

## Глава 12

### РАСЧЕТ ЭЛЕКТРИЧЕСКОГО И МАГНИТНОГО ПОЛЕЙ. ДВИЖЕНИЕ ЗАРЯЖЕННЫХ ЧАСТИЦ

Основная задача электродинамики состоит в расчете электрического и магнитного полей и их взаимодействия с окружающими телами при заданном распределении зарядов и электрических токов. Ее решение предполагает нахождение потенциала, напряженности и индукции электромагнитного поля, определение силы взаимодействия заряженных тел, расчет движения заряженных частиц [1; 2; 4; 6-8; 9; 11; 12].

#### 12.1. Расчет электрического поля

Многие задачи электростатики требуют нахождения напряженности и потенциала различных точек электростатического поля, создаваемого системой зарядов в окружающем пространстве. Пусть вдали от проводников расположены несколько точечных зарядов  $q_s$ ,  $s = 1, 2, \dots, N$ . Чтобы рассчитать распределение потенциала, используется программа, которая перебирает все точки некоторой плоскости, задаваемые целочисленными параметрами  $i$  и  $j$  (глава 3). При этом вычисляются расстояния от данной точки до каждого заряда и алгебраическая сумма потенциалов:

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

Чтобы визуализировать распределения потенциала вдоль плоскости экрана, на нем ставятся точки, цвет которых зависит от величины  $\varphi_{i,j}$  [5; 6].

Для расчета электростатического поля одного или нескольких электродов (то есть протяженных тел), их заменяют совокупностью близко

расположенных точечных зарядов  $q_s$ . Величины  $q_s$  и их расположение подбирают так, чтобы получить заданную поверхностную плотность заряда. Воспользуемся этим методом, чтобы промоделировать поле двух разноименно заряженных отклоняющих пластин **электронно-лучевой трубки**. Используется программа ТР-1; результаты вычислений представлены на рис. 12.1.1. Границы разноцветных областей соответствуют эквипотенциальным поверхностям. На рис. 12.1.2 представлен результат расчета распределения потенциала на плоскости, содержащей положительно заряженный стержень и отрицательно заряженный шарик.

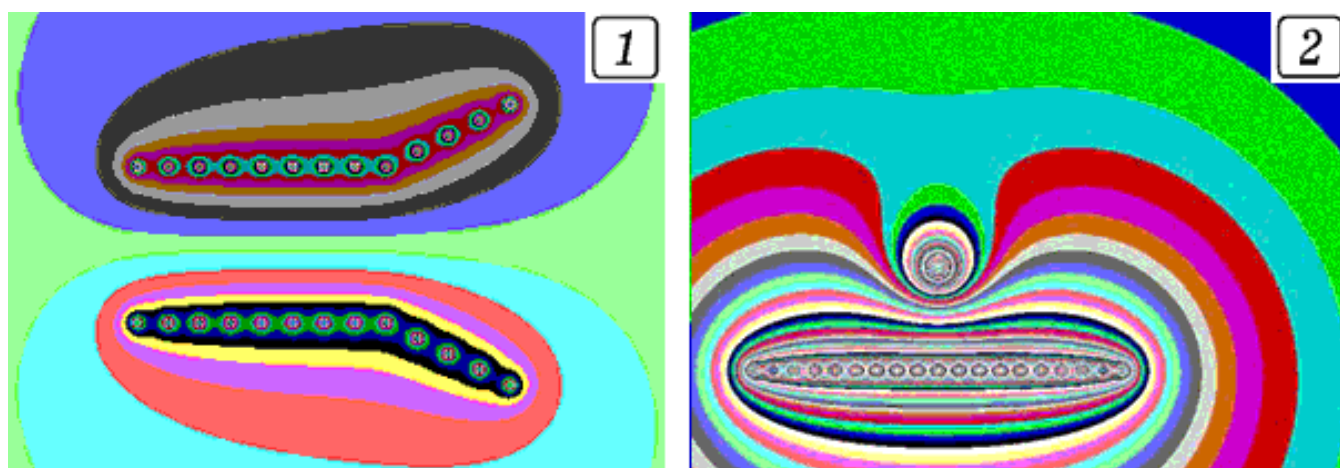


Рис. 12.1. Электроды моделируются совокупностью точечных зарядов.

Формулы для силы Кулона и силы гравитационного притяжения имеют похожий вид, поэтому электростатическое поле рассчитывается подобно гравитационному полю. При этом следует помнить об одном отличии: электрические заряды бывают двух знаков, причем одноименные заряды отталкиваются, а массивные тела всегда притягиваются. В главе 2 (параграф 2.5) показано, как путем интегрирования по объему рассчитать силу гравитационного притяжения шарообразного тела и материальной точки, а также неоднородного диска и стержня. Соответствующие задачи по электростатике решаются аналогично.

Для определения напряженности электростатического поля в различных узлах двумерной сетки используется метод, описанный в главе 3.

Построение силовых линий осуществляется **методом градиентного спуска**, который рассмотрен ниже.

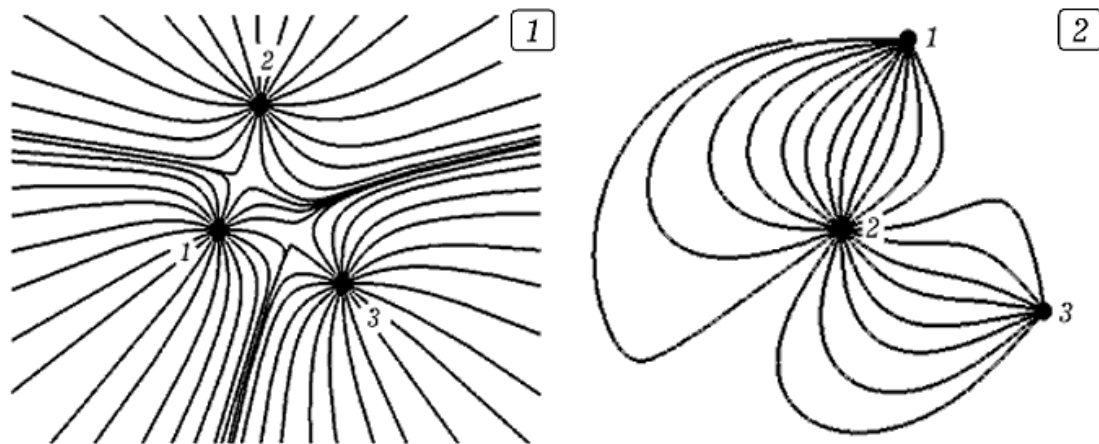


Рис. 12.2. Расчет силовых линий электростатического поля

Пусть необходимо построить силовую линию напряженности поля  $\vec{E}$ , созданного тремя точечными зарядами, которая проходит через точку  $A_0(x_0, y_0)$  и имеет низкий потенциал. Так как  $\vec{E} = -grad\varphi$ , то вектор напряженности поля направлен в сторону наиболее быстрого убывания потенциала. В процедуре *Raschet* (программа ПР-2) сначала вычисляется потенциал  $\varphi$  в точке  $A_0$ . В цикле по  $i$  перебираются точки  $B_i$ , которые лежат на окружности с центром  $A_0$  и радиусом  $d$  и имеют координаты  $(x_0 + d \cos(i\Delta\alpha), y_0 + d \sin(i\Delta\alpha))$ , где  $\Delta\alpha$  - шаг изменения угла. При этом рассчитываются расстояния от  $B_i$  до точечных зарядов, вычисляются потенциалы  $\varphi_{1i}$  точек  $B_i$  и разности  $\varphi_{1i} - \varphi$ . Это позволяет определить угол  $\alpha_m$ , соответствующий наиболее быстрому возрастанию потенциала. На экране ставится точка  $A_1$  с координатами  $(x_0 + d \cos \alpha_m, y_0 + d \sin \alpha_m)$ , после чего все начинается снова: в цикле перебираются все точки  $B_i$ , равноудаленные от  $A_1$  на расстояние  $d$ , и т. д. Программа ПР-2 совершает заданное число шагов (около 500), после чего построение данной силовой линии прекращается. Основная часть программы содержит цикл по  $j$ , в

нем задаются координаты точки  $A_0$ , расположенной вблизи отрицательного заряда, от которой начинает строиться новая силовая линия. Результаты расчетов поля трех точечных зарядов показаны на рис. 12.2.

Силовые линии можно построить иначе: из начальной точки  $A_0$  запустить легкую частицу-маркер, которая будет перемещаться в направлении напряженности  $\vec{E}$ . Этот способ рассмотрен в параграфе 12.5.

## 12.2. Перемещение заряженных частиц в электрических полях

Многие задачи электродинамики требуют расчета движения зарядов в электромагнитном поле. В главе 7 описаны методы моделирования перемещения частиц в электрическом и магнитном полях (параграфы 7.4 и 7.5), а также движения двух частиц, взаимодействующих друг с другом. Рассмотрим расчет движения заряженных частиц в поле протяженных тел (электродов), образующих **электронную линзу**, позволяющую фокусировать электронные пучки и получать электронно-оптические изображения.

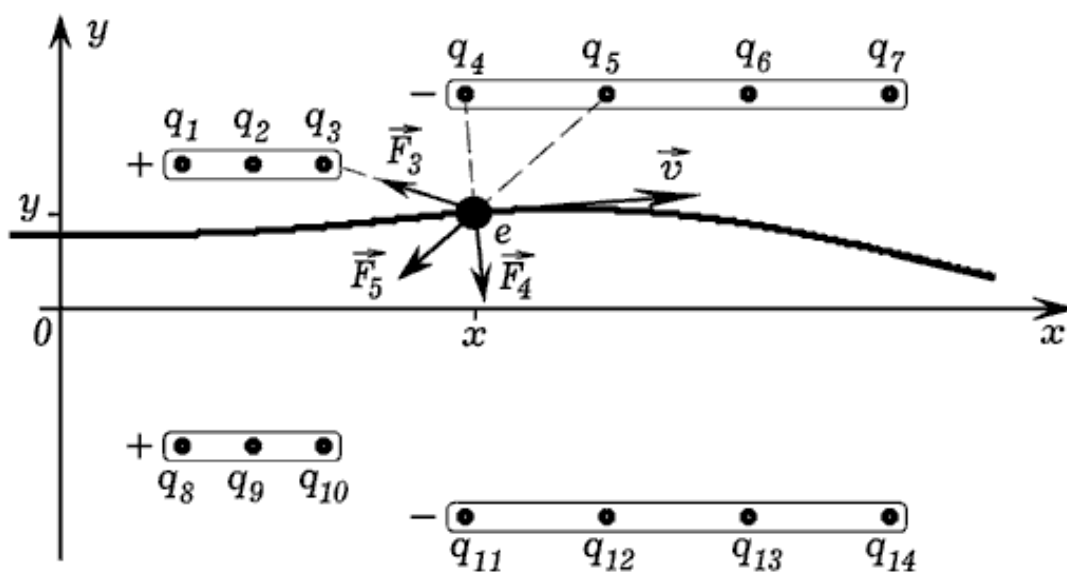


Рис. 12.3. К расчету траектории движения электрона в линзе

Приближенно рассчитаем движение электрона в поле двух коаксиальных цилиндрических электродов, имеющих разные заряды. Ограничимся рассмотрением двумерного движения в поле, создаваемом системой точечных зарядов (рис. 12.3). Один цилиндрический электрод моделируется зарядами  $q_1, q_2, q_3, q_8, q_9, q_{10}$ ; другой - зарядами  $q_4, q_5, q_6, q_7, q_{11}, q_{12}, q_{13}, q_{14}$ . Пусть левый цилиндр малого радиуса имеет положительный потенциал, а правый - отрицательный. При движении электрона необходимо учесть все силы, действующие на него со стороны зарядов  $q_1, q_2, \dots, q_{14}$ , найти проекции результирующей силы на оси координат, рассчитать проекции ускорения, скорости и координаты частиц в момент  $t + 1$ . Для этого используются формулы:

$$L_i^t = \sqrt{(x^t - a_i)^2 + (y^t - b_i)^2}, \quad i = 1, 2, \dots, N,$$

$$F_x^{t+1} = -\sum_{i=1}^N k \frac{eq_i}{(L_i^t)^2} \cdot \frac{(x^t - a_i)}{L_i^t}, \quad F_y^{t+1} = -\sum_{i=1}^N k \frac{eq_i}{(L_i^t)^2} \cdot \frac{(y^t - b_i)}{L_i^t},$$

$$v_x^{t+1} = v_x^t + F_x^{t+1} \Delta\tau / m, \quad x^{t+1} = x^t + v_x^{t+1} \Delta\tau,$$

$$v_y^{t+1} = v_y^t + F_y^{t+1} \Delta\tau / m, \quad y^{t+1} = y^t + v_y^{t+1} \Delta\tau.$$

Здесь  $a_i, b_i$  - координаты зарядов  $q_i$  ( $i = 1, 2, \dots, N$ ),  $L_i$  - расстояние от электрона до заряда  $q_i$ ,  $e$  - модуль заряда электрона,  $F_x, F_y$  - проекции результирующей силы, действующей на электрон со стороны всех  $N$  зарядов на оси координат. Используется программа ПР-3, результаты моделирования фокусировки электронных пучков представлены на рис. 12.4. Аналогичным методом можно рассчитать движение электронов в поле отклоняющих пластин, на которое подано изменяющееся напряжение [5].

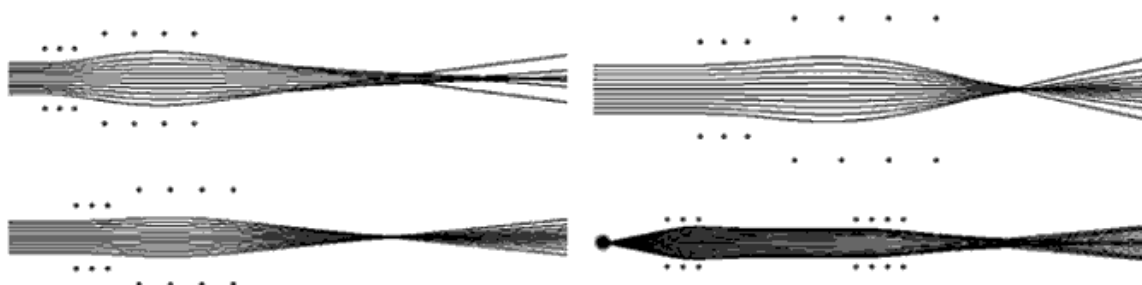


Рис. 12.4. Фокусировка электронного пучка электростатической линзой

Другой тип задач состоит в нахождении распределения заряженных частиц, которое соответствует минимуму потенциальной энергии. Допустим, к прямоугольному проводящему заряженному телу приблизили заряженный шар. Необходимо промоделировать распределение свободных заряженных частиц, если тело и шар имеют: 1) заряды одного знака; 2) заряды противоположных знаков [5].

При внесении проводника во внешнее электростатическое поле заряда  $Q$  свободные заряды в проводнике перераспределяются так, что суммарная потенциальная энергия  $U$  их взаимодействия друг с другом и внешним полем становится минимальной. Это явление электростатической индукции. Потенциальная энергия  $U_i$  взаимодействия заряда  $q_i$  с другими зарядами и общая энергия системы  $U$  равны:

$$U_i = \sum_{j=1, j \neq i}^N k \frac{q_i q_j}{r_{ij}}, \quad U = \frac{1}{2} \sum_{i=1}^N U_i + \sum_{i=1}^N k \frac{Q q_i}{r_{i0}}.$$

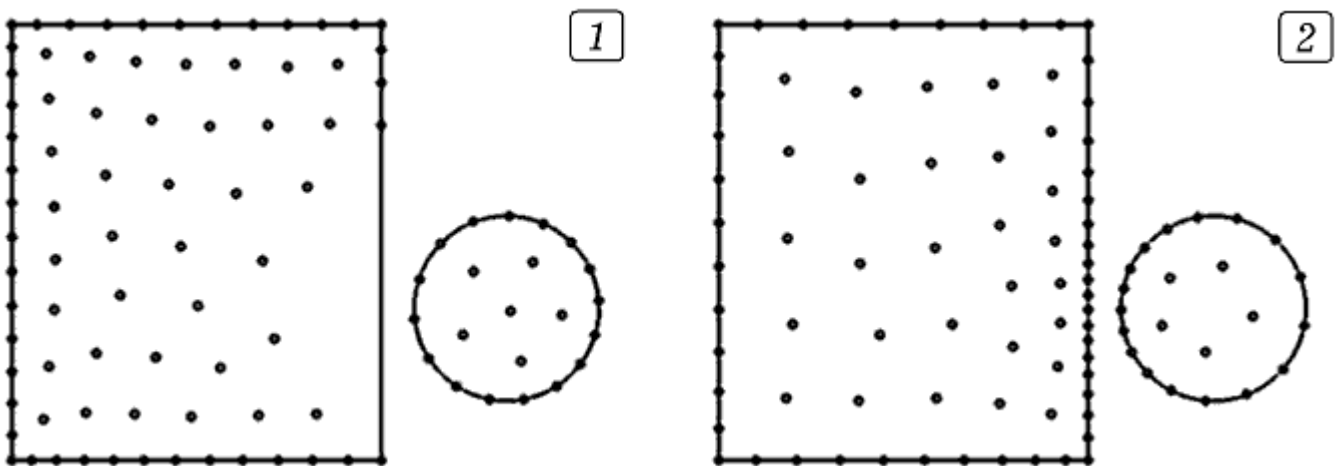


Рис. 12.5. Распределение свободных зарядов в заряженных проводниках

Все частицы разделяются на две группы: 1) принадлежащие телу 1 и имеющие заряд  $q_1$ ; 2) принадлежащие телу 2 и имеющие заряд  $q_2$ . Используется программа ПР-4, в которой вычисляется суммарная потенциальная энергия взаимодействия  $U$  всех зарядов и ищется такое их распределение, при котором  $U$  минимально. Для этого каждый заряд сдви-

гается на небольшую случайную величину и вычисляется новое значение  $U$ . Если оно уменьшилось, то изменение принимается, а иначе - отвергается. На рис. 12.5 представлены результаты моделирования, если заряды тел одноименные (рис. 12.5.1) и разноименные (рис. 12.5.2).

### 12.3. Решение уравнения Пуассона для однородной среды

Интерес представляет собой проблема нахождения распределения потенциала в случае, когда электростатическое поле в неоднородной среде создается протяженными заряженными телами, рядом с которыми находятся проводники. Эта стационарная задача подобна задаче о расчете формы упругой пленки, натянутой на каркас и деформируемой заданным образом; она требует решения уравнения Пуассона.

Пусть в ограниченной области заданы распределение плотности заряда  $\rho(x, y, z)$ , диэлектрическая проницаемость среды  $\varepsilon(x, y, z)$  и потенциал  $\varphi(x, y, z)_S$  вдоль границы  $S$  рассматриваемой области (включая поверхности проводников). Необходимо рассчитать распределение потенциала  $\varphi(x, y, z)$  внутри рассматриваемой области [1; 4; 8].

Нахождение потенциалов различных точек поля требует решения уравнения Пуассона. Запишем его для однородной двумерной среды:

$$\Delta\varphi = \frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} = -\frac{\rho(x, y)}{\varepsilon\varepsilon_0}.$$

При отсутствии зарядов получаем уравнение Лапласа:  $\Delta\varphi = 0$ .

Расчет электростатического поля состоит в определении функции  $\varphi(x, y)$ , превращающей уравнение Пуассона в истинное высказывание во всех точках вне проводников и равной заданным значениям  $\varphi(x, y)_S$  на поверхностях проводников и границе области. Это **задача Дирихле**: необходимо решить дифференциальное уравнение в частных производных в



замкнутой области  $\Omega$  при известном распределении искомой функции  $\varphi(x, y)$  на границе области.

Для поиска  $\varphi(x, y)$  используют **метод сеток**: 1) от непрерывной области  $\Omega$  переходят к сеточной области  $\Omega_{\Delta x, \Delta y}$ ; 2) для узлов сетки  $\Omega_{\Delta x, \Delta y}$  составляют конечно-разностные уравнения; 3) исходя из начальных и граничных условий, задают начальные значения искомой функции  $\varphi = \varphi(x, y)$  и ее значения в граничных узлах сетки  $\Omega_{\Delta x, \Delta y}$ ; 4) решают получившуюся систему конечно-разностных уравнений, определяя значения функции  $\varphi(x, y)$  в узлах сетки. Переход от решаемого ДУЧП к конечно-разностному уравнению осуществляется путем замены производных на их конечно-разностные аппроксимации (здесь  $\Delta x = \Delta y = h$ ):

$$\frac{\varphi_{i-1,j} - 2\varphi_{i,j} + \varphi_{i+1,j}}{h^2} + \frac{\varphi_{i,j-1} - 2\varphi_{i,j} + \varphi_{i,j+1}}{h^2} = -\frac{\rho_{i,j}}{\varepsilon\varepsilon_0},$$

$$\varphi_{i,j} = (\varphi_{i+1,j} + \varphi_{i-1,j} + \varphi_{i,j+1} + \varphi_{i,j-1} + \frac{\rho_{i,j}h^2}{\varepsilon\varepsilon_0})/4.$$

Допустим, имеется проводящая пластина, рядом с которой расположен заостренный проводник и точечный заряд. Потенциалы пластины и острия имеют разные знаки. Для расчета распределения потенциала в случае, когда среда однородная, используется программа ТР-5. В результате большого числа итераций получается распределение потенциала, изображенное на рис. 12.6.1. Между острием и пластиной линии равного потенциала расположены очень близко, напряженность велика.

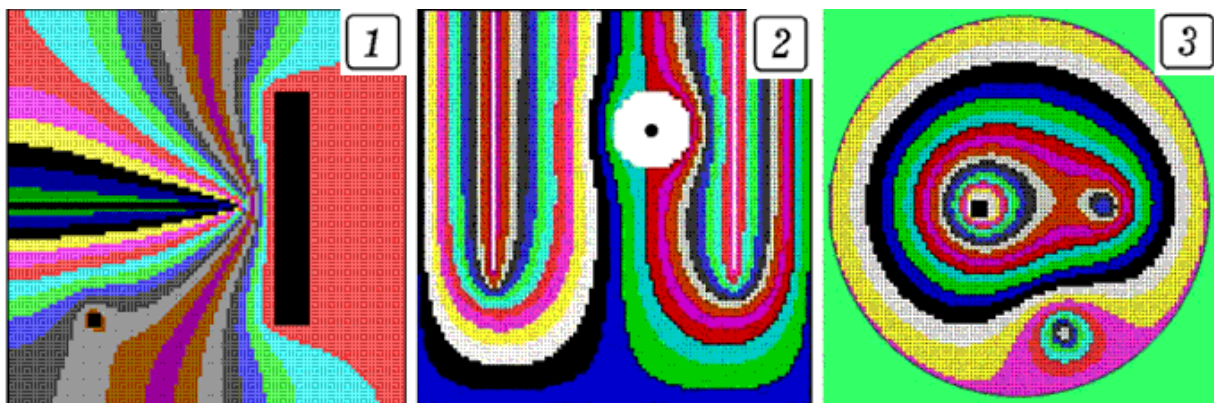


Рис. 12.6. Результаты расчета распределения потенциала



Программа ПР-5 позволяет решить разные задачи. Например, на рис. 12.6.2 представлено распределение потенциала электрического поля двух параллельных пластин с  $\varphi = 100$  В, между которыми находится металлический шарик с потенциалом  $\varphi_1 = 20$  В. На рис. 12.6.3 приведены результаты моделирования электрического поля, созданного бесконечной металлической трубой, внутри которой параллельно ее оси расположены три линейных проводника, имеющих различные потенциалы [5; 6].

## 12.4. Решение уравнения Пуассона для неоднородной среды

Допустим, имеется неоднородная прямоугольная пластина из диэлектрика с диэлектрической проницаемостью  $\varepsilon = \varepsilon(x, y)$ . Задана плотность распределения заряда  $\rho(x, y)$  и граничные условия  $\varphi(x, y)_S$ . Распределение потенциала должно удовлетворять уравнению [2; 9]:

$$\frac{\partial}{\partial x} \left( \varepsilon(x, y) \frac{\partial \varphi}{\partial x} \right) + \frac{\partial}{\partial y} \left( \varepsilon(x, y) \frac{\partial \varphi}{\partial y} \right) = - \frac{\rho(x, y)}{\varepsilon_0},$$

$$\frac{\partial \varepsilon}{\partial x} \frac{\partial \varphi}{\partial x} + \varepsilon \frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial \varepsilon}{\partial y} \frac{\partial \varphi}{\partial y} + \varepsilon \frac{\partial^2 \varphi}{\partial y^2} = - \frac{\rho}{\varepsilon_0}.$$

В конечных разностях получаем:

$$\frac{\varepsilon_{i+1,j} - \varepsilon_{i-1,j}}{2h} \frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2h} + \varepsilon_{i,j} \frac{\varphi_{i-1,j} - 2\varphi_{i,j} + \varphi_{i+1,j}}{h^2} +$$

$$+ \frac{\varepsilon_{i,j+1} - \varepsilon_{i,j-1}}{2h} \frac{\varphi_{i,j+1} - \varphi_{i,j-1}}{2h} + \varepsilon_{i,j} \frac{\varphi_{i,j-1} - 2\varphi_{i,j} + \varphi_{i,j+1}}{h^2} = - \frac{\rho_{i,j}}{\varepsilon_0},$$

$$\varphi_{i,j} = (\varphi_{i+1,j} + \varphi_{i-1,j} + \varphi_{i,j+1} + \varphi_{i,j-1} + \rho_{i,j} h^2 / \varepsilon_0 \varepsilon_{i,j}) / 4 + [(\varepsilon_{i+1,j} - \varepsilon_{i-1,j}) \cdot (\varphi_{i+1,j} - \varphi_{i-1,j}) + (\varepsilon_{i,j+1} - \varepsilon_{i,j-1}) \cdot (\varphi_{i,j+1} - \varphi_{i,j-1})] / (16 \varepsilon_{i,j}).$$

Для решения рассматриваемой стационарной задачи используется релаксационный метод последовательных приближений: во всех внут-

ренних узлах сетки задаются произвольные исходные значения искомой функции  $\varphi = \varphi(x_i, y_j)$ ; значения функции во внешних узлах должны соответствовать граничным условиям. Осуществляется первая итерация, в ходе которой перебираются все внутренние узлы сетки и, исходя из начальных значений, определяют новые уточненные значения функции  $\varphi = \varphi(x_i, y_j)$ . Затем осуществляются второе, третье, ... приближения, причем результаты  $i$ -й итерации используются в качестве исходных для  $i+1$ -й итерации. В результате получающиеся значения  $\varphi = \varphi(x_i, y_j)$  приближаются к истинному распределению потенциала  $\varphi = \varphi(x, y)$ .

Для реализации этого алгоритма требуется зарезервировать в памяти ЭВМ три двумерных массива размерностью  $N \times M$ , элементами которых являются числа типа `real` или `single`. Один массив для  $\varepsilon_{i,j}$ , два других - для  $\varphi_{i,j}^t$  и  $\varphi_{i,j}^{t+1}$ . Это не позволяет в среде Free Pascal создать сетку с большим числом узлов. Увеличить число узлов можно, задав диэлектрическую проницаемость целыми числами. Вместо двух массивов для  $\varphi_{i,j}^t$  и  $\varphi_{i,j}^{t+1}$  на различных временных слоях следует создать один, но в этом случае, чтобы избежать накапливания ошибок, придется перебирать элементы в четырех различных направлениях [5; 6].

Алгоритм А-1 решения уравнения Пуассона состоит в последовательном переборе узлов сетки слева направо и справа налево, сверху вниз и снизу вверх, в ходе которого вычисляются значения потенциала  $\varphi_{i,j}$ . Перед каждым проходом следует учесть граничные условия задачи, то есть приравнять потенциалы на границе двумерной области к некоторой величине или функции координат  $\varphi(x, y)_S$ . Если внутри области имеются точки с заданным потенциалом  $\varphi_0$ , то необходимо каждый раз приравнивать их потенциал к  $\varphi_0$ .

## Алгоритм А-1

```

ПРОЦЕДУРА Sreda; {--
  ДЛЯ i:=1 ДО N ДЕЛАТЬ {-
    ДЛЯ j:=1 ДО M ДЕЛАТЬ {e[i,j]:=1+0.015*j*i; } -} --}
ПРОЦЕДУРА Gran; {--
  ДЛЯ i:=1 ДО N ДЕЛАТЬ { fi[i,1]:=-350; fi[i,M]:=350;
  fi[i,2]:=-350; fi[i,M-1]:=350; fi[2,i]:=-350+10*i;
  fi[N-1,i]:=-350+10*i; fi[1,i]:=-350+10*i;
  fi[N,i]:=-350+10*i;}
  ДЛЯ j:=15 ДО 25 ДЕЛАТЬ { fi[55,j]:=0; } --}
ПРОЦЕДУРА Raschet; {-- q:=0;
  ЕСЛИ ((i>20) И (i<30) И (j>30) И (j<35)) ТО q:=500;
  ЕСЛИ ((i>40) И (i<55) И (j>40) И (j<55)) ТО q:=-300;
  fi[i,j]:=(fi[i+1,j]+fi[i-1,j]+fi[i,j+1]+fi[i,j-1]+q/
  e[i,j])/4+((e[i-1,j]-e[i+1,j])*(fi[i+1,j]-fi[i-1,j])+(e[i,
  j-1]-e[i,j+1])*(fi[i,j+1]-fi[i,j-1]))/(16*e[i,j]); --}
ПРОЦЕДУРА Draw; {--
  ЗАДАТЬ ЦВЕТ round(fi[i,j]/10) mod 15;
  ПОСТРОИТЬ КВАДРАТ (i*4+50,j*4,i*4+54,j*4+4); --}
НАЧАЛО ПРОГРАММЫ
ВЫЗОВ ПРОЦЕДУРЫ Sreda;
ПОВТОРЯТЬ ДО НАЖАТИЯ НА ПРОБЕЛ {-- k:=k+1;
  ВЫЗОВ ПРОЦЕДУРЫ Gran;
  ДЛЯ i:=2 ДО N-1 ДЕЛАТЬ {-
    ДЛЯ j:=2 ДО M-1 ДЕЛАТЬ { ВЫЗОВ ПРОЦЕДУРЫ Raschet; } -}
  ВЫЗОВ ПРОЦЕДУРЫ Gran;
  ДЛЯ j:=M-1 ДО 2 ДЕЛАТЬ {-
    ДЛЯ i:=N-1 ДО 2 ДЕЛАТЬ { ВЫЗОВ ПРОЦЕДУРЫ Raschet; } -}
  ЕСЛИ k mod 10=0 ТО { ДЛЯ i:=2 ДО N-1 ДЕЛАТЬ {-
    ДЛЯ j:=2 ДО M-1 ДЕЛАТЬ { ВЫЗОВ ПРОЦЕДУРЫ Draw; } -}
  --}
КОНЕЦ ПРОГРАММЫ

```

Используемая программа ТР-6 содержит: 1) процедуру Sreda, в которой задается диэлектрическая проницаемость среды  $\varepsilon_{i,j}$  в узлах сетки; 2) процедуру Gran, в которой учитывается распределение потенциала

вдоль границы области; 3) процедуру *Raschet*, в которой задается распределение заряда и вычисляется потенциал в различных узлах  $i, j$ ; 4) процедуру *Draw*, выводящую результат вычислений на экран; 5) основную часть программы, в которой осуществляется вызов перечисленных процедур в требуемом порядке.

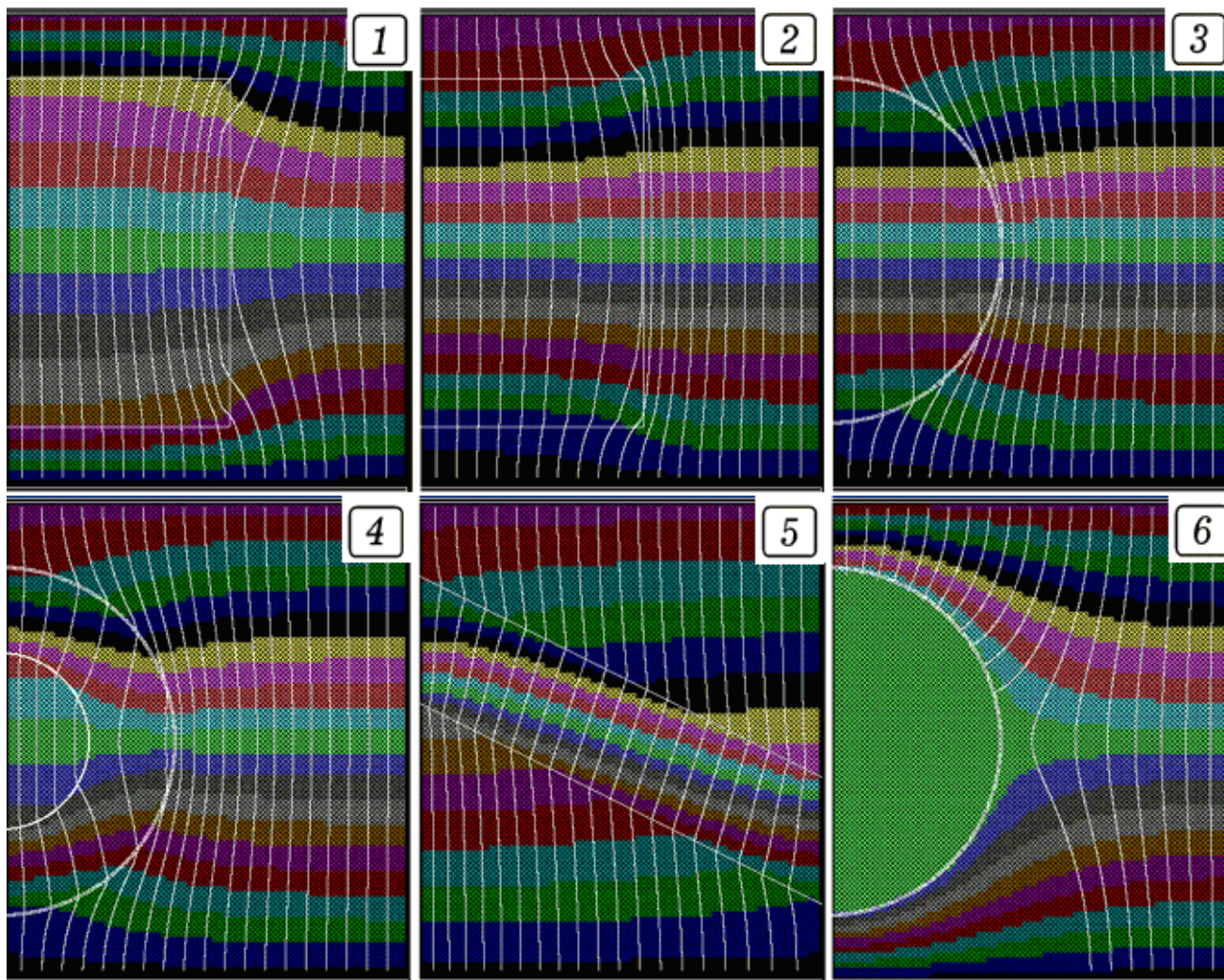


Рис. 12.6. Результаты расчета электростатического поля

С помощью программы ТР-6, задавая различные распределения зарядов, потенциалов, диэлектрической проницаемости среды  $\varepsilon(x, y)$  и граничные условия, можно решить широкий круг задач. Например, рассчитать электрическое поле, создаваемое двумя параллельными разноименно заряженными пластинами, в случаях, когда в пространство между ними внесен: 1) прямоугольный брусок с диэлектрической проницаемостью  $\varepsilon$

меньшей, чем у окружающей среды  $\varepsilon_{cp}$  (рис. 12.6.1); 2) прямоугольный брусок с  $\varepsilon > \varepsilon_{cp}$  (рис. 12.6.2); 3) цилиндр с  $\varepsilon > \varepsilon_{cp}$  (рис. 12.6.3); 4) труба с  $\varepsilon > \varepsilon_{cp}$  (рис. 12.6.4); 5) пластина с  $\varepsilon > \varepsilon_{cp}$ , расположенная под углом (рис. 12.6.5); 6) металлический цилиндр (рис. 12.6.6). Программа ПР-7 рассчитывает линии равного потенциала и силовые линии электрического поля в случае, когда в пространство между заряженными пластинами внесена труба с  $\varepsilon > \varepsilon_{cp}$ . Из рисунков видно, что на границе раздела сред с различными  $\varepsilon$  происходит преломление силовых линий; причем в средах с большим  $\varepsilon$  они располагаются реже (напряженность меньше). При внесении в электрическое поле проводящего цилиндра (потенциалы всех его точек равны), силовые линии оказываются перпендикулярными к поверхности. Метод построения силовых линий описан в параграфе 12.5. Аналогичным образом решается задача для магнитного поля.

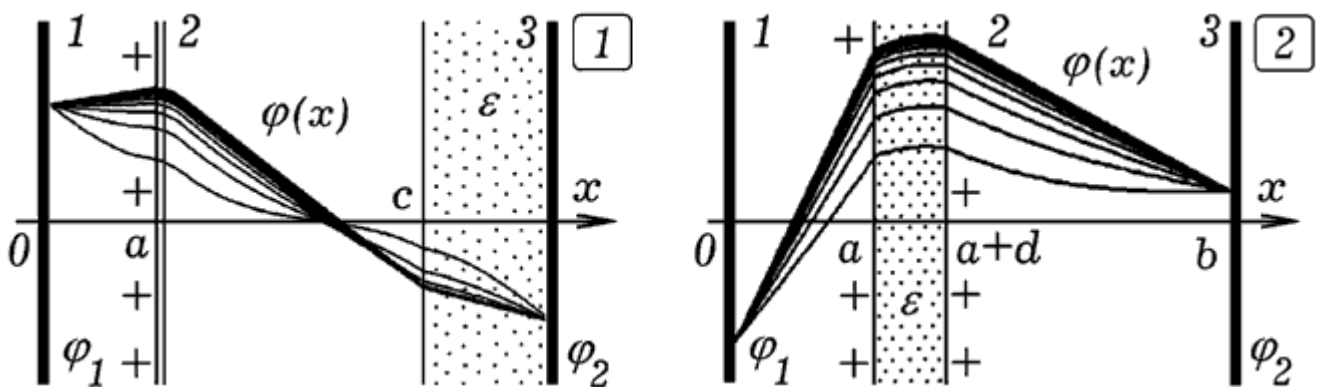


Рис. 12.7. Расчет потенциала поля в одномерном случае

Рассмотренный выше подход позволяет решить одномерную задачу, например, рассчитать распределение потенциала электростатического поля  $\varphi(x)$  между двумя бесконечно большими пластинами 1 и 3, расположенными перпендикулярно оси  $Ox$  на расстоянии  $b$  друг от друга (рис. 12.7.1). Потенциалы пластин  $\varphi_1$  и  $\varphi_2$  известны. Между ними находится пластина 2 из диэлектрика, заряженная положительно или отрицательно. Рядом с пластиной 3 имеется слой диэлектрика с проницаемо-



стью  $\varepsilon$ . Необходимо задать потенциалы на границах области, плотность распределения заряда  $\rho$  и зависимость  $\varepsilon = \varepsilon(x)$ . На рис. 12.7.2 показано распределение потенциала вдоль оси  $Ox$  в случае, когда между металлическими пластинами имеется толстая пластина с диэлектрической проницаемостью  $\varepsilon$ , внутри которой равномерно "размазан" электрический заряд ( $\rho = const$ ). На рис. 12.7 показан не только искомый график  $\varphi = \varphi(x)$ , но и несколько предшествующих ему приближений; это помогает понять сущность метода последовательных итераций.

## 12.5. Расчет электрического поля в полярной и цилиндрической системах координат

Выше рассмотрены методы решения уравнения Пуассона в декартовых координатах. Теперь рассчитаем распределение потенциала в двумерной области, имеющей форму четверти круга, для чего решим уравнение Пуассона в полярных координатах  $r$  и  $\alpha$  (рис. 12.8). Граничные условия зададим так: 1) граница  $AB$ :  $(d\varphi/d\alpha)_{AB} = 0$ ,  $\varphi_{i,0} = \varphi_{i,1}$ ; 2) граница  $BC$ :  $\varphi_{BC} = 20$  В,  $\varphi_{N,j} = 20$  В; 3) граница  $CD$ :  $(d\varphi/d\alpha)_{CD} = 0$ ,  $\varphi_{i,M+1} = \varphi_{i,M}$ ; 4) граница  $DA$ :  $(d\varphi/d\alpha)_{DA} = 0$ ,  $\varphi_{0,j} = 0$  В. Потенциалы двух точек поддерживаются постоянными:  $\varphi_{20,20} = \varphi_1 = 50$  В и  $\varphi_{50,70} = \varphi_2 = -80$  В. Уравнение Пуассона в полярных координатах и соответствующее конечно-разностное уравнение имеют вид [3; 8]:

$$\frac{1}{r} \frac{\partial \varphi}{\partial r} + \frac{\partial^2 \varphi}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2 \varphi}{\partial \alpha^2} = -\frac{q(x, y)}{\varepsilon \varepsilon_0},$$

$$\frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2r\Delta r} + \frac{\varphi_{i+1,j} - 2\varphi_{i,j} + \varphi_{i-1,j}}{\Delta r^2} + \frac{\varphi_{i,j+1} - 2\varphi_{i,j} + \varphi_{i,j-1}}{r^2 \Delta \alpha^2} = -\frac{q_{i,j}}{\varepsilon \varepsilon_0},$$

$$2 \left( \frac{1}{r^2 \Delta \alpha^2} + \frac{1}{\Delta r^2} \right) \varphi_{i,j} = \frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2r\Delta r} + \frac{\varphi_{i+1,j} + \varphi_{i-1,j}}{\Delta r^2} + \frac{\varphi_{i,j-1} + \varphi_{i,j+1}}{r^2 \Delta \alpha^2} + \frac{q_{i,j}}{\varepsilon \varepsilon_0}.$$

Отсюда выражают  $\varphi_{i,j}$ . Для решения задачи используется программа ПР-4. Результаты вычисления распределения потенциала в двумерной области при заданных граничных условиях представлены на рис. 12.8.2. Программа выполняет последовательность итераций, получающиеся значения потенциалов постепенно приближаются к искомым значениям, которые соответствуют точному решению задачи.

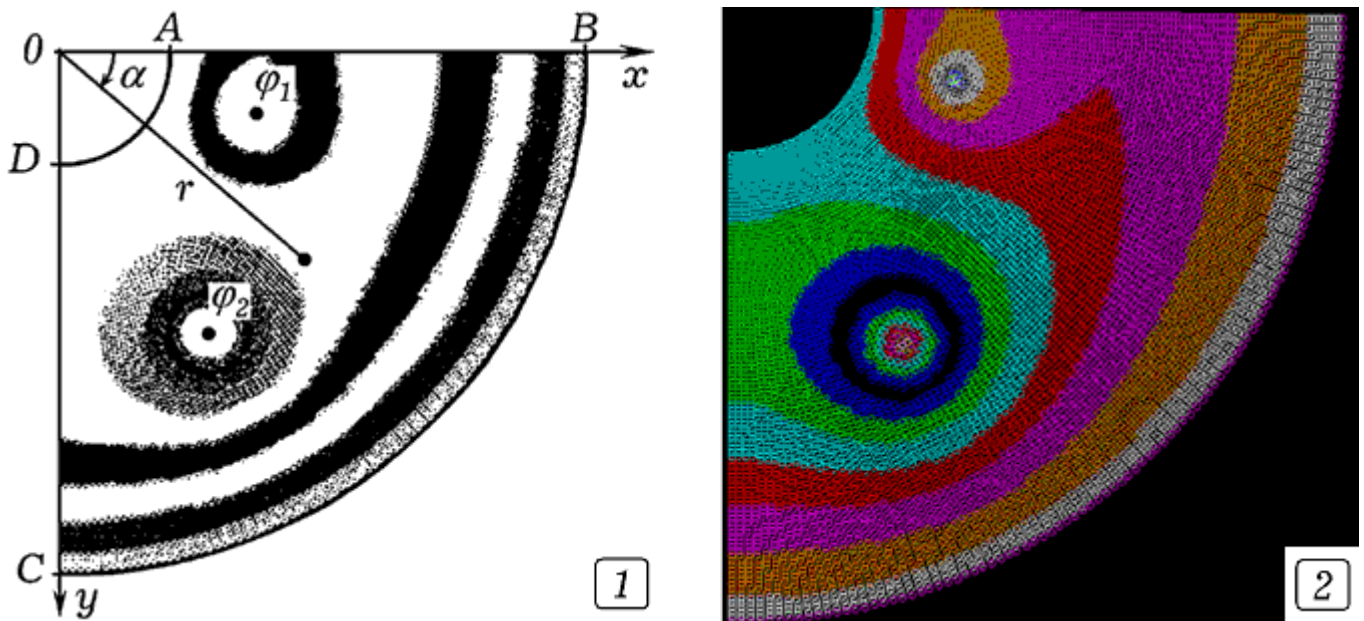


Рис. 12.8. Расчет электрического поля в сферических координатах

Теперь рассчитаем электростатическое поле в пространстве между двумя электродами 1 и 2 при наличии заряженного шара 3 (рис. 12.9.1), используя цилиндрические координаты. Распределение потенциала должно удовлетворять уравнению Пуассона:

$$\frac{1}{r} \frac{\partial \varphi}{\partial r} + \frac{\partial^2 \varphi}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2 \varphi}{\partial \alpha^2} + \frac{\partial^2 \varphi}{\partial z^2} = -\frac{\rho}{\epsilon \epsilon_0}.$$

Электроды обладают цилиндрической симметрией, то же самое относится к распределению заряда и граничным условиям. Поэтому искомое распределение потенциала не зависит от координаты  $\alpha$  и  $\partial^2 \varphi / \partial \alpha^2 = 0$ . Задача сводится к нахождению распределения потенциала в плоскости осевого сечения  $xOz$ . В конечных разностях получаем:



$$\frac{\varphi_{i+1,k} - \varphi_{i-1,k}}{2r\Delta r} + \frac{\varphi_{i-1,k} - 2\varphi_{i,k} + \varphi_{i+1,k}}{\Delta r^2} + \frac{\varphi_{i,k-1} - 2\varphi_{i,k} + \varphi_{i,k+1}}{\Delta z^2} = -\frac{\rho_{i,k}}{\varepsilon\varepsilon_0}.$$

Будем считать, что  $\Delta r = \Delta z = h$ . Выразим  $\varphi_{i,k}$ :

$$\varphi_{i,k} = \frac{\varphi_{i+1,k} - \varphi_{i-1,k}}{8r} h + \frac{\varphi_{i-1,k} + \varphi_{i+1,k} + \varphi_{i,k-1} + \varphi_{i,k+1}}{4} + \frac{\rho_{i,k}}{4\varepsilon\varepsilon_0} h^2.$$

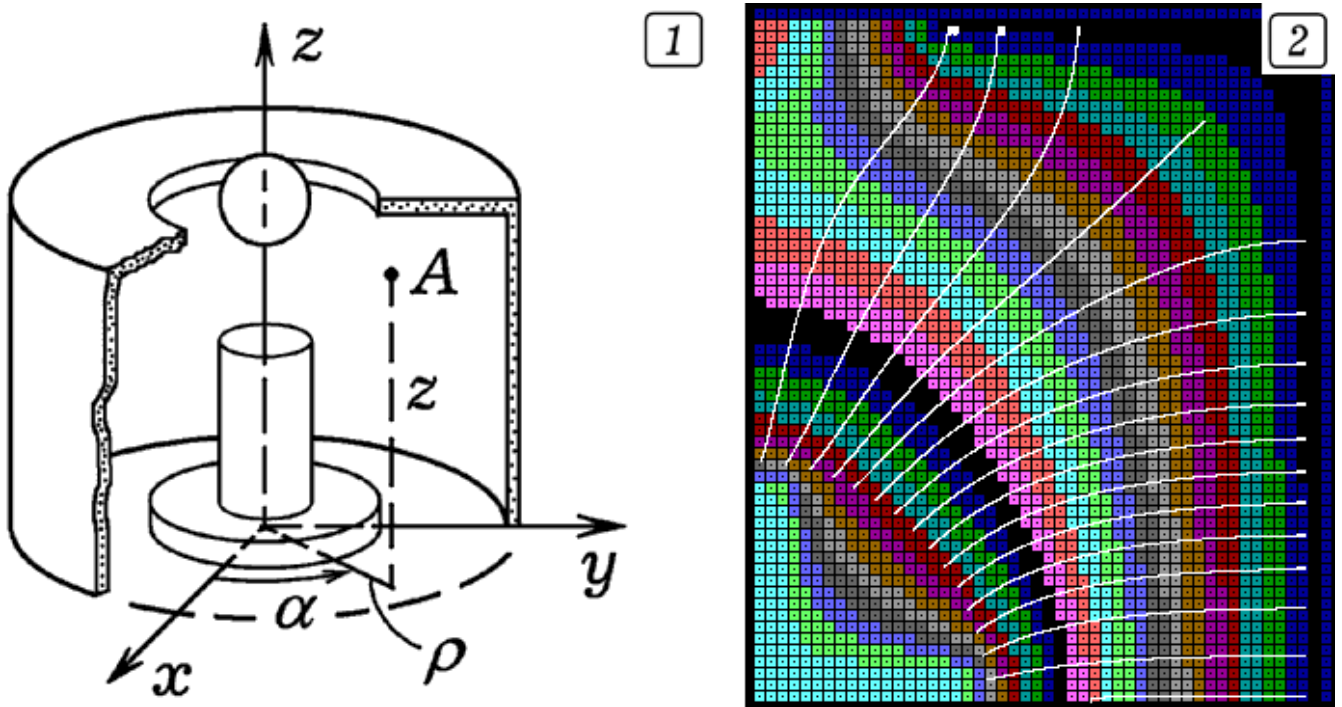


Рис. 12.9. Расчет электрического поля в цилиндрических координатах

Используется программа ТР-9, в которой перебираются все узлы двумерной сетки и с каждой итерацией пересчитываются значения потенциала. После 5000 итераций осуществляется построение силовых линий. Для этого в пространство между электродами запускаются невесомые частицы-маркеры, которые перемещаются вдоль силовых линий и рисуют свою траекторию. Напряженность  $\vec{E}$  равна градиенту потенциала  $\varphi$ . Для определения направления силовой линии в точке с координатами  $x, z$  используются формулы:

$$E_x = -\frac{1}{2} \left( \frac{\varphi_{i+1,k} - \varphi_{i,k}}{h} + \frac{\varphi_{i+1,k+1} - \varphi_{i,k+1}}{h} \right),$$

$$E_y = -\frac{1}{2} \left( \frac{\varphi_{i,k+1} - \varphi_{i,k}}{h} + \frac{\varphi_{i+1,k+1} - \varphi_{i+1,k}}{h} \right).$$

Результаты расчета электростатического поля представлены на рис. 12.9.2. Силовые линии перпендикулярны эквипотенциальным поверхностям и направлены от положительного электрода к отрицательному.

## 12.6. Расчет магнитного поля проводников с током

Сначала решим простую задачу и построим силовые линии магнитного поля, созданного параллельными проводниками с током (рис. 12.10).

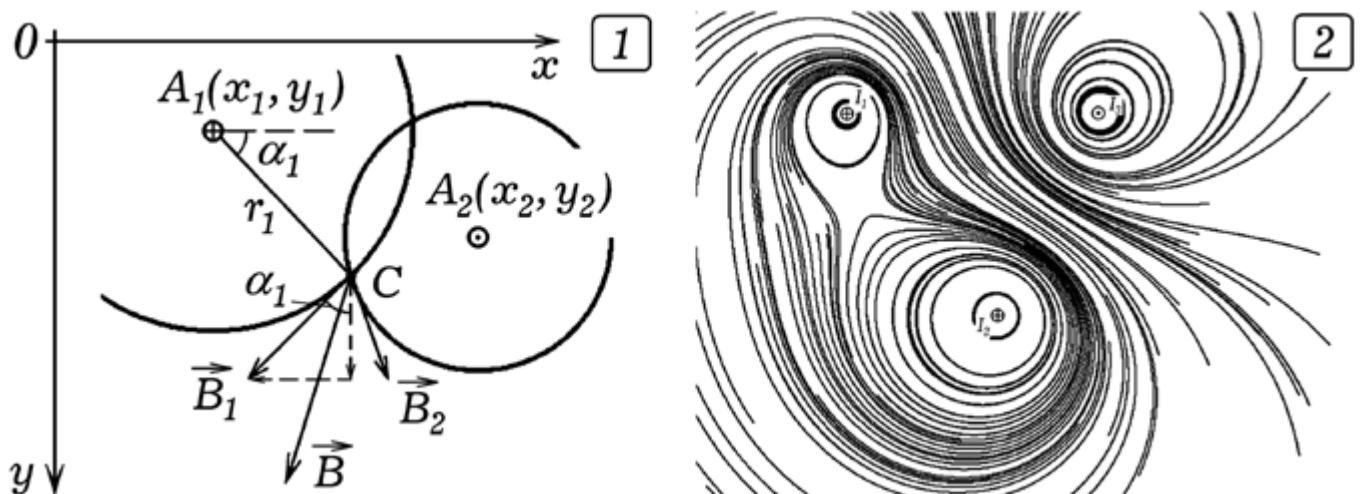


Рис. 12.10. Силовые линии магнитного поля трех параллельных токов

Допустим, два проводника с токами  $I_1$  и  $I_2$  пересекают плоскость  $xOy$  в точках  $A_1$  и  $A_2$  с координатами  $(x_1, y_1)$  и  $(x_2, y_2)$ . Индукция магнитного поля в точке  $C$  (рис. 12.10.1) может быть найдена по формулам:

$$\vec{B} = \vec{B}_1 + \vec{B}_2, \quad B_x = -B_1 \sin \alpha_1 - B_2 \sin \alpha_2, \quad B_y = B_1 \cos \alpha_1 + B_2 \cos \alpha_2,$$

$$\cos \alpha_1 = (x - x_1) / r_1, \quad \sin \alpha_1 = (y - y_1) / r_1, \quad B_1 = kI_1 / r_1,$$

$$\cos \alpha_2 = (x - x_2) / r_2, \quad \sin \alpha_2 = (y - y_2) / r_2, \quad B_2 = kI_2 / r_2,$$

$$r_1 = \sqrt{(x - x_1)^2 + (y - y_1)^2}, \quad r_2 = \sqrt{(x - x_2)^2 + (y - y_2)^2}.$$

Используемая программа ТР-10 случайным образом выбирает точку  $D_0$ , определяет проекции вектора индукции  $\vec{B}$  в этой точке и в его направлении строит отрезок единичной длины, находя точку  $D_1$ . Затем все повторяется снова. Выполнив 10000 шагов, программа снова случайно выбирает точку  $D_0$  и начинает строить новую силовую линию и т. д. В результате получается совокупность силовых линий магнитного поля (рис. 12.10.2).

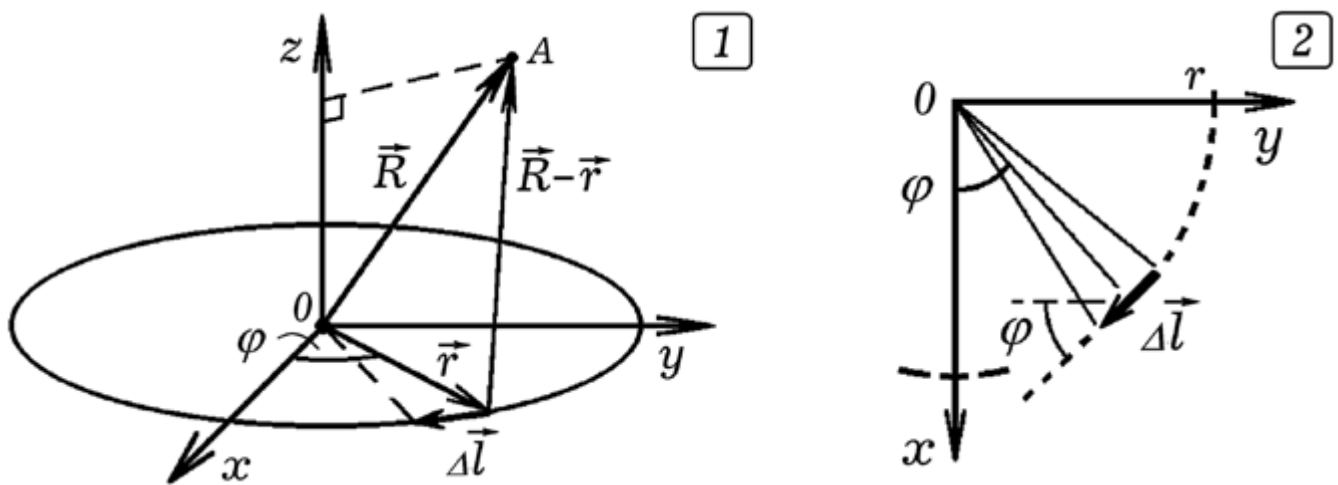


Рис. 12.11. К расчету индукции магнитного поля витка с током

Рассмотрим виток с током радиусом  $r$ , лежащий в плоскости  $xOy$  (рис. 12.11.1). Разобьем его на элементы длиной  $\Delta l = r\Delta\varphi$ ; координаты вектора  $\Delta\vec{l}_s$  (рис. 12.11.2) равны  $(r\sin\varphi_s\Delta\varphi, -r\cos\varphi_s\Delta\varphi, 0)$ ,  $\varphi_s = s\Delta\varphi$ . Расположение элемента  $s$ -го витка  $\Delta\vec{l}_s$  и точки  $A$ , в которой определяется индукция магнитного поля  $\vec{B}$ , задается векторами  $\vec{r}(r\cos\varphi_s, r\sin\varphi_s, 0)$  и  $\vec{R}(R_x, R_y, R_z)$ . По закону Био-Савара-Лапласа

$$\vec{B}(x, y, z) = \frac{\mu\mu_0}{4\pi} \sum_{s=1}^N \frac{I \cdot [\Delta\vec{l}_s, (\vec{r} - \vec{R})]}{|\vec{r} - \vec{R}|^3},$$

где  $\mu_0$  - магнитная постоянная,  $I$  - сила тока. Можно записать:

$$\Delta\vec{l}_s = \Delta l \sin\varphi_s \vec{i} - \Delta l \cos\varphi_s \vec{j} = r\Delta\varphi \sin\varphi_s \vec{i} - r\Delta\varphi \cos\varphi_s \vec{j},$$

$$\vec{r} = r \cos \varphi_s \vec{i} + r \sin \varphi_s \vec{j} + 0 \vec{k}, \quad \vec{R} = R_x \vec{i} + R_y \vec{j} + R_z \vec{k}.$$

В силу симметрии для всех точек плоскости  $yOz$   $B_x = 0$ . В программе ТР-11, рассчитывающей магнитное поле, перебираются узлы двумерной сетки на плоскости  $yOz$  и вычисляются проекции  $B_z$  и  $B_y$ :

$$B_z = \frac{\mu\mu_0}{4\pi} \sum_{s=1}^N \frac{(\Delta l_{sx} \Delta y_s - \Delta l_{sy} \Delta x_s)}{\Delta R_s^3}, \quad B_y = \frac{\mu\mu_0}{4\pi} \sum_{s=1}^N \frac{(\Delta l_{sz} \Delta x_s - \Delta l_{sx} \Delta z_s)}{\Delta R_s^3},$$

где  $\Delta x_s = r \cos \varphi_s - R_x$ ,  $\Delta y_s = r \sin \varphi_s - R_y$ ,  $\Delta z_s = 0 - R_z$  - проекции вектора  $\Delta \vec{R}_s = \vec{R} - \vec{r}_s$ , соединяющего элемент витка  $\Delta \vec{l}_s$  с точкой  $A$ ; его длина равна  $\Delta R_s = \sqrt{\Delta x_s^2 + \Delta y_s^2 + \Delta z_s^2}$ . Если обмотка содержит несколько витков, то соответствующие проекции вектора  $\vec{B}$  суммируются [5; 6].

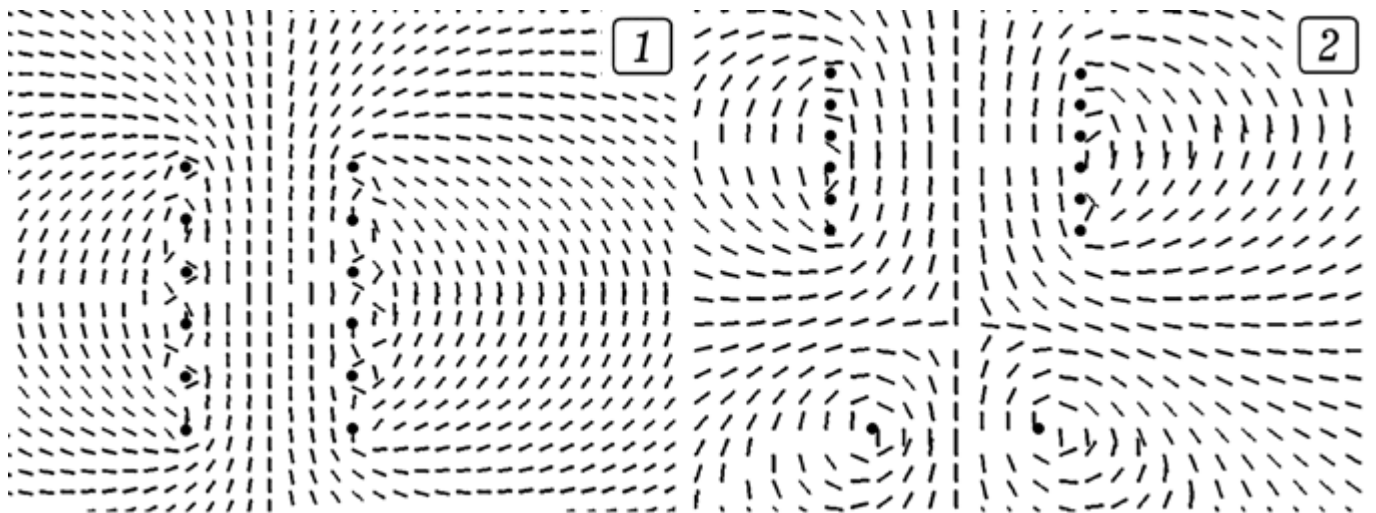


Рис. 12.12. Магнитное поле, создаваемое витками с током

На рис. 12.12 показан результат расчета магнитного поля, создаваемого шестью витками с током. Рис. 12.2.2 соответствует случаю, когда обмотка из шести витков расположена соосно с одиночным витком, причем по ним текут токи во встречных направлениях.

Теперь построим силовые линии магнитного поля, созданного двумя обмотками с током. Для этого выберем начальную точку  $A_0$ , в ней определим направление индукции магнитного поля  $\vec{B}$ , построим небольшой отрезок в направлении вектора  $\vec{B}$  и перейдем в новую точку  $A_1$ . Ес-

ли все это повторить многократно, то получится силовая линия. Предлагается самостоятельно написать программу, решающую эту задачу [5]. При этом можно рассмотреть два случая, когда токи в обмотках текут в одном (рис. 12.13.1) или в противоположных направлениях (рис. 12.13.2).

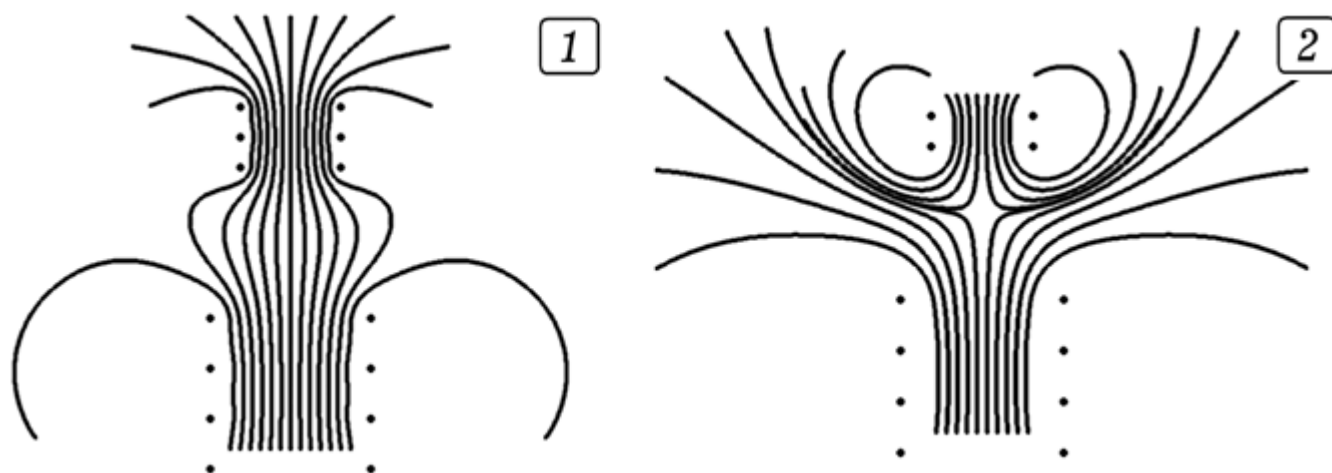


Рис. 12.13. Силовые линии магнитного поля двух обмоток

## 12.7. Распространение электромагнитной волны в волноводе

Электромагнитная волна представляет собой колебания электрического и магнитного полей, распространяющиеся в пространстве. При этом вектора  $\vec{E}$  и  $\vec{H}$  колеблются во взаимно перпендикулярных плоскостях и удовлетворяют волновым уравнениям:

$$\frac{\partial^2 E}{\partial x^2} + \frac{\partial^2 E}{\partial y^2} + \frac{\partial^2 E}{\partial z^2} = \frac{1}{v^2} \frac{\partial^2 E}{\partial t^2}, \quad \frac{\partial^2 H}{\partial x^2} + \frac{\partial^2 H}{\partial y^2} + \frac{\partial^2 H}{\partial z^2} = \frac{1}{v^2} \frac{\partial^2 H}{\partial t^2}.$$

В главе 8 проанализированы модели одномерной и двумерной волн, получающиеся в результате решения волнового уравнения.

Рассмотрим двумерную модель распространения электромагнитной волны по волноводу. Будем считать, что на оси прямоугольного волновода находится источник, создающий электромагнитную волну, у которой вектор  $\vec{E}$  колеблется перпендикулярно плоскости экрана. Эта задача ана-

логична задаче о распространении механической волны вдоль длинной пластины с закрепленными краями. При этом достаточно рассчитать  $\vec{E}$  в одной половине волновода, а при выводе на экран прорисовать обе.

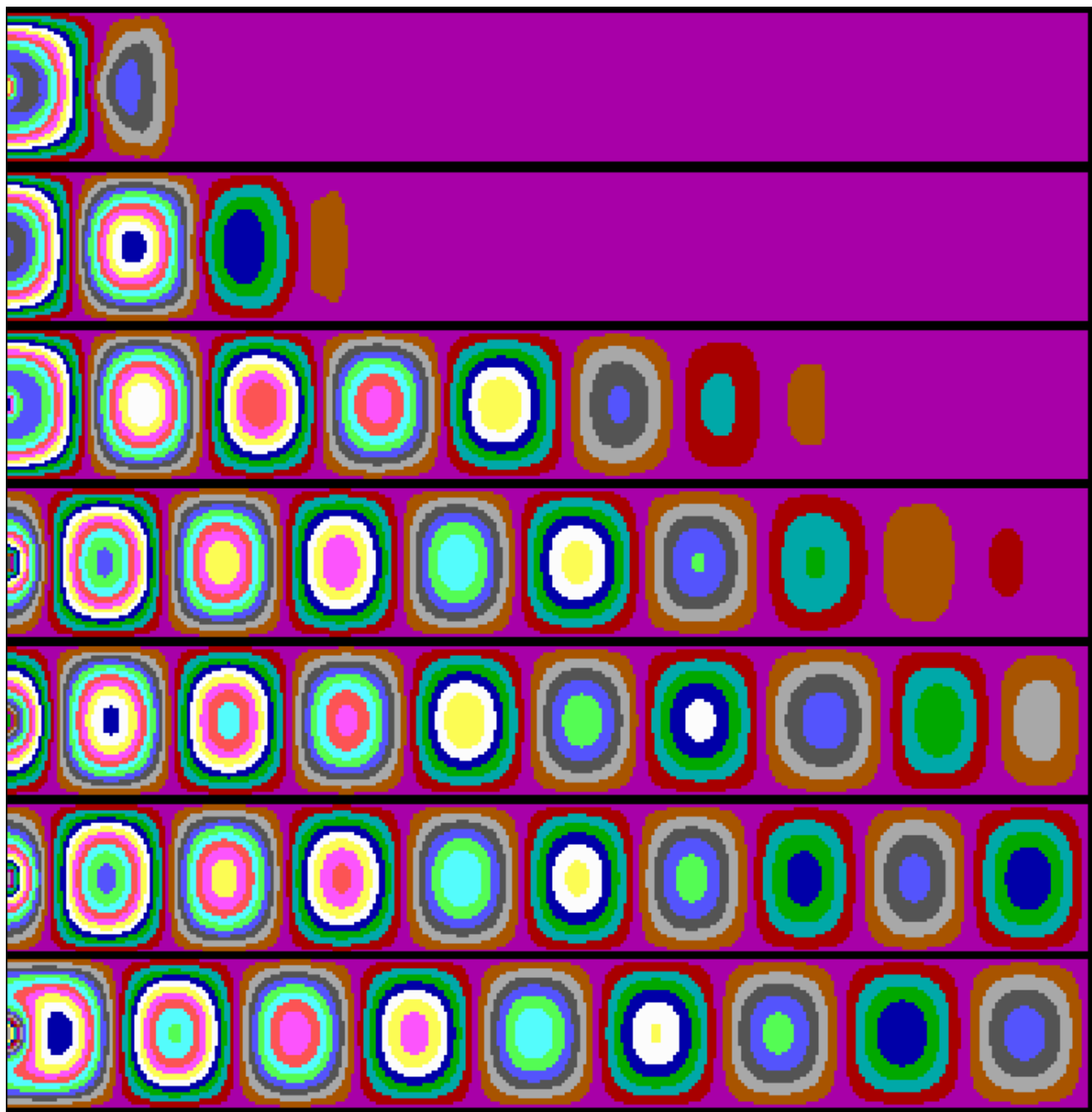


Рис. 12.14. Распространение электромагнитной волны в волноводе, правый конец которого закрыт. Образование стоячей волны

На рис. 12.14 показаны состояния системы в последовательные моменты времени. Видно, как волна распространяется по волноводу слева направо, доходит до правого закрытого конца и отражается. Возникает

стоячая волна, при этом в соседних пучностях вектор  $\vec{E}$  направлен в противоположные стороны (к нам или от нас). Вектор  $\vec{H}$  колеблется в плоскости рисунка. Известны и другие способы решения этой задачи, например, в книге [2, с. 135-141] рассмотрен способ решения системы уравнений для  $\vec{E}$  и  $\vec{H}$  на "сдвинутых сетках".

## Приложение к главе 12

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 el_pole_otkl_plastin;
uses dos, crt, graph;
const dt=0.002; M=8;
a: array[1..26]of integer=(20,22,24,26,28,30,32,34,36,
38, 40,42,44,20,22,24,26,28,30,32,34,36,38,40,42,44);
b: array[1..26]of integer=(5,5,5,5,5,5,5,5,5,5,6,7,8,9,
-5,-5,-5,-5,-5,-5,-5,-5,-5,-5,-6,-7,-8,-9);
var U,vy,F,Fx,Fy,l,t: real;
x,y,k,i,Gd,Gm,cvet: integer; q: array[1..26]of real;
BEGIN
Gd:=Detect; InitGraph(Gd, Gm,'c:\bp\bgi');
For i:=1 to 13 do q[i]:=20;
For i:=14 to 26 do q[i]:=-20;
For x:=1 to 640 do For y:=1 to 480 do begin
  U:=0; For i:=1 to 26 do begin
    l:=sqrt(sqr((x-M*a[i])/1)+sqr((y-M*b[i]-240)/1));
    If l=0 then l:=0.01; U:=U+q[i]/l; end;
    cvet:=abs(round(U+1000) mod 15);
    putpixel(x,y,cvet); end;
Repeat until KeyPressed; CloseGraph;
END.

```



## Программа ПР-2

```

program silovie_linii_el_polya;
uses crt, graph; const d=1;
var x0,y0,amax,max,fil,fi,a,r1,r2,r3,x,y,pi: real;
EC,DV,MV,i,j,q1,q2,q3,x1,y1,x2,y2,x3,y3,k: integer;
Label metka;
Procedure Raschet;
begin
Repeat r1:=sqrt((x-x1)*(x-x1)+(y-y1)*(y-y1));
r2:=sqrt((x-x2)*(x-x2)+(y-y2)*(y-y2));
r3:=sqrt((x-x3)*(x-x3)+(y-y3)*(y-y3));
fi:=(q1/r1)+(q2/r2)+(q3/r3); max:=0; x0:=x; y0:=y;
For i:=1 to 90 do begin a:=2*pi*i/90;
x:=x0+cos(a); y:=y0+sin(a);
r1:=sqrt((x-x1)*(x-x1)+(y-y1)*(y-y1));
r2:=sqrt((x-x2)*(x-x2)+(y-y2)*(y-y2));
r3:=sqrt((x-x3)*(x-x3)+(y-y3)*(y-y3));
fil:=(q1/r1)+(q2/r2)+(q3/r3);
If fil-fi>max then begin max:=fil-fi; amax:=a; end;
end; k:=k+1; x:=x0+cos(amax); y:=y0+sin(amax);
circle(round(x),round(y),1); until k>500;
end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
pi:=arctan(1)*4; x1:=250; y1:=240; q1:=-10;
x2:=300; y2:=100; q2:=20; x3:=400; y3:=300; q3:=10;
circle(x1,y1,2); circle(x2,y2,2); circle(x3,y3,2);
For j:=1 to 20 do begin x:=250+d*cos(0.314*j);
y:=240+d*sin(0.314*j); Raschet; k:=0; end;
Repeat until Keypressed; CloseGraph;
END.

```

## Программа ПР-3

```

program electr_linza;
uses crt, graph;
const dt=0.002; Ms=30; q1=5; q2=-23; m=1;
a: array[1..14]of integer=(20,21,22,24,26,28,30,20,
21,22,24,26,28,30);
b: array[1..14]of integer=(2,2,2,3,3,3,3,-2,-2,-2,
-3,-3,-3,-3);

```

```

var x,y,vx,vy,F,Fx,Fy,l,l1,t,v,a1: real;
k,k1,i,Gd, Gm,n: integer; q: array[1..14]of real;
begin v:=15{25}; Randomize;
Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
For i:=1 to 3 do q[i]:=q1; For i:=4 to 7 do q[i]:=q2;
For i:=8 to 10 do q[i]:=q1;
For i:=11 to 14 do q[i]:=q2; a1:=-0.15;
Repeat y:=0.2*k1-1; x:=15; inc(k1); a1:=0;
  vx:=v*cos(a1); vy:=v*sin(a1);
  Repeat Fx:=0; Fy:=0; t:=t+dt;
    For i:=1 to 14 do begin
      l:=sqr(x-a[i])+sqr(y-b[i]);
      Fx:=Fx-q[i]*(x-a[i])/l/sqrt(l);
      Fy:=Fy-q[i]*(y-b[i])/l/sqrt(l); end;
      vx:=vx+Fx*dt/m; vy:=vy+Fy*dt/m;
      x:=x+vx*dt; y:=y+vy*dt; inc(k);
      putpixel(round(Ms*x)-450,300-round(Ms*y),15);
      For i:=1 to 14 do circle(Ms*a[i]-450,300-Ms*b[i],2);
    until (KeyPressed)or(k>2500); k:=0;
until (KeyPressed)or(k1>10);
Repeat until KeyPressed; CloseGraph;
END.

```

Программа ПР-4

```

program pereraspr_zaradov;
uses crt, graph;
const n=100; m=78; X0=395; Y0=235; R=70;
var i,j,k,s,DV,MV,EC: integer; x1,y1,U,U0,l,d: real;
  x,y,q: array[0..N+1] of real;
Procedure Energiya;
begin U:=0; For j:=1 to N do begin
  l:=sqrt(sqr(x[i]-x[j])+sqr(y[i]-y[j]));
  If l=0 then l:=1000; U:=U+200*q[i]*q[j]/l; end; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
For i:=1 to N do begin x[i]:=random(280)+20;
  y[i]:=random(330)+20; q[i]:=-1;
  If i>m then begin x[i]:=random(R)+X0-R/2;
    y[i]:=random(R)+Y0-R/2; q[i]:=2; end; end;
Repeat k:=k+1; i:=round(random(N+1));

```

```

Energiya; U0:=U; x1:=x[i]; y1:=y[i];
x[i]:=x[i]+random(600)/100-3;
y[i]:=y[i]+random(600)/100-3;
If i<=m then begin If x[i]<20 then x[i]:=20;
  If x[i]>300 then x[i]:=300; If y[i]<20 then y[i]:=20;
  If y[i]>350 then y[i]:=350; end;
  If (i>m)and(sqr(x[i]-X0)+sqr(y[i]-Y0)>R*R) then begin
  d:=sqrt(sqr(x[i]-X0)+sqr(y[i]-Y0))/R;
  x[i]:=X0+(x[i]-X0)/d; y[i]:=Y0+(y[i]-Y0)/d; end;
  Energiya; If U>=U0 then begin x[i]:=x1;y[i]:=y1; end;
If k mod 1000=0 then begin cleardevice; k:=0;
  rectangle(20,20,300,350); circle(X0,Y0,R);
  For i:=1 to N do begin
    circle(round(x[i]),round(y[i]),2);
    circle(round(x[i]),round(y[i]),3);
    end; delay(5); end;
until KeyPressed; CloseGraph;
END.

```

Программа ПР-5

```

program el_pole_odnorodn_sreda;
{$N+} uses crt, graph; const n=100; m=100;
var i,ii,j,jj,k,DV, MV : integer;
fi: array[0..N+1,0..M+1] of single; q : single;
Procedure Raschet;
begin
If ((i>55)and(i<65)and(j>45)and(j<55))
      then q:=3 else q:=0;
If ((i>15)and(i<25)and(j>15)and(j<25)) then q:=-4;
If ((i>35)and(i<45)and(j>15)and(j<20)) then q:=-6;
fi[i,j]:=0.25*(fi[i+1,j]+fi[i-1,j]+fi[i,j+1]+
fi[i,j-1])+q; end;
Procedure Gran_usl;
begin for i:=1 to N do
begin fi[i,2]:=0; fi[i,M-1]:=2*i;
  fi[2,i]:=0; fi[N-1,i]:=2*i; end;
For i:=60 to 65 do For j:=70 to 75 do fi[i,j]:=0; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat inc(k);
  For i:=2 to N-1 do

```

```

    For j:=2 to M-1 do Raschet; Gran_usl;
  For j:=2 to M-1 do
    For i:=2 to N-1 do Raschet; Gran_usl;
  If k mod 50=0 then
    For i:=2 to N-1 do For j:=2 to M-1 do
      begin setcolor(round(abs(5-fi[i,j]/10)));
        rectangle(i*4+50,j*4,i*4+54,j*4+4);
        rectangle(i*4+51,j*4+1,i*4+53,j*4+3); k:=0;
      end;
  until KeyPressed; CloseGraph;
END.

```

Программа ПР-6

```

program el_pole_neodnorodn_sreda;
{$N+} uses crt, graph; const N=70; M=70;
var i, j, ii, jj, k, q, DV, MV :integer;
fi, e: array[1..N, 1..M] of single;
Procedure Sreda;
begin For i:=1 to N do For j:=1 to M do e[i,j]:=1+0.015*j*i;
end;
Procedure Gran;
begin
For i:=1 to N do begin fi[i,1]:=-350; fi[i,M]:=350;
  fi[1,i]:=fi[2,i]; fi[N,i]:=fi[N-1,i]; end;
For j:=15 to 25 do fi[55,j]:=0; end;
Procedure Raschet;
begin q:=0;
If ((i>20) and (i<30) and (j>30) and (j<35)) then q:=500;
If ((i>40) and (i<55) and (j>40) and (j<55)) then q:=-300;
fi[i,j]:=(fi[i+1,j]+fi[i-1,j]+fi[i,j+1]+fi[i,j-1]+q/
e[i,j])/4+((e[i-1,j]-e[i+1,j])*(fi[i+1,j]-fi[i-1,j])
+(e[i,j-1]-e[i,j+1])*(fi[i,j+1]-fi[i,j-1]))/
(16*e[i,j])); end;
Procedure Draw;
begin setcolor(round(abs(2+fi[i,j]/30)));
  rectangle(i*4+50,j*4,i*4+54,j*4+4);
  rectangle(i*4+51,j*4+1,i*4+53,j*4+3); end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi'); Sreda;
Repeat inc(k);
  Gran; For i:=2 to N-1 do For j:=2 to M-1 do Raschet;

```

```

Gran; For j:=M-1 downto 2 do
    For i:=N-1 downto 2 do Raschet;
If k mod 50=0 then
    For i:=2 to N-1 do For j:=2 to M-1 do Draw;
until KeyPressed; CloseGraph;
END.

```

## Программа ПР-7

```

{$N+} program truba_el_pole;
uses crt, graph; const N=80; M=95; R=34; dt=0.1; h=1;
var i,j,q,s,DV,MV: integer; x,y,Ex,Ey:real;
fi,e: array[0..N+1,0..M+1] of single;
k: Longint; u: boolean;
Procedure Sreda;
begin For i:=1 to N do For j:=1 to M do begin
    u:=(sqr(i)+sqr(j-48)<R*R)and(sqr(i)+sqr(j-48)>R*R/4);
    {u:=(i<45)and(abs(j-48)<R);}
    If u then e[i,j]:=3 else e[i,j]:=1; end; end;
Procedure Gran;
begin For i:=1 to N do begin fi[i,1]:=100;
    fi[i,M]:=-100; end; For j:=1 to M do begin
    fi[1,j]:=fi[2,j]; fi[N,j]:=fi[N-1,j]; end; end;
Procedure Raschet;
begin fi[i,j]:=(fi[i+1,j]+fi[i-1,j]+fi[i,j+1]+fi[i,
j-1]+q/e[i,j])/4+((e[i-1,j]-e[i+1,j])*(fi[i+1,j]-fi
[i-1,j])+(e[i,j-1]-e[i,j+1])*(fi[i,j+1]-fi[i,j-1]))
/(16*e[i,j])); end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Sreda; x:=20; y:=10;
Repeat inc(k);
If k<2E+3 then begin
Gran; For i:=2 to N-1 do For j:=2 to M-1 do Raschet;
Gran; For j:=M-1 downto 2 do
    For i:=N-1 downto 2 do Raschet;
If k mod 200=0 then begin
    For i:=2 to N-1 do For j:=2 to M-1 do
    begin setcolor(round(10+fi[i,j]/10)mod 15);
    circle(50+i*5,j*5,2); circle(50+i*5,j*5,1); end;
end; end;
If (k>2E+3)and(x<5*80) then begin

```

```

If (y>5*93) then begin inc(s); x:=20*s; y:=10; end;
i:=round(int(x/5)); j:=round(int(y/5));
Ex:=- (fi[i+1,j]-fi[i,j]+fi[i+1,j+1]-fi[i,j+1])/h/2;
Ey:=- (fi[i,j+1]-fi[i,j]+fi[i+1,j+1]-fi[i+1,j])/h/2;
x:=x+Ex*dt; y:=y+Ey*dt;
setcolor(white); circle(50+round(x),round(y),1); end;
If k=2E+3 then begin
    setcolor(white); circle(50,48*5,R*5);
    circle(50,48*5,round(R*5/2+2)); end;
until KeyPressed; CloseGraph;
END.

```

Программа ПР-8

```

Program el_pole_polyarnie_koordinat;
uses crt, graph;
const N=100; M=100; a=0.4; R0=3; dr=0.1; dt=0.005;
var i,j,k,Gd,Gm: integer; q,dal,al,pi,r: real;
fi: array[0..N+1,0..M+1] of real;
BEGIN
Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
pi:=arctan(1)*4; dal:=pi/M/2;
Repeat
    For i:=1 to N do For j:=1 to M do begin r:=R0+i*dr;
        fi[20,20]:=50; fi[50,70]:=-80; fi[N,j]:=20;
        fi[i,j]:=((fi[i+1,j]-fi[i-1,j])/2/r/dr+(fi[i,j+1]+
            fi[i,j-1])/r/r/dal/dal+(fi[i-1,j]+fi[i+1,j])/dr/dr
            +q)/2/(1/r/r/dal/dal+1/dr/dr);
        fi[i,0]:=fi[i,1]; fi[i,M+1]:=fi[i,M]; end;
    If k mod 10=0 then For i:=1 to N do For j:=1 to M do
        begin r:=30*(R0+i*dr); al:=dal*j;
            setcolor(round(abs(fi[i,j]/5+3))); circle(100+
                round(r*cos(al)),20+round(r*sin(al)),2);
            end; inc(k);
until KeyPressed; CloseGraph;
END.

```

Программа ПР-9

```

{$N+}program Potenc_cilindr_koordinati;
uses crt, graph;

```

```

const N=50; M=60; h=1; dt=0.01;
var i,j,k,DV,MV: integer; r,C,Ex,Ey,x,y : single;
t:Longint; f,f1: array[0..N+1,0..M+1] of single;
procedure Raschet;
begin r:=h*i+0.001;
If i*i+(M-k)*(M-k)<64 then C:=1 else C:=0;
f1[i,k]:=(f[i-1,k]+f[i+1,k]+f[i,k-1]+f[i,k+1]+(f[i+1,k]-f[i-1,k])*h/2/r+C)/4; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi'); x:=30;
Repeat inc(t); If t<5000 then begin
For i:=0 to N do For k:=0 to M do begin
If (i<4)and(k<20)then f[i,k]:=30;
If (i<20)and(k<4)then f[i,k]:=30;
If (i>N-2)or((k>M-2)and(i>15)) then f[i,k]:=-30;
f[i,M]:=f[i,M-1]; f[i,0]:=f[i,1]; f[0,k]:=f[1,k]; end;
For i:=1 to N-1 do For k:=1 to M-1 do Raschet;
For i:=0 to N do For k:=0 to M do f[i,k]:=f1[i,k]; end;
If (t>5000)and(j<15) then begin If f[i,k]<-29.5 then
begin inc(j); x:=21*cos(j/10); y:=21*sin(j/10); t:=5000;
end; i:=round(x); k:=round(y);
Ex:=- (f[i+1,k]-f[i,k]+f[i+1,k+1]-f[i,k+1])/2;
Ey:=- (f[i,k+1]-f[i,k]+f[i+1,k+1]-f[i+1,k])/2;
x:=x+Ex*dt; y:=y+Ey*dt; setcolor(white);
circle(300+round(6*x),400-round(6*y),1); end;
If (t<5000)and(t mod 100=0) then begin
For i:=1 to N do For k:=1 to M do begin
setcolor(abs(round((f[i,k]+30)/2)mod 14));
rectangle(300+6*i,400-6*k,304+6*i,404-6*k);
rectangle(301+6*i,401-6*k,303+6*i,403-6*k); end; end;
until KeyPressed; CloseGraph;
END.

```

Программа ПР-10

```

program magnitn_pole_toka;
uses crt, graph;
var DrV,MV,i,x1,y1,x2,y2,x3,y3,k : integer;
    cosa1,sina1,cosa2,sina2,cosa3,sina3,I1,I2,I3,
    B,B1,B2,B3,Bx,By,r1,r2,r3,x,y,pi: real;
BEGIN DrV:=Detect; InitGraph(DrV,MV,'c:\bp\bgi');
pi:=arctan(1)*4; x1:=150; y1:=100;

```



```

x2:=400; y2:=100; x3:=300; y3:=300;
circle(x1,y1,2); circle(x2,y2,2); circle(x3,y3,2);
x:=300; I1:=1; I2:=-2; I3:=2;
Randomize;
For i:=1 to 80 do begin
  k:=0; x:=round(random(64))*10;
  y:=round(random(48))*10;
  Repeat k:=k+1;
    r1:=sqrt((x-x1)*(x-x1)+(y-y1)*(y-y1));
    r2:=sqrt((x-x2)*(x-x2)+(y-y2)*(y-y2));
    r3:=sqrt((x-x3)*(x-x3)+(y-y3)*(y-y3));
    B1:=I1/r1; cosa1:=(x-x1)/r1; sina1:=(y-y1)/r1;
    B2:=I2/r2; cosa2:=(x-x2)/r2; sina2:=(y-y2)/r2;
    B3:=I3/r3; cosa3:=(x-x3)/r3; sina3:=(y-y3)/r3;
    Bx:=-B1*sina1-B2*sina2-B3*sina3;
    By:=B1*cosa1+B2*cosa2+B3*cosa3;
    B:=sqrt(Bx*Bx+By*By);
    x:=x+0.1*Bx/B; y:=y+0.1*By/B;
    putpixel(round(x),round(y),15);
  until (k>10000)or KeyPressed; end;
Repeat until KeyPressed; CloseGraph;
END.

```

Программа ПР-11

```

program magn_pole_vitki;
uses crt, graph;
const N=100; pi=3.1415; r=80;
var S,k, ii,jj,i,j,DV, MV : integer; Rx,Ry,Rz: integer;
Bx,By,Bz,dfi,deltax,deltay,deltaz,deltaR,dlx,dly,dlz, alpha:
real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
For ii:=-25 to 25 do For jj:=-10 to 30 do begin
  Rx:=0; Ry:=20*ii; Rz:=20*jj; Bz:=0; By:=0;
  For k:=0 to 5 do For i:=1 to N do begin
    dfi:=2*pi/N; dlx:=r*sin(dfi*i)*dfi;
    dly:=-r*cos(dfi*i)*dfi; deltax:=r*cos(dfi*i)-Rx;
    deltay:=r*sin(dfi*i)-Ry; deltaz:=50*k-Rz;
    deltaR:=sqrt(deltax*deltax+deltay*deltay+
    deltaz*deltaz); Bz:=(dlx*deltay-dly*deltax)/
    (deltaR*deltaR*deltaR)+Bz; By:=(dlz*deltax-

```

```

    dlx*deltaz)/(deltaR*deltaR*deltaR)+By;
    If By<>0 then alpha:=arctan(Bz/By); end;
    line(250+Ry,400-Rz,250+round(Ry+12*cos(alpha)),
        400-round(Rz+12*sin(alpha))); end;
For k:=0 to 5 do begin circle(250+r,400-50*k,3);
    circle(250-r,400-50*k,3); end;
Repeat until KeyPressed; CloseGraph;
END.

```

## Программа ПР-12

```

program em_volna_volnovod;
{$N+} uses crt, graph;
const N=350; M=25; h=1; dt=0.002; vv=4; a=2E-5;
var i,j,DV,MV: integer; t: single; k:Longint;
    xi,eta: array[1..N,1..M] of single;
Procedure Form1;
begin eta[i,j]:=(1-a)*eta[i,j]+vv*(xi[i,j+1]-
    2*xi[i,j]+xi[i,j-1])/h/h*dt; end;
Procedure Form2;
begin eta[i,j]:=(1-a)*eta[i,j]+vv*(xi[i+1,j]-
    2*xi[i,j]+xi[i-1,j])/h/h*dt; end;
Procedure Draw;
begin For i:=2 to N-1 do For j:=2 to M-1 do
begin setcolor(round(20+xi[i,j]/3)mod 15);
    rectangle(10+i*2,350+j*2,13+i*2,353+j*2);
    rectangle(11+i*2,351+j*2,12+i*2,352+j*2);
    rectangle(10+i*2,354-j*2,13+i*2,357-j*2);
    rectangle(11+i*2,355-j*2,12+i*2,356-j*2); end;
end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
setbkcolor(15); setcolor(8);
Repeat t:=t+dt; xi[2,2]:=60*sin(t*0.3); inc(k);
For i:=2 to N-1 do For j:=2 to M-1 do Form1;
For i:=2 to N-1 do For j:=2 to M-1 do
    xi[i,j]:=xi[i,j]+eta[i,j]*dt;
For j:=M-1 downto 2 do For i:=N-1 downto 2 do Form2;
For i:=2 to N-1 do For j:=2 to M-1 do
    xi[i,j]:=xi[i,j]+eta[i,j]*dt;
For j:=1 to M do xi[1,j]:=xi[2,j];
For i:=1 to N do xi[i,1]:=xi[i,2];

```

```
If k mod 2000=0 then begin Draw; k:=0; end;  
until KeyPressed; CloseGraph;  
END.
```

## Список литературы

1. Гулд Х., Тобочник Я. Компьютерное моделирование в физике. В 2 ч. Ч. 1. М.: Мир, 1990. 349 с.
2. Ращиков В. И., Рошаль А. С. Численные методы решения физических задач: учеб. пособие. СПб.: Издательство "Лань", 2005. 208 с.
3. Калиткин Н. Н. Численные методы. БХВ-Петербург, 2011. 592 с.
4. Кунин С. Вычислительная физика. М.: Мир, 1992. 518 с.
5. Майер Р. В. Задачи, алгоритмы, программы: электронное учеб. пособие. Глазов: Глазов. гос. пед. ин-т, 2012. URL: <http://maier-rv.glazov.net>
6. Майер Р. В. Компьютерное моделирование физических явлений. Глазов: Глазов. гос. пед. ин-т, 2009. 112 с.
7. Саранин В. А. Избранные задачи электростатики: Старые заблуждения и новые решения. Lambert Academic Publishing, 2013. 171 с.
8. Федоренко Р. П. Введение в вычислительную физику: учеб. пособие для вузов. М.: Изд-во Моск. физ.-техн. ин-та, 1994. 528 с.
9. Giordano N. J. Computational Physics. New Jersey, Prentice Hall, 1997. 419 p.
10. Phillipson P. E., Schuster P. Modeling by Nonlinear Differential Equations: Dissipative and Conservative Processes. World Scientific Publishing, 2009. 225 p.
11. Woolfson M. M., Pert G. J. An Introduction to Computer Simulation. Oxford University Press, 1999. 311 p.
12. Saranin V. A. About behavior of electrostatic pendulum near conducting or dielectric plate // Journal of Electrostatics. 2014. № 72. P. 235-241.