

Глава 11

РАСЧЕТ ТЕЧЕНИЯ ЖИДКОСТИ. КОНВЕКЦИЯ

Гидродинамика изучает законы движения жидкостей и их взаимодействие с твердыми телами. Основная задача состоит в определении поля скоростей, поля плотностей и поля давлений жидкости, на которую действуют внешние силы. В настоящей главе моделируются различные течения несжимаемых жидкостей, включая конвективные течения. Анализируемые модели предполагают численное решение уравнения Навье-Стокса и уравнения теплопроводности [1-3; 7; 8; 11-13; 15; 18-20].

11.1. Вытекание жидкости из сосуда произвольной формы

Рассмотрим сосуд заданной формы, наполненный идеальной (невязкой) жидкостью. Выясним, как будет изменяться уровень свободной поверхности жидкости, если открыть отверстие на дне сосуда. Торричелли установил, что скорость невязкой жидкости, вытекающей из сосуда, равна $v_2 = \sqrt{2gy}$, где y - высота уровня жидкости. Быстрота ее уменьшения $v_1 = -dy/d\tau$ зависит от формы сосуда или функции $S = S(y)$, выражающей связь между площадью S сечения сосуда горизонтальной плоскостью и ее координатой y . Если сосуд произвольной формы заменить сосудом, форма которого является фигурой вращения с той же зависимостью $S = S(y)$, то функции скоростей истечения идеальной жидкости $v_2(\tau)$ и понижения уровня поверхности $v_1(\tau)$ от времени не изменятся.

Итак, задача сводится к расчету вытекания жидкости из сосуда, боковые стенки которого получены в результате вращения некоторой кривой вокруг вертикальной оси симметрии (рис. 11.1.1). В этом случае любое

горизонтальное сечение сосуда плоскостью $y = \text{const}$ представляет собой круг радиусом $R = R(y)$. Пусть s - площадь сечения отверстия, $v_1 = -dy/d\tau$ - скорость опускания уровня жидкости. Из уравнения неразрывности струи $Sv_1 = sv_2$ следует, что $v_1 = sv_2/S = sv_2/(\pi R^2(y))$. За время $\Delta\tau$ вытекает объем жидкости $\Delta V = Sv_1\Delta\tau = sv_2\Delta\tau$. Высота жидкости в сосуде понижается на величину $\Delta y = -v_1\Delta\tau = -sv_2\Delta\tau/S(y)$.

Рассмотрим дискретные моменты времени $\tau = t\Delta\tau$, где $t = 0, 1, 2, \dots$. Зная высоту уровня жидкости y^t в момент t , можно определить скорость вытекания $v_2^t = (2gy^t)^{1/2}$ и соответствующее понижение уровня: $\Delta y^t = -sv_2^t\Delta\tau/(\pi R^2(y^t))$. Отсюда высота уровня в следующий момент $t+1$ равна $y^{t+1} = y^t + \Delta y^t = y^t - sv_2^t\Delta\tau/(\pi R^2(y^t))$.

Алгоритм А-1

НАЧАЛО ПРОГРАММЫ

```
y:=5*2*3.14; S2:= 0.1; dt:= 0.01;
```

```
ПОКА y>0 ПОВТОРЯТЬ {---
```

```
    v2:=SQRT(2*9.8*y); R:=1.3+SIN(y); S1:=3.14*R*R
```

```
    v1:=-S2*v2/S1; y:=y+v1*dt; t:=t+dt;
```

```
    ВЫВОД ЗНАЧЕНИЙ y, v1, v2, t НА ЭКРАН  --}
```

```
КОНЕЦ ПРОГРАММЫ
```

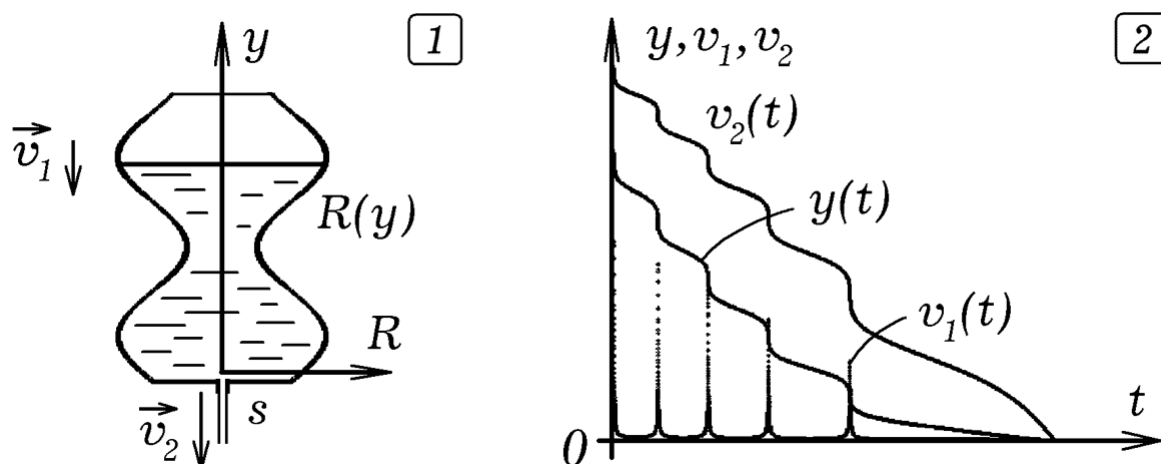


Рис. 11.1. Понижение уровня жидкости при ее вытекании из сосуда

Промоделируем вытекание жидкости из гипотетического сосуда, радиус горизонтального сечения которого связан с высотой y зависимостью $R = 1,3 + \sin(y)$ (рис. 11.1.1). Получим графики зависимостей $y(t)$, $v_1(t)$, $v_2(t)$ в случае, когда начальная высота $y_0 \approx 31,5$ м. Для этого используется программа ПР-1 (Приложение). Получающиеся графики $y(t)$, $v_1(t)$, $v_2(t)$ изображены на рис. 11.1.2. В узких местах сосуда скорость v_1 опускания уровня жидкости достигает максимума, высота уровня y быстро уменьшается. В случае когда сосуд цилиндрический, то есть $R = \text{const}$ и $S_1 = \text{const}$, графиком $y(t)$ является ветвь параболы, а скорости $v_1(t)$ и $v_2(t)$ уменьшаются пропорционально времени.

11.2. Потенциальное течение жидкости

Существует два метода описания движения жидкостей и газов. В **методе Лагранжа** течение жидкости рассматривается как движение N частиц, имеющих определенные координаты и скорости: $\vec{r}_i = f(\vec{r}_{0i}, \tau)$, $i = 1, \dots, N$. Его развитием являются метод крупных частиц (глава 9), метод частиц в ячейках, метод частиц-маркеров, метод сглаженных частиц и др. [5, с. 179-198; 12, с. 293-309; 13, с. 301-304, 359-362; 20, с. 211-256]. Основным в гидродинамике является **метод Эйлера**, требующий расчета поля скоростей $\vec{v}(\vec{r}, \tau)$, поля плотностей $\rho(\vec{r}, \tau)$ и давлений $p(\vec{r}, \tau)$ путем решения системы дифференциальных уравнений [1-8; 11-13; 16-20]. Познакомимся с этим методом на примере расчета установившегося потенциального течения идеальной жидкости.

Потенциальный характер течения означает, что линии тока незамкнуты, то есть течение невихревое (во всех точках $\text{rot } \vec{v} = 0$). При этом поле скоростей является потенциальным; можно ввести функцию $\varphi(x, y, z)$, называемую **потенциалом скорости** [4; 12; 13], такую, что

$$\vec{v} = -\text{grad}\varphi = -\frac{\partial\varphi}{\partial x}\vec{i} - \frac{\partial\varphi}{\partial y}\vec{j} - \frac{\partial\varphi}{\partial z}\vec{k} = -\frac{\partial\varphi}{\partial\vec{r}}.$$

Так как источники течения отсутствуют, то

$$\text{div}\vec{v} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} = 0.$$

Отсюда следует $\text{div}(\text{grad}\varphi) = 0$, или

$$\Delta\varphi = \frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} + \frac{\partial^2\varphi}{\partial z^2} = 0.$$

Итак, потенциал скорости удовлетворяет **уравнению Лапласа**.

Ограничимся рассмотрением плоского течения несжимаемой жидкости, при котором все ее частицы перемещаются параллельно некоторой плоскости xOy . Частицы, лежащие на одном перпендикуляре к этой плоскости, имеют равные скорости. Их проекции $v_x = f_1(x, y, \tau)$, $v_y = f_2(x, y, \tau)$, $v_z = 0$. Введем **функцию тока** $\psi(x, y, \tau)$ такую, что

$$v_x = \frac{\partial\psi}{\partial y}, \quad v_y = -\frac{\partial\psi}{\partial x}.$$

Множество точек, для которых $\psi(x, y, \tau) = \text{const}$ при фиксированном τ в силу стационарности течения, образует **линию тока**. При этом семейство линий равного потенциала скорости $\varphi(x, y, \tau) = \text{const}$ ортогонально (перпендикулярно) совокупности линий тока $\psi(x, y, \tau) = \text{const}$. Для функции тока $\psi(x, y, \tau)$ также можно записать уравнение Лапласа:

$$\Delta\psi = \frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} = 0.$$

При расчете течения следует учитывать **граничные условия**, определяющие значения рассчитываемых функций и их производных на границах твердых тел, обтекаемых жидкостью. Для идеальной жидкости проекция v_n скорости жидкости на нормаль $\vec{n}$ к поверхности тела, в системе отсчета, связанной с телом, равна 0: $v_n = (\partial\varphi / \partial\vec{n})_S = 0$ [12; 13]. Запишем уравнение Лапласа в конечных разностях:

$$\frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{\Delta x^2} + \frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{\Delta y^2} = 0,$$

$$\psi_{i,j} = (\psi_{i+1,j} + \psi_{i-1,j} + \psi_{i,j+1} + \psi_{i,j-1})/4.$$

Эта стационарная задача решается **методом последовательных приближений**: задается исходное распределение искомой функции $\psi(x, y)$, соответствующее граничным условиям, а затем осуществляется последовательность итераций, в ходе которых пересчитываются значения функции $\psi(x, y)$ в узлах сетки [6; 17; 20].

Рассчитаем установившееся потенциальное течение идеальной жидкости по трубе прямоугольного сечения и построим линии тока. Внутри трубы имеются выступы и различные препятствия заданной формы. Для этого используется программа ТР-2, содержащая два цикла, в которых последовательно перебираются все узлы сетки и пересчитываются значения $\psi_{i,j}$ [9; 10]. Граничные условия задаются в процедуре *Gran_usl*, верхней границе соответствует $\psi_{i,1} = -200$, а нижней - $\psi_{i,M} = 200$; расчеты выполняются в процедуре *Raschet*. Через каждые 20 итераций результат вычислений выводится на экран с помощью процедуры *Draw*. После запуска программы на экране монитора появляется результат вычислений функции тока $\psi(x, y)$ через 20 итераций, затем через 40 итераций и т. д. Области с различными $\psi(x, y)$ изображаются разными цветами; их границы соответствуют линиям тока (рис. 11.2).

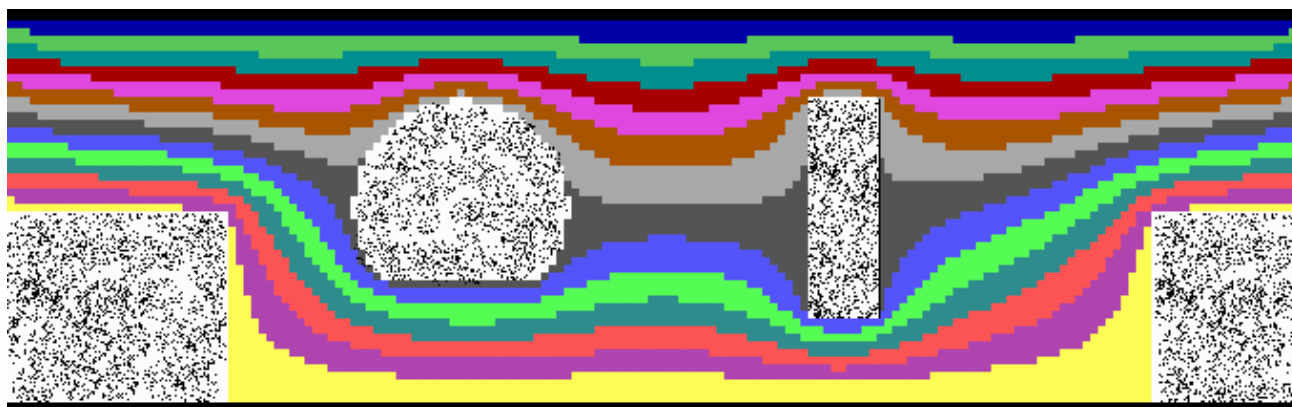


Рис. 11.2. Результаты расчета потенциального течения жидкости

Задача на построение линий тока может быть решена иначе. Допустим, между двумя параллельными пластинами течет идеальная жидкость; внутри потока перпендикулярно течению расположен цилиндр и уступ. Решение задачи состоит из двух этапов: 1) решение уравнения Лапласа для функции тока, определение поля скоростей; 2) введение в поток частиц-маркеров, которые будут перемещаться вдоль линий тока. Координаты маркеров вычисляются по формулам: $v_x = \partial\psi / \partial y$, $v_y = -\partial\psi / \partial x$, $x^{t+1} = x^t + (v_{i,j+1} - v_{i,j-1})\Delta\tau / 2h$, $y^{t+1} = y^t - (v_{i+1,j} - v_{i-1,j})\Delta\tau / 2h$, где $h = \Delta x = \Delta y$. Используется программа ПР-3, результаты вычислений представлены на рис. 11.3.

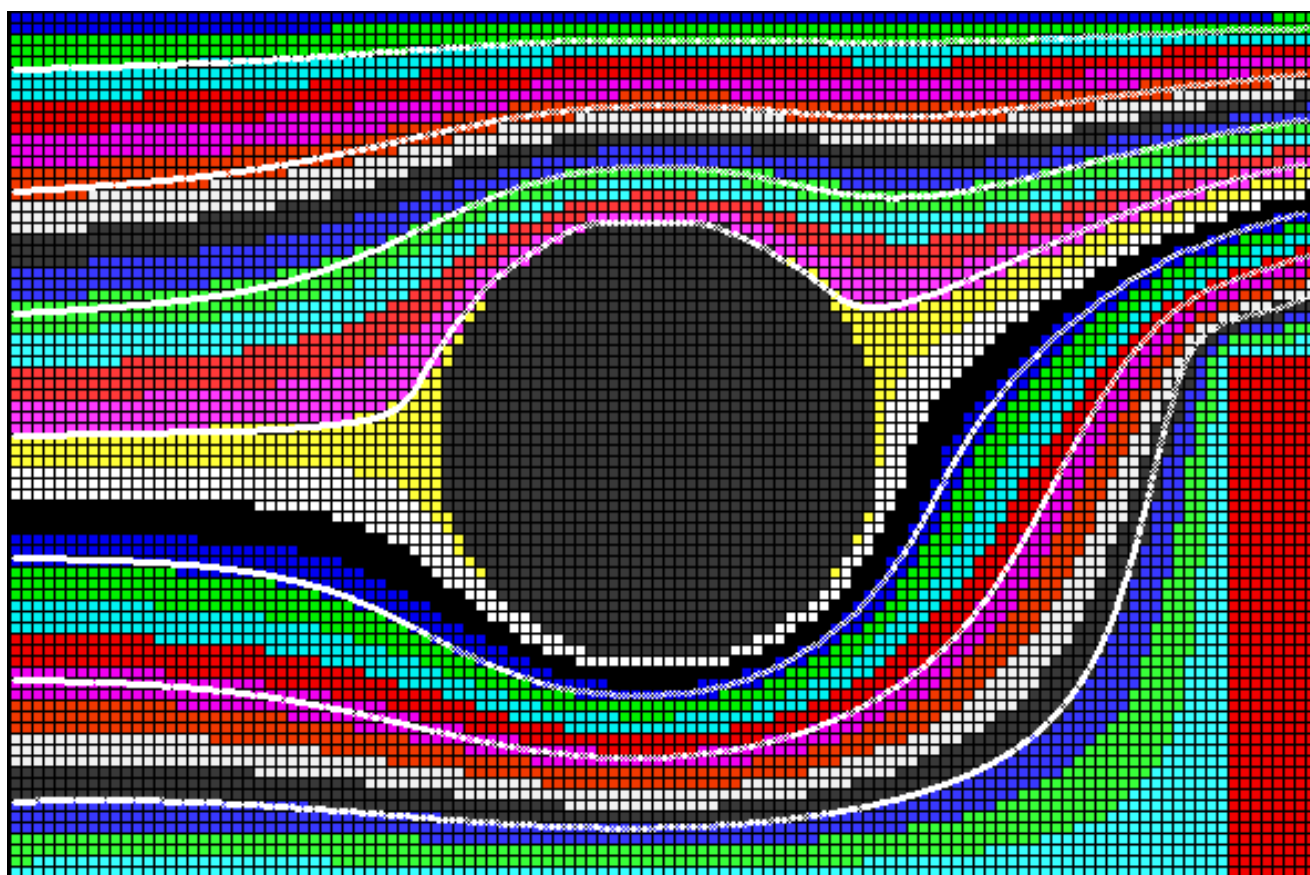


Рис. 11.3. Результаты моделирования течения идеальной жидкости

11.3. Течение вязкой жидкости в одномерном случае. Конвекция

Другой класс задач гидродинамики состоит в расчете стационарного течения вязкой жидкости [1-5; 18-20]. Движущееся тело увлекает за собой прилегающие к его поверхности слои жидкости, которые тоже приходят в движение. Вязкость относится к явлениям переноса и описывается уравнением, похожим на уравнение теплопроводности (глава 10).

Пусть неоднородная жидкость движется в пространстве между горизонтальными дном и пластиной, перемещающейся с некоторой скоростью в горизонтальном направлении. Ось координат x направим вертикально вверх (рис. 11.4) и запишем уравнение движения вязкой жидкости для одномерного случая в конечных разностях [1; 3; 4; 7; 12; 13]:

$$\frac{\partial v_z}{\partial \tau} = \frac{\partial}{\partial x} \left(\nu(x) \frac{\partial v_z}{\partial x} \right) - \frac{1}{\rho} \frac{\partial p}{\partial z} = \frac{\partial \nu}{\partial x} \cdot \frac{\partial v_z}{\partial x} + \nu \frac{\partial^2 v_z}{\partial x^2} - \frac{1}{\rho} \frac{\partial p}{\partial z},$$

$$\frac{v_i^{t+1} - v_i^t}{\Delta \tau} = \frac{v_{i+1} - v_{i-1}}{2\Delta x} \cdot \frac{v_{i+1}^t - v_{i-1}^t}{2\Delta x} + v_i^t \frac{v_{i+1}^t - 2v_i^t + v_{i-1}^t}{\Delta x^2} - \frac{1}{\rho} \frac{\partial p}{\partial z},$$

$$v_i^{t+1} = v_i^t + \frac{v_{i+1} - v_{i-1}}{2\Delta x} \cdot \frac{v_{i+1}^t - v_{i-1}^t}{2\Delta x} \Delta \tau + v_i^t \frac{v_{i+1}^t - 2v_i^t + v_{i-1}^t}{\Delta x^2} \Delta \tau - \frac{C}{\rho} \Delta \tau,$$

где $C = \partial p / \partial z$. Используемая программа ТР-4 содержит цикл, в котором последовательно перебираются узлы сетки и вычисляются значения v_i .

Конкретизируем задачу. В плоском сосуде находятся две несмешивающиеся жидкости с вязкостями ν_1 и ν_2 ; толщина слоя первой жидкости d_1 , а второй - d_2 . На свободной поверхности второй жидкости находится пластина. Определим поле скоростей жидкости $v(x)$, если: 1) пластина движется равноускоренно с ускорением $a = const$; 2) пластина сначала движется равноускоренно с ускорением $a = const$, а затем равномерно со скоростью $v = const$; 3) пластина совершает гармонические колебания влево-вправо; 4) пластина движется влево, а дно сосуда - вправо [10].

Для одномерного случая получаем [3; 4; 18]:

$$\frac{dv_z}{d\tau} = \nu \frac{\partial^2 v_z}{\partial x^2} - \frac{1}{\rho} \frac{\partial p}{\partial z}, \quad v_{i,j}^{t+1} = v_{i,j}^t + \nu \frac{v_{i-1,j}^t - 2v_{i,j}^t + v_{i+1,j}^t}{\Delta x^2} \Delta \tau - \frac{C}{\rho} \Delta \tau.$$

Используется программа ПР-4, в ней следует задать $C = \partial p / \partial z = 0$ [9; 10]. Рассмотрим результаты вычислений для двух случаев: 1) скорость верхней пластины растет пропорционально времени, $v_1 > v_2$ (рис. 11.4.1); 2) верхняя пластина сначала увеличивает свою скорость, затем движется равномерно, $v_1 < v_2$ (рис. 11.4.2). Чтобы промоделировать другие явления, необходимо соответствующим образом задать граничные условия.

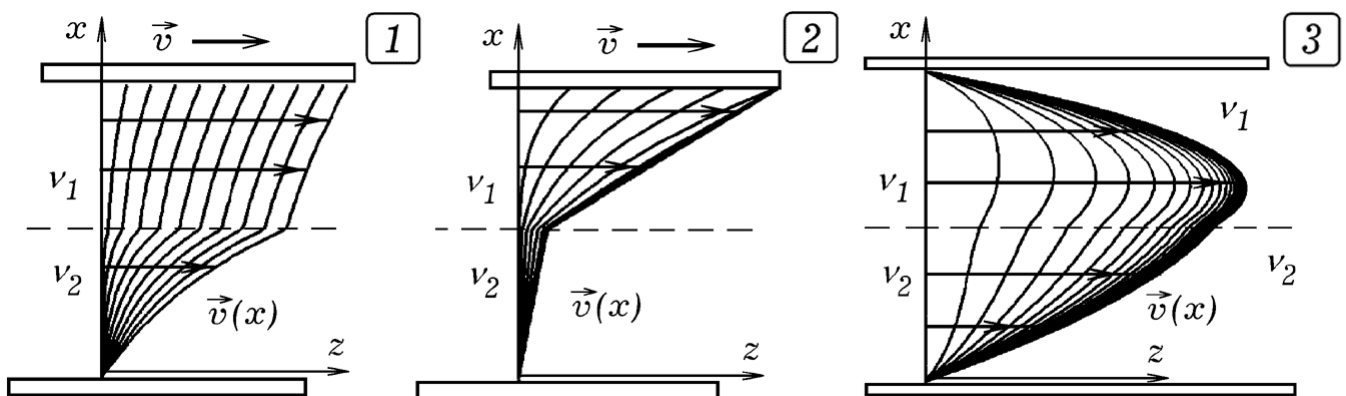


Рис. 11.4. Результаты расчета течения вязкой неоднородной жидкости

Допустим, в пространстве между двумя горизонтальными пластинами большого размера находятся две несмешивающиеся жидкости с различными вязкостями и плотностями. Рассчитаем скорости различных слоев жидкости, если на концах канала поддерживается постоянная разность давлений $C = \partial p / \partial z \neq 0$, а течение ламинарное. Используется программа ПР-4 [10], результат представлен на рис. 11.4.3.

Рассмотрим простую задачу на расчет **конвективных потоков жидкости** [2, с. 32-208; 15; 16]. В большом сосуде, наполненном жидкостью, установлены две вертикальные пластины, расстояние d между которыми много меньше их размеров. Температуры пластин постоянны и равны $T_1 = 0$ и $T_2 > 0$. Считая, что плотность жидкости прямо пропорциональна температуре, рассчитаем ее установившееся течение (поле скоростей $v(x)$). Направим ось Ox перпендикулярно пластинам, а ось Oz - верти-

кально. При $t \rightarrow \infty$ установится стационарное распределение температуры $T(x) = T_2 x / d$ (рис. 11.5.1). Запишем одномерное уравнение конвекции:

$$\frac{\partial v_z}{\partial t} = \nu \frac{\partial^2 v_z}{\partial x^2} + G(T - T_0), \text{ где } v = v_z,$$

где T_0 - температура жидкости в сосуде. В конечных разностях получим:

$$v_i^{t+1} = v_i^t + \nu \frac{v_{i-1}^t - 2v_i^t + v_{i+1}^t}{\Delta x^2} \Delta \tau + G(T_i - T_0) \Delta \tau.$$

Программа ПР-5, используемая для решения этой задачи, содержит цикл по времени, в котором перебираются все узлы одномерной сетки и вычисляются значения скорости v_i на итерационном шаге $t+1$. Получающееся распределение скоростей релаксирует к истинному распределению $v = v(x)$. В случае когда $T_0 = (T_1 + T_2)/2$, профиль скоростей получается симметричным (рис. 11.5.1). Если $T_0 < (T_1 + T_2)/2$, то слой поднимающейся вверх жидкости имеет большую ширину (рис. 11.5.2).

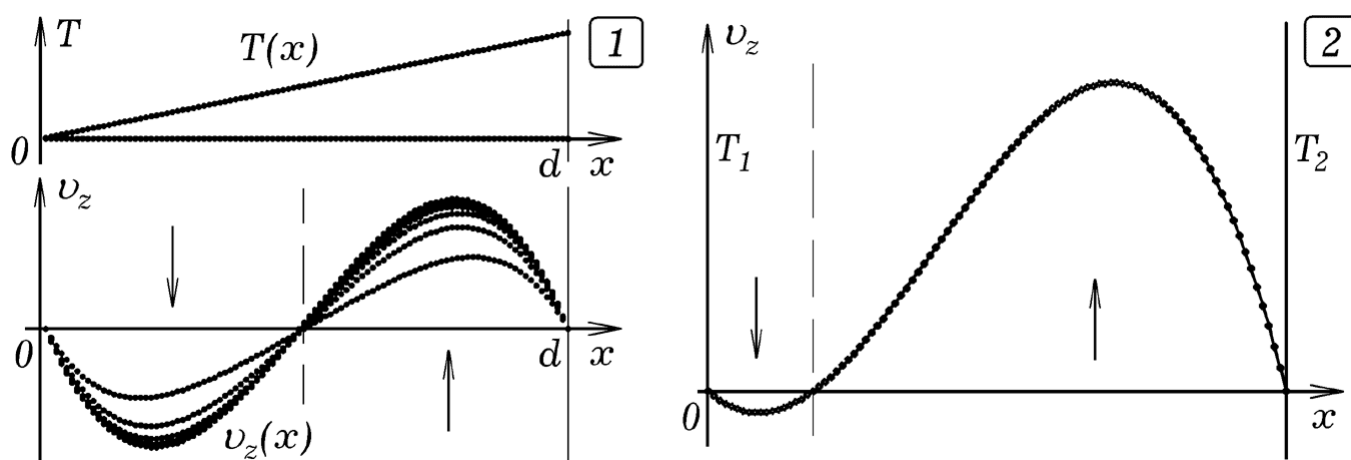


Рис. 11.5. Расчет течения между пластинами с разными температурами

11.4. Двумерное течение вязкой жидкости

Теперь проанализируем установившееся безвихревое течение вязкой жидкости в длинной трубе (канале) постоянного сечения, внутри которой движется бесконечно длинное тело. Это течение инвариантно по отноше-

нию к переносам в направлении движения. Если ось Oz совпадает с направлением течения, а оси Ox и Oy лежат в плоскости, перпендикулярной течению, то можно показать, что выполняется соотношение [1; 3]:

$$\frac{dv_z}{d\tau} = v \left(\frac{\partial^2 v_z}{\partial x^2} + \frac{\partial^2 v_z}{\partial y^2} \right) - \frac{1}{\rho} \frac{\partial p}{\partial z}.$$

где $\nu = \eta / \rho$ - кинематическая вязкость жидкости, $\partial p / \partial z$ - градиент давления. В конечных разностях получаем [3; 11; 16]:

$$v_{i,j}^{t+1} = v_{i,j}^t + \nu \left(\frac{v_{i-1,j}^t - 2v_{i,j}^t + v_{i+1,j}^t}{\Delta x^2} + \frac{v_{i,j-1}^t - 2v_{i,j}^t + v_{i,j+1}^t}{\Delta y^2} - \frac{C}{\rho} \right) \Delta \tau.$$

В качестве примера рассмотрим течение жидкости в узком длинном канале сложного профиля, по которому движется длинное тело, имеющее профиль корабля. Задача решается с помощью программы ТР-6 [9; 10], в ней необходимо правильно задать граничные условия. Слои вязкой жидкости, прилегающие к поверхностям твердых тел, имеют одинаковую с ним скорость (условие прилипания). Если имеется свободная поверхность, то скорость частиц этой поверхности равна скорости частиц, расположенных слоем ниже. Результаты вычислений представлены на рис. 11.6.1. Программа также позволяет рассчитать течение жидкости в канале сложного профиля, в который погружены две длинные пластины, движущиеся в противоположных направлениях (рис. 11.6.2).

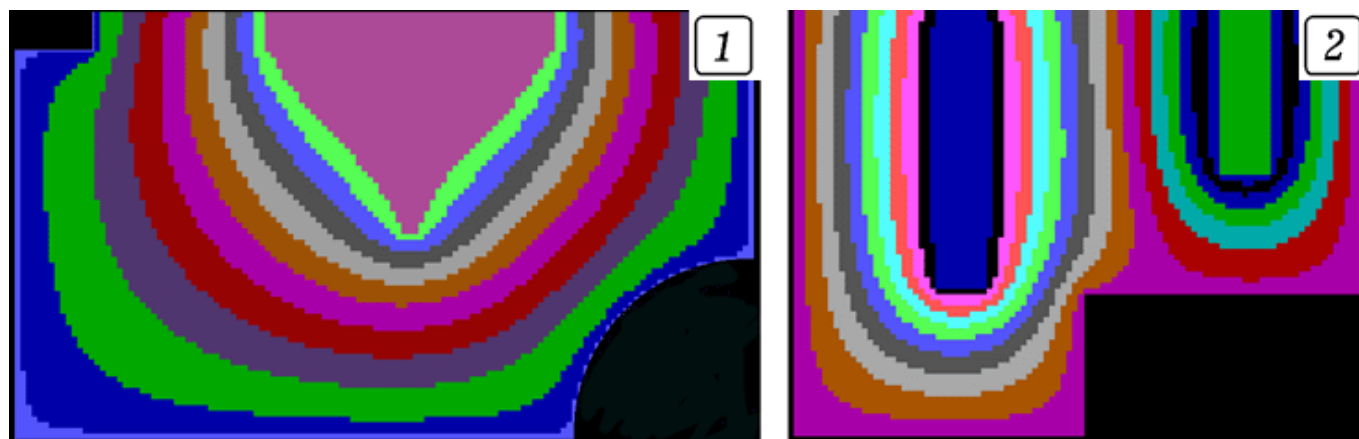


Рис. 11.6. Течение вязкой жидкости, обтекающей движущиеся тела

На рис. 11.7.1 показан результат расчета течения в длинной трубе сложного профиля при наличии разности давлений на концах трубы (во всех точках поперечного сечения). Аналогично рассчитывается поле скоростей в трубе, внутри которой параллельно оси расположен неподвижный стержень (рис. 11.7.2). Эти задачи являются стационарными и могут быть решены более простым способом. Так как $\partial v_z / \partial \tau = 0$, то

$$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = \frac{C}{\rho}, \quad \frac{v_{i-1,j}^t - 2v_{i,j}^t + v_{i+1,j}^t}{\Delta x^2} + \frac{v_{i,j-1}^t - 2v_{i,j}^t + v_{i,j+1}^t}{\Delta y^2} = \frac{C}{\rho},$$

$$v_{i,j} = (v_{i+1,j} + v_{i-1,j} + v_{i,j+1} + v_{i,j-1} - Ch^2 / \rho) / 4,$$

где $h = \Delta x = \Delta y$ - шаг сетки. Так как жидкость вязкая, то вблизи стенки трубы (на границе рассматриваемой области) $v = 0$.

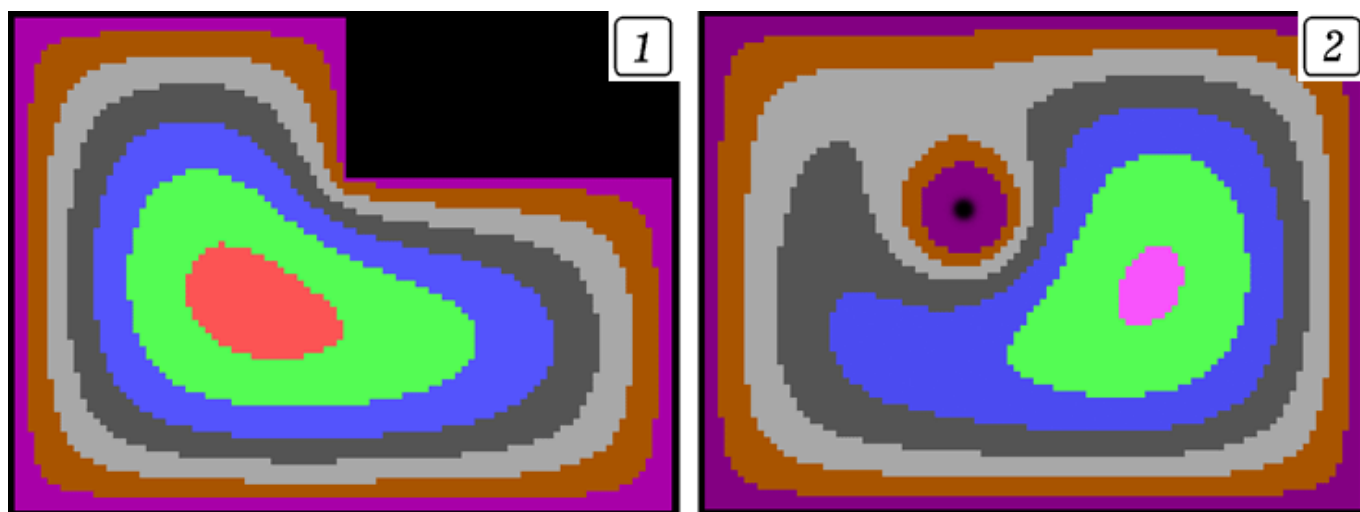


Рис. 11.7. Течение жидкости в трубе при наличии разности давлений

Проанализируем другую задачу. Поток несжимаемой вязкой жидкости набегаёт на бесконечно длинный цилиндр, течение ламинарное. В направлении течения создана разность давлений. Рассчитаем поле скоростей, построим линии тока.

Решение сводится к расчету поля скоростей в одном квадранте декартовой системы координат xOy (рис. 11.8). Из уравнений Навье-Стокса для двумерного поля скоростей частиц несжимаемой жидкости [16-20]:

$$\frac{dv_x}{dt} = \nu \left(\frac{\partial^2 v_x}{\partial x^2} + \frac{\partial^2 v_x}{\partial y^2} \right) - \frac{1}{\rho} \frac{\partial p}{\partial x}, \quad \frac{dv_y}{dt} = \nu \left(\frac{\partial^2 v_y}{\partial x^2} + \frac{\partial^2 v_y}{\partial y^2} \right).$$

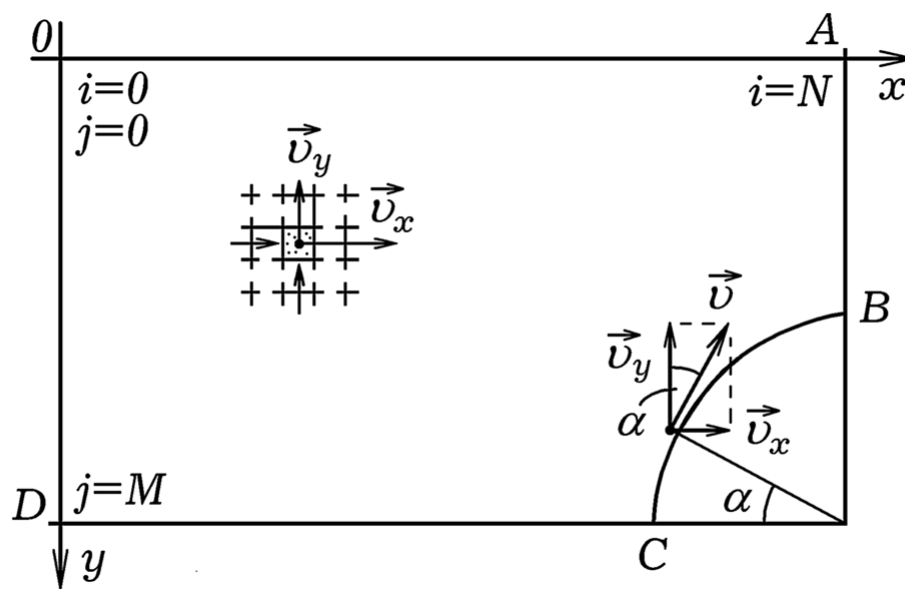


Рис. 11.8. К расчету поля скоростей в потоке вязкой жидкости

Так как отсутствуют источники, то $\text{div } \vec{v} = \partial v_x / \partial x + \partial v_y / \partial y = 0$. Расчет течения сводится к нахождению распределения скоростей, удовлетворяющему записанным дифференциальным уравнениям и граничным условиям (рис. 11.8): 1) на границе OA $v_x = V_\infty$, $v_y = 0$; 2) на границе AB $v_y = 0$; 3) на границе BC $v_n = 0$, $v = v_\tau$, $v_x = v \sin \alpha$, $v_y = -v \cos \alpha$; 4) на границе CD $v_y = 0$; 5) на границе DO $v_x = V_\infty$, $v_y = 0$ [1, с. 20-27, с. 80-83].

В конечных разностях получаем:

$$\begin{aligned} v_{xi,j}^{t+1} &= v_{xi,j}^t + (\nu(v_{xi-1,j}^t + v_{xi+1,j}^t + v_{xi,j-1}^t + v_{xi,j+1}^t - 4v_{xi,j}^t) / h^2 - C / \rho) \Delta \tau, \\ v_{yi,j}^{t+1} &= v_{yi,j}^t + \nu(v_{yi-1,j}^t + v_{yi+1,j}^t + v_{yi,j-1}^t + v_{yi,j+1}^t - 4v_{yi,j}^t) \Delta \tau / h^2, \\ \Delta x &= \Delta y = h, \quad v_{x1,j} = V_\infty, \quad v_{y1,j} = 0, \quad v_{yN,j} = 0, \quad v_{yN,j} = v_{yN-1,j}, \\ v_{xi,1} &= V_\infty \quad \text{при} \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, M. \end{aligned}$$

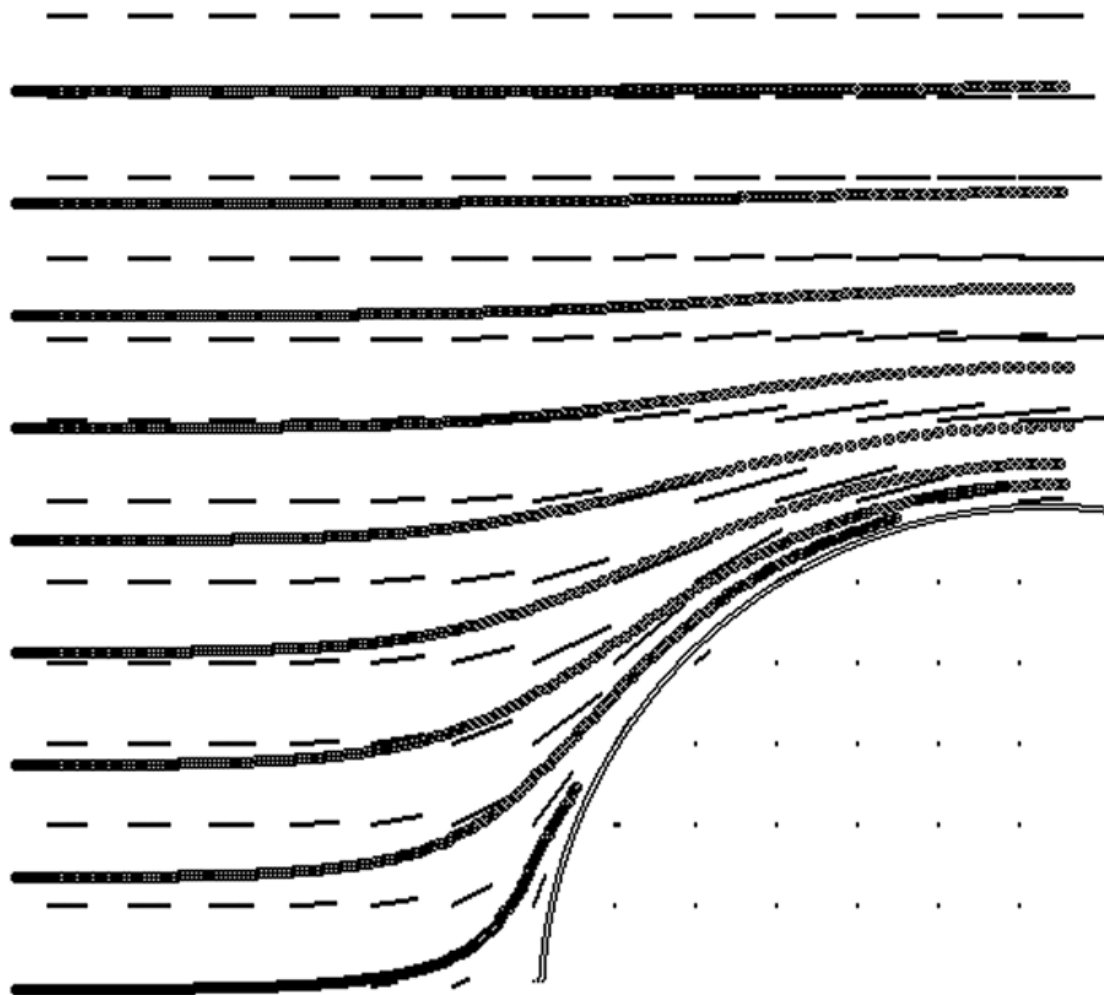


Рис. 11.9. Обтекание цилиндра вязкой жидкостью

Так как во всех узлах сетки $v_x \geq 0$, $v_y \leq 0$, то из уравнения $\text{div } \vec{v} = 0$ следует $v_{xi,j} - v_{yi,j} = v_{xi+1,j} - v_{yi,j+1}$. В самом деле, суммарный поток вектора скорости через нижнюю и левую стенки квадратной ячейки должен быть равен суммарному потоку через верхнюю и правую стенки. Для границы СВ (значение $(N-i)^2 + (M-j)^2$ лежит в интервале от R^2 до $(R + \Delta R)^2$) должно выполняться: $v_{xi,j} = v(M-j)/R$, $v_{yi,j} = -v(N-i)/R$, где $v = \sqrt{v_{xi,j}^2 + v_{yi,j}^2}$. Используемая программа ПР-7 содержит вложенные циклы по i и j , в которых перебираются все узлы двумерной сетки и вычисляются значения проекций скорости. Затем отдельно для каждого узла находится коэффициент $s = (v_{xi-1,j} - v_{yi,j+1}) / (v_{xi,j} - v_{yi,j})$ и пересчитываются проекции скорости так, чтобы они удовлетворяли условию

$\operatorname{div} \vec{v} = 0$ [9]. После расчета поля скоростей программа строит вектора $\vec{v}$ в узлах сетки (рис. 11.9). Чтобы получить линии тока, в левую часть потока вводятся легкие частицы-маркеры, которые смещаются в направлении течения жидкости: $x^{t+1} = x^t + v_{xi,j} \Delta \tau$, $y^{t+1} = y^t + v_{yi,j} \Delta \tau$.

11.5. Вихревое движение вязкой жидкости

Важная проблема вычислительной гидродинамики состоит в расчете движения жидкости с образованием вихрей. Классической является двумерная задача о течении вязкой несжимаемой жидкости в прямоугольной полости с движущейся крышкой. Она формулируется так: вязкая несжимаемая жидкость целиком заполняет двумерную прямоугольную полость (каверну), накрытую сверху твердой пластиной, движущейся с постоянной скоростью $\vec{V}$ (рис. 11.10). Стенки полости непроницаемы. Необходимо рассчитать поле скоростей [3, с. 148-156].

Из уравнений Навье-Стокса для плоского течения несжимаемой вязкой жидкости следует [1-4; 7; 8; 11-13; 18-20]:

$$\begin{aligned} \frac{\partial v_x}{\partial t} + v_x \frac{\partial v_x}{\partial x} + v_y \frac{\partial v_x}{\partial y} &= -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left(\frac{\partial^2 v_x}{\partial x^2} + \frac{\partial^2 v_x}{\partial y^2} \right), \\ \frac{\partial v_y}{\partial t} + v_x \frac{\partial v_y}{\partial x} + v_y \frac{\partial v_y}{\partial y} &= -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left(\frac{\partial^2 v_y}{\partial x^2} + \frac{\partial^2 v_y}{\partial y^2} \right), \quad \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} = 0. \end{aligned}$$

Введем функцию тока $\psi(x, y, t)$ такую, что: $v_x = \partial \psi / \partial y$, $v_y = -\partial \psi / \partial x$.

Течение двумерное, вихрь скорости $\vec{\omega}$ по модулю равен (рис. 11.10):

$$\omega = -\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} = -\frac{\partial^2 \psi}{\partial x^2} - \frac{\partial^2 \psi}{\partial y^2},$$

а по направлению он перпендикулярен плоскости рисунка. Учтем, что нет перепада давлений: $\partial p / \partial x = \partial p / \partial y = 0$. Если первое уравнение из систе-

мы Навье-Стокса продифференцировать по y , второе продифференцировать по x , а затем уравнения вычесть, то получится:

$$\frac{\partial \omega}{\partial t} + v_x \frac{\partial \omega}{\partial x} + v_y \frac{\partial \omega}{\partial y} = \nu \left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right).$$

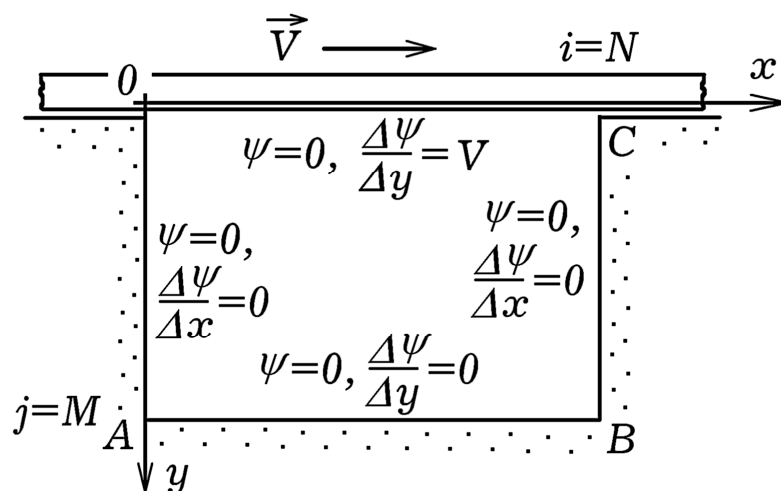


Рис. 11.10. Расчет течения жидкости в каверне с подвижной крышкой

Последние два уравнения и составляют систему "функция тока - вихрь скорости". Расчетную область покроем двумерной сеткой и запишем уравнения в конечных разностях [3; 7; 11-13]:

$$\begin{aligned} \psi_{i,j} &= (\psi_{i-1,j} + \psi_{i+1,j} + \psi_{i,j-1} + \psi_{i,j+1} + \omega_{i,j} h^2) / 4, \\ v_{xi,j} &= (\psi_{i,j+1} - \psi_{i,j-1}) / 2h, \quad v_{yi,j} = -(\psi_{i+1,j} - \psi_{i-1,j}) / 2h, \\ \omega_{i,j}^{t+1} &= \omega_{i,j}^t - v_{xi,j}^t \frac{\omega_{i+1,j}^t - \omega_{i-1,j}^t}{2h} \Delta \tau - v_{yi,j}^t \frac{\omega_{i,j+1}^t - \omega_{i,j-1}^t}{2h} \Delta \tau + \\ &+ \nu \frac{\omega_{i-1,j}^t + \omega_{i+1,j}^t + \omega_{i,j-1}^t + \omega_{i,j+1}^t - 4\omega_{i,j}^t}{h^2} \Delta \tau. \end{aligned}$$

Результат расчета поля скоростей зависит от граничных условий, которые задаются так (рис. 11.10): 1) на левой границе OA $\psi = 0$, $\partial \psi / \partial x = 0$, вихрь скорости $\omega_{1,j} = 2\psi_{2,j} / h^2$; 2) на нижней границе AB $\psi = 0$, $\partial \psi / \partial y = 0$, вихрь скорости $\omega_{i,M} = 2\psi_{i,M-1} / h^2$; 3) на правой границе BC

$\psi = 0$, $\partial\psi/\partial x = 0$, вихрь скорости $\omega_{N,j} = 2\psi_{N-1,j}/h^2$; 4) на верхней границе CO $\psi = 0$, $\partial\psi/\partial y = V$, вихрь $\omega_{i1} = 2\psi_{i,2}/h^2 - 2V/h$ [2; 7; 19].

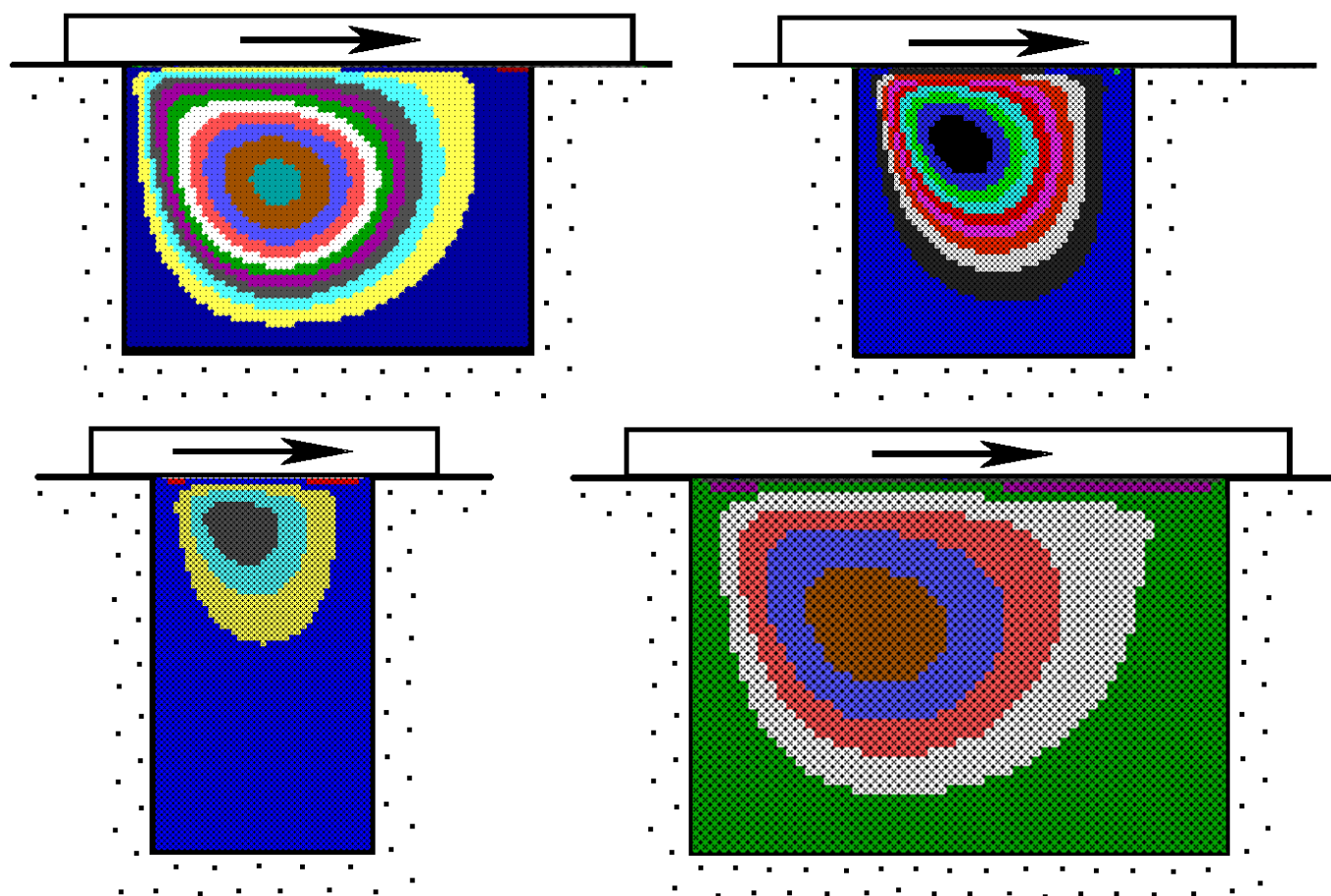


Рис. 11.11. Течение жидкости в полости с движущейся крышкой

Используется программа ТР-8 [9], результаты ее работы для каверн прямоугольной формы представлены на рис. 11.11. Для расчета поля скоростей необходимо, чтобы программа выполнила более 1000 итераций. На рисунках показано получающееся распределение функции тока; границы разноцветных областей соответствуют линиям тока. Видно, что центр образующегося вихря смещен в сторону, противоположную движению крышки. В нижней части глубокой полости жидкость остается практически неподвижной. В принципе несложно построить линии тока, запустив внутрь полости частицы-маркеры, смещающиеся в направлении движения жидкости.

Анализируемая программа позволяет исследовать вихревое течение жидкости в каверне с подвижной крышкой при различных скоростях движения крышки, размерах каверны, вязкости жидкости. С ее помощью также можно рассчитать поле скоростей для каверны с движущейся крышкой и дном (рис. 11.12). Хорошо видно, что когда скорости крышки и дна направлены противоположно, сначала образуются два вихря, закрученных в одном направлении, а затем они через некоторое время объединяются в один (рис. 11.12.1).

В случае когда крышка и дно движутся в одном направлении с неравными скоростями, возникают два противоположно направленных вихря разных размеров (рис. 11.12.2). На экране монитора получается цветное изображение, границы разноцветных областей соответствуют линиям тока. Для получения стационарного решения необходимо, чтобы программа выполнила более 1500 итераций. Изображение на экране обновляется через 200 итераций. Результаты численного моделирования вихревого течения жидкости в кавернах различного профиля с подвижной крышкой представлены на рис. 11.13.

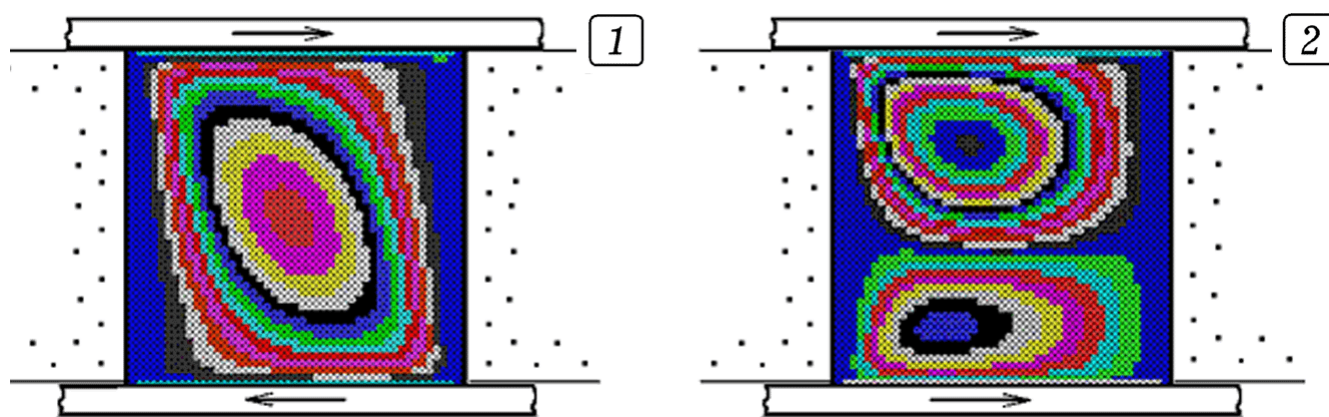


Рис. 11.12. Течение жидкости в каверне с подвижными дном и крышкой

Программа ТР-8 рассчитывает функцию тока в каждом узле сетки. Можно добиться того, чтобы она вычисляла проекции скорости $v_x = \partial\psi / \partial y$, $v_y = -\partial\psi / \partial x$ и для каждого узла строила вектор скорости $\vec{v}$ частиц жидкости (программа ТР-9). Это позволит определить поле ско-

ростей для каверны с подвижной крышкой (рис 11.14.1) и для каверны с движущимися дном и крышкой (рис 11.14.2) [9], а также изучить зависимость течения жидкости от ее плотности, вязкости, размеров полости и скорости движения крышки.

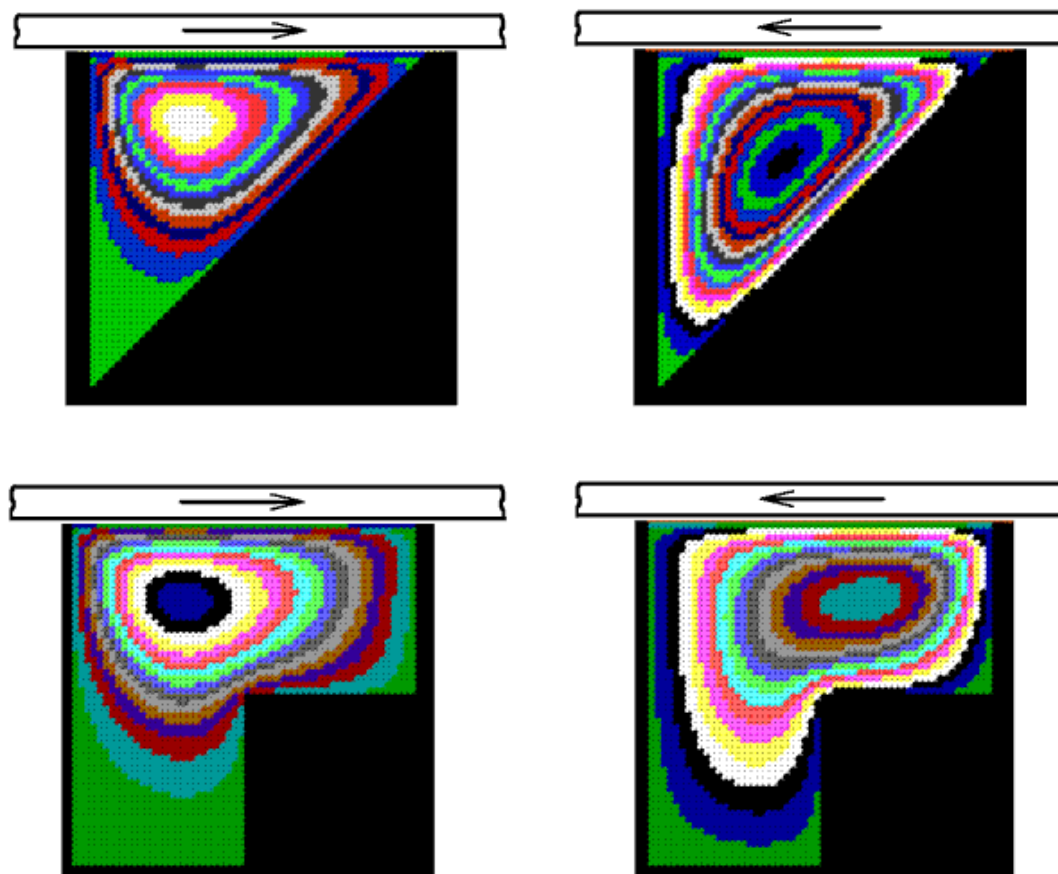


Рис. 11.13. Течение жидкости в треугольной каверне и каверне с выступом

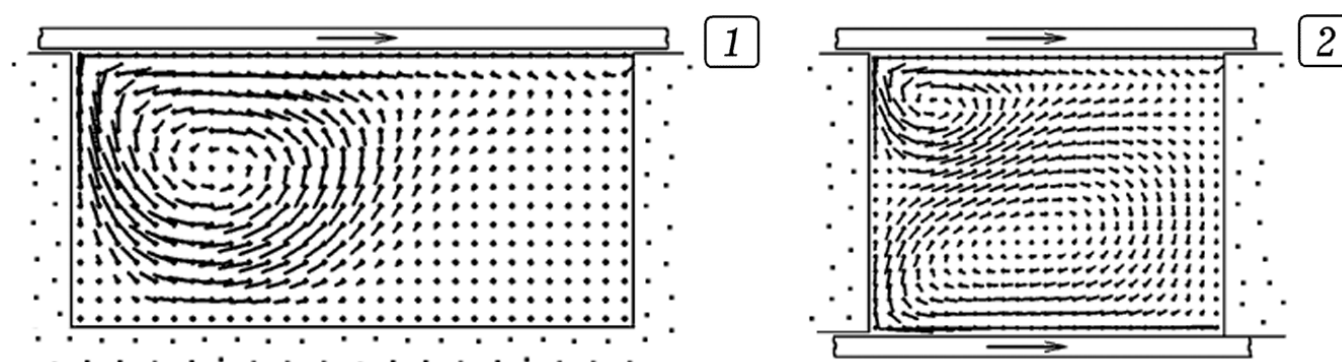


Рис. 11.14. Поле скоростей в каверне с движущимися дном и крышкой

Рассмотренный выше метод позволяет исследовать вихревое течение жидкости за препятствием и в каверне без крышки. Программа ТР-10

моделирует образование вихрей при обтекании вязкой жидкостью одного и двух препятствий (рис. 11.15). Как заданы граничные условия, ясно из процедуры *Gr_usl*. Обсуждаемая программа позволяет, например, рассчитать вихри за пластиной, на которую набегают поток жидкости [7, с. 182-193]. Так как течение жидкости симметрично относительно горизонтальной плоскости, то достаточно решить задачу для верхней полу-плоскости.

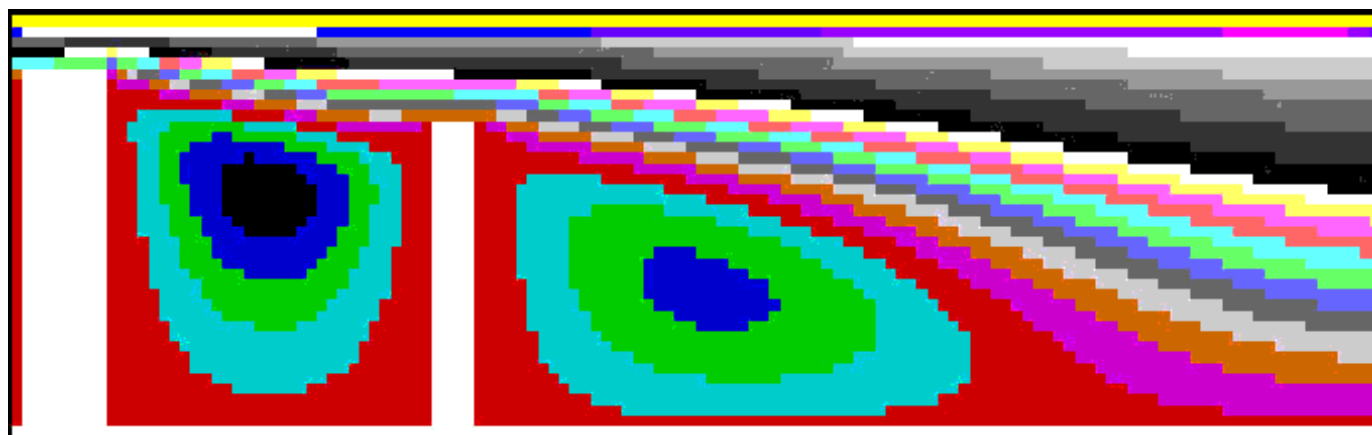


Рис. 11.15. Вихревое течение жидкости за препятствием и в каверне

11.6. Конвективное движение жидкости в двумерном случае

Особый интерес представляет собой проблема моделирования **свободной конвекции жидкости**. Пусть в поле тяжести имеется полость, наполненная жидкостью, и создана разность температур между различными стенками полости, либо внутри нее находится источник тепла или холода. При этом возникает конвекция – перенос теплоты потоками вещества. Она обусловлена тем, что нагретые слои жидкости имеют меньшую плотность и поднимаются вверх, а холодные – опускаются вниз. Чтобы найти распределения скорости и температуры жидкости, устанавливающиеся в результате ее конвективного движения и теплообмена, следует

совместно решить систему из уравнений Навье-Стокса в переменных "функция тока - вихрь скорости" и уравнения теплопроводности [1; 2; 15]:

$$\frac{\partial \omega}{\partial t} = \nu \left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right) - \frac{\partial \omega}{\partial x} \frac{\partial \psi}{\partial y} + \frac{\partial \omega}{\partial y} \frac{\partial \psi}{\partial x} - G \frac{\partial T}{\partial x},$$

$$\frac{\partial T}{\partial t} = k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) - \frac{\partial \psi}{\partial y} \frac{\partial T}{\partial x} + \frac{\partial \psi}{\partial x} \frac{\partial T}{\partial y} + \frac{q(x, y)}{\rho c}, \quad \omega = -\frac{\partial^2 \psi}{\partial x^2} - \frac{\partial^2 \psi}{\partial y^2},$$

где T - температура, q - плотность мощности источников тепла, слагаемое $-G\partial T/\partial x$ учитывает подъем нагретых слоев жидкости (ось Oy направлена вниз). Для численного решения этой системы уравнений строят конечно-разностную схему и задают граничные условия для ψ , ω и T [1; 2; 15]. Программа ПР-11 позволяет промоделировать гравитационную конвекцию жидкости в прямоугольной полости в следующих случаях: 1) при подогреве снизу, что приводит к возникновению валков (рис. 11.16), аналогичных **ячейкам Бенара**; 2) при подогреве сбоку (рис. 11.17), а также при подогреве слева и охлаждении справа; 3) при наличии источника тепла или холода внутри жидкости (рис. 11.18) [9].

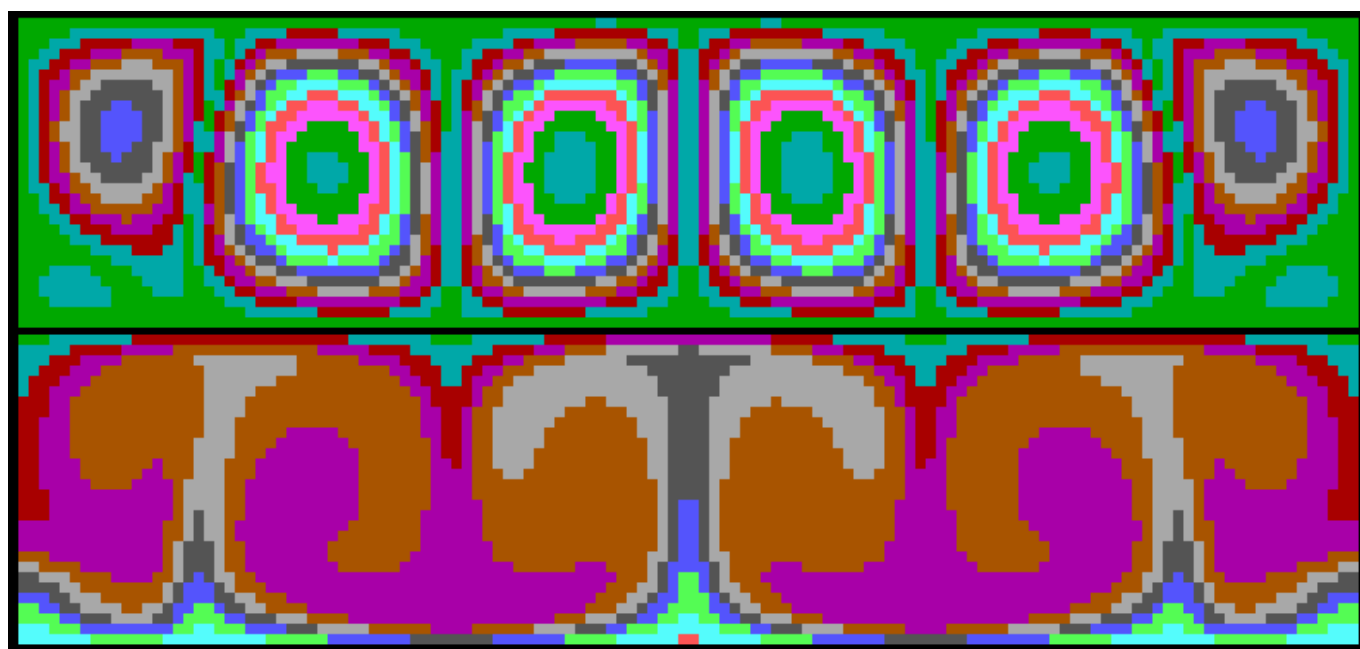


Рис. 11.16. Моделирование конвекции и образования ячеек Бенара

Граничные условия задаются в процедуре *Gr_usl*. Компьютерная модель позволяет провести целую серию вычислительных экспериментов, изучая конвекцию при различных вязкости, теплопроводности, плотности жидкости, размерах полости, температурах ее стенок, мощностях и положениях источников тепла (холода). Течение нестационарное, рис. 11.16-11.19 соответствуют некоторому моменту времени τ . Например, на рис. 11.16 сверху показано распределение функции тока (границы разноцветных областей – линии тока), а снизу – распределение температуры (границы – изотермы) в случае, когда дно сосуда нагрето. На рис. 11.17 показан результат расчета поля скоростей (слева) и поля температур (справа) в некоторый момент времени в случае, когда нагревается левая стенка полости. Хорошо видно, что возникает вихревое движение жидкости, более нагретые слои поднимаются по левой стенке, затем смещаются вправо и опускаются у правой стенки. Средняя температура повышается. При запуске программы на экране монитора возникают цветные динамические анимации; статические картинки не позволяют передать красоту моделируемых явлений в полной мере.

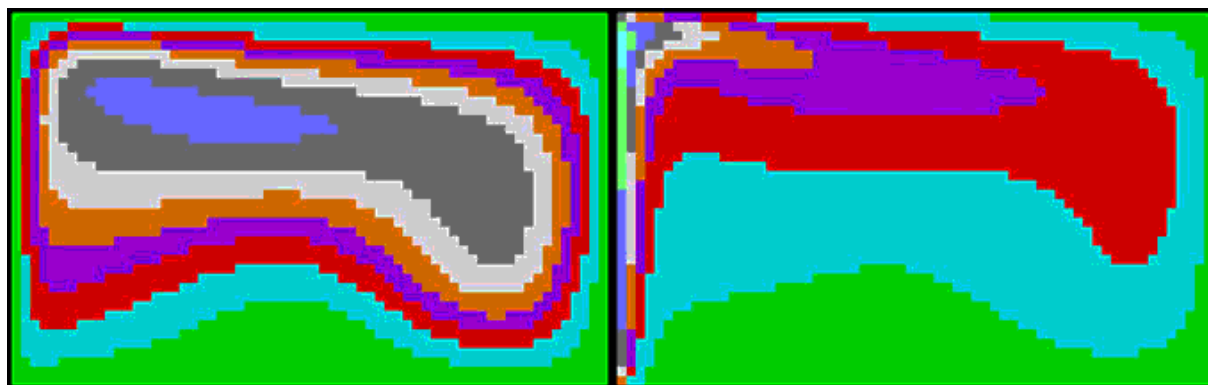


Рис. 11.17. Результаты моделирования конвекции (подогрев слева)

На рис. 11.18 моделируется конвективное движение жидкости, когда внутри нее находится небольшой источник тепла. Нагретые слои жидкости (газа), поднимаясь вверх, образуют вихрь, похожий на "гриб" атомного взрыва. Несложно промоделировать конвективные потоки жидкости в

случае, когда источник тепла смещен в сторону или когда внутри полости имеется источник тепла и источник холода [9]. Для того чтобы промоделировать горение в двумерной среде (рис. 11.19), необходимо учесть, что при высоких температурах газ находится в плазменном состоянии и излучает энергию, причем теряемая мощность пропорциональна T^β , где $\beta \approx 3$ (глава 10). Эта энергия не идет на нагревание слоев газа. В программе ТР-11 необходимо активизировать закомментированный оператор. При этом если температура узла превышает некоторое пороговое значение T_{KR} , то считается, что в этом узле находится поглотитель тепла, мощность которого составляет bT^β .

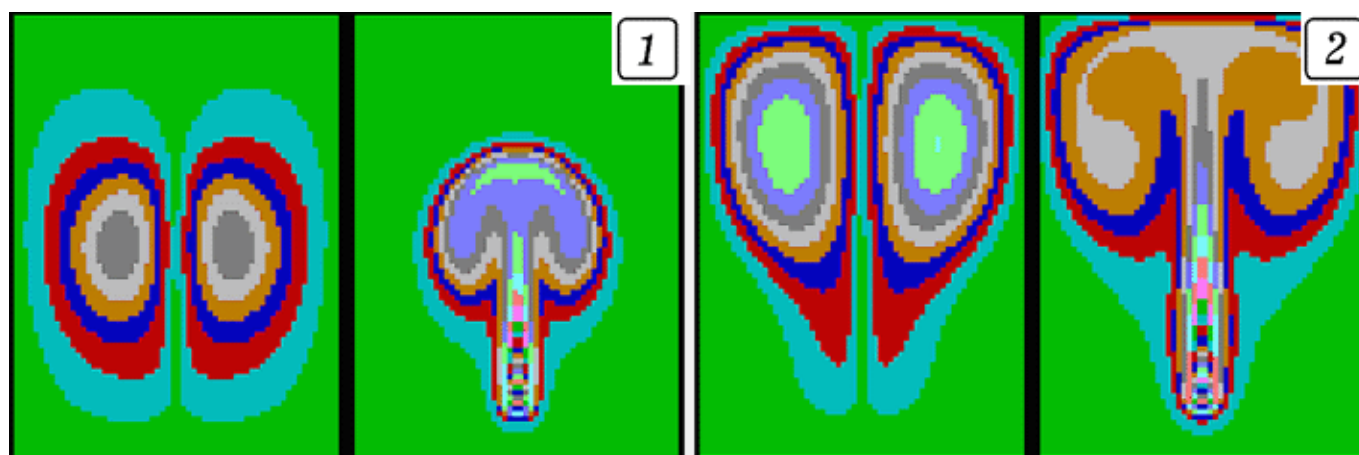


Рис. 11.18. Конвекция от небольшого источника тепла внутри сосуда

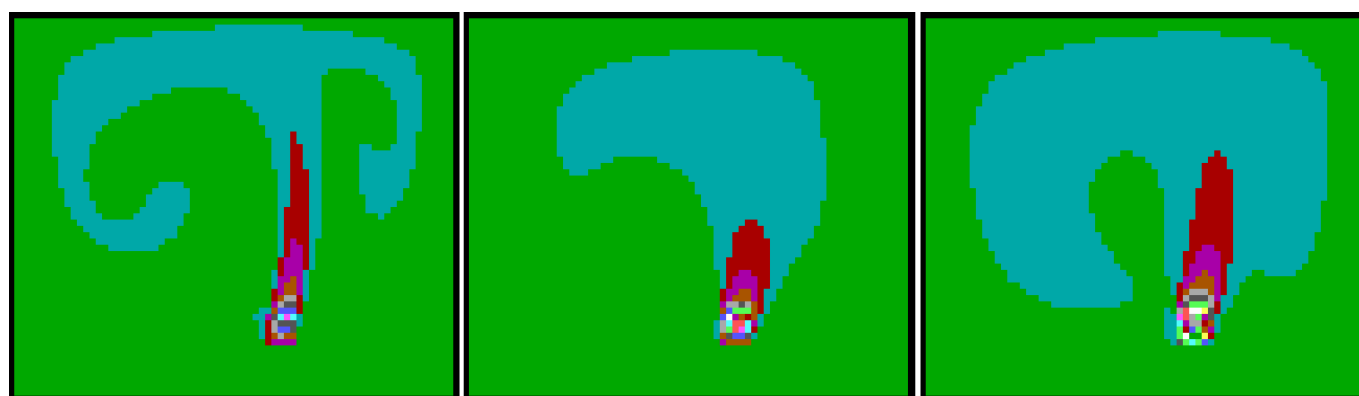


Рис 11.19. Моделирование горения в двумерной среде

Общие идеи моделирования рассмотренных выше явлений изложены в научной и учебной литературе [1-20]. Некоторые из них обсуждались

профессором В. А. Сараниным [14] на конференции "Учебный физический эксперимент: Проблемы и решения", проводимой в Глазовском педагогическом институте.

Приложение к главе 11

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

```
program Vitekanie_iz_sosuda;
uses crt, graph;
const s2=0.1; dt=0.005;
var DV,MV: integer; y,dy,R,t,v1,v2,s1:real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
y:=5*6.28; line(0,300,300,300);
Repeat v2:=sqrt(2*9.8*y); R:=1.3+SIN(y);
s1:=3.14*R*R; dy:=-s2*v2*dt/s1;
y:=y+dy; v1:=-dy/dt; t:=t+dt;
circle(10+round(2*t),300-round(5*y),1);
circle(10+round(2*t),300-round(10*v2),1);
circle(10+round(2*t),300-round(10*v1),1);
until (y<0)or(KeyPressed);
Repeat until KeyPressed; CloseGraph;
END.
```

Программа ПР-2

```
program Potencialnoe_techenie;
{$N+}uses crt, graph;
const n=170; m=50;
var i,ii,j,jj,k,DV,MV: integer;
    psi: array[1..N, 1..M] of real;
Procedure Raschet;
begin psi[i,j]:=(psi[i+1,j]+psi[i-1,j]+
                psi[i,j+1]+psi[i,j-1])/4; end;
Procedure Gran_usl;
```

```

begin
For i:=1 to N do begin psi[i,2]:=-200; psi[i,M-1]:=200;
      psi[i,1]:=-200; psi[i,M]:=200; end;
For j:=1 to 25 do begin
      psi[2,j]:=-208+16*j; psi[N-1,j]:=-208+16*j;
      psi[1,j]:=-208+16*j; psi[N,j]:=-208+16*j; end;
For i:=1 to N do For j:=1 to M do
If (j<35)and((i-60)*(i-60)+(j-25)*(j-25)<200)
      then psi[i,j]:=0;
For i:=1 to N do For j:=1 to M do
If (abs(j-25)<15)and(abs(i-110)<5) then psi[i,j]:=0;
For i:=1 to N do For j:=1 to M do
If (j>25)and((i<30)or(i>150))then psi[i,j]:=200; end;
Procedure Draw;
begin
setcolor(round((psi[i,j]+200)/30)+1);
If ((j<35)and((i-60)*(i-60)+(j-25)*(j-25)<200)) or
      ((j>25)and((i<30)or(i>150)))or((abs(j-25)<15)
      and(abs(i-110)<5)) then setcolor(15);
rectangle(i*4+5,j*4,i*4+8,j*4+3);
rectangle(i*4+6,j*4+1,i*4+7,j*4+2);
end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat inc(k); Gran_usl;
      For i:=2 to N-1 do For j:=2 to M-1 do Raschet;
      Gran_usl;
      For j:=M-1 downto 2 do For i:=N-1 downto 2 do Raschet;
      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-3

```

program Potenc_techen_linii_toka;
uses crt, graph;
const n=120; m=80; dt=0.5;
var i,j,k,s,DV,MV: integer;
psi: array[1..N, 1..M] of real;

```

```

x,y,vx,vy:array[1..8] of real;
Procedure Raschet;
begin psi[i,j]:=(psi[i+1,j]+psi[i-1,j]+
                psi[i,j+1]+psi[i,j-1])/4; end;
Procedure Gran_usl;
begin For j:=1 to M do begin psi[1,j]:=j-1;
psi[N,j]:=8/3*(j-1); end;
For i:=1 to N do begin psi[i,1]:=0; psi[i,M]:=80; end;
For i:=1 to N do For j:=1 to M do begin
If (i-60)*(i-60)+(j-40)*(j-40)<400 then psi[i,j]:=40;
If (i>110)and(j>31) then psi[i,j]:=80; end; end;
Procedure Draw1;
begin setcolor(round(psi[i,j]/3)mod 15);
If (i-60)*(i-60)+(j-40)*(j-40)<400 then setcolor(15);
rectangle(i*5,j*5,i*5+3,j*5+3);
rectangle(i*5+1,j*5+1,i*5+2,j*5+2); end;
Procedure Draw2;
begin For s:=1 to 8 do begin
If (i=round(x[s]/5+1))and(j=round(y[s]/5)) then
begin vx[s]:=psi[i,j+1]-psi[i,j-1];
vy[s]:=psi[i+1,j]-psi[i-1,j];
x[s]:=x[s]+vx[s]*dt; y[s]:=y[s]-vy[s]*dt;
circle(round(x[s]),round(y[s]),2);
end; end; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi'); randomize;
For s:=1 to 8 do begin x[s]:=10; y[s]:=55*s-20; end;
For i:=2 to N-1 do For j:=2 to M-1 do psi[i,j]:=40;
Repeat inc(k); Gran_usl;
  For i:=2 to N-1 do
    For j:=2 to M-1 do Raschet; Gran_usl;
  For j:=M-1 downto 2 do
    For i:=N-1 downto 2 do Raschet;
  If (k mod 100=0)and(k<800) then
    For i:=2 to N-1 do For j:=2 to M-1 do Draw1;
  If k>800 then
    For i:=2 to N-1 do For j:=2 to M-1 do Draw2;
until KeyPressed; CloseGraph;
END.

```

Программа ПР-4

```

program Odnomernoe_pole_scorostei_vyazkost;
uses crt, graph;
const n=90; h=1; dt=0.05; p=0.02;
var k,ii,i,DV,MV : integer;
rho,t: real; nu,v,vv: array[0..N+1] of real;
Procedure Raschet;
begin If (i<45) then rho:=0.7 else rho:=1.4;
vv[i]:=v[i]+(nu[i+1]-nu[i-1])*(v[i+1]-
v[i-1])*dt/h/h/4+nu[i]*(v[i+1]-2*v[i]+
v[i-1])*dt/h/h+p*dt/rho; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
For i:=1 to N do
  If (i<45) then nu[i]:=0.7 else nu[i]:=1.3;
Repeat k:=k+1; t:=t+0.01; v[1]:=0; v[N]:=0;
For i:=1 to N do v[i]:=vv[i];
  For i:=2 to N-1 do Raschet;
If k mod 2000=0 then For i:=1 to N do
  begin line(round(100+v[i]*16),i*4+50,
            round(100+v[i+1]*16),i*4+54); k:=0; end;
until KeyPressed; CloseGraph;
END.

```

Программа ПР-5

```

Program Konvekciya_odnomernoe_techenie;
Uses crt, graph;
Const n=100; dt=0.2; h=1; Tsr=4;
Var i,k,DV,MV : integer; T,v: array[1..N] of real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
v[1]:=0; v[N]:=0; line(50,0,50,480); line(550,0,550,480);
For i:=1 to N do begin T[i]:=0.1*i; circle(50+5*i,140,2);
circle(50+5*i,140-round(10*T[i]),2); end;
Repeat inc(k); For i:=2 to N-1 do v[i]:=v[i]+
2*(v[i-1]-2*v[i]+v[i+1])/h/h*dt+0.3*(T[i]-Tsr)*dt;
For i:=N-1 downto 2 do v[i]:=v[i]+
2*(v[i-1]-2*v[i]+v[i+1])/h/h*dt+0.3*(T[i]-Tsr)*dt;
If k mod 500=0 then For i:=1 to N do begin k:=0;
  circle(50+5*i,320-round(v[i]),2);

```

```

    circle(50+5*i,320-round(v[i]),1); end;
until (KeyPressed); Repeat until KeyPressed; CloseGraph;
END.

```

Программа ПР-6

```

Program Obtekanie_korablya_vyazkost;
uses crt, graph;
const N=90; M=58; h=1; dt=0.01; p=0;
var  ii,jj,kk,i,j,DV, MV: integer;
      vv,v: array[0..N+1, 0..M+1] of real;
Procedure Gr_usl;
begin
  For i:=1 to N do For j:=1 to M do v[i,j]:=vv[i,j];
  For i:=2 to N-1 do For j:=2 to M-1 do begin
    If (j<5)and(abs(5-i)<5) then v[i,j]:=0;
    If (j<40)and(abs(i-50)<1+20*sqr(cos(j/25)))
      then v[i,j]:=300; end;
  For i:=1 to N do v[i,M]:=0;
  For j:=1 to M do begin v[1,j]:=0; v[N,j]:=0; end;
end;
Procedure Raschet;
begin
  vv[i,j]:=v[i,j]+(v[i,j+1]-2*v[i,j]+v[i,j-1])*dt/(h*h)
    +(v[i+1,j]-2*v[i,j]+v[i-1,j])*dt/(h*h)+p; end;
procedure Draw;
begin setcolor(round(v[i,j]/30)+1);
  If v[i,j]<1 then setcolor(9);
  rectangle(i*5,j*5,i*5+4,j*5+4);
  rectangle(i*5+1,j*5+1,i*5+3,j*5+3);
  rectangle(i*5+2,j*5+2,i*5+3,j*5+3);
end;
BEGIN
DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat Gr_usl;
  For i:=1 to N-1 do For j:=2 to M-1 do Raschet;
  Gr_usl; For j:=M-1 downto 2 do
  For i:=N-1 downto 2 do Raschet;
  For i:=1 to N do vv[i,1]:=vv[i,2]; kk:=kk+1;
  If kk/30=round(kk/30) then
    For i:=2 to N-1 do For j:=2 to M-1 do Draw;

```

```
until KeyPressed; CloseGraph;
END.
```

Программа ПР-7

```
program Obtekanie_cilindra_vyazkost;
{$N+}uses crt, graph;
const n=40; m=40; dt=0.2; h=1; p=0.4;
var i,j,DV,MV: integer; s,v2,x,y: single;
vx,vy,vvx,vvy: array[0..N+1,0..M+1] of single; k:longint;
Procedure Gr_usl;
begin For i:=1 to N do For j:=1 to M do begin
    vx[1,j]:=20; vy[1,j]:=0; vy[N,j]:=0;
    vx[N,j]:=vx[N-1,j]; vx[i,1]:=20;
    If (i-N)*(i-N)+(j-M)*(j-M)<360 then
    begin vx[i,j]:=0; vy[i,j]:=0; end;
    If ((i-N)*(i-N)+(j-M)*(j-M)>360) and
    ((i-N)*(i-N)+(j-M)*(j-M)<405) then begin
    v2:=sqrt(sqr(vx[i,j])+sqr(vy[i,j]));
    vx[i,j]:=v2*(M-j)/20; vy[i,j]:=-v2*(N-i)/20; end; end;
end;
Procedure Ur1;
begin vvz[i,j]:=vx[i,j]+(vx[i+1,j]-2*vz[i,j]+vx[i-1,j]+
    vx[i,j+1]-2*vz[i,j]+vx[i,j-1])*dt+p;
    vvy[i,j]:=vy[i,j]+(vy[i+1,j]-2*vvy[i,j]+vy[i-1,j]+
    vy[i,j+1]-2*vvy[i,j]+vy[i,j-1])*dt; end;
Procedure Ur2;
begin For i:=1 to N do For j:=1 to M do begin
    s:=(vvz[i-1,j]-vvy[i,j+1])/(vvz[i,j]-vvy[i,j])+0.001;
    vx[i,j]:=vvz[i,j]*s; vy[i,j]:=vvy[i,j]*s; end; end;
Procedure Draw1;
begin If (i mod 3=0) and (j mod 3=0) then line(12*i,12*j,
    12*i+round(vx[i,j]/1),12*j+round(vy[i,j]/1)); end;
Procedure Draw2;
begin i:=round(x/12); j:=round(y/12);
x:=x+vx[i,j]*0.1; y:=y+vy[i,j]*0.1;
circle(round(x),round(y),2); end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat k:=k+1; Gr_usl;
For i:=1 to N do For j:=1 to M do Ur1; Ur2; Gr_usl;
For j:=M downto 1 do For i:=N downto 1 do Ur1; Ur2;
```

```

If (k mod 10=0)and(k<100) then begin cleardevice;
circle(12*N,12*M,226); circle(12*N,12*M,224);
For i:=1 to N do For j:=1 to M do Draw1; end;
If k mod 200=0 then begin y:=50*k/200+20; x:=20; end;
If k>100 then Draw2;
until (KeyPressed)or(k>1.96E+3);
Repeat until KeyPressed;
CloseGraph;
END.

```

Программа ПР-8

```

program vihrevoe_techenie_v_kaverne;
{$N+}uses crt, graph;
const n=80; m=60; dt=0.02; h=1; nu=1.1;
var i,j,k,DV,MV: integer; vx,vy: single;
    psi,w: array[0..N+1,0..M+1] of single;
Procedure Gr_usl;
begin For i:=1 to N do For j:=1 to M do begin
psi[0,j]:=0; psi[1,j]:=0; psi[N,j]:=0; psi[N+1,j]:=0;
psi[i,0]:=0; psi[i,1]:=0; psi[i,2]:=2; psi[i,M]:=0;
psi[i,M+1]:=0;
w[1,j]:=-2*psi[2,j]; w[i,1]:=-2*psi[i,2]+1;
w[N,j]:=-2*psi[N-1,j]; w[i,M]:=-2*psi[i,M-1];
end; end;
Procedure Url1;
begin psi[i,j]:=(psi[i-1,j]+psi[i+1,j]+
psi[i,j-1]+psi[i,j+1]+w[i,j]*h*h)/4; end;
Procedure Ur2;
Begin vx:=psi[i,j+1]-psi[i,j];
vy:=psi[i,j]-psi[i+1,j];
w[i,j]:=w[i,j]-vx*(w[i+1,j]-w[i-1,j])*dt/2/h-
vy*(w[i,j+1]-w[i,j-1])*dt/2/h+nu*(w[i-1,j]-
4*w[i,j]+w[i+1,j]+w[i,j-1]+w[i,j+1])/h/h*dt; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat k:=k+1;
    Gr_usl; For i:=1 to N do For j:=1 to M do Url1;
    Gr_usl; For i:=1 to N do For j:=1 to M do Ur2;
    Gr_usl; For i:=N downto 1 do
        For j:=M downto 1 do Url1;
    Gr_usl; For i:=N downto 1 do

```

```

        For j:=M downto 1 do Ur2;
    If k mod 200=0 then begin cleardevice; k:=0;
        For i:=0 to N+1 do For j:=0 to M+1 do begin
            setcolor(abs(round(psi[i,j]/4)+1));
            circle(i*6+10,j*6,1); circle(i*6+10,j*6,2);
            circle(i*6+10,j*6,3); end; end;
until KeyPressed; CloseGraph;
END.

```

Программа ПР-9

```

program Vihri_v_kaverne_pole_scorostei;
uses crt, graph;
const n=52; m=42; dt=0.05; h=1; nu=0.3;
var i,j,k,DV,MV: integer; v2: single;
    psi,w,vx,vy: array[0..N+1,0..M+1] of single;
Procedure Gr_usl;
begin For i:=1 to N do For j:=1 to M do begin
psi[0,j]:=0; psi[1,j]:=0; psi[N,j]:=0; psi[N+1,j]:=0;
psi[i,0]:=0; psi[i,1]:=0; psi[i,2]:=2; psi[i,M]:=0;
psi[i,M+1]:=0; w[1,j]:=-2*(psi[2,j]-psi[1,j]);
w[i,1]:=-2*psi[i,2]+1; w[N,j]:=-2*psi[N-1,j];
w[i,M]:=-2*psi[i,M-1]; end; end;
Procedure Ur1;
begin
psi[i,j]:=(psi[i-1,j]+psi[i+1,j]+
psi[i,j-1]+psi[i,j+1]+w[i,j]*h*h)/4; end;
Procedure Ur2;
Begin vx[i,j]:=-psi[i,j+1]+psi[i,j];
vy[i,j]:=psi[i+1,j]-psi[i,j];
w[i,j]:=w[i,j]-vx[i,j]*(w[i+1,j]-w[i-1,j])*dt/2/h-
vy[i,j]*(w[i,j+1]-w[i,j-1])*dt/2/h+nu*(w[i-1,j]-
4*w[i,j]+w[i+1,j]+w[i,j-1]+w[i,j+1])/h/h*dt; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat k:=k+1;
    Gr_usl; For i:=1 to N do For j:=1 to M do Ur1;
    Gr_usl; For i:=2 to N-1 do For j:=2 to M-1 do Ur2;
    Gr_usl; For i:=N downto 1 do
        For j:=M downto 1 do Ur1;
    Gr_usl; For i:=N-1 downto 2 do
        For j:=M-1 downto 2 do Ur2;

```

```

If k mod 500=0 then begin cleardevice;
  For i:=1 to N-1 do For j:=1 to M-1 do begin
    If (i mod 2=0)and(j mod 2=0) then
      begin circle(i*10,j*10,2); line(i*10,j*10,
        i*10+round(vx[i,j]*30),j*10+round(vy[i,j]*30));
      end; end; end;
until KeyPressed; CloseGraph;
END.

```

Программа ПР-10

```

program Vihri_za_vistupami;
{$N+}uses crt, graph;
const n=140; m=50; dt=0.02; h=1; nu=2.0;
      a=4; b=10; c=5; a1=40; b1=45; c1=10;
var i,j,k,DV,MV,EC: integer;
      psi,w: array[-1..N+2,-1..M+2]of single;
Procedure Gr_usl;
begin For j:=0 to M+1 do For i:=0 to N+1 do begin
psi[0,j]:=psi[1,j]; psi[N+1,j]:=psi[N,j];
psi[i,0]:=0; psi[i,1]:=20; psi[i,M]:=0;
If (i>a)and(i<b)and(j>c) then psi[i,j]:=0;
If j>c then begin w[a,j]:=2*psi[a-1,j]/h/h;
w[b,j]:=2*psi[b+1,j]/h/h; end;
If (i>a)and(i<b) then w[i,c]:=2*psi[c,j-1]/h/h;
If (i>a1)and(i<b1)and(j>c1) then psi[i,j]:=0;
If j>c1 then begin w[a1,j]:=2*psi[a1-1,j]/h/h;
w[b1,j]:=2*psi[b1+1,j]/h/h; end;
If (i>a1)and(i<b1) then w[i,c1]:=2*psi[c1,j-1]/h/h;
w[0,j]:=0; w[N+1,j]:=0; w[i,M+1]:=0; w[i,0]:=0; end; end;
Procedure Ur1;
begin psi[i,j]:=(psi[i-1,j]+psi[i+1,j]+
psi[i,j-1]+psi[i,j+1]-w[i,j]*h*h)/4; end;
Procedure Ur2;
begin w[i,j]:=w[i,j]+(psi[i,j+1]-psi[i,j-1])*(w[i+1,j]
-w[i-1,j])*dt/4/h+(-psi[i+1,j]+psi[i-1,j])*(w[i,j+1]-
w[i,j-1])*dt/4/h+nu*(w[i-1,j]-4*w[i,j]+w[i+1,j]+
w[i,j-1]+w[i,j+1])/h/h*dt; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat Gr_usl; For i:=1 to N do For j:=1 to M do Ur1;
      Gr_usl; For i:=1 to N do For j:=1 to M do Ur2;

```

```

Gr_usl; For i:=N downto 1 do For j:=M downto 1 do Ur1;
Gr_usl; For i:=1 to N do For j:=1 to M do Ur2;
inc(k); If k mod 50=0 then begin cleardevice; k:=0;
For i:=1 to N do For j:=1 to M do begin
setcolor(round(abs(psi[i,j]*2))+4);
If psi[i,j]=0 then setcolor(15);
rectangle(i*6+10,j*6,i*6+15,j*6+5);
rectangle(i*6+12,j*6+2,i*6+13,j*6+3); end; end;
until KeyPressed; CloseGraph;
END.

```

Программа ПР-11

```

{$N+}program Konvekciya;
uses crt, graph;
const N=50; M=60; dt=0.0025; h=1; nu=13.1;
var i,j,k,DV,MV:integer; q,q2: single;
psi,w,T:array[0..N+1,0..M+1] of single;
Procedure Gr_usl;
begin For i:=0 to N+1 do For j:=0 to M+1 do begin
T[i,0]:=0; T[i,M+1]:=0; T[0,j]:=0; T[N+1,j]:=0;
T[i,1]:=0; T[i,M]:=0; T[1,j]:=0; T[N,j]:=0;
psi[0,j]:=0; psi[N+1,j]:=0;
psi[i,0]:=0; psi[i,M+1]:=0; w[i,M]:=-2*psi[i,M]/h/h;
w[1,j]:=-2*psi[1,j]/h/h; w[N,j]:=-2*psi[N,j]/h/h;
w[i,1]:=-2*psi[i,1]/h/h; w[0,j]:=0; w[N+1,j]:=0;
w[i,M+1]:=0; w[i,0]:=0; end; end;
Procedure Ur1;
begin psi[i,j]:=(psi[i-1,j]+psi[i+1,j]+
psi[i,j-1]+psi[i,j+1]+w[i,j]*h*h)/4; end;
Procedure Ur2;
begin w[i,j]:=w[i,j]-(psi[i,j+1]-psi[i,j-1])*(w[i+1,j]-
w[i-1,j])*dt/4/h+(psi[i+1,j]-psi[i-1,j])*(w[i,j+1]-
w[i,j-1])*dt/4/h+nu*(w[i-1,j]-4*w[i,j]+w[i+1,j]+
w[i,j-1]+w[i,j+1])/h/h*dt-0.0002*(T[i+1,j]-T[i-1,j]); end;
Procedure Ur3;
begin q:=0;
{If t[i,j]>0.2 then q:=-t[i,j]*t[i,j]*t[i,j]/50000;}
If (abs(i-30)<3)and(abs(j-50)<3) then q:=350;
T[i,j]:=T[i,j]+1.3*(T[i-1,j]-4*T[i,j]+T[i+1,j]+
T[i,j-1]+T[i,j+1])/h/h*dt-

```

```
(psi[i,j+1]-psi[i,j-1])*(T[i+1,j]-T[i-1,j])*dt/4/h+
(psi[i+1,j]-psi[i-1,j])*(T[i,j+1]-T[i,j-1])*dt/4/h+q*dt;
end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat Gr_us1; For i:=1 to N do For j:=1 to M do Ur1;
Gr_us1; For i:=N downto 1 do For j:=M downto 1 do Ur1;
Gr_us1; For i:=1 to N do For j:=1 to M do Ur2;
Gr_us1; For i:=N downto 1 do For j:=M downto 1 do Ur2;
Gr_us1; For i:=1 to N do For j:=1 to M do Ur3;
Gr_us1; For i:=N downto 1 do For j:=M downto 1 do Ur3;
inc(k); If k mod 200=0 then begin cleardevice; k:=0;
For i:=1 to N do For j:=1 to M do begin
    setcolor(abs(round(psi[i,j]/20))mod 12+2);
    rectangle(i*6,j*6,i*6+4,j*6+4);
    rectangle(i*6+1,j*6+1,i*6+2,j*6+2);
    setcolor(abs(round(T[i,j]/60))mod 14+2);
    rectangle(i*6+300,j*6,i*6+305,j*6+5);
    rectangle(i*6+301,j*6+1,i*6+304,j*6+4); end; end;
until KeyPressed; CloseGraph;
END.
```

Список литературы

1. Берковский Б. М., Полевиков В. К. Вычислительный эксперимент в конвекции. Мн.: Университетское, 1988. 167 с.
2. Гебхарт Б., Джалурия Й., Махаджан Р. и др. Свободноконвективные течения, тепло- и массообмен. В 2 кн. Кн. 1. М.: Мир, 1991. 678 с.
3. Госмен А. Д., Пан В. М., Ранчел А. К., Сполдинг Д. Б. и др. Численные методы течения вязкой жидкости. М.: Мир, 1972. 324 с.
4. Елизарова Т. Г. Математические модели и численные методы в динамике жидкости и газа. Подходы, основанные на системах квазигидродинамических уравнений. Лекции. М.: Физический факультет МГУ, 2005. 224с.
5. Рашиков В. И., Рошаль А. С. Численные методы решения физических задач: учеб. пособие. СПб.: Издательство "Лань", 2005. 208 с.

6. Калиткин Н. Н. Численные методы. БХВ-Петербург, 2011. 592 с.
7. Кунин С. Вычислительная физика. М.: Мир, 1992. 518 с.
8. Лаврентьев М. А., Шабат Б. В. Проблемы гидродинамики и их математические модели. М.: Наука, 1973. 416 с.
9. Майер Р. В. Задачи, алгоритмы, программы: электронное учеб. пособие. Глазов: Глазов. гос. пед. ин-т, 2012. URL: <http://maier-rv.glazov.net>
10. Майер Р. В. Компьютерное моделирование физических явлений. Глазов: Глазов. гос. пед. ин-т, 2009. 112 с.
11. Моделирование гидродинамических течений: учеб.-метод. пособие. М.: 2006. 72 с. URL: <http://window.edu.ru/resource/497/70497>
12. Поттер Д. Вычислительные методы в физике. М.: Мир, 1975. 392 с.
13. Роуч ТТ. Вычислительная гидродинамика. М.: Мир, 1980. 616 с.
14. Саранин В. А., Иванов Ю. В. Равновесие жидкостей и его устойчивость. М.-Ижевск: НИЦ "Регулярная и хаотическая динамика", 2009. 172 с.
15. Тарунин Е. Л. Вычислительный эксперимент в задачах свободной конвекции. Иркутск: Изд-во Иркутск. ун-та, 1990. 228 с.
16. Темам Р. Уравнение Навье-Стокса. Теория и численный анализ. М.: Мир, 1981. 408 с.
17. Тихонов А. Н., Самарский А. А. Уравнения математической физики. М.: Наука, 1966. 724 с.
18. Федоренко Р. П. Введение в вычислительную физику: учеб. пособие для вузов. М.: Изд-во Моск. физ.-техн. ин-та, 1994. 528 с.
19. Филатов Е. Ю., Ясинский Ф. Н. Математическое моделирование течений жидкостей и газов: учеб. пособие / Государственное образовательное учреждение высшего профессионального образования "Ивановский государственный энергетический университет имени В. И. Ленина". Иваново, 2007. 84 с.
20. Woolfson M. M., Pert G. J. An Introduction to Computer Simulation. Oxford University Press, 1999. 311 p.