

Глава 10

ЯВЛЕНИЯ ПЕРЕНОСА. АВТОВОЛНОВЫЕ ПРОЦЕССЫ

К явлениям переноса относятся диффузия (перенос массы), теплопроводность (перенос тепла) и внутреннее трение (перенос импульса). Это необратимые процессы, описываемые похожими уравнениями [1; 3-6; 12-16]. Мы в основном будем обсуждать явление теплопроводности, понимая, что в случае диффузии и внутреннего трения задача решается аналогично и отличается лишь интерпретацией результатов.

10.1. Уравнение теплопроводности для однородной среды

Рассмотрим однородную прямоугольную пластину. Пусть задано начальное распределение температуры $T = T(x, y)$, а также мощность q_k и координаты x_k, y_k источников тепла. Определены **краевые условия**: край пластины в некоторых точках имеет постоянную температуру, а в некоторых точках теплоизолирован. Необходимо рассчитать распределение температуры по поверхности пластины в момент времени τ .

Запишем **уравнение теплопроводности** для анизотропной пластины:

$$\frac{\partial T}{\partial \tau} = k_x \frac{\partial^2 T}{\partial x^2} + k_y \frac{\partial^2 T}{\partial y^2} + \frac{q(x, y)}{c\rho},$$

где k_x и k_y - коэффициенты температуропроводности вдоль осей Ox и Oy , q - плотность мощности источников тепла, c - удельная теплоемкость, ρ - плотность среды. Для изотропной пластины $k_x = k_y$. Чтобы решить это уравнение численным методом, рассмотрим трехмерную область Ω в пространстве, образованном координатными осями x , y и

осью времени τ . Построим сетку $\Omega_{\Delta x, \Delta y, \Delta \tau}$ с пространственным шагом $h = \Delta x = \Delta y$ и шагом по времени $\Delta \tau$. При этом мы как бы разбиваем пластину на большое количество элементов квадратной формы, образующих N столбцов и M строк. Каждый элемент однозначно задается парой чисел i, j . От непрерывного времени τ переходим к дискретному времени t так, что $\tau = t\Delta \tau, t = 0, 1, 2, \dots$ [3-6].

Допустим, в некоторый дискретный момент времени t температура элемента i -го столбца j -й строки была равна $T_{i,j}^t$. Чтобы найти ее значение в следующий момент времени $t+1$ (то есть $\tau + \Delta \tau$), запишем уравнение теплопроводности в конечных разностях:

$$\frac{T_{i,j}^{t+1} - T_{i,j}^t}{\Delta \tau} = k_x \frac{T_{i-1,j}^t - 2T_{i,j}^t + T_{i+1,j}^t}{\Delta x^2} + k_y \frac{T_{i,j-1}^t - 2T_{i,j}^t + T_{i,j+1}^t}{\Delta y^2} + \frac{q_{i,j}}{c\rho}.$$

Отсюда следует выразить $T_{i,j}^{t+1}$. Для решения этой задачи необходимо задать два двумерных массива размерностью $N \times M$: один - для температуры $T_{i,j}^t$ в узлах сетки в момент t и другой - для температуры $T_{i,j}^{t+1}$ в следующий момент $t+1$. Используется программа ПР-1 (Приложение). Она содержит цикл по времени t с вложенными в него двумя циклами по i и по j , в которых перебираются все элементы пластины и вычисляются их температуры в следующий момент $\tau + \Delta \tau$ [7; 8].

На рис. 10.1.1 представлены результаты расчета температурного поля при наличии источника тепла и источника холода, а на рис. 10.1.2 - при наличии двух источников тепла (или холода). В обоих случаях пластина имеет отверстие (в форме квадрата или круга), на границах которого поддерживается постоянная температура. Нижний край пластины теплоизолирован, поэтому изотермы пересекают его под прямым углом. Остальные края пластины имеют постоянную температуру.

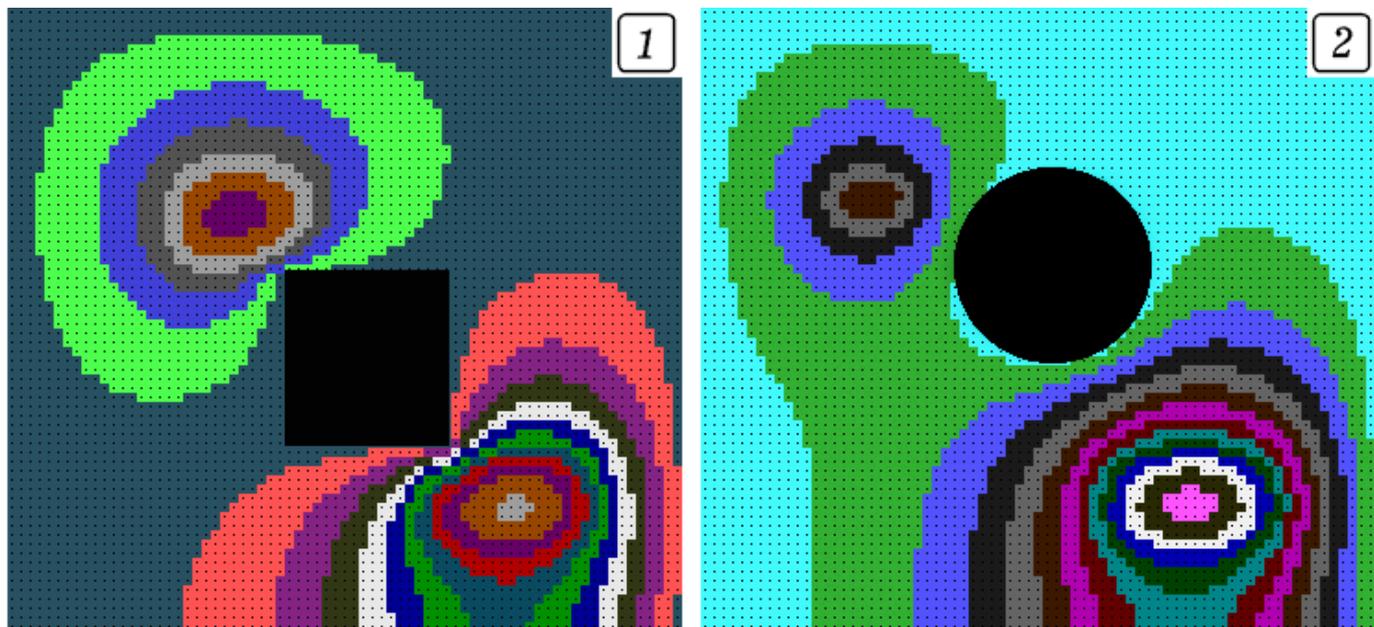


Рис. 10.1. Задача о теплопроводности пластины

Аналогично решается задача в одномерном случае. Допустим, однородный стержень разрезали на две половины, левую нагрели до температуры T_1 , а правую - до температуры T_2 , после чего половинки привели в соприкосновение. Рассчитаем температуру различных точек стержня в последующие моменты времени.

Рассматриваемое явление подобно процессу выравнивания концентрации молекул одной из двух жидкостей за счет диффузии. Если оба конца стержня, получившегося после совмещения половинок, поддерживаются при постоянных температурах T_1 и T_2 , то система стремится к стационарному состоянию, в котором температура линейно зависит от координаты (рис. 10.2.1). Если правый конец стержня теплоизолирован, а левый поддерживается при фиксированной температуре T_1 , то температура всех точек стержня будет повышаться, стремясь к T_1 (рис. 10.2.2).

На рис. 10.2.3 представлены результаты расчетов, когда в начальный момент температура средней части стержня равна T_1 , а во всех остальных точках - T_2 . Температура концов стержня поддерживается постоянной и равна T_2 . Видно, что с течением времени за счет теплообмена происходит выравнивание температуры различных точек среды. Аналогичное выравнивание

нивание концентраций молекул жидкости происходит в процессе диффузии. Несложно промоделировать ситуацию, когда вблизи центра стержня имеется несколько источников тепла положительной и отрицательной мощности, а начальное распределение температуры $T^0(x)$ вдоль стержня задано как-то иначе.

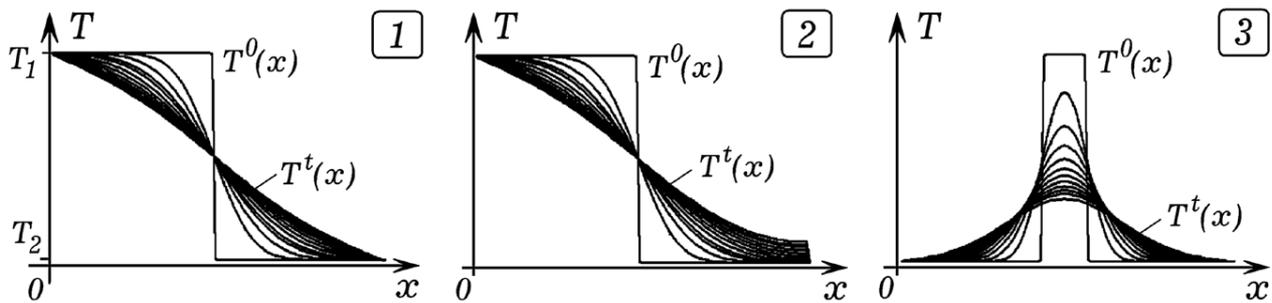


Рис. 10.2. Решение задачи о теплопроводности стержня

10.2. Уравнение адвекции-диффузии и его решение

Адвекцией называется перенос вещества (или тепла) в горизонтальном направлении. Рассмотрим задачу о газовой атаке. Пусть в момент t в точке $A(x, y)$, расположенной над поверхностью земли, начинают выделяться вредные вещества (ВВ) со скоростью q (рис. 10.3). Параллельно поверхности земли дует ветер со скоростью v . Необходимо промоделировать это явление и рассчитать концентрацию $C(t)$ вредных веществ в точках B, C, D, ... в различные моменты времени t [12-16].

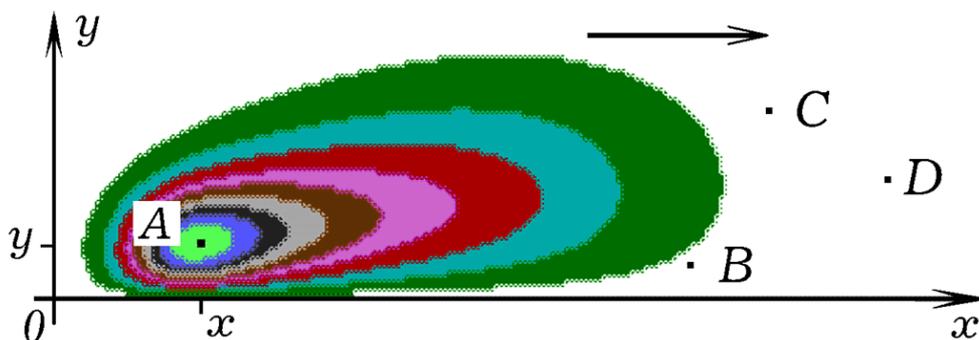


Рис. 10.3. К задаче о газовой атаке

Запишем уравнение адвекции-диффузии в декартовых координатах:

$$\frac{\partial C}{\partial t} = D \left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} \right) - v \frac{\partial C}{\partial x} + q(x, y, t).$$

В конечных разностях получаем:

$$C_{i,j}^{t+1} = C_{i,j}^t + k \frac{C_{i-1,j}^t + C_{i+1,j}^t - 4C_{i,j}^t + C_{i,j-1}^t + C_{i,j+1}^t}{h^2} \Delta \tau + v \frac{C_{i,j}^t - C_{i+1,j}^t}{h} \Delta \tau + q_{i,j}^t \Delta \tau,$$

где $h = \Delta x = \Delta y$. Используется программа ТР-2, в ней последовательно перебираются все узлы двумерной сетки и рассчитываются значения концентрации $C_{i,j}$ в следующий момент времени $t + 1$. Результаты компьютерного моделирования представлены на рис. 10.4. Области, имеющие различные концентрации ВВ, закрашены разными цветами.

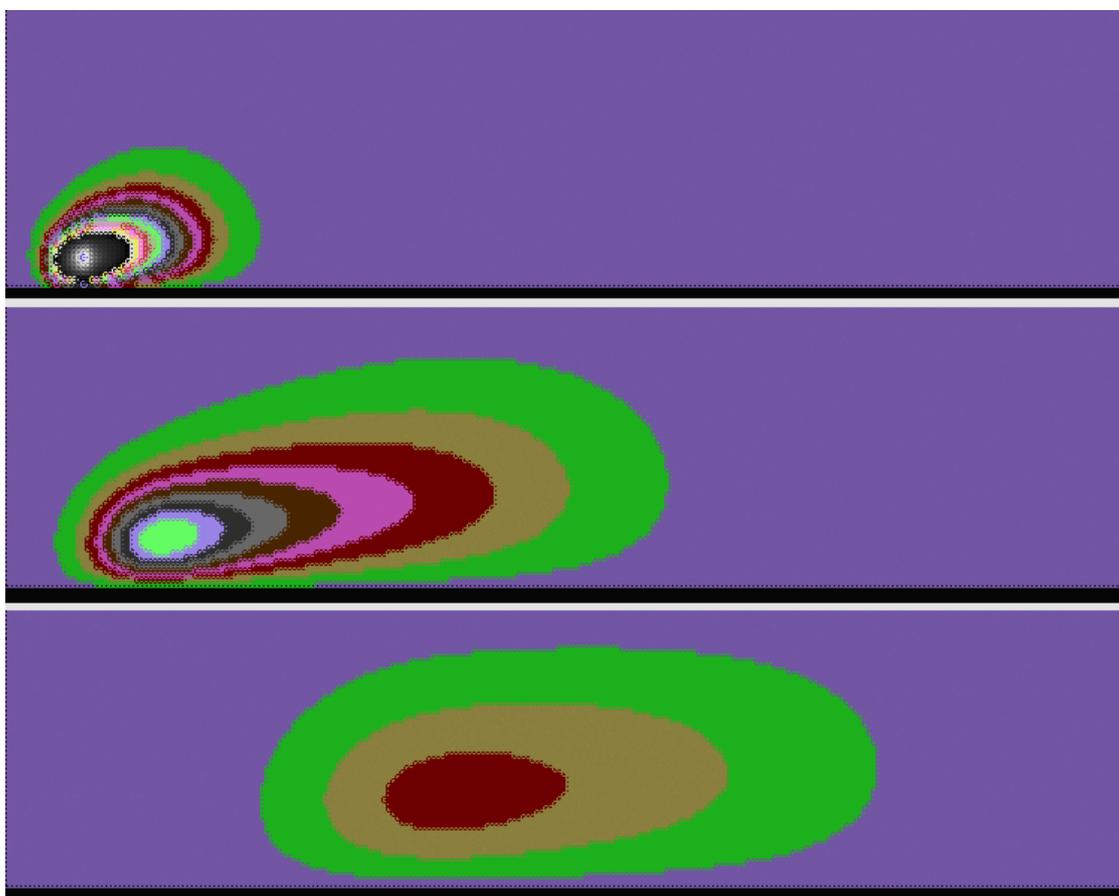


Рис. 10.4. Образование и перемещение облака ВВ с течением времени

Рассмотренная компьютерная модель позволяет изучить изменение концентрации $ВВ$ в различных точках среды, например, в точках $B(50,23)$, $C(90,23)$, $D(130,23)$, $E(170,23)$, $F(210,23)$, $G(250,23)$, лежащих на одной горизонтали. На рис. 10.5 представлены получающиеся графики изменения концентрации вредных веществ в перечисленных точках с течением времени. Если облако с $ВВ$ проходит через точку наблюдения, то концентрация $ВВ$ растет, достигает максимума, а затем снижается.

На первый взгляд кажется, что чем дальше от источника A находится точка наблюдения, тем позже до нее дойдет облако с $ВВ$ и тем ниже максимальное значение концентрации $ВВ$ в этой точке. Графики на рис. 10.5 показывают, что не всегда максимальное значение концентрации $ВВ$ в точке наблюдения тем больше, чем ближе она к источнику: это зависит от скорости ветра, коэффициента диффузии, расстояния от точки наблюдения до поверхности земли и т. д. Так, точка D находится дальше, чем B и C , но максимальный уровень концентрации $ВВ$ в ней выше.

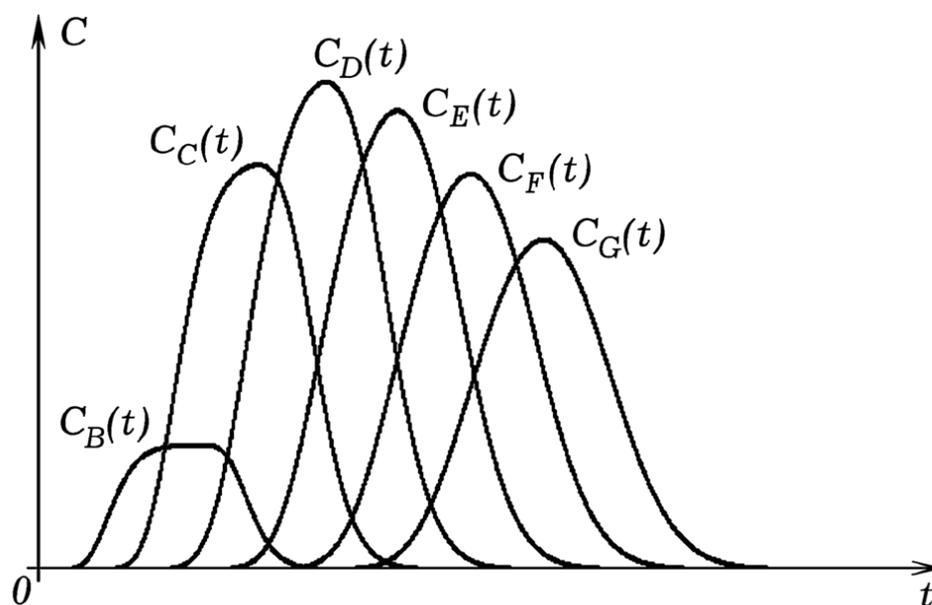


Рис. 10.5. Изменение концентраций $ВВ$ в точках B, C, D, E, F, G

Проанализируем еще одну задачу. Между двумя параллельными пластинами течет без завихрений вязкая жидкость. Внутри потока имеется источник тепла известной мощности. Гравитационное поле отсутствует,

поэтому конвекции нет. Рассчитаем поле скоростей и температур. Для этого ось Ox направим в направлении течения, а ось Oy - перпендикулярно пластинам. Распределение скоростей и температур может быть найдено из уравнений переноса [9-12]:

$$\frac{dv_x}{dt} = \nu \frac{\partial^2 v_x}{\partial y^2} - \frac{1}{\rho} \frac{\partial p}{\partial x}, \quad \frac{\partial T}{\partial t} + v_x \frac{\partial T}{\partial x} = k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + \frac{q(x, y)}{c\rho}.$$

Для моделирования этого явления используется программа ПР-3 [7], содержащая цикл по времени, в котором сначала рассчитывается поле скоростей (первые 300 итераций), а затем поле температур. Вблизи поверхности трубы скорость жидкости равна 0, а на оси трубы скорость максимальна (рис. 10.6). Нагретые слои жидкости смещаются вдоль оси Ox .

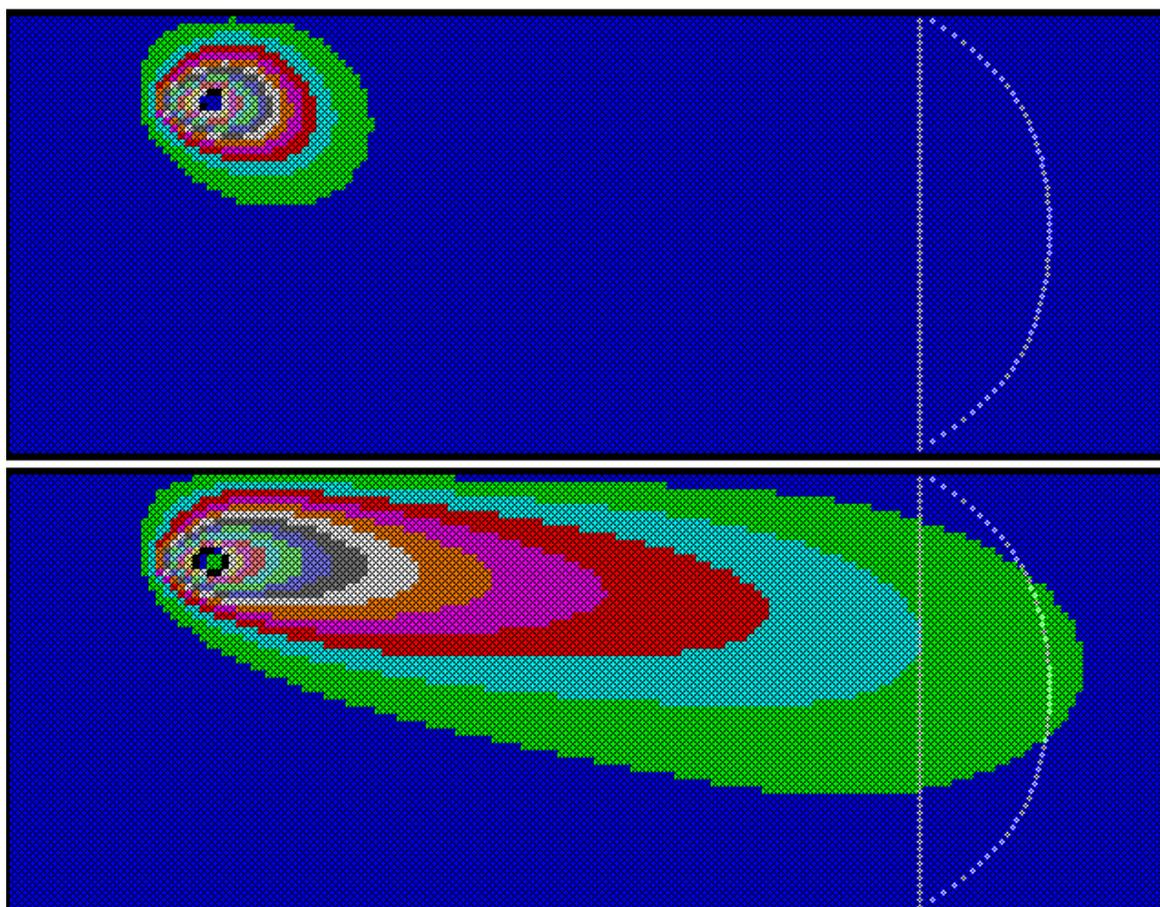


Рис. 10.6. Результаты расчета поля температур и скоростей

Если нагретое тело поместить в газообразную среду, находящуюся в однородном поле тяжести, то нагретые и более легкие слои газа будут

подниматься вверх. Пусть нагреватель представляет собой систему параллельных стержней, лежащих в горизонтальной плоскости и разогреваемых протекающим по ним током. Он установлен в жидкой (газообразной) среде, над ним находятся твердые тела с низким коэффициентом теплопроводности. Ось Ox направим вправо, а ось Oy вертикально вниз; при этом нагретые стержни перпендикулярны плоскости xOy .

Построим сетку $\Omega_{\Delta x, \Delta y, \Delta \tau}$. Из-за подъема теплого воздуха вверх, тепло от нижних узлов сетки передается верхним лучше, чем от верхних нижним. Будем приближенно считать, что если градиент температуры направлен вниз, то есть $T_{i,j+1} > T_{i,j-1}$, то температура $T_{i,j}$ элемента с координатами i и j в следующий момент $t+1$ повышается на $b(T_{i,j+1} - T_{i,j-1})\Delta \tau$, в противном случае она понижается на ту же величину. На рис. 10.7 представлены результаты моделирования; хорошо видны восходящие потоки жидкости или газа (похожие на языки пламени), которые поднимаются вверх и касаются внесенных в среду твердых тел.

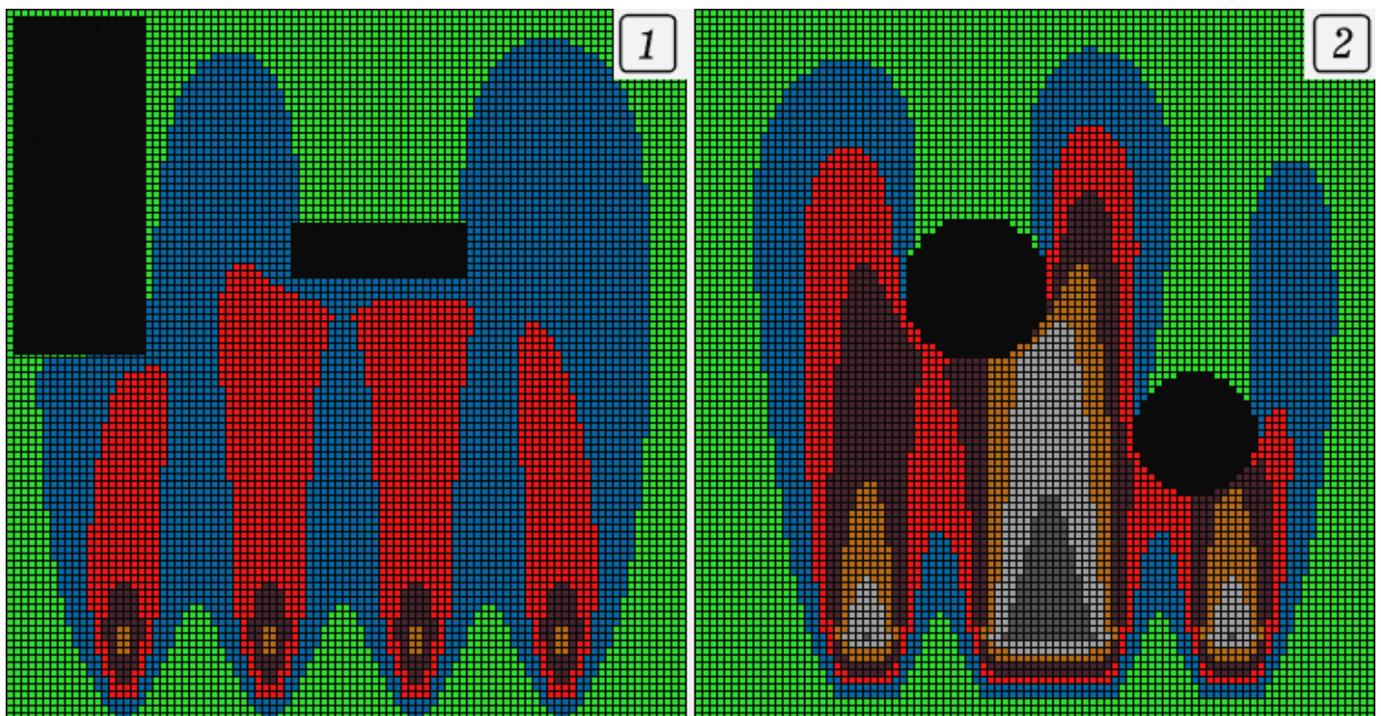


Рис. 10.7. Моделирование теплопроводности в газах

10.3. Решение уравнения теплопроводности для неоднородной среды

Имеется неоднородная пластина. Задано начальное распределение температуры, известны координаты и мощности источников тепла и холода. Определены граничные условия [3; 12], например, края пластины теплоизолированы или поддерживаются при постоянной температуре. Чтобы рассчитать распределение температуры по поверхности пластины в последующие моменты времени, запишем уравнение теплопроводности для неоднородной среды:

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

$$\frac{\partial T}{\partial \tau} = \frac{\partial k}{\partial x} \frac{\partial T}{\partial x} + k \frac{\partial^2 T}{\partial x^2} + \frac{\partial k}{\partial y} \frac{\partial T}{\partial y} + k \frac{\partial^2 T}{\partial y^2} + \frac{q}{c\rho}.$$

Для моделирования теплопроводности неоднородной пластины необходимо задать три двумерных массива размерностью $N \times M$: один - для коэффициента температуропроводности $k_{i,j}$, другой - для температуры $T_{i,j}^t$ в узлах сетки в момент t и третий - для температуры $T_{i,j}^{t+1}$ в следующий момент $t+1$. Все эти величины являются действительными числами, поэтому при большом количестве узлов требуется зарезервировать значительный объем памяти. При программировании на языке Pascal выйти из этой ситуации можно различными способами: 1) использовать указатели; 2) если коэффициент температуропроводности меняется не плавно, а скачком, то объявить массив для $k_{i,j}$ массивом целых чисел; 3) использовать метод переменных направлений и записывать результаты расчета $T_{i,j}^{t+1}$ на следующем временном слое $t+1$ в тот же массив, что и на предыдущем слое t ; 4) использовать другую среду программирования, в которой возможно организовать большой массив [7; 8].

Более подробно рассмотрим третий подход: результаты расчета температуры записываются в тот же массив $T_{i,j}$. Это не совсем правильно: для расчета температуры $T_{20,10}^{t+1}$ 20-го элемента 10 строки требуются значения $T_{19,10}^t, T_{20,10}^t, T_{21,10}^t$ на предыдущем временном слое t , а переменная $T[19,10]$ на этом шаге уже содержит значение $T_{19,10}^{t+1}$. При достаточно малом шаге по времени $\Delta\tau$ значения $T_{19,10}^{t+1}$ и $T_{19,10}^t$ отличаются незначительно. Однако при многократном прохождении элементов 10 строки слева направо ошибки накапливаются. В результате при симметричном начальном распределении температуры и источников тепла получающееся поле температур не обладает симметрией. Исправить этот недостаток можно, поочередно изменяя направление счета: сначала - строчки сверху вниз, элементы в строках - слева направо; затем - строчки снизу вверх, элементы - справа налево и т. д.

Метод переменных направлений предполагает поочередное использование уравнения:

$$T_{i,j}^{t+1} = T_{i,j}^t + \left(\frac{k_{i+1,j} - k_{i-1,j}}{2\Delta x} \cdot \frac{T_{i+1,j}^t - T_{i-1,j}^t}{2\Delta x} + k_{i,j} \frac{T_{i-1,j}^t - 2T_{i,j}^t + T_{i+1,j}^t}{\Delta x^2} + q_{i,j}^t \right) \Delta\tau$$

и второго аналогичного уравнения по j . Применяется алгоритм А-1.

Алгоритм А-1

```

ПРОЦЕДУРА Raschet; {==
ЕСЛИ (i>32)И(i<38)И(j>32)И(j<38) ТО q:=2 ИНАЧЕ q:=0;
ЕСЛИ НАПРАВ=1 ТО T[i,j]:=
    T[i,j]+(k[i,j+1]-k[i,j-1])*(T[i,j+1]-T[i,j-1])*dt/(4*dx*dx)+
        k[i,j]*(T[i,j-1]-2*T[i,j]+T[i,j+1])*dt/(dx*dx)+q*dt
ИНАЧЕ T[i,j]:=
    T[i,j]+(k[i+1,j]-k[i-1,j])*(T[i+1,j]-T[i-1,j])*dt/(4*dx*dx)+
        k[i,j]*(T[i-1,j]-2*T[i,j]+T[i+1,j])*dt/(dx*dx)+q*dt;
==}

```

НАЧАЛО ПРОГРАММЫ

ДЛЯ i:=1 ДО N ДЕЛАТЬ {A=

```

ДЛЯ j:=1 ДО M ДЕЛАТЬ {B=
    ЕСЛИ (j>30) И (j<40) И (i>20) И (i<30) ТО T[i,j]:=200
                                ИНАЧЕ T[i,j]:=0; =B} =A}
ДЛЯ i:=1 ДО N ДЕЛАТЬ {C=
    ДЛЯ j:=1 ДО M ДЕЛАТЬ {D=
        ЕСЛИ (i<35) И (j<35) ТО k[i,j]:=1 ИНАЧЕ k[i,j]:=1.5; =D} =C}
ПОВТОРЯТЬ ДО НАЖАТИЯ НА КЛАВИШУ {E= time:=time+1;
    ДЛЯ i:=2 ДО N-1 ДЕЛАТЬ {F= ДЛЯ j:=2 ДО M-1 ДЕЛАТЬ
        {G= НАПРАВ:=2; ВЫЗОВ ПРОЦЕДУРЫ Raschet; =G} =F}
    ДЛЯ j:=M-1 ДО 2 ДЕЛАТЬ {H= ДЛЯ i:=N-1 ДО 2 ДЕЛАТЬ
        {I= НАПРАВ:=1; ВЫЗОВ ПРОЦЕДУРЫ Raschet; =I} =H}
    ЕСЛИ time mod 10=0 ТО ДЛЯ i:=2 ДО N-1 ДЕЛАТЬ {J=
        ДЛЯ j:=2 ДО M-1 ДЕЛАТЬ {K= УСТАНОВИТЬ ЦВЕТ T[i,j];
            ПОСТАВИТЬ ТОЧКУ (i,j); =K} =J}
=E}
КОНЕЦ ПРОГРАММЫ
    
```

В трехмерном или одномерном случаях задача решается по такой же схеме. Следует помнить, что уравнение теплопроводности описывает и другие явления переноса: диффузию (перенос вещества) и вязкость (перенос импульса). Поэтому для многих задач на теплопроводность можно найти соответствующую задачу, связанную с расчетом концентрации вещества при диффузии или поля скоростей в потоке вязкой жидкости.

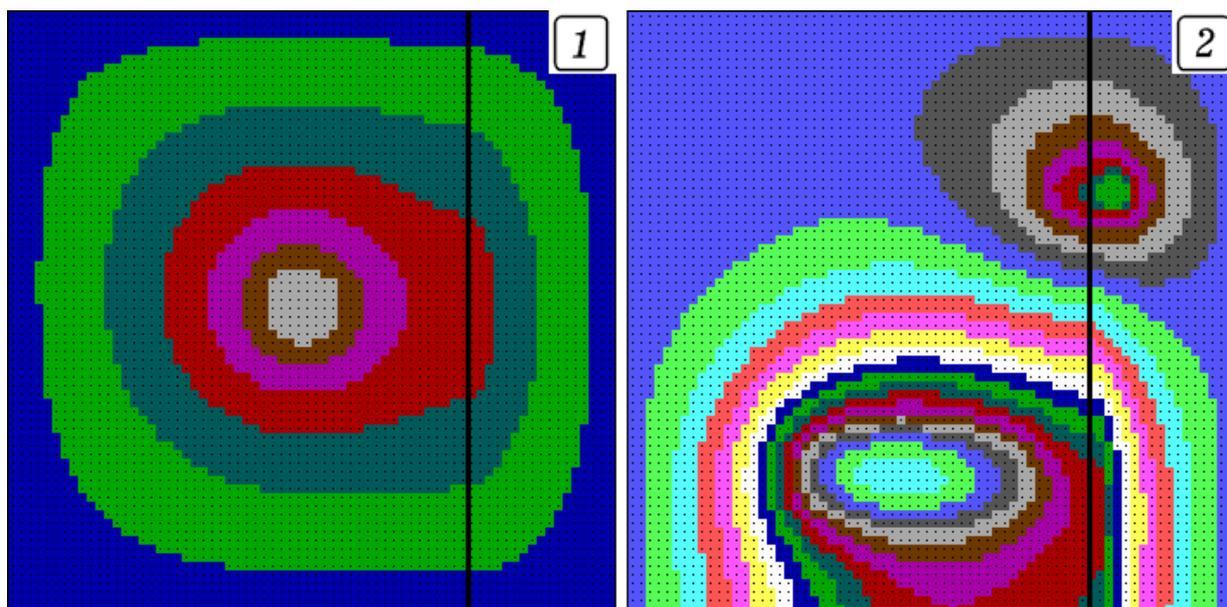


Рис. 10.8. Результаты моделирования теплопроводности пластины

На рис. 10.8 представлены результаты расчетов температурного поля для неоднородной пластины, коэффициент температуропроводности которой равен

$$k_{i,j} = \begin{cases} 0,1 + 0,002i^2 + 0,05j, & \text{если } i \leq 55, \\ 2, & \text{если } i > 55. \end{cases}$$

Используется программа ПР-5. Если все края пластины поддерживаются при постоянной температуре, то изотермы параллельны краю (рис. 10.8.1). На рис. 10.8.2 действуют источники тепла и холода, нижний край пластины теплоизолирован ($T_{i,M}^t = T_{i,M-1}^t$ для всех i), поэтому изотермы пересекают его под прямым углом. Черная вертикальная линия соответствует $i = 55$.

10.4. Расчет поля температур в цилиндрической и сферической системах координат

Пусть по цилиндрической трубе радиуса R течет жидкость; ее скорость на оси трубы равна v_0 . Внутри трубы расположен кольцевой электронагреватель радиуса R_1 . Нагреватель включают на некоторое время t , а затем выключают. Необходимо рассчитать распределение температуры внутри трубы в последующие моменты времени.

Ось Oz совместим с осью трубы, а ось r направим перпендикулярно ее стенке. Пренебрежем трением жидкости о нагреватель. Скорость слоев, удаленных от оси трубы на расстояние r , равна $v(r) = v_0(1 - r^2/R^2)$. Задача обладает цилиндрической симметрией, запишем **уравнение теплопроводности в цилиндрической системе координат** [10-16]:

$$\frac{\partial T}{\partial t} = D \left(\frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial z^2} \right) - v(r) \frac{\partial T}{\partial z} + \frac{q(z, r, t)}{c\rho}.$$

Используется программа ПР-6; результаты вычислений поля температур через некоторое время после включения нагревателя представлены на

рис. 10.9. Видно, что нагреватель повышает температуру движущихся мимо него слоев жидкости; нагретый объем смещается вправо.

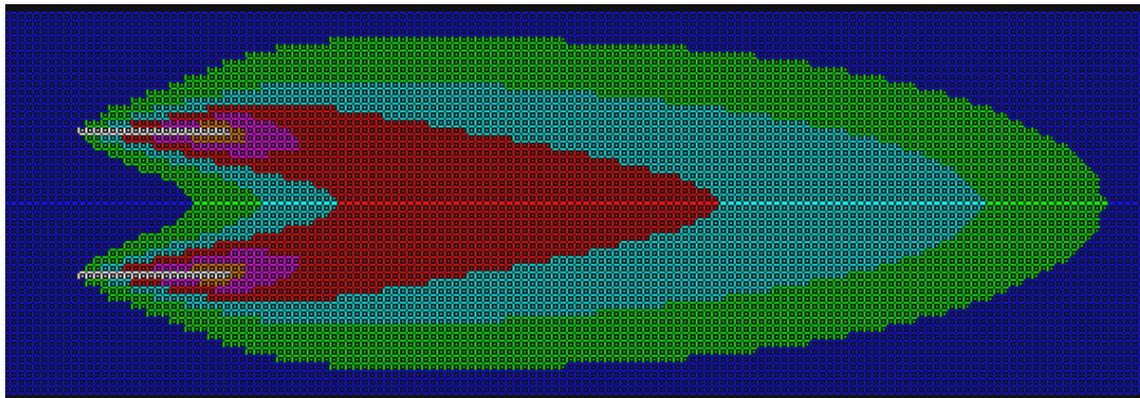


Рис. 10.9. Поле температур при включении кольцевого нагревателя

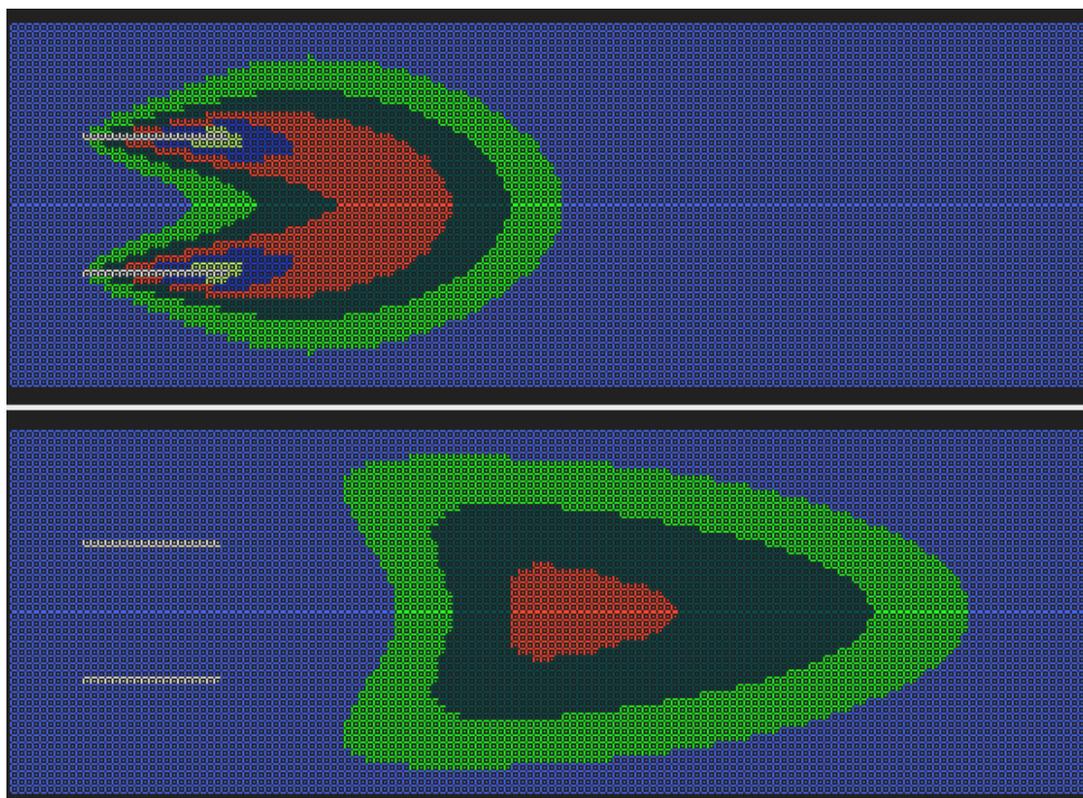


Рис. 10.10. Результат кратковременного включения нагревателя

При кратковременном включении нагревателя образуется нагретая область, которая также смещается в направлении движения жидкости и уменьшается в размерах (рис. 10.10). Решая задачу таким способом, мы пренебрегаем силами вязкого трения, действующими со стороны кольце-

вого нагревателя на движущиеся слои жидкости. Если в программу внести небольшие изменения, то можно промоделировать ситуацию, в которой нагревается небольшой участок трубы внешним источником тепла. Чтобы вместо источника тепла получить источник холода, необходимо поменять знак перед мощностью q .

Теперь решим задачу о теплопроводности шара в сферических координатах. Допустим, металлический шар нагрели до температуры T_0 , а потом опустили в вязкую жидкость (конвекция отсутствует), имеющую температуру T_1 . Шар некоторое время охлаждается, температура его поверхности уменьшается. После этого шар вынули из жидкости и поместили в газообразную среду с низкой теплопроводностью. За счет теплообмена с центральной частью шара температура его поверхности повышается, а температура в центре понижается. Через некоторое время из-за теплообмена с газообразной средой температура поверхности начинает медленно уменьшаться, шар охлаждается. Промоделируем это явление, рассчитаем температуру различных точек шара и среды.

В данном случае температурное поле обладает сферической симметрией. Запишем уравнение теплопроводности в сферических координатах [4; 12; 13]:

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

Так как поле температур не зависит от меридиональной координаты θ и азимутальной координаты φ , то получаем:

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

Отсюда следует конечно-разностное уравнение:

$$T_i^{t+1} = T_i^t + k \left(\frac{T_{i+1}^t - 2T_i^t + T_{i-1}^t}{\Delta r^2} + \frac{2}{r_i} \frac{T_i^t - T_{i-1}^t}{\Delta r} \right) \Delta \tau + \frac{q_i^t}{c\rho} \Delta \tau.$$

При решении задачи следует учесть **граничные условия сопряжения**: если между двумя телами (шаром и вязкой средой) имеется идеальный тепловой контакт, то: 1) их температуры на поверхности контакта одинаковы; 2) по закону сохранения энергии тепловой поток, выходящий из шара, равен тепловому потоку, входящему в среду.

Используется программа ПР-7. Чтобы учесть граничные условия сопряжения, введем функцию $q(t)$, учитывающую теплопередачу от шара к среде. Будем считать, что на поверхности шара (точка с координатой $r = R$) имеется поглотитель тепла с мощностью $q_1(t) = -\alpha(T_n - T_c) < 0$, а в ближайшей точке среды имеется источник тепла с той же по модулю мощностью $q(t) = \alpha(T_n - T_c) > 0$. Здесь T_n и T_c - температуры поверхности шара и прилегающего к ней слоя среды, α - коэффициент теплоотдачи.

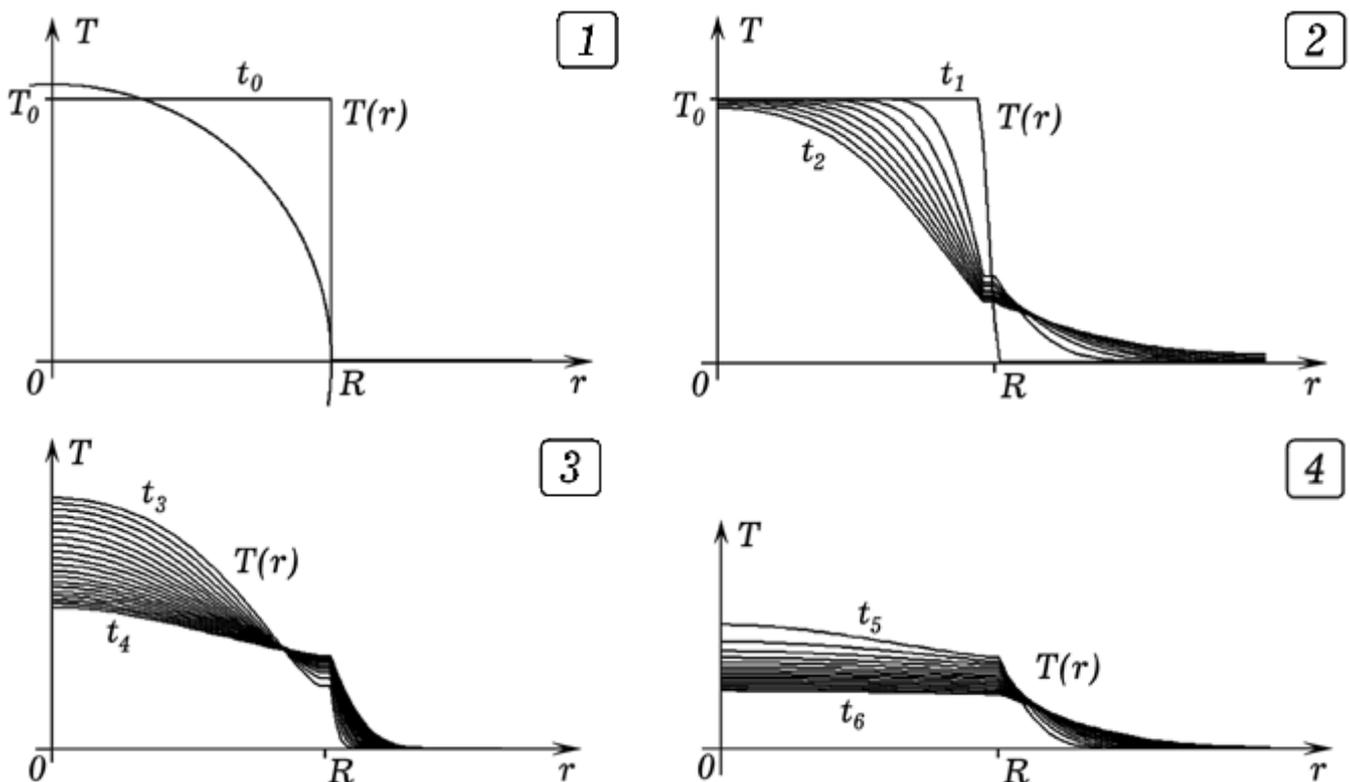


Рис. 10.11. Результаты расчета поля температур при охлаждении шара

На рис. 10.11.1 показано начальное распределение температуры. Пока шар находится внутри жидкости, он охлаждается (рис. 10.11.2), темпера-

тура поверхности становится ниже, чем в центре. Когда шар оказывается в воздухе, то температура поверхности повышается за счет теплообмена с центральной частью шара; теплообмен с окружающей средой происходит существенно медленнее (рис. 10.11.3). Через некоторое время температура поверхности шара начинает уменьшаться из-за теплообмена с окружающей средой (рис. 10.11.4). Если жидкость, в которую опускают нагретый шар, имеет небольшую вязкость, то возникают конвективные потоки. При этом можно приближенно считать, что температура всех точек жидкости одинакова и равна некоторой средней температуре T_{cp} . Решение задачи упрощается.

10.5. Одномерная двухфазная задача Стефана

Особый интерес представляют собой компьютерные модели распространения тепла в средах с изменяющимся фазовым состоянием. Среди них задача о промерзании грунта (решена Стефаном в 1889 г.), задача о кристаллизации расплава при погружении в него пластины, задача о нарастании ледяного покрова, металлургическая задача об остывании расплавленных тел и образовании слитка, задача об отвердевании земного шара из расплавленного состояния и другие.

Рассмотрим **одномерную двухфазную задачу Стефана** [17]. Имеется теплопроводящая среда, находящаяся в жидком и твердом состояниях, заполняющая полупространство $x > 0$. В начальный момент ее температура меньше температуры плавления $T_{пл}$. В точках с координатами $x \leq 2$ имеются источники тепла, под воздействием которых происходит плавление на границе раздела фаз, протекающее при температуре $T_{пл}$. Необходимо найти распределение температуры в последовательные моменты времени и определить расположение границы фаз.

Задача сводится к решению уравнения теплопроводности

$$\frac{\partial T}{\partial \tau} = k(x) \frac{\partial^2 T}{\partial x^2} + \frac{q(x, y)}{c\rho}, \text{ где } k = \begin{cases} k_1, & \text{если } x < \xi(t), \\ k_2, & \text{если } x \geq \xi(t), \end{cases}$$

совместно с **условием Стефана**, которое следует из уравнения теплового баланса [17]:

$$\lambda\rho \frac{\partial \xi}{\partial x} = - \left(k_1 \frac{\partial T_1(x, t)}{\partial x} - k_2 \frac{\partial T_2(x, t)}{\partial x} \right)_{x=\xi(t)},$$

где λ - удельная теплота плавления, ρ - плотность среды, $x = \xi(t)$ - уравнение границы, разделяющей жидкую и твердую фазы, k_1, k_2 - коэффициенты температуропроводности жидкой и твердой фазы соответственно, T_1 и T_2 - температуры вблизи границы раздела $x = \xi(t)$ со стороны жидкой и твердой фаз.

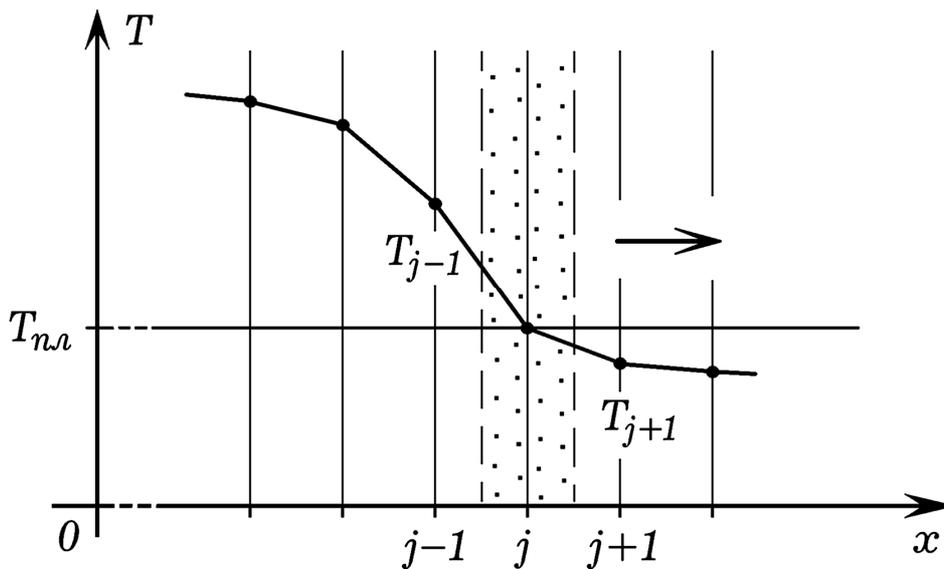


Рис. 10.12. Одномерная двухфазная задача Стефана

Рассмотрим решение задачи Стефана для одномерной среды длиной L , когда температура левого конца все время выше $T_{пл}$, правый конец теплоизолирован, а граница раздела фаз смещается вправо (рис. 10.12). На каждом шаге определяется число узлов j , в которых $T_i > T_{пл}$, $i \leq j$. Температура в j -том узле незначительно отличается от температуры плавления.

ния $T_{нл}$, поэтому переменной T_j присваивается значение $T_{нл}$. Количество теплоты, приходящее в j -тый узел, на шаге t

$$dq^t = \left| k_{j-1} \frac{T_j^t - T_{j-1}^t}{h} - k_{j+1} \frac{T_{j+1}^t - T_j^t}{h} \right|.$$

Здесь k_{j-1} и k_{j+1} - коэффициенты теплопроводности слева и справа от границы раздела двух фаз.

Если количество теплоты dq^t , поступающее в j -й слой, меньше заданного значения теплоты плавления $q_{нл} = \lambda \rho \Delta x$, которую необходимо затратить, чтобы расплавить слой толщиной $h = \Delta x$ единичной площади, то фронт плавления не смещается. ЭВМ вычисляет температуру во 2-м, 3-м, ..., $(j-1)$ -м, а затем в $(j+1)$ -м, $(j+2)$ -м, ..., $(N-1)$ -м узлах. Температура T_j остается равной $T_{нл}$; значение dq^t накапливается в переменной dq . Если dq^t превышает $q_{нл}$, то j -й слой плавится и граница раздела жидкой и твердой фаз смещается вправо на 1 узел. Величина dq^t уменьшается на затраченную энергию $q_{нл}$. Переменной T_{j+1} присваивается значение $T_{нл}$, и все повторяется снова.

Рассмотренный алгоритм реализован в программе ТР-8. В нашем случае теплопроводность жидкой фазы в 3 раза больше, чем у твердой фазы. На рис. 10.13.1 показаны получающиеся распределения температуры $T(x)$ в последовательные моменты времени, если температура левого конца одномерной среды неограниченно растет. На рис. 10.13.2 приведен результат, соответствующий ситуации, при которой температура левого конца стержня повышается до некоторой величины, а затем остается постоянной. В обоих случаях время плавления зависит от количества теплоты $q_{нл}$, необходимого для плавления слоя толщиной $h = \Delta x$ единичной площади, взятого при температуре плавления.

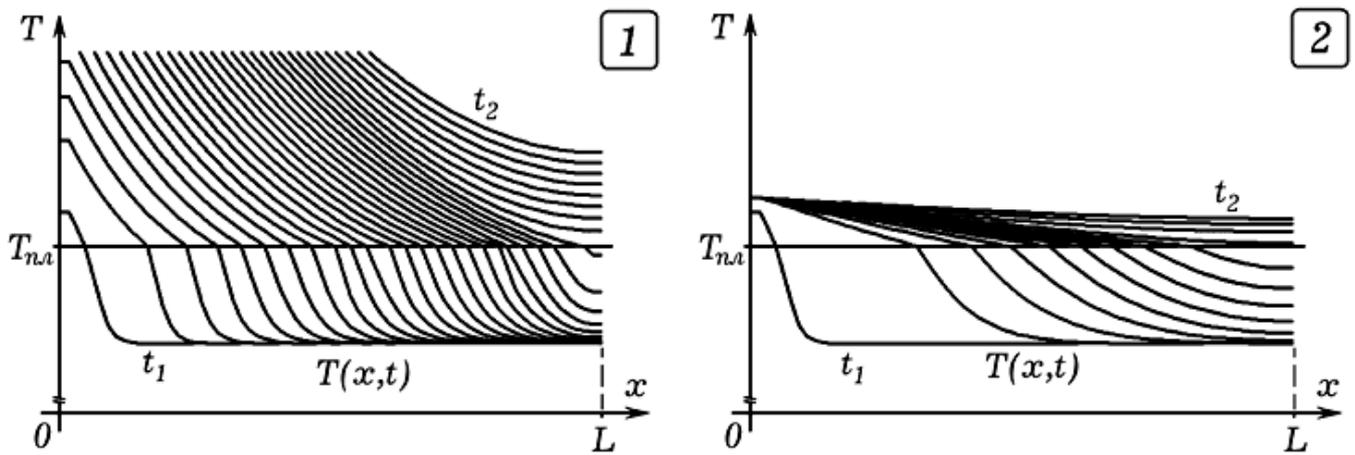


Рис. 10.13. Моделирование плавления стержня (задача Стефана)

10.6. Нелинейное уравнение теплопроводности

К нелинейным задачам теплопроводности относятся: 1) задачи, сводящиеся к уравнениям теплопроводности, коэффициенты которых зависят от температуры; 2) задачи, в которых мощности источников и/или граничные условия являются нелинейными функциями температуры. Такие ситуации возникают, если температура изменяется в широких пределах, и коэффициенты в уравнении меняются в 10 и более раз (горение, процессы, происходящие внутри звезд, и т. д.). Так как перенос тепла при этом происходит в первую очередь за счет излучения, то обычно считают, что коэффициент температуропроводности $k(T) = k_0 T^\sigma$, а плотность мощности источников тепла $q(T) = q_0 T^\beta$, где $\sigma > 0$ и $\beta > 1$. Нелинейное уравнение теплопроводности приобретает вид [10-13]:

$$\frac{\partial T}{\partial \tau} = \frac{\partial}{\partial x} \left(k(T) \frac{\partial T}{\partial x} \right) + q_0 T^\beta = \frac{\partial k}{\partial x} \frac{\partial T}{\partial x} + k \frac{\partial^2 T}{\partial x^2} + q_0 T^\beta.$$

Запишем это уравнение в конечных разностях:

$$\frac{T_i^{t+1} - T_i^t}{\Delta \tau} = \frac{k_{i+1} - k_{i-1}}{2\Delta x} \cdot \frac{T_{i+1}^t - T_{i-1}^t}{2\Delta x} + k_i \frac{T_{i-1}^t - 2T_i^t + T_{i+1}^t}{\Delta x^2} + q_0 (T_i^t)^\beta.$$

$$T_i^{t+1} = T_i^t + \left(\frac{k_{i+1} - k_{i-1}}{2\Delta x} \cdot \frac{T_{i+1}^t - T_{i-1}^t}{2\Delta x} + k_i \frac{T_{i-1}^t - 2T_i^t + T_{i+1}^t}{\Delta x^2} + q_0 (T_i^t)^\beta \right) \Delta \tau.$$

На рис. 10.14 представлен результат решения нелинейного уравнения

$$\frac{\partial T}{\partial t} = k(T) \frac{\partial^2 T}{\partial x^2}, \quad \text{при } k(T) = T^2 / 2,$$

в случае когда температура элемента в центре среды из $N = 200$ элементов изменяется по закону $T_{100}(t) = 10\sqrt{t}$ (программа ПР-9). При этом возникает фронт температуры, распространяющийся с конечной скоростью, образуется **температурная волна**, которая захватывает все большую часть среды (рис. 10.14.1). Эта задача рассмотрена в книге [12, с. 588]. Незначительно изменив программу, можно промоделировать распространение температурной волны, образующейся, когда в центре среды начинает действовать источник тепла, а затем его мощность падает до 0.

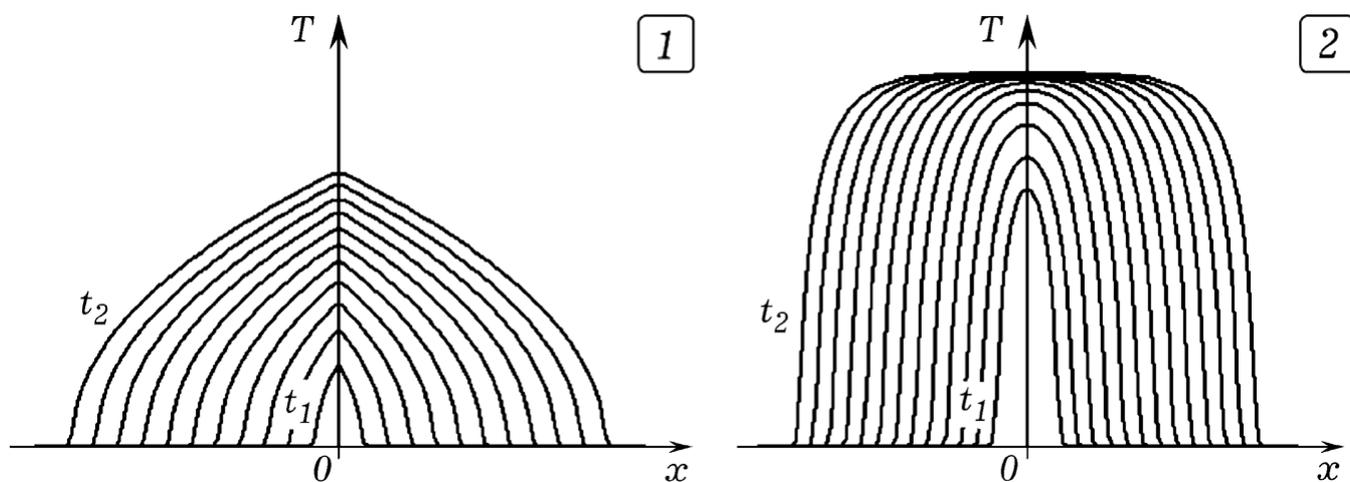


Рис. 10.14. Температурные волны, возникающие в нелинейных средах.

Промоделируем горение в одномерной среде. Будем считать, что при увеличении температуры выделяемое количество тепла растет по закону $q_1(T) = 0,05T$, а излучаемое количество тепла - по закону: $q_2(T) = 0,02T^2$, поэтому уравнение приобретает вид:

$$\frac{\partial T}{\partial t} = k(T) \frac{\partial^2 T}{\partial x^2} + 0,05T - 0,02T^2, \text{ при } k(T) = T^2/2 + 0,01.$$

Чтобы "поджечь" среду, в течение некоторого времени будем увеличивать температуру центрального элемента по закону: $T_{100}(t) = \sqrt{t}$. Результаты моделирования приведены на рис. 10.14.2 (программа ПР-10). Так как при увеличении температуры излучаемая энергия q_2 растет быстрее выделяемой q_1 , то при некоторой температуре наступает динамическое равновесие: $q_1 = q_2$, повышение температуры прекращается. При этом вдоль оси Ox перемещается фронт температуры, область горения расширяется.

Нелинейное уравнение теплопроводности моделирует образование **тепловых структур**. При определенных условиях процесс горения оказывается локализован внутри некоторой области, в то время как вне ее температура равна 0. В течение ограниченного времени **обострения** температура возрастает в сотни раз, стремясь к бесконечности. Возникает упорядоченность, формируется локализованная структура, независимая от начальных условий, происходит самоорганизация.

Теперь промоделируем процесс горения однородного стержня, концы которого поддерживаются при постоянной температуре. Стержень нагревается в центральной части, мощность источника $q_1 = const$. Будем считать, что при температуре ниже температуры возгорания T' температуропроводность $k = 2$. После возгорания коэффициент температуропроводности пропорционален квадрату температуры: $k(T) = 4T^2 + 2$. На горящих участках выделяется тепло; плотность мощности источников тепла задается уравнением: $q = 0,1(T - T')^3$. Используется программа ПР-11 [8]. При ее запуске температура в центре стержня растет, достигает критического значения T' , после чего начинается горение (рис. 10.15.1).

Изменим условия задачи и рассмотрим горение в одномерной нелинейной среде, левый конец которой теплоизолирован, а правый находится в тепловом контакте с термостатом (имеет постоянную температуру).

Допустим, что коэффициент температуропроводности прямо пропорционален температуре: $k(T) = aT$. Количество теплоты, выделяющееся при горении, пропорционально температуре, а тепло, теряемое вследствие теплообмена с внешней средой, зависит от температуры следующим образом: $q = b_1T - b_2T^2$. В начальный момент средняя часть стержня нагрета.

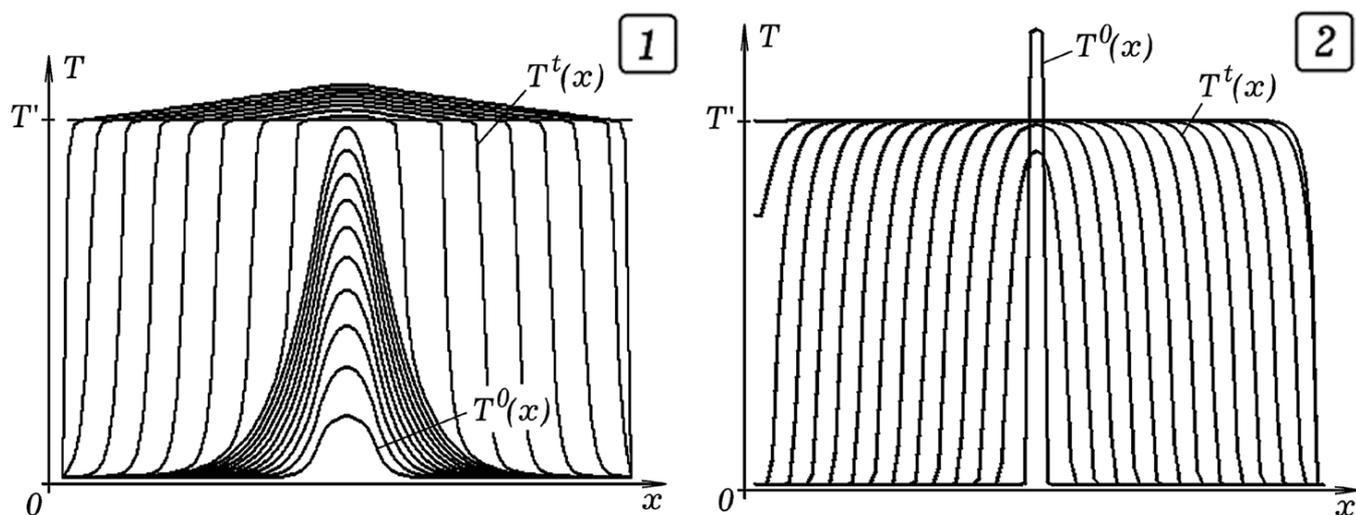


Рис. 10.15. Возгорание и горение одномерной среды

Используемая программа представлена в [8, с. 84], результаты моделирования - на рис. 10.15.2. Температура в центре стержня достаточно высока для того, чтобы произошло возгорание среды. С ростом температуры излучаемая в пространство мощность растет пропорционально T^2 , а выделяющаяся мощность - пропорционально T . Поэтому температура достигает некоторого значения, после чего остается постоянной (наступает динамическое равновесие). Процесс горения охватывает новые элементы стержня, область с высокой температурой расширяется.

10.7. Моделирование автоволновых процессов

Допустим, имеется двумерная активная среда, элементы которой могут находиться в трех состояниях: 1) возбужденном; 2) невозбужден-

ном; 3) в состоянии **рефрактерности**. Например, поле, на котором быстро растет трава. Если траву поджечь (высокая температура является активатором), то начнется реакция горения, сопровождающаяся выгоранием травы и выделением дыма (ингибитора, препятствующего горению). Рассмотренная в главе 4 модель автоволновых процессов, основанная на использовании метода клеточных автоматов, не совсем корректна из-за своей дискретности. В реальных средах происходят плавные изменения концентраций **активатора** и **ингибитора**, описываемые уравнениями теплопроводности и диффузии. Получающаяся **двухкомпонентная модель автоволнового процесса** позволяет изучить распространение автоволн и их аннигиляцию, образование однорукавной волны и т. д.

Построим математическую модель. В случае распространения огня по полю, на котором быстро растет трава, активатором является температура T : когда она достигает критической температуры возгорания T_{KP} , трава загорается. Концентрация ингибитора v тем больше, чем больше дыма и меньше травы в данной точке. Когда концентрация ингибитора достигает порогового значения v_{KP} (много дыма, вся трава выгорела), химическая реакция прекращается. Роль концентрации ингибитора играет величина $u(x, y) = u_0 - v(x, y)$, где u_0 - максимальная плотность травы, а v лежит в интервале от 0 до u_0 . Для одномерной среды получаем систему из двух дифференциальных уравнений [2, с. 44-81]:

$$\frac{\partial T}{\partial t} = a \frac{\partial^2 T}{\partial x^2} + P(T, u), \quad \frac{\partial u}{\partial t} = b \frac{\partial^2 u}{\partial x^2} + Q(T, u) + ru(u_0 - u),$$

которые похожи на уравнения теплопроводности. Когда трава горит ($T_i > T_{KP}$ и $u > u_{KP}$), выделяется мощность $P = P_1$, а количество травы уменьшается со скоростью Q . При горении температура среды не может превышать bT_{KP} . Если трава сгорела, но температура элемента активной среды еще высока ($T > 0$), он теряет энергию, его температура быстро уменьшается за счет охлаждения, выделяющаяся мощность $P = P_2$ отрицательна. Когда элемент среды охладился ($T < T_{KP}$), а трава не выросла

($u < u_0$), она продолжает расти, что учитывается слагаемым $ru(u_0 - u)$, где u_0 - концентрация травы после того, как ее рост прекратился. При приближении u к u_0 скорость роста травы уменьшается. В химических растворах происходит теплопроводность и диффузия реагирующих веществ; чтобы это учесть, следует задать коэффициенты a и $b > 0$.

Для численного решения рассмотренной выше системы двух дифференциальных уравнений необходимо заменить все производные их конечно-разностными аппроксимациями и выразить T_i^{t+1} и u_i^{t+1} в дискретный момент времени $t + 1$:

$$T_i^{t+1} = T_i^t + a \frac{T_{i-1}^t - 2T_i^t + T_{i+1}^t}{\Delta x^2} \Delta t + P_i^t \Delta t,$$

$$u_i^{t+1} = u_i^t + b \frac{u_{i-1}^t - 2u_i^t + u_{i+1}^t}{\Delta x^2} \Delta t + Q_i^t \Delta t + ru_i(u_0 - u_i) \Delta t.$$

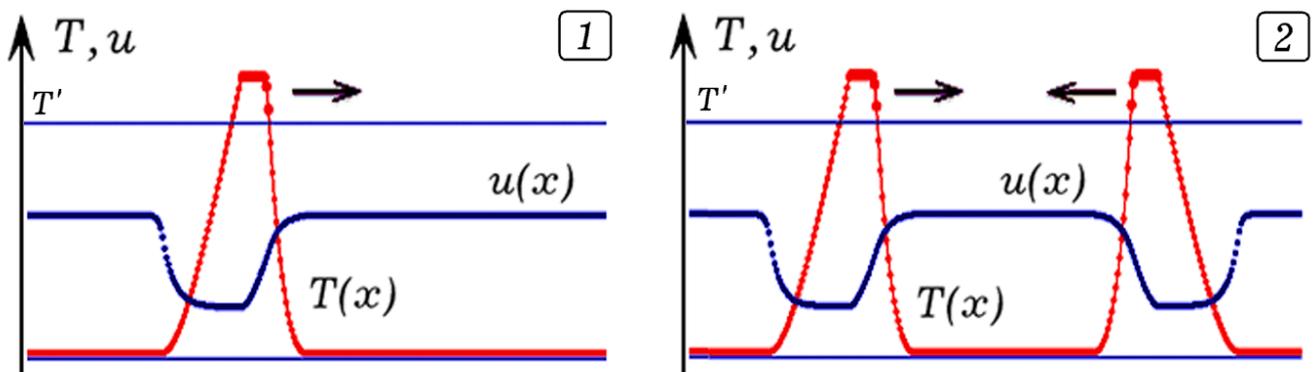


Рис. 10.16. Распространение и аннигиляция одномерных автоволн

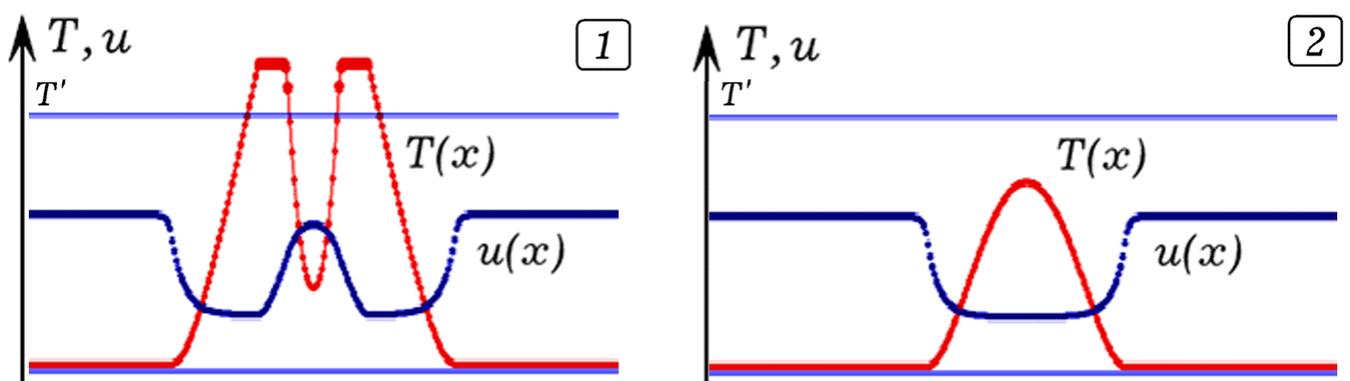


Рис. 10.17. Аннигиляция двух автоволн в одномерных средах

Используется программа ПР-12 [7], в ней подобраны следующие параметры $P_1 = 2$, $P_2 = -0,1$, $Q = -0,07$, $r = 0,003$, $b = 1,2$. На рис. 10.16 и 10.17 представлены результаты моделирования распространения автоволны в одномерной среде и аннигиляции двух автоволн.

Теперь методом компьютерного моделирования исследуем процесс распространения автоволны в двумерной активной среде, огибание автоволной препятствия, а также образование спиральной автоволны. Допустим, по полю, на котором быстро растет трава, распространяется фронт огня. Исходя из двухкомпонентной модели, запишем дифференциальные уравнения для активатора (температуры) T и концентрации травы u :

$$\frac{\partial T}{\partial t} = a \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + P(T, u), \quad \frac{\partial u}{\partial t} = Q(T, u).$$

Разобьем прямоугольную область на элементы, каждый из которых характеризуется величинами $T_{i,j}$ и $u_{i,j}$. Используется программа ПР-13, результаты моделирования представлены на рис. 10.18-10.21. Для того чтобы получить сферическую волну, элементы внутри круга небольшого радиуса переводят в возбужденное состояние, их как бы "поджигают", задавая начальную температуру выше T_{KP} .

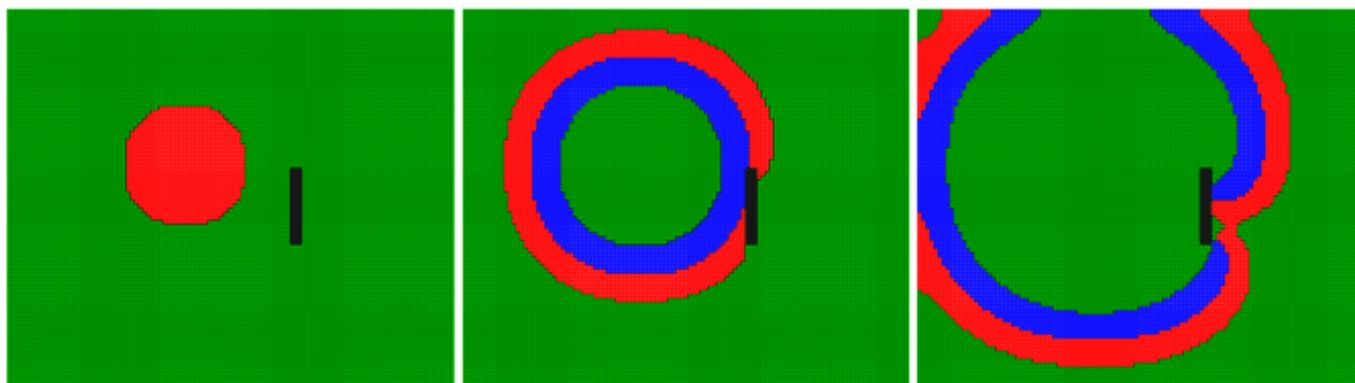


Рис. 10.18. Огибание сферической автоволной препятствия

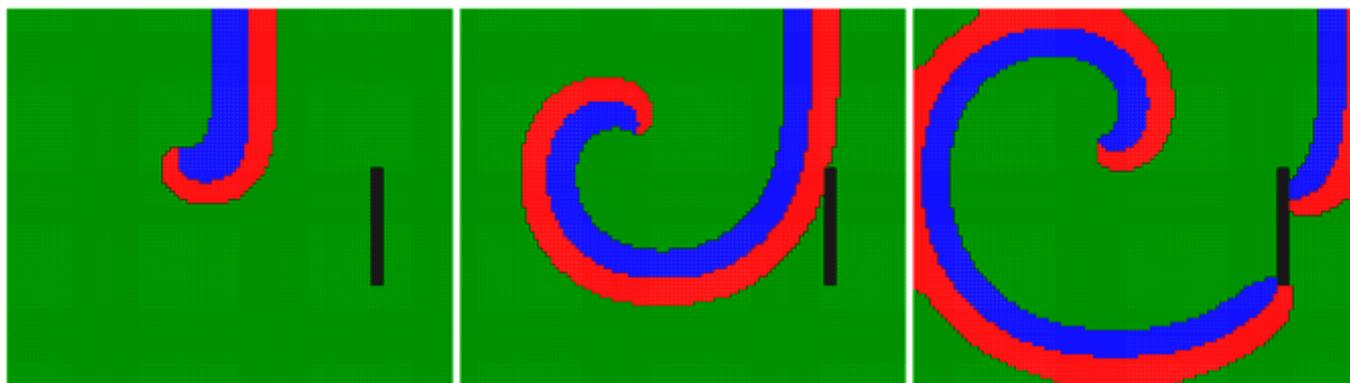


Рис. 10.19. Однорукавная спиральная автоволна

Из рис. 10.18 видно, как образовавшаяся автоволна огибает препятствие (группу элементов черного цвета с низкой теплопроводностью) и продолжает распространяться дальше. Для получения спиральной однорукавной автоволны задают фронт волны, оборванный в центре экрана (рис. 10.19). Из-за краевого эффекта происходит изгибание фронта автоволны, он принимает форму спирали. При наличии препятствия фронт разрывается еще в одном месте.

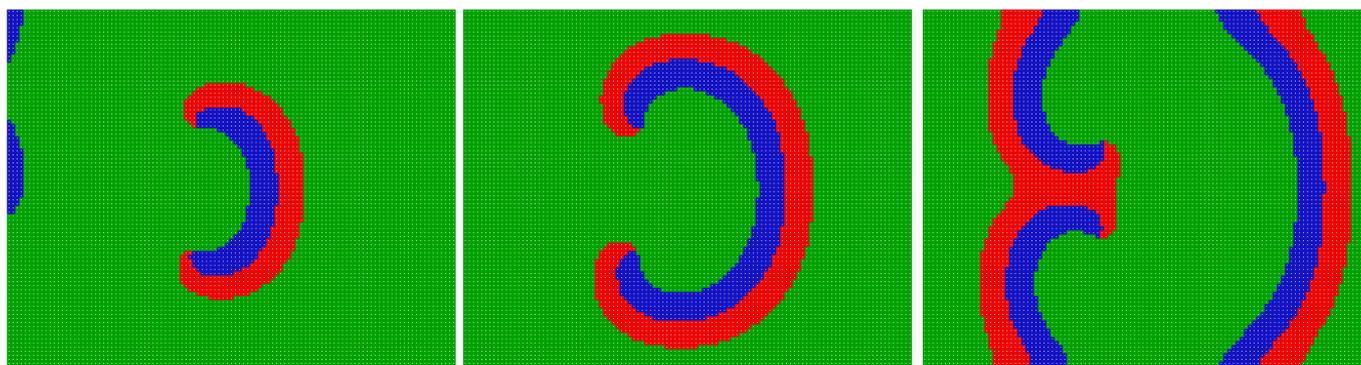


Рис. 10.20. Периодический автоволновой процесс

На рис. 10.20 показана эволюция небольшого участка фронта возбуждения. Происходит периодическое образование замкнутого фронта возбуждения, который по мере удаления от источника становится круговым. Чтобы получить двухрукавную волну (рис. 10.21), необходимо задать начальные условия следующим образом (программа ПР-13):

```
If (i>55) and (i<65) and (j<50) then t[i,j]:=20*(i-55);
```

```
If (i>50) and (i<56) and (j<50) then begin  
u[i,j]:=round(0.3*ukr); t[i,j]:=0.99*tkr; end;  
If (i>55) and (i<65) and (j>50) then t[i,j]:=20*(65-i);  
If (i>64) and (i<75) and (j>50) then begin  
u[i,j]:=round(0.3*ukr); t[i,j]:=0.99*tkr; end;
```

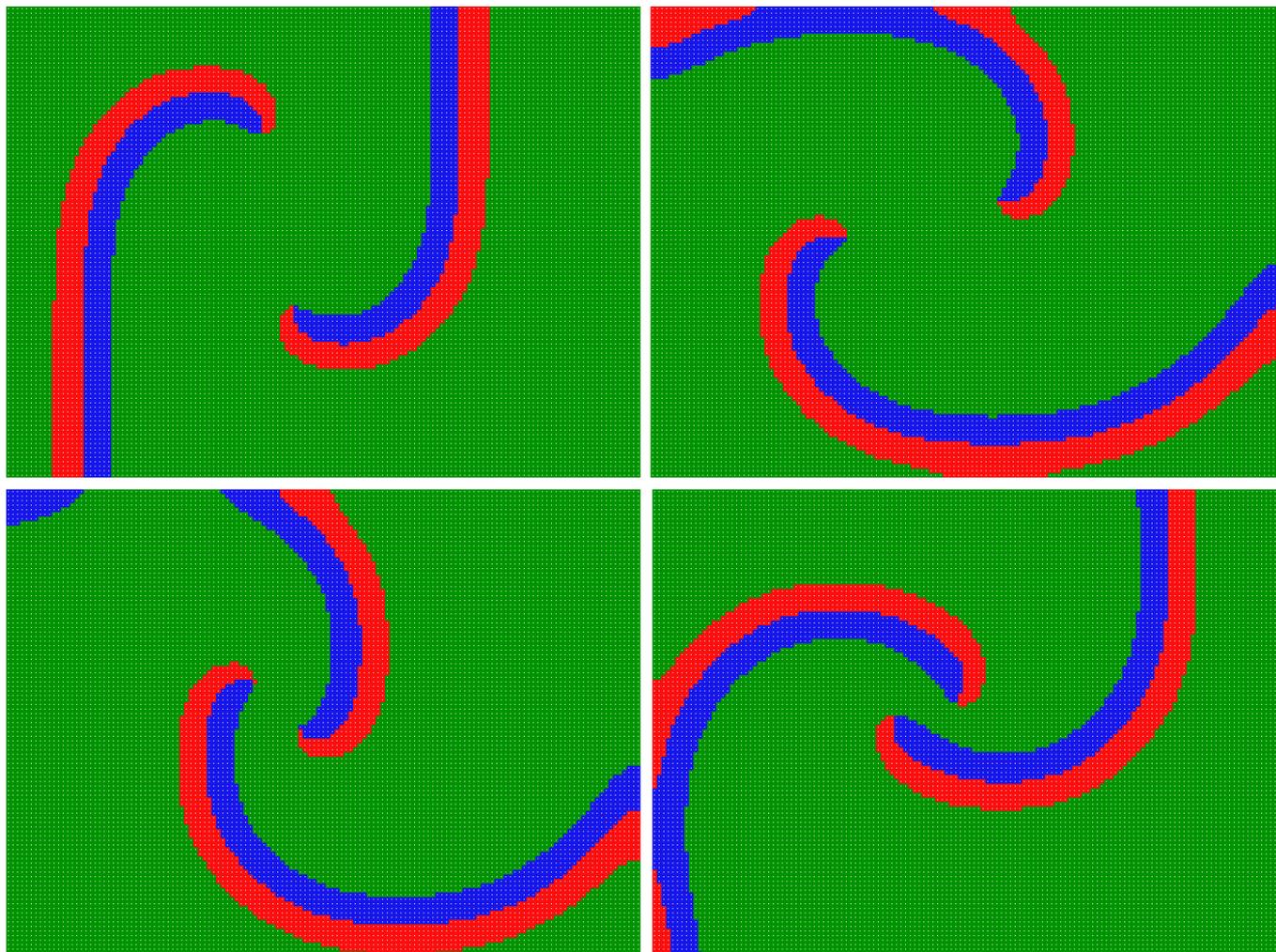


Рис. 10.21. Двухрукавная автоволна

Приложение к главе 10

В приложении представлены тексты программ, которые позволяют промоделировать рассмотренные выше явления. Они написаны в средах Borland Pascal 7.0 и Free Pascal 1.0.10.

Программа ПР-1

```

{$N+} program Teploprovodnost_plastini;
uses crt, graph;
const n=75; m=70; h=1; dt=0.005;
var i,j,C,kk,DV,MV: integer; q, a : single;
tt,t: array[0..N+1,0..M+1]of single; uslovie: boolean;
Procedure Istoch;
begin
If ((i>15)and(i<25))and((j>20)and(j<25))
            then q:=6 else q:=0;
If ((i>45)and(i<65))and((j>50)and(j<60)) then q:=4;
end;
Procedure Nach_uslov;
begin
For i:=1 to N do For j:=1 to M do
    begin uslovie:=((j>45)and(j<65)and(i>10)and(i<30));
    If uslovie=true then t[i,j]:=24 else t[i,j]:=1; end; end;
Procedure Raschet;
begin Istoch;
{If (abs(i-40)<10)and(abs(j-40)<10) then a:=0
            else a:=0.7;}
If sqr(i-40)+sqr(j-30)<100 then a:=0 else a:=0.7;
tt[i,j]:=t[i,j]+a*(t[i,j+1]-2*t[i,j]+t[i,j-1])*dt/(h*h)
+a*(t[i+1,j]-2*t[i,j]+t[i-1,j])*dt/(h*h)+q*dt; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi'); Nach_uslov;
Repeat inc(kk);
For i:=2 to N-1 do For j:=2 to M-1 do begin
Raschet; end;
For i:=2 to N-1 do For j:=2 to M-1 do t[i,j]:=tt[i,j];
For i:=1 to N do t[i,M]:=t[i,M-1];
If kk mod 200=0 then begin kk:=0;
    For i:=2 to N-1 do For j:=2 to M-1 do
    begin C:=round(100-t[i,j]/20)mod 15+1; setcolor(C);
    rectangle(i*5+50,j*5,i*5+54,j*5+4);
    rectangle(i*5+51,j*5+1,i*5+53,j*5+3); end;
    setcolor(black); circle(250,150,53); end;
until KeyPressed; CloseGraph;
END.

```

Программа ПР-2

```

{$N+} program Gazovayz_ataka;
uses crt, graph;
const N=250; M=60; dt=0.01; h=1; D=0.9;
var DV,MV,i,j,t: integer; q,v: single;
C: array[0..N+1,0..M+1] of single;
Procedure Raschet;
begin
If (i>10) and (i<20) and (j>50) and (t<5000)
    then q:=5 else q:=0;
C[i,j]:=C[i,j]+D*(C[i+1,j]-2*C[i,j]+C[i-1,j]+C[i,j+1]-
2*C[i,j]+C[i,j-1])*dt/h/h+q+v*(C[i,j]-C[i+1,j])/h*dt;
end;
BEGIN v:=0.5; DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat t:=t+1;
    For i:=1 to N do For j:=1 to M do Raschet;
    For j:=M downto 1 do For i:=N downto 1 do Raschet;
    For j:=1 to M do C[N+1,j]:=C[N,j];
    For i:=1 to N do C[1,j]:=C[0,j]; {delay(10);}
    If t mod 100=0 then begin {cleardevice;}
    For i:=1 to N do For j:=1 to M do begin
    setcolor(round(C[i,j]/400)+8);circle(3*i,50+3*j,2); end;
    circle(round(t/40),450-round(2*C[50,23]),2);
    circle(round(t/40),450-round(2*C[90,23]),2);
    circle(round(t/40),450-round(2*C[130,23]),2);
    circle(round(t/40),450-round(2*C[170,23]),2);
    circle(round(t/40),450-round(2*C[210,23]),2);
    circle(round(t/40),450-round(2*C[250,23]),2); end;
until KeyPressed; CloseGraph;
END.

```

Программа ПР-3

```

{$N+} program Tecenie_vodi_istochnik_teplo;
uses crt, graph; const n=160; m=60; dt=0.2; h=1;
var i,j,k,DV,MV,EC: integer; q: single;
T: array[0..N+1,0..M+1] of single;
v: array[0..M+1] of single;
Procedure Nach_usl;
begin For i:=1 to N do For j:=1 to M do

```

```

begin T[i,j]:=1; end; end;
Procedure Resh1;
begin v[0]:=0; v[M]:=0;
  v[j]:=v[j]+(v[j+1]-2*v[j]+v[j-1])*dt/h/h+0.1; end;
Procedure Resh2;
begin If (i>20)and(i<30)and(j>10)and(j<15)
      then q:=3 else q:=0;
T[i,j]:=T[i,j]+(T[i+1,j]-2*T[i,j]+T[i-1,j]+T[i,j+1]-
2*T[i,j]+T[i,j-1])*dt/h/h+q+0.01*v[j]*(T[i,j]-T[i+1,j])*dt;
end;
procedure Draw;
begin setcolor(round(T[i,j]/6)+1);
  circle(4*i,50+4*j,1); circle(4*i,50+4*j,2);
  setcolor(15); circle(500+round(v[j]),50+4*j,1);
  circle(500,50+4*j,1); end;
BEGIN
DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi'); Nach_usl;
Repeat k:=k+1;
If k<300 then begin For j:=1 to M do Resh1;
  For j:=M downto 1 do Resh1; end else begin
  For i:=1 to N do For j:=1 to M do Resh2;
  For j:=M downto 1 do For i:=N downto 1 do Resh2; end;
If k mod 20=0 then begin cleardevice;
  For i:=1 to N do For j:=1 to M do Draw; end;
until KeyPressed; CloseGraph;
END.

```

Программа ПР-4

```

{$N+} program Fakeli;
uses crt, graph; const n=95; m=100; dt=0.02; h=1;
var ii,jj,kk,i,j,DV,MV : integer;
  t,tt: array[1..N,1..M] of single; q,a,b: single;
procedure Reshen;
begin q:=0; a:=0.9; b:=0.01;
If (i mod 20>15)and((j>87)and(j<100)) then q:=100;
If sqr(i-40)+sqr(j-40)<100 then
  begin a:=0.03; b:=0; end;
tt[i,j]:=t[i,j]+a*(t[i,j+1]-2*t[i,j]+t[i,j-1])*dt/h
  +a*(t[i+1,j]-2*t[i,j]+t[i-1,j])*dt/h+q*dt;
tt[i,j]:=tt[i,j]+(t[i,j+1]-t[i,j-1])*b;

```

```

end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat inc(kk);
For i:=2 to N-1 do For j:=2 to M-1 do Reshen;
For i:=2 to N-1 do For j:=2 to M-1 do t[i,j]:=tt[i,j];
For i:=N-1 downto 2 do For j:=M-1 downto 2 do Reshen;
For i:=2 to N-1 do For j:=2 to M-1 do t[i,j]:=tt[i,j];
For i:=1 to N do t[i,1]:=t[i,2];
If kk mod 15=0 then For i:=2 to N-1 do
  For j:=2 to M-1 do begin kk:=0;
    setcolor(round(t[i,j]/150+2) mod 15);
    If sqr(i-40)+sqr(j-40)<100 then begin
      setcolor(black); end;
    rectangle(i*4+50,j*4,i*4+52,j*4+2);
    rectangle(i*4+50,j*4,i*4+51,j*4+1); end;
until KeyPressed; CloseGraph;
END.

```

Программа ПР-5

```

{$N+}program Teploprovodnost_neodnorodnaya_plastina;
uses crt, graph;
const n=72; m=72; h=1; dt=0.02;
var  kk,i,j,dir,DV,MV: integer; q,a,b,bb : single;
     k,t : array[1..N,1..M] of single;
Procedure Raschet;
begin q:=0;
If (abs(i-36)<5) and (abs(j-36)<5) then q:=10;
If dir=1 then
t[i,j]:=t[i,j]+(k[i,j+1]-k[i,j-1])*(t[i,j+1]-t[i,j-1])*dt/(4*h*h)+
k[i,j]*(t[i,j-1]-2*t[i,j]+t[i,j+1])*dt/(h*h)+q*dt else
t[i,j]:=t[i,j]+(k[i+1,j]-k[i-1,j])*(t[i+1,j]-t[i-1,j])*dt/(4*h*h)+k[i,j]*(t[i-1,j]-2*t[i,j]+t[i+1,j])*dt/(h*h)+q*dt; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
For i:=1 to N do For j:=1 to M do
  begin k[i,j]:=0.1+0.002*i*i+0.05*j;
    If (i>55) then k[i,j]:=2; end;
Repeat inc(kk); dir:=0;
For i:=2 to N-1 do For j:=2 to M-1 do Raschet; dir:=1;

```

```

For j:=M-1 downto 2 do For i:=M-1 downto 2 do Raschet;
For i:=1 to N do t[i,M]:=t[i,M-1];
If kk mod 50=0 then
  For i:=2 to N-1 do For j:=2 to M-1 do begin
    setcolor(abs(round(t[i,j]/25+8)) mod 15+1);
    rectangle(i*5+50,j*5,i*5+54,j*5+4);
    rectangle(i*5+51,j*5+1,i*5+53,j*5+3); end;
  line(55*5+50,0,55*5+50,600);
until KeyPressed; CloseGraph;
END.

```

Программа ПР-6

```

{$N+} program Teploprovodnost_truba_techenie;
Uses crt, graph;
Const N=150; M=25; dt=0.005; h=1; a=0.3;
Var i,j,k,DV,MV: integer; q,v,F: single;
T: array[0..N+1,0..M+1] of single;
Procedure Raschet;
Begin If (i>10)and(i<30)and(j=10)and(k<5E+4) then
      q:=150 else q:=0;
v:=0.5*(1-j*j/M/M); F:=a*(T[i,j+1]-T[i,j])/h/(j+0.5);
T[i,j]:=T[i,j]+a*(T[i+1,j]-2*T[i,j]+T[i-1,j]+T[i,j+1]-
-2*T[i,j]+T[i,j-1])*dt/h/h+F*dt+q*dt-v*(T[i+1,j]-
T[i,j])/h*dt; end;
procedure Draw;
begin setcolor(round(T[i,j]/160)+1);
circle(4*i,150+4*j,2); circle(4*i,150+4*j,1);
circle(4*i,153-4*j,2); circle(4*i,153-4*j,1);
setcolor(white);
If (i>10)and(i<30)and(j=10) then begin
circle(4*i,150+4*j,2); circle(4*i,150+4*j,1);
circle(4*i,153-4*j,2); circle(4*i,153-4*j,1);
end; end;
BEGIN
DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat k:=k+1;
  For i:=1 to N do For j:=1 to M do Raschet;
  For j:=M downto 1 do For i:=N downto 1 do Raschet;
  For j:=1 to M do begin
    T[N+1,j]:=T[N,j]; T[0,j]:=T[1,j]; end;

```

```

For i:=1 to N do T[i,0]:=T[i,1];
If k mod 500=0 then begin k:=0;
For i:=1 to N do For j:=1 to M do Draw; end;
until KeyPressed; CloseGraph;
END.

```

Программа ПР-7

```

{$N+} program Teploprovodnost_shar;
uses crt, graph;
const n=100; m=1; h=1; dt=0.01;
var i,j,time,DV,MV: integer; f,tt,k1,k2,a1: single;
t1,t: array[1..N] of single;
Procedure Raschet;
begin If (tt>450)and(tt<450+dt) then begin a1:=0.05;
k2:=0.02; For i:=51 to 99 do t[i]:=5; end;
For i:=2 to 49 do begin
If i=49 then f:=-a1*(T[49]-T[51]) else f:=0;
t1[i]:=t[i]+k1*((t[i+1]-2*t[i]+t[i-1]))/(h*h)+
2/i*(t[i]-t[i-1])/h)*dt+f; end; T1[50]:=T1[49];
For i:=51 to 99 do begin
If i=51 then f:=a1*(T[49]-T[51]) else f:=0;
t1[i]:=t[i]+k2*((t[i+1]-2*t[i]+t[i-1]))/(h*h)+
2/i*(t[i]-t[i-1])/h)*dt+f; end; t1[1]:=t1[2];
t1[100]:=t1[99]; For i:=1 to N do t[i]:=t1[i]; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi'); a1:=0.9;
For i:=1 to N do If i<50 then t[i]:=30 else t[i]:=1;
t[49]:=25; t[50]:=15; t[51]:=5; k1:=0.3; k2:=0.9;
Repeat tt:=tt+dt; inc(time); Raschet;
If time mod 500=1 then begin cleardevice; time:=2;
For i:=2 to N do begin setcolor(round(t[i]/5)+1);
If i<50 then circle(400,120,2*i);
line(i*5+20,420-round(8*t[i]),(i-1)*5+20,420-round(8*t[i-
1])); end; line(20,420,500,420); line(20,0,20,420); end;
until KeyPressed; CloseGraph;
END.

```

Программа ПР-8

```

{$N+} program Zadacha_Stefana;
uses crt, graph;
const N=70; tp=1; h=1; dt=0.01; q_pl=10; M=1000;

```

```

var fl,i,j,j1,DV,MV: integer;
dq,q1,q2,a,time,tt1,tt2: single; kk: Longint;
tt,t,q: array[-1..N+1] of single;
k: array[-1..N+1] of single; Label mm;
Procedure Raschet1;
begin If i=2 then q1:=0.05 else q1:=0;
If t[2]>1.05 then q1:=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])/(4*h*h)*dt
+q1*dt+q[i]*dt; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
For i:=1 to N do t[i]:=0.9;
Repeat inc(kk); time:=time+dt;
For i:=-1 to N+1 do If t[i]>tp then k[i]:=3
                        else k[i]:=1.0;
If (kk<1000) or (j>=N-3) then
  For i:=2 to N-1 do Raschet1 else
  begin
    For i:=1 to N-1 do
      If (t[i]>=tp) and (t[i+1]<tp) then j:=i;
      tt1:=t[i]; tt2:=t[i+1]; t[j]:=tp;
      dq:=abs(k[j-1]*(t[j]-t[j-1])/h)-
          abs(k[j+1]*(t[j+1]-t[j])/h)+dq;
    For i:=1 to N do q[i]:=0;
    If dq>=q_p1 then begin
      For i:=2 to j-1 do Raschet1;
      tt[j]:=abs(tp+tt[j-1])/2; tt[j+1]:=tp;
      For i:=2 to j-1 do q[i]:=-abs(q_p1/(j-2));
      dq:=abs(dq-q_p1);
      For i:=j+2 to N-1 do Raschet1; goto mm; end;
      If dq<q_p1 then begin
        For i:=2 to j-2 do Raschet1;
        tt[j-1]:=(tt[j-2]+tp)/2;
        tt[j]:=tp; tt[j+1]:=(tt[j+2]+Tp)/2;
        For i:=j+2 to N-1 do Raschet1; end;
      mm: end;
    For i:=2 to N-1 do t[i]:=tt[i];
    t[1]:=t[2]; t[N]:=t[N-1];
    If kk mod 10000=50 then begin {cleardevice;}

```

```

For i:=2 to N do begin line(i*8+10,1200-round(M*t[i]),
(i-1)*8+10,1200-round(M*t[i-1])); end; end;
line(20,1200-round(M*tp),620,1200-round(M*tp));
line(20,500,620,500);
until KeyPressed; CloseGraph;
END.

```

Программа ПР-9

```

{$N+} program Teplovaya_volna;
Uses crt, graph; Const n=200; tk=3; h=1; dt=0.0001;
Var i,j,kk,DV,MV : integer; q1,q2,a,vr : single;
tt,t: array[0..N+1] of single; k: array[0..N+1] of single;
Procedure Raschet;
begin
if i=100 then t[i]:=10*sqrt(vr); k[i]:=0.5*t[i]*t[i];
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])/(4*h*h)*dt; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat kk:=kk+1; vr:=vr+dt;
For i:=2 to N-1 do Raschet;
For i:=N-1 downto 2 do Raschet;
For i:=2 to N-1 do t[i]:=tt[i];
t[1]:=t[2]; t[N]:=t[N-1];
If kk mod 10000=0 then begin {cleardevice;} kk:=0;
For i:=2 to N do begin line(i*3+20,400-
round(5*t[i]),(i-1)*3+20,400-round(5*t[i-1])); end;
line(100*3+20,400,100*3+20,0); end;
until (vr>1200{12})or(KeyPressed);
Repeat until KeyPressed; CloseGraph;
END.

```

Программа ПР-10

```

{$N+} program Gorenje_odnomern_sreda;
uses crt, graph; const n=200; tk=3; h=1; dt=0.001;
var ii,jj,i,j,DV,MV,EC: integer; q,a,vr: single;
tt,t: array[0..N+1]of single;
k: array[0..N+1]of single; kk: Longint;
Procedure Raschet;
begin If (vr<10)and(i=100) then t[i]:=sqrt(vr);
k[i]:=t[i]*t[i]/5+0.01; q:=0.05*T[i]-0.02*T[i]*T[i];

```

```

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])/(4*h*h)*dt+q*dt; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat kk:=kk+1; vr:=vr+dt;
  For i:=2 to N-1 do Raschet;
  For i:=N-1 downto 2 do Raschet;
  For i:=2 to N-1 do t[i]:=tt[i];
  t[1]:=t[2]; t[N]:=t[N-1];
  If kk mod 30000=0 then begin {cleardevice;} kk:=0;
    For i:=2 to N do begin
      line(i*3+20,400-round(90*t[i]),
        (i-1)*3+20,400-round(90*t[i-1])); end;
    line(100*3+20,400,100*3+20,0); end;
until (kk>60000) or (KeyPressed);
Repeat until KeyPressed; CloseGraph;
END.

```

Программа ПР-11

```

{$N+} program Gorenje_odnomern_sreda_2;
uses crt, graph; const n=200; tk=5; h=1; dt=0.001;
var ii,jj,kk,i,j,DV,MV: integer; q,qq,a: single;
t: array[1..N] of single; k: array[1..N] of single;
Procedure Param_sred;
begin For j:=1 to N do begin
  If t[j]>tk then a:=4 else a:=0;
  k[j]:=a*t[j]*t[j]+0.5; end; end;
Procedure Raschet;
begin If t[i]>tk then q:=0.1*(t[i]-tk)*(t[i]-tk)
  *(t[i]-tk){-1E-4*t[i]*t[i]*t[i]} else q:=0;
  If (i>97) and (i<103) then qq:=0.3 else qq:=0;
  t[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])/(4*h*h)*dt+(q+qq)*dt; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat inc(kk); Param_sred;
  For i:=2 to N-1 do Raschet;
  For i:=N-1 downto 2 do Raschet;
  If kk mod 1500=0 then begin {cleardevice;} kk:=0;
  For i:=2 to N do begin line(i*4+20,400-round(50*t[i]),
    (i-1)*4+20,400-round(50*t[i-1]));end; end;
    line(20,400-round(50*tk),800,400-round(50*tk));

```

```

t[1]:=0; t[N]:=t[N-1];
until KeyPressed; CloseGraph;
END.

```

Программа ПР-12

```

{$N+} program Avtovolni_difuravnenie;
uses crt, graph;
const n=320; M=40; t0=10; tkr=5;
u0=3; ukr=1; h=1; dt=0.01;
var time,i,j,DV,MV,EC,k : integer; a,b,r,P,Q : real;
    t,u: array[1..N] of single;
    gor: array[1..N] of integer;
Procedure Raschet; {--- Vichislenie temperaturi ---}
begin P:=0; Q:=0;
    If (t[i]>tkr)and(u[i]>ukr) then gor[i]:=1
        else gor[i]:=0; {trava sgoraet}
    If gor[i]=1 then begin P:=2; Q:=-0.07; end;
    If (gor[i]=0)and(t[i]>0) then begin P:=-0.1; end;
    If (gor[i]=0)and(u[i]<u0)and(t[i]<0.2) then
        r:=0.003 else r:=0; {trava rastet}
    If (t[i]>1.2*tkr) then t[i]:=1.2*tkr;
    t[i]:=t[i]+a*(t[i+1]-2*t[i]+t[i-1])*dt/(h*h)+P*dt;
    { u[i]:=u[i]+Q*dt+r*u[i]*(u0-u[i])/5;}
    u[i]:=u[i]+b*(u[i+1]-2*u[i]+u[i-1])*dt/(h*h)+
        Q*dt+r*u[i]*(u0-u[i]);
end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
a:=6; b:=3; setbkcolor(15);
For i:=1 to N do begin u[i]:=u0;
If (i>1)and(i<20)or(i>300)and(i<N) then t[i]:=t0; end;
Repeat inc(time);
    t[1]:=t[2]; t[N]:=t[N-1]; u[1]:=u[2]; u[N]:=u[N-1];
    For i:=2 to N-1 do Raschet;
    For i:=N-1 downto 2 do Raschet;
    If time mod 100=0 then begin cleardevice;
        For i:=2 to N do begin setcolor(12);
            circle(i*2,400-round(M*t[i]),1);
            circle(i*2,400-round(M*t[i]),2);
            line(i*2,400-round(M*t[i]),2*(i-1),400-
                round(M*t[i-1]));

```

```

    If gor[i]=1 then
        circle(i*2,400-round(M*t[i]),4);
        setcolor(9); circle(i*2,400-round(M*u[i]),1);
        circle(i*2,400-round(M*u[i]),2);
    end; end; line(0,405,640,405);
    line(0,400-round(M*tkr),640,400-round(M*tkr));
until (time>100000) or (KeyPressed);
Repeat until KeyPressed; CloseGraph;
END.

```

Программа ПР-13

```

{$N+} program Avtovolni_dvumern_sreda;
uses crt, graph;
const n=110; M=90; t0=10; tkr=5;
      u0=65000; ukr=32000; h=1; dt=0.05;
var time,i,j,k,DV,MV,EC: integer; a,P,Q,uu: single;
    t: array[0..N+1,0..M+1] of single;
    u: array[0..N+1,0..M+1] of word; gor: boolean;
Procedure Raschet;
begin P:=0; Q:=0;
If (t[i,j]>tkr) and (u[i,j]>ukr) then
    gor:=true else gor:=false; {trava sgoraet}
If gor=true then begin P:=1.9; Q:=-1600; end;
If (gor=false) and (t[i,j]>0) then P:=-0.25;
If (gor=false) and (u[i,j]<u0) and (t[i,j]<0.9*tkr)
    then Q:=500000; {trava rastet}
t[i,j]:=t[i,j]+a*(t[i+1,j]+t[i-1,j]+t[i,j+1]
    -4*t[i,j]+t[i,j-1])*dt/(h*h)+P*dt;
uu:=u[i,j]+Q*dt;
If t[i,j]>2*tkr then t[i,j]:=2*tkr;
If uu>=u0 then u[i,j]:=u0;
If (uu>1) and (uu<u0) then u[i,j]:=round(uu);
If uu<1 then u[i,j]:=1;
If (i=92) and (j>40) and (j<70) then a:=0.1 else a:=0.8;
end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
setbkcolor(15); a:=0.8;
For i:=1 to N do For j:=1 to M do begin
    u[i,j]:=u0; t[i,j]:=0;
    If sqr(i-45)+sqr(j-40)<100

```

```

        then t[i,j]:=2*t0; {odinochnaya volna}
(* If (i>45)and(i<55)and(j<40)
        then t[i,j]:=2*t0; {odnorukavnaya volna}
    If (i>40)and(i<46)and(j<40) then begin
        u[i,j]:=round(0.3*ukr); t[i,j]:=0.9*tkr; end; *)
    end;
Repeat inc(time);
For i:=1 to N do For j:=1 to M do begin
    u[1,j]:=u[2,j]; u[N,j]:=u[N-1,j];
    u[i,1]:=u[i,2]; u[i,M]:=u[i,M-1];
    t[1,j]:=t[2,j]; t[N,j]:=t[N-1,j];
    t[i,1]:=t[i,2]; t[i,M]:=t[i,M-1]; end;
For i:=2 to N-1 do For j:=2 to M-1 do Raschet;
For i:=N-1 downto 2 do
    For j:=M-1 downto 2 do Raschet;
If time mod 100=1 then begin cleardevice;
    For i:=2 to N do For j:=2 to M-1 do begin
        setcolor(9); If u[i,j]>ukr then setcolor(2);
        If (t[i,j]>tkr)and(u[i,j]>ukr) then setcolor(12);
        If u[i,j]<ukr then setcolor(9);
        If(i>90)and(i<94)and(j>40)and(j<70) then setcolor(8);
        rectangle(i*5,j*5,i*5+4,j*5+4);
        rectangle(i*5+1,j*5+1,i*5+3,j*5+3); end; end;
until KeyPressed; CloseGraph;
END.

```

Список литературы

1. Булавин Л. А., Выгорницкий Н. В., Лебовка Н. И. Компьютерное моделирование физических систем. Долгопрудный: Издательский Дом "Интеллект", 2011. 352 с.
2. Васильев В. А., Романовский Ю. М., Яхно В. Г. Автоволновые процессы. М.: Наука, 1987. 240 с.
3. Ращиков В. И., Рошаль А. С. Численные методы решения физических задач: учеб. пособие. СПб.: Издательство "Лань", 2005. 208 с.

4. 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.
5. Калиткин Н. Н. Численные методы. БХВ-Петербург, 2011. 592 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. Рудяк В. Я. Математические модели природных явлений и технологических процессов. Новосибирск: Изд-во НГТУ, 2003. 181 с.
10. Компьютеры и нелинейные явления: Информатика и современное естествознание / авт. пред. А. А. Самарский. М.: Наука, 1988. 176 с.
11. Компьютеры, модели, вычислительный эксперимент. Введение в информатику с позиций математического моделирования / авт. пред. А. А. Самарский. М.: Наука, 1988. 176 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. Giordano N. J. Computational Physics. New Jersey, Prentice Hall, 1997. 419 p.
15. Phillipson P. E., Schuster P. Modeling by Nonlinear Differential Equations: Dissipative and Conservative Processes. World Scientific Publishing, 2009. 225 p.
16. Woolfson M. M., Pert G. J. An Introduction to Computer Simulation. Oxford University Press, 1999. 311 p.
17. <http://en.wikipedia.org/wiki/Stefan-problem>

[ВВЕРХ](#)