задача 7. В вертикальную стену вмонтирована горизонтальная балка (консольное закрепление). К ее концу привязан трос с грузом m_1 , перекинутый через неподвижный блок. К средней части балки подвешен груз m_2 . Необходимо рассчитать форму балки в состоянии равновесия.

От реальной конструкции перейдем к конечно-элементной аппроксимации. Сечение балки разобьем на четырехугольные элементы. Потенциальная энергия системы:

$$U = \frac{k}{2} \sum_{i=1}^6 (l_i - L_i)^2 = \frac{k}{2} [(l_1 - L_1)^2 + (l_2 - L_2)^2 + \dots + (l_6 - L_6)^2],$$

где L_i, l_i ($i=1,2,\dots,6$) - длины четырех сторон и двух диагоналей четырехугольного элемента в ненапряженном и деформированном состояниях, k - их коэффициент жесткости. При этом приближенно считается, что этот конечный элемент эквивалентен четырехугольнику, вершины которого соединены упругими стержнями. В отдельной процедуре рассчитывается суммарная энергия элементов 1-2-13-14, 2-3-12-13, 3-4-11-12 и т. д. К этому значению U прибавляется потенциальная энергия грузов $m_1gh_1 + m_2gh_2$. Результаты моделирования - на рис. 3.8.2 и 3.8.3.

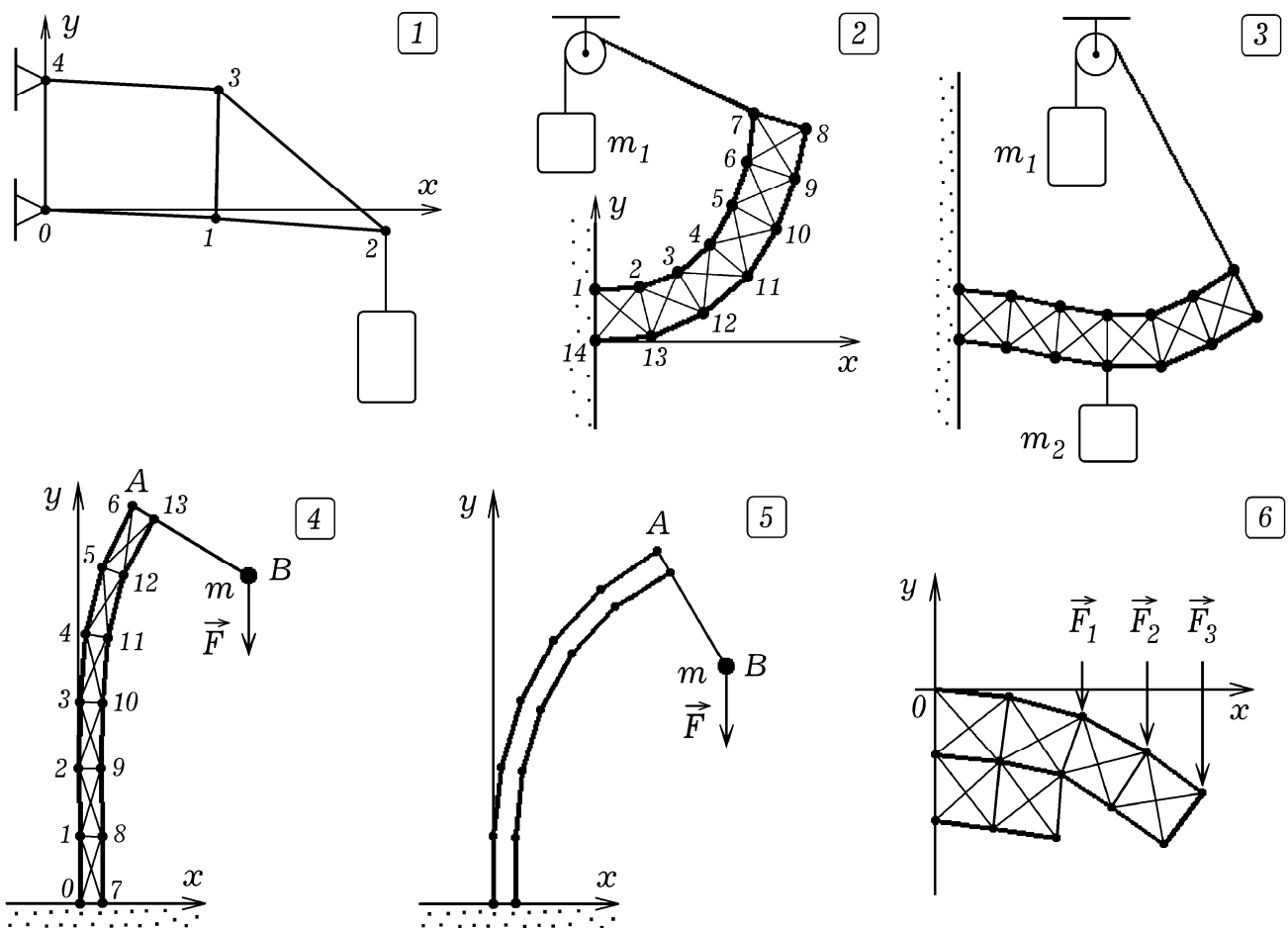


Рис. 3.8. Результаты моделирования деформации упругих тел

Задача 8. Упругий вертикальный стержень закреплен за нижний конец. К его верхней части прикреплена горизонтальная жесткая балка АВ длиной L , на конце которой расположен груз массы m . Рассчитайте форму стержня.

Пронумеруем вершины, как показано на рис. 3.8.4. Используется программа ПР-6. Координаты вершин четырехугольных элементов сохранены в массивах $x[i]$, $y[i]$. С помощью функции $W(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ вычисляется потенциальная энергия 6 стержней, соответствующих четырехугольнику с вершинами α_i , $i = 1, 2, 3, 4$. В процедуре Pot_en рассчитывается суммарная потенциальная энергия всей системы, включая энергию груза на балке АВ (рис. 3.8.5). Аналогично решается следующая задача.

Задача 9. К упругой балке, консольно закрепленной в стене, приложены силы F_1 , F_2 и F_3 , направленные вертикально вниз. Необходимо рассчитать форму балки. Результат моделирования - на рис. 3.8.6.

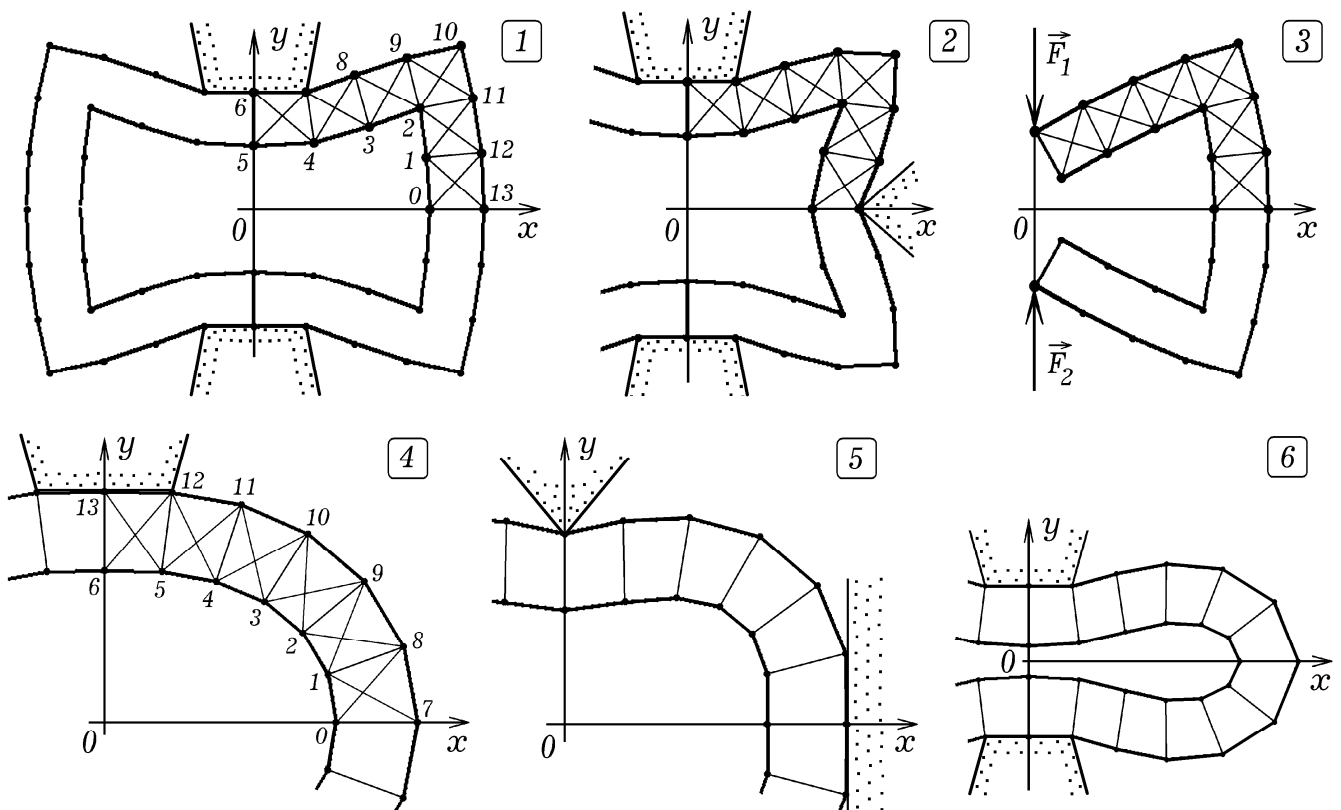


Рис. 3.9. Использование метода конечных элементов

Задача 10. Труба прямоугольного сечения из упругого материала сжимается с противоположных сторон. Рассчитайте ее форму при различных нагрузках.

Для решения задачи достаточно рассмотреть деформацию четверти трубы. Разбиение на элементы, нумерация вершин и результаты моделирования представлены на рис. 3.9.1 и 3.9.2. Листинг используемой программы ПР-7 приведен в Приложении. Тем же способом рассчитывается форма швеллера или П-образной пластины (рис. 3.9.3) и цилиндрической трубы, к которым приложена механическая нагрузка (рис. 3.9.4, 3.9.5 и 3.9.6).

3.8. Точность, сходимость и устойчивость численного метода

Поведение динамических систем описывается функциями непрерывного аргумента. Используемые же для создания компьютерных моделей цифровые ЭВМ дискретны. В самом деле, память ЦЭВМ имеет конечное число разрядов и может отобразить конечное множество чисел. Кроме того, программа ЦЭВМ состоит из отдельных команд, каждая из которых – отдельный акт преобразования информации. Таким образом, **цифровая ЭВМ в отличие от аналоговой в принципе не может плавно изменять свое состояние:** за один “оборот” цикла делается элементарный шаг по пути к решению задачи. Уменьшение шага приводит к увеличению машинного времени, требуемого для решения задачи. Выход состоит в использовании ЦЭВМ с большей разрядностью и высоким быстродействием.

При численном решении дифференциальных уравнений используемые алгоритмы удовлетворяют требованию **сходимости:** при увеличении числа итераций получающиеся результаты вычислений стремятся к истинному решению задачи. Например, ЭВМ рассчитывает форму упругой пленки, натянутой на некоторый каркас. Обходя все узлы сетки, она пере-

считывает смещение элементов, выполняя первую итерацию, затем вторую итерацию и т. д. Используемый алгоритм должен быть таким, чтобы с каждой итерацией искомые смещения пленки в узлах сетки асимптотически стремились к истинным значениям. Понятно, что предпочтительнее тот алгоритм, скорость сходимости которого выше, так как он позволяет решить задачу за меньшее число шагов.

Важной характеристикой компьютерной модели является ее **точность**. Она зависит от ряда факторов: 1) от степени обусловленности задачи; 2) от адекватности системы математических уравнений оригиналу (объекту исследования); 3) от точности метода, используемой вычислительной схемы. Допустим, задача сводится к решению системы линейных алгебраических уравнений $a_1x + b_1y = c_1$ и $a_2x + b_2y = c_2$. Каждому уравнению соответствует прямая на плоскости XOY , а решению системы – точка пересечения этих прямых. Если задача плохо обусловлена, то эти прямые, пересекаясь, образуют небольшой угол. При этом небольшие изменения коэффициентов приведут к существенным изменениям получающихся результатов; координаты точки пересечения будут определены с большой погрешностью. В случае когда задача хорошо обусловлена, графики пересекаются практически под прямым углом и координаты точки пересечения могут быть определены достаточно точно.

Оценить погрешность данного вычислительного алгоритма можно, сравнив даваемое им решение с результатами, полученными аналитически. Например, математически строго можно доказать, что в центральном поле гравитационных сил планета при определенных начальных условиях движется по замкнутой траектории (эллипсу). Если же используемая компьютерная модель дает иной результат (траектория не замкнута), значит расчеты выполнены неверно. Другой способ проверки адекватности модели заключается в проведении нескольких вычислительных экспериментов при различных значениях шага пространственно-временной сетки. Если получаются сильно отличающиеся результаты, значит компьютерная модель нуждается в доработке.

Важно, чтобы использующийся численный метод был **устойчив**. Это означает, что на любом этапе вычислительного процесса погрешности за счет взаимной компенсации уменьшаются и приводят к меньшей конечной ошибке. Иными словами, накопления погрешностей не происходит. При сгущении пространственно-временной сетки ошибки, возникающие в процессе округления, не должны вызывать искажения решения. **Вычислительная устойчивость алгоритма** также означает, что результат вычислений непрерывно зависит от параметров системы, внешнего воздействия на исследуемую систему, начальных или граничных условий. Например, изменение начальной скорости тела на 1% должно приводить к изменению времени прохождения заданного расстояния примерно на 1%.

Разностные схемы решения уравнений могут быть абсолютно неустойчивые, абсолютно устойчивые и условно устойчивые. **Абсолютно неустойчивые** схемы при сколь угодно малых величинах шага h и Δt с каждым оборотом цикла "набирают" погрешность, что приводит к результату, сильно отличающемуся от истинного. **Абсолютно устойчивые** схемы не "раскачиваются" ни при каких значениях шага, но при малых h и Δt дают более точный результат. **Условно устойчивые** схемы при одних значениях шага h и Δt ведут себя устойчиво, а при других значениях - неустойчиво.

В качестве примера условно устойчивой схемы рассмотрим задачу о теплопроводности стержня. Запишем конечно-разностное уравнение:

$$T_{i,j}^{t+1} = T_{i,j}^t + \frac{k\Delta\tau}{\Delta x^2} (T_{i-1}^t - 2T_i^t + T_{i+1}^t).$$

Пусть $T_{i-1}^t = 0$, $T_i^t = 1$, $T_{i+1}^t = 0$, тогда из предыдущей формулы следует:

$$T_{i,j}^{t+1} = 1 + \frac{k\Delta\tau}{\Delta x^2} (0 - 2 \cdot 1 + 0) = 1 - 2 \frac{k\Delta\tau}{\Delta x^2}, \text{ причем } \Delta x, \Delta t > 0.$$

Из физических соображений получается, что в следующий дискретный момент времени $t + 1$ температура в i -ом узле должна лежать в интервале

[0; 1]: $0 < T_i^{t+1} < 1$. Получаем: $0 < 1 - 2k\Delta\tau / \Delta x^2$, $2k\Delta\tau / \Delta x^2 < 1$. Поэтому рассматриваемый метод устойчив, когда $\Delta\tau < \Delta x^2 / (2k)$.

Приложение к главе 3

В приложении представлены тексты программ, позволяющие решить рассмотренные выше задачи. Они написаны в средах Borland Pascal 7.0, Free Pascal 1.0.10.

Программа ПР-1

```
uses crt, graph;
var r1,r2,r3,x1,x2,x3,y1,y2,y3,fi,q1,q2,q3 :real;
Gd,Gm,i,j :integer;
BEGIN Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi\');
  q1:=15; q2:=-23; q3:=8; {q1:=8; q2:=46;} x1:=180.5;
  y1:=180.5; x2:=400.5; y2:=220.5; x3:=240.5; y3:=340.5;
  circle(round(x1),round(y1),3);
  circle(round(x2),round(y2),3);
  For i:=0 to 640 do For j:=0 to 480 do begin
    r1:=sqrt(sqr(x1-i)+sqr(y1-j));
    r2:=sqrt(sqr(x2-i)+sqr(y2-j));
    r3:=sqrt(sqr(x3-i)+sqr(y3-j)); fi:=q1/r1+q2/r2+q3/r3;
    {putpixel(i,j,round((fi+1)*20));}
    If round(fi/0.05)-fi/0.05<0.002 then putpixel(i,j,15);
  end;
  Repeat until Keypressed; CloseGraph;
END.
```

Программа ПР-2

```
uses crt, graph;
var EC,D,DrV,MV,i,j,q1,q2,q3,x1,y1,x2,y2,x3,y3 : integer;
cosa1,cosa2,cosa3,sina1,sina2,sina3,a,a1,a2,r1,r2,r3,
x,y,Ex,Ey,pi: real; Label metka;
BEGIN DrV:=Detect; InitGraph(DrV,MV,'c:\bp\bgi');
  pi:=arctan(1)*4; x1:=150; y1:=240; q1:=15;
```

```

x2:=300; y2:=100; q2:=-20; x3:=450; y3:=400; q3:=10;
circle(x1,y1,2);circle(x2,y2,2);circle(x3,y3,2);
circle(x1,y1,4);circle(x2,y2,4);circle(x3,y3,4);
circle(x1,y1,6);circle(x2,y2,6);circle(x3,y3,6);
For i:=1 to 100 do For j:=1 to 100 do begin
x:=35*i; y:=35*j; r1:=sqrt((x-x1)*(x-x1)+(y-y1)*(y-y1));
r2:=sqrt((x-x2)*(x-x2)+(y-y2)*(y-y2));
r3:=sqrt((x-x3)*(x-x3)+(y-y3)*(y-y3));
If (r1=0)or(r2=0)or(r3=0) then goto metka;
cosa1:=(x-x1)/r1; sina1:=(y-y1)/r1;
cosa2:=(x-x2)/r2; sina2:=(y-y2)/r2;
cosa3:=(x-x3)/r3; sina3:=(y-y3)/r3;
Ex:=q1*cosa1/(r1*r1)+q2*cosa2/(r2*r2)+q3*cosa3/(r3*r3);
Ey:=q1*sina1/(r1*r1)+q2*sina2/(r2*r2)+q3*sina3/(r3*r3);
If Ex<>0 then a:=arctan(Ey/Ex) else a:=pi/2;
line(35*i-round(12*cos(a)),35*j-round(12*sin(a)),
      35*i+round(12*cos(a)),35*j+round(12*sin(a)));
line(35*i-round(12*cos(a))+1,35*j-round(12*sin(a)),
      35*i+round(12*cos(a))+1,35*j+round(12*sin(a)));
metka: end; Repeat until Keypressed; CloseGraph;
END.

```

Программа ПР-3

```

uses crt, graph;
const n=100; m=1; h=1; dt=0.001;
var ii,jj,kk,i,j,DV,MV,EC: integer; q,q1 :real;
    tt,t: array[0..N+1] of real;
    k: array[0..N+1] of real;
BEGIN
DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
EC:=GraphResult; If EC <> grOK then Halt(1);
For i:=1 to N do
  If (i>18)and(i<22) then t[i]:=5 else t[i]:=0.1;
For j:=1 to N do
  If j>20 then k[j]:=1.8 else k[j]:=1;
Repeat
  For i:=2 to N-1 do begin
    If (i>80)and(i<83) then q:=0.5 else q:=0;
    tt[i]:=t[i]+k[i]*(t[i+1]-2*t[i]+t[i-1]))*dt/(h*h)+

```

```

        (k[i+1]-k[i-1])*(t[i+1]-t[i-1])*dt/(4*h*h)+q*dt;
    end;
For i:=2 to N-1 do t[i]:=tt[i];
t[1]:=0.1; t[N]:=t[N-1];
If kk/3000=round(kk/3000) then
    For i:=2 to N do line(i*5+20,420-round(80*t[i]),
        (i-1)*5+20,420-round(80*t[i-1])); inc(kk);
until KeyPressed; CloseGraph;
END.

```

Программа ПР-4

```

uses crt, graph;
const N=100; M=100; a=0.4; R0=3; dr=0.1; dt=0.005;
var i,j,k,Gd,Gm: integer; q,dal,al,pi,r: real;
T: array[0..N+1,0..M+1] of real;
BEGIN pi:=arctan(1)*4; dal:=pi/M/2;
    Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
    Repeat For i:=1 to N do For j:=1 to M do begin
        r:=R0+i*dr; q:=0;
        If (abs(I-40)<6)and(abs(j-70)<6) then q:= -80;
        T[i,j]:=T[i,j]+a*((T[i+1,j]-T[i-1,j])/2/r/dr+
            (T[i-1,j]-2*T[i,j]+T[i+1,j])/dr/dr+(T[i,j-1]-
            2*T[i,j]+T[i,j+1])/dal/dal/r/r)*dt+q*dt;
        T[1,j]:=-50; T[N,j]:=90; T[i,1]:=T[i,2];
        T[i,M]:=T[i,M-1]; end;
        If k mod 100=0 then For i:=1 to N do
            For j:=1 to M do begin r:=30*(R0+i*dr);
                al:=dal*j; setcolor(round(abs(T[i,j]/5+3)));
                circle(100+round(r*cos(al)),20+round(r*sin(al)),2);
            end; inc(k);
    until KeyPressed; CloseGraph;
END.

```

Программа ПР-5

```

{$N+}
uses dos, crt, graph;
const N=20; b=20; b1=100; k=400; g=10; pi=3.1415;
var U,U0,U1,x1,y1,l,l1,dU: Real; i,j,t,Gd,Gm: integer;
    x,y,m: array [1..N] of single; Energiya: string;
Procedure Energy;

```

```

var i: integer;
begin U:=0;
For i:=2 to N do begin
  l:=sqrt(sqr(x[i]-x[i-1])+sqr(y[i]-y[i-1]));
  If l>b then dU:=k*sqr(l-b)/2;
  U:=U+dU+m[i]*g*y[i]; end;
  l1:=sqrt(sqr(x[N]-400)+sqr(y[N]-60));
  U:=U+k*sqr(l1-b1)/2; end;
BEGIN Randomize;
Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
For i:=1 to N do begin x[i]:=20*i; y[i]:=-2*i;
  If i<N then m[i]:=2 else m[i]:=10; end;
Repeat inc(t);
  For i:=2 to N do begin
    x1:=x[i]; y1:=y[i]; Energy; U0:=U;
    x[i]:=x[i]+random(100)/25-2;
    y[i]:=y[i]+random(100)/25-2; Energy; U1:=U;
    If U1>U0 then begin x[i]:=x1; y[i]:=y1; end;
    end; Energy; Str(U,Energiya);
  If t mod 500=1 then begin cleardevice;
    For i:=1 to N-1 do begin
      circle(10+round(x[i]),150-round(y[i]),3);
      line(10+round(x[i]),150-round(y[i]),
          10+round(x[i+1]),150-round(y[i+1])); end;
      line(10+round(x[N]),150-round(y[N]),410,150-60);
      OuttextXY(10,450,Energiya); end;
until KeyPressed; CloseGraph;
END.

```

Программа ПР-6

```

uses crt, graph; Const N=14; M=2;
var dl1,dl2,dl3,dl4,dl5,dl6,U,U1,k,c,x1,y1: real;
v,i,j,DV,MV: integer; x,y,a,b: array[-1..N+1] of real;
Function W(i1,i2,i3,i4:integer): real;
begin dl1:=sqrt(sqr(x[i1]-x[i2])+sqr(y[i1]-y[i2]))-10;
dl2:=sqrt(sqr(x[i2]-x[i3])+sqr(y[i2]-y[i3]))-30;
dl3:=sqrt(sqr(x[i3]-x[i4])+sqr(y[i3]-y[i4]))-10;
dl4:=sqrt(sqr(x[i4]-x[i1])+sqr(y[i4]-y[i1]))-30;
dl5:=sqrt(sqr(x[i1]-x[i3])+sqr(y[i1]-y[i3]))-c;

```

```

dl6:=sqrt(sqr(x[i2]-x[i4])+sqr(y[i2]-y[i4]))-c;
W:=k*(sqr(dl1)+sqr(dl2)+sqr(dl3)+sqr(dl4)+sqr(dl5)
+sqr(dl6)); end;
Procedure Pot_En;
begin U:=W(1,8,7,0)+W(2,9,8,1)+W(3,10,9,2)+W(4,11,10,3)+
W(5,12,11,4)+W(6,13,12,5)+200*(y[13]-5*(y[6]-y[13])); end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi'); Randomize;
For i:=0 to 6 do begin x[i]:=0; y[i]:=30*i; end;
For i:=7 to 13 do begin j:=i-7; x[i]:=10; y[i]:=30*j; end;
k:=400; c:=sqrt(sqr(x[0]-x[8])+sqr(y[0]-y[8]));
Repeat Pot_En; U1:=U;
  x1:=x[13]+5*(x[13]-x[6]); y1:=y[13]-5*(y[6]-y[13]);
  For i:=0 to N do begin a[i]:=x[i]; b[i]:=y[i];
  x[i]:=x[i]+random(100)/1000-0.05;
  y[i]:=y[i]+random(100)/1000-0.05; end; y[0]:=0; x[0]:=0;
  y[7]:=0; x[7]:=10; If x[1]<0 then x[1]:=0;
  If x[3]<0 then x[3]:=0; If x[5]<0 then x[5]:=0; Pot_En;
  If U>U1 then begin For i:=0 to 13 do begin
    x[i]:=a[i]; y[i]:=b[i]; end; end;
  inc(v); If v>5000 then v:=0;
  If (U<U1)and(v mod 50=0) then begin cleardevice;
  For i:=0 to 13 do begin
    circle(90+round(x[i]*M),400-round(y[i]*M),2);
    line(90+round(x[i]*M),400-round(y[i]*M),
      90+round(x[i-1]*M),400-round(y[i-1]*M)); end;
  circle(90+round(x1*M),400-round(y1*M),5); end;
until KeyPressed; CloseGraph;
END.

```

Программа ПР-7

```

uses crt, graph; const N=13; g=9.8; m=5;
c: array[0..N]of real=(30,30,30,20,10,0,0,10,20,30,40,40,40,
40); d: array[0..N]of real=(0,10,20,20,20,20,30,30,30,30,30,
20,10,0);
var dl1,dl2,dl3,dl4,dl5,dl6,U,U1,k: real;
i,DV,MV: integer; x,y,a,b: array[0..N] of real;
Function W(i1,i2,i3,i4:integer):real;
begin dl1:=sqrt(sqr(x[i1]-x[i2])+sqr(y[i1]-y[i2]))-10;
dl2:=sqrt(sqr(x[i2]-x[i3])+sqr(y[i2]-y[i3]))-10;

```

```

dl3:=sqrt(sqr(x[i3]-x[i4])+sqr(y[i3]-y[i4]))-10;
dl4:=sqrt(sqr(x[i4]-x[i1])+sqr(y[i4]-y[i1]))-10;
dl5:=sqrt(sqr(x[i1]-x[i3])+sqr(y[i1]-y[i3]))-14.1;
dl6:=sqrt(sqr(x[i2]-x[i4])+sqr(y[i2]-y[i4]))-14.1;
W:=k*(2*sqr(dl1)+sqr(dl2)+2*sqr(dl3)+sqr(dl4)+sqr(dl5)
+sqr(dl6)); end;
Procedure Pot_En;
begin U:=W(0,1,12,13)+W(1,2,11,12)+W(2,9,10,11)+W(2,3,8,9)+
W(3,4,7,8)+W(4,5,6,7)+40*y[6]{+60*x[13]}; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi'); Randomize;
For i:=0 to N do begin x[i]:=c[i]; y[i]:=d[i]; end; k:=20;
Repeat
  Pot_En; U1:=U;
  For i:=0 to N do begin a[i]:=x[i]; b[i]:=y[i];
    x[i]:=x[i]+random(20)/100-0.1;
    y[i]:=y[i]+random(20)/100-0.1; end;
  x[5]:=0; x[6]:=0; y[0]:=0; y[13]:=0;
  If y[7]>y[6] then y[7]:=y[6]; Pot_En; If U>U1 then begin
  For i:=0 to N do begin x[i]:=a[i]; y[i]:=b[i]; end; end;
  If U<U1 then begin cleardevice;
  For i:=1 to N do begin
    circle(20+round(x[i]*M),300-round(y[i]*M),2);
    line(20+round(x[i]*M),300-round(y[i]*M),
      20+round(x[i-1]*M),300-round(y[i-1]*M)); end; end;
until KeyPressed; CloseGraph;
END.

```

Список литературы

1. Галагер Р. Метод конечных элементов: Основы. М.: Мир, 1984. 428 с.
2. Гулд Х., Тобочник Я. Компьютерное моделирование в физике. В 2 ч. Ч. 2. М.: Мир, 1990. 400 с.
3. Дульнев Г. Н., Парфенов В. Г., Сигалов А. В. Применение ЭВМ для решения задач теплообмена: учеб. пособие для теплофизич. и теплоэнергетич. спец. вузов. М.: Высш. шк., 1990. 207 с.

4. Ильина В. А., Силаев П. К. Численные методы для физиков-теоретиков. II. Москва-Ижевск: Институт компьютерных исследований, 2004. 118 с.
5. Коннор Дж., Бреббиа К. Метод конечных элементов в механике жидкости. Л.: Судостроение, 1979. 264 с.
6. Кунин С. Вычислительная физика. М.: Мир, 1992. 518 с.
7. Майер Р. В. Задачи, алгоритмы, программы: электронное учеб. пособие. Глазов: Глазов. гос. пед. ин-т, 2012. URL: <http://maier-rv.glazov.net>
8. Майер Р. В. Компьютерное моделирование физических явлений. Глазов: Глазов. гос. пед. ин-т, 2009. 112 с.
9. Поттер Д. Вычислительные методы в физике. М.: Мир, 1975. 392 с.
10. Рябенский В. С. Введение в вычислительную математику: учеб. пособие. М.: Физматлит, 2000. 296 с.
11. Сегерлинд Л. Применение метода конечных элементов. М.: Мир, 1979. 392 с.
12. Тихонов А. Н., Самарский А. А. Уравнения математической физики. М.: Наука, 1966. 724 с.
13. Федоренко Р. П. Введение в вычислительную физику: учеб. пособие для вузов. М.: Изд-во Моск. физ.-техн. ин-та, 1994. 528 с.
14. Черноусько Ф. Л., Баничук Н. В. Вариационные задачи механики и управления (Численные методы). М.: Наука, 1973. 238 с.
15. Шуп Т. Решение инженерных задач на ЭВМ: Практическое руководство. М.: Мир, 1982. 238 с.

[ВВЕРХ](#)