Майер Р.В., Глазовский пединститут

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ:
2. НЕПРЕРЫВНО-ДЕТЕРМИНИРОВАННЫЕ МОДЕЛИ ДИНАМИЧЕСКИХ СИСТЕМ С КОНЕЧНЫМ ЧИСЛОМ СТЕПЕНЕЙ СВОБОДЫ

Различают динамические и стохастические системы. Динамические системы описываются алгебраическими или дифференциальными уравнениями, а характеризующие их величины изменяются непрерывно так, что небольшое относительное изменение внешнего воздействия, входного сигнала или параметра приводит к сопоставимому изменению состояния системы. Например: движение материальной точки в силовом поле, электромагнитные колебания в колебательном контуре, изменение численности популяции, изменения концентрации того или иного вещества в химических реакциях и т.д. При изучении подобных систем используется непрерывно-детерминированный подход, а соответствующие ему математические схемы называются D-схемами (от англ. dynamic). Анализ систем с небольшим числом степеней свободы требует совместного решения небольшого числа уравнений [1, 6]. В ряде случаев решение задачи может быть найдено только численными методами.

2.1. Модели, требующие решения алгебраических уравнений

В некоторых случаях компьютерное моделирование изучаемой динамической системы не требует вычисления производных и интегралов, а сводится к решению алгебраических уравнений, исследованию функции одного или нескольких аргументов графическими методами (построение графика, поверхности, силовых линий, линий тока, линий равного уровня, потенциала, температуры, световых лучей и т.д.). Используемая компьютерная программа должна содержать один или несколько вложенных циклов, в которых осуществляется многократное выполнение одной и той же вычислительной процедуры, связанной с нахождением значения функции в узлах одномерной или многомерной сетки. Результаты вычислений выводятся на экран в числовом или графическом виде: печатается массив чисел, либо строятся графики, поверхности, изолинии, линии наибыстрейшего спуска и т.д.

Достаточно часто изучение того или иного процесса требует решения алгебраического уравнения $f(x)=g(x)$, которое можно записать в виде $F(x)=f(x)-g(x)=0$. Для решения уравнений и систем уравнений используют прямые (точные) и итерационные (приближенные) методы [5]. В вычислительной математике под точным решением понимают решение с точностью до погрешности округления, которое выдала бы идеальная ЭВМ с бесконечной разрядностью машинного слова, работающая по точным формулам. **Прямые методы** дают точное решение, причем число арифметических операций может быть оценено заранее. Например, для точного решения квадратного уравнения сначала определяют дискриминант, а затем находят корни с тем количеством знаков, на которое рассчитан компьютер. **Итерационные методы** имеют следующие особенности: 1) получается приближенное решение с заданной точностью; 2) для его нахождения требуется выполнить большое количество итераций (приближений); 3) требующееся число операций заранее неизвестно.

В случае, когда функция $F(x)=f(x)-g(x)$ является полиномом первой, второй или третьей степени, применяются аналитические методы нахождения точного значения корня. Если же $F(x)$ -- полином четвертой и более высокой степени, тригонометрическая или трансцендентная функция, то используются приближенные (численные) методы. При этом решение состоит из двух этапов: 1) локализация корня, то есть приближенное установление числового интервала, внутри которого он находится; 2) уточнение корня путем уменьшения содержащего его интервала до требуемого значения ε.

Для отделения корней можно использовать теорему: если значения функции $F(x)$ на концах интервала $[a;\ b]$ имеют разные знаки (при этом $F(a)\cdot F(b)<0$), то внутри этого интервала содержатся не менее одного корня уравнения $F(x)=0$.

Наиболее простой способ численного решения уравнения состоит в использовании **метода табуляции.** Его алгоритм заключается в следующем: 1) локализовать корень уравнения $F(x)=0$, то есть найти интервал $[a;\ b]$, внутри которого график функции $y=F(x)$ однократно пересекает ось абсцисс; 2) протабулировать функцию $F(x)=f(x)-g(x)$ при дискретных значениях аргумента $x_i=a+ih$, где $h=\Delta x$ -- шаг изменения аргумента, $i=1,2,3,..$; 3) найти интервал $[x_i, x_{i+1}]$, внутри которого функция $y=F(x)$ пересекает ось абсцисс (при этом значения $F(x_i)$ и $F(x_{i+1})$ имеют противоположные знаки); 4) если ширина интервала $[x_i, x_{i+1}]$ пре-

вышает требуемую точность ε, то считать, что $a = x_i$, $b = x_{i+1}$ и перейти к операции 2, табулируя функцию с меньшим шагом.

Более удобным является **метод половинного деления**. Он состоит в следующем: 1) локализуют корень уравнения $F(x) = 0$, определяя содержащий его интервал $[a;\ b]$; 2) отрезок $[a;\ b]$ делят точкой $c = (a+b)/2$ пополам; 3) если $F(c) = 0$, то значение $x = c$ и есть корень уравнения; если это не так, то из двух отрезков $[a;\ c]$ и $[c;b]$ выбирается тот, на границах которого функция $y = F(x)$ имеет противоположные знаки ($F(a) \cdot F(b) < 0$); 4) если ширина отрезка, содержащего корень, превышает заданную точность ε, то снова повторяют операции 2 и 3, то есть делят выбранный отрезок пополам и выбирают тот, что содержит корень. В противном случае, -- приближенное значение корня равно одному из границ отрезка. В качестве примера рассмотрим решение трансцендентного уравнения $x^2 = e^{-x}$, которое можно записать как $e^{-x} - x^2 = 0$. Соответствующий алгоритм А-1, записанный в псевдокоде, представлен ниже. Существуют и другие методы численного решения подобных уравнений [5, 12]: метод простой итерации, метод касательных, метод секущих, метод парабол и т.д.

Алгоритм А-1

```
НАЧАЛО ПРОГРАММЫ
    a:=0; b:=1; eps:=0.000001
    m: x:=a; y1:=exp(-x)-x*x;
    x:=b; y2:=exp(-x)-x*x;
    c:=(b+a)/2; x:=c; y3:=exp(-x)-x*x;
    ПЕЧАТЬ a, b, c
    ЕСЛИ y1*y3>0 ТО a:=c ИНАЧЕ b:=c;
    ЕСЛИ b-a>eps ТО ПЕРЕЙТИ К МЕТКЕ m;
    ПЕЧАТЬ “КОРЕНЬ ЛЕЖИТ В ИНТЕРВАЛЕ ”, a, b;
КОНЕЦ ПРОГРАММЫ
```

В некоторых случаях создание компьютерной модели предполагает решение системы алгебраических уравнений. Например, расчет цепи постоянного тока из нескольких контуров с источниками ЭДС методом Кирхгофа требует решения системы алгебраических уравнений, число которых равно количеству ветвей. Чтобы рассчитать трехфазную цепь необходимо решить систему из нескольких уравнений в комплексных числах. Для решения систем линейных уравнений также используются **точные** (конечные) и **приближенные итерационные** (бесконечные) методы. Точные методы позволяют получить точное решение с помощью конечного

числа операций. К ним относятся метод Крамера, метод исключения Гаусса, состоящий в приведении матрицы коэффициентов к треугольному виду. Приближенные методы предполагают построение итерационного процесса, который дает последовательность значений, при определенных условиях сходящуюся к точному решению системы. Обычно ограничиваются первыми 100 или 1000 итераций, что позволяет определить приближенное решение с заданной точностью.

Методом Гаусса можно решить любую систему линейных алгебраических уравнений, однако в случаях, когда уравнения имеют достаточно много нулевых коэффициентов, итерационные методы позволяют получить результат за меньшее количество шагов и требуют меньший объем памяти. Для решения системы нелинейных уравнений используются приближенные методы: метод простой итерации, метод Ньютона [12].

2.2. Численное решение обыкновенных дифференциальных уравнений

Поведение динамических систем описывается функциями непрерывного аргумента. Но цифровая ЭВМ обрабатывает информацию дискретно: ее программа состоит из отдельных команд, циклов, подпрограмм, процедур, каждая из которых осуществляет отдельный акт преобразования исходных данных. Выход состоит в применении **метода сеток**, предполагающего дискретизацию области изменения аргументов, в замене функции непрерывного аргумента функцией дискретного аргумента.

Не редко исследуемое явление или процесс описывается **обыкновенным дифференциальным уравнением** (ОДУ), связывающим независимую переменную x, искомую функцию y и несколько ее первых производных: $F(x, y, y', ..., y^{(n)}) = 0$, где n -- порядок старшей производной. Функция $y = f(x)$ называется решением ОДУ, если при ее подстановке диффуравнение превращается в истинное высказывание. Любое ОДУ имеет бесконечно много решений, для выбора искомого необходимо учесть **начальные условия.**

Для численного решения подобных уравнений используется **метод конечных разностей**, который состоит в следующем. Область непрерывного изменения аргументов заменяют сеткой и переходят к функциям дискретного аргумента. Производные, входящие в диффуравнения, заменяют соответствующими им конечно-разностными аппроксимациями, а

интегралы -- суммами с большим, но конечным числом слагаемых, в которых складываются небольшие, но конечные величины. Использование этого метода для решения диффуравнений приводит к тому, что получается система алгебраических уравнений, содержащих значения искомой функции в узлах сетки.

Как известно, производная функции $y = f(x)$ -- это предел отношения приращения функции $\Delta y = f(x+\Delta x) - f(x)$ к приращению аргумента Δx в случае, когда приращение аргумента стремится к 0:

$$y' = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} = \frac{dy}{dx}.$$

При этом дифференциал dx -- бесконечно малое приращение аргумента, а dy -- соответствующая ему линейная часть приращения функции. Производная характеризует быстроту изменения функции и крутизну графика в данной точке x. Она равна тангенсу угла между касательной и положительным направлением оси абсцисс. Частная производная $\partial f / \partial x_1$ функции многих переменных $f(x_1,..., x_n)$ по переменной x_1 в точке $(x_1,..., x_n)$ есть предел приращения функции $f(x_1 + \Delta x_1,..., x_n) - f(x_1,..., x_n)$ к приращению аргумента Δx_1 при фиксированных значениях остальных независимых переменных.

Для численного дифференцирования используется **метод сеток**: области непрерывного изменения аргументов функции $y = f(x_1,..., x_n, \tau)$ заменяют конечным множеством узлов, образующих одномерную или многомерную **пространственно–временную сетку** [2, 3, 5, 15]. От функции непрерывного аргумента переходят к функции дискретного аргумента, приближенно вычисляют ее значения на различных временных слоях, находят производные и интегралы. При этом бесконечно малые приращения функции $y = f(x_1,..., x_n, \tau)$ и приращения ее аргументов заменяются малыми, но конечными разностями.

Пусть задана функция $y = y(x)$. Разобьем интервал $[a;\ b]$ на элементарные отрезки длиной $h = \Delta x$, получив конечное множество узлов сетки $x_i = a + i\Delta x$, где $i = 1,2,...N$, а N -- число узлов. При этом мы переходим от непрерывной области Ω к сетке $\Omega_{\Delta x}$, от функции непрерывного аргумента к функции дискретного аргумента $y_i = y(x_i)$. Запишем для нее ряд Тейлора:

$$y(x) = y(x_i) + y'_i \frac{(x - x_i)}{1!} + y''_i \frac{(x - x_i)^2}{2!} + ... + y_i^{(k)} \frac{(x - x_i)^k}{k!},$$

$$y(x_{i+1}) = y(x_i) + y'(x_i)\frac{h}{1!} + y''(x_i)\frac{h^2}{2!} + \ldots + y^{(k)}(x_i)\frac{h^k}{2!},$$

где $h = \Delta x = x - x_i$. Ограничиваясь первыми двумя слагаемыми в правой части, получаем приближенное равенство: $y(x_{i+1}) = y(x_i) + y'(x_i)\Delta x$. Оно же следует из определения производной:

$$y' = \frac{dy}{dx} = \lim_{\Delta x \to 0}\frac{\Delta y}{\Delta x} \approx \frac{\Delta y}{\Delta x} = \frac{y_{i+1} - y_i}{\Delta x}, \qquad y_{i+1} = y_i + y'(x_i)\Delta x.$$

Левая и правая разностные производные первого порядка в точке с координатой x_i могут быть найдены так (рис. 1.1):

$$y'(x_i)_- = \frac{y(x_i) - y(x_{i-1})}{\Delta x},\ y'(x_i)_+ = \frac{y(x_{i+1}) - y(x_i)}{\Delta x},$$

Центральная разностная производная первого порядка равна (рис. 1.2):

$$y'(x_i) = \frac{y(x_{i+1}) - y(x_{i-1})}{2\Delta x}.$$

Для второй производной получим:

$$y''(x_i) = \frac{y'(x_{i+1}) - y'(x_i)}{\Delta x} = \frac{y(x_{i+1}) - y(x_i)}{\Delta x^2} - \frac{y(x_i) - y(x_{i-1})}{\Delta x^2},$$

$$y''(x_i) = \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{\Delta x^2}.$$

Чем меньше шаг сетки Δx, тем выше точность найденных производных.

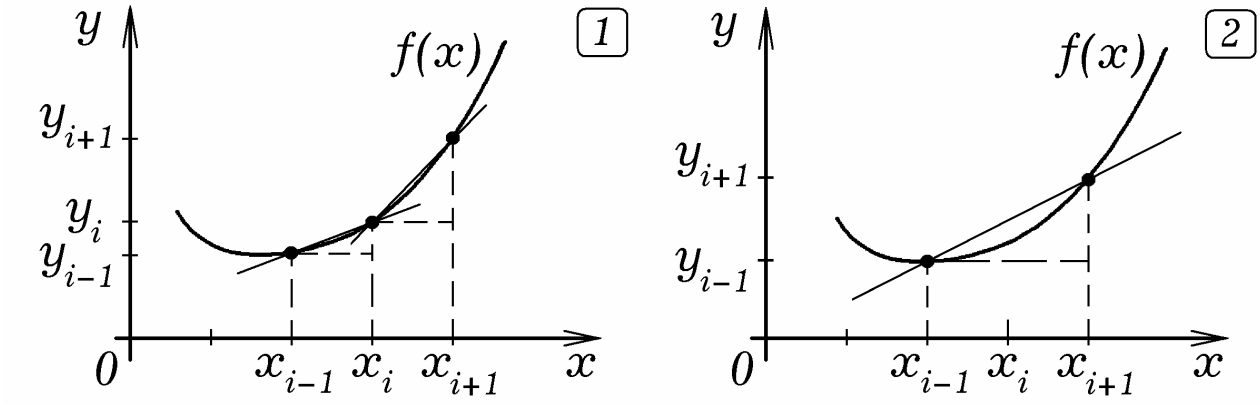

Рис. 1. Разностная аппроксимация первой производной.

Рассмотрим **нестационарную задачу**: по известному начальному состоянию системы в момент τ_0 и закону ее эволюции $x = f(x, \dot{x}, \tau)$ необходимо определить состояние системы $x(\tau)$ в последующие моменты времени τ. В этом случае аргументом является время τ, поэтому временную область дискретизируют сеткой $\tau = \tau_0 + t\Delta\tau$, $t = 0,1,2,\ldots$ Задача решается методом конечных разностей: сначала определяется состояние системы $x_1 = x(\tau_0 + \Delta\tau)$ в момент времени $\tau_0 + \Delta\tau$, затем оно рассматрива-

ется как начальное и устанавливается состояние системы $x_2 = x(\tau_0 + 2\Delta\tau)$ в момент $\tau_0 + 2\Delta\tau$, после этого находится $x_3 = x(\tau_0 + 3\Delta\tau)$ и т.д. Например, при решении ОДУ первого порядка: $\dot{x}(\tau) - f(x(\tau), \tau) = 0$ получаются следующие конечно-разностные уравнения: $(x^{t+1} - x^t)/\Delta\tau = f(x^t, t)$, $x^{t+1} = x^t + f(x^t, t)\Delta\tau$.

Другим методом решения ОДУ является метод Рунге-Кутта. Сущность метода Рунге-Кутта четвертого порядка выражается следующими формулами:

$$y_{i+1} = y_i + (k_1 + 2k_2 + 2k_3 + k_4)/6,$$
$$k_1 = h \cdot f(x_i, y_i), \qquad k_2 = h \cdot f(x_i + \Delta x/2, y_i + k_1/2),$$
$$k_3 = h \cdot f(x_i + \Delta x/2, y_i + k_2/2), \quad k_4 = h \cdot f(x_i + \Delta x, y_i + k_3).$$

Строго обоснование этого метода можно найти в учебнике по численным методам [4, 5].

ОДУ второго порядка $a\ddot{x}(\tau) + b\dot{x}(\tau) + kx(\tau) - f(t) = 0$ приводится к системе из двух дифференциальных уравнений первого порядка:

$$\dot{x}(\tau) - \upsilon(\tau) = 0, \qquad a\dot{\upsilon}(\tau) + b\upsilon(\tau) + kx(\tau) - f(\tau) = 0,$$

которые решаются методами, рассмотренными выше. Можно поступить иначе: все производные заменить конечно-разностными аппроксимациями, а затем выразить x^{t+1}. Например, для уравнения незатухающих колебаний $\ddot{x} + \omega_0^2 x = 0$ получаем: $(x^{t-1} - 2x^t + x^{t+1})/\Delta t^2 + \omega_0^2 x^t = 0$. Отсюда следует: $x^{t+1} = 2x^t - x^{t-1} - \omega_0^2 x^t \Delta\tau^2$, $\upsilon^t = (x^{t+1} - x^{t-1})/(2\Delta\tau)$. Это **метод Верле**. Для начала итерационного процесса следует найти x^{t+1} по формуле: $x^{t+1} = x^t + \upsilon^t \Delta\tau$.

Для интегрирования уравнений движения типа $\ddot{x} = F/m = a_x$ также можно использовать **алгоритм Верле в скоростной форме**, выражающийся уравнениями:

$$x^{t+1} = x^t + \upsilon_x^t \Delta\tau + a_x^t \Delta\tau^2/2, \quad \upsilon_x^{t+1} = \upsilon_x^t + (a_x^{t+1} + a_x^t)\Delta\tau/2,$$

Он является самостартующимся и имеет более высокий порядок точности по времени, чем алгоритм Эйлера. Существуют и другие методы численного решения ОДУ [3, 4, 5, 8, 11, 13, 14]. Среди них уточненный и исправленный методы Эйлера, метод прогонки, метод прогноза и коррекции (предиктор-корректорные методы Адамса), метод пристрелки, методы с автоматическим выбором шага и другие.

2.3. Примеры решения обыкновенных диффуравнений численными методами

Часто моделирование объекта требует решения **задачи Коши**, состоящей в нахождении функции $y = f(x)$ удовлетворяющей некоторому обыкновенному дифференциальному уравнению $F(x, y, y', ..., y^{(n)}) = 0$ и начальным условиям $y(0) = y_0$, $y'(0) = y_1$, $y''(0) = y_2$, …, $y^{(n-1)} = y_{n-1}$. Число начальных условий n равно порядку диффуравнения. Рассмотрим несколько задач, требующих решения обыкновенных дифференциальных уравнений методом конечных разностей.

Задача 1. Движение материальной точки описывается уравнением $x(t) = 3t^3 - 3t^2 + 4$. Определите скорость и ускорение (первую $\dot{x}$ и вторую $\ddot{x}$ производные функции $x(t)$) в момент времени 2 с. Найдите пройденный путь (интеграл I данной функции) за интервал от 1 до 3 с методом трапеций. Решите задачу аналитически и сравните результаты.

Для решения этой задачи используется программа ПР-1, работающая по алгоритму А-2. При уменьшении шага Δt получающиеся значения производной и интеграла стремятся к некоторым предельным значениям, которые совпадают с аналитически найденными значениями производной $\dot{x}$ и интеграла I функции.

Алгоритм А-2

```
ЗАДАТЬ ФУНКЦИЮ Funct:=t*t*t-t*t+3;
----------------------
НАЧАЛО ПРОГРАММЫ
   t:=3; h:=0.001;
   y1:=Funct(t-h); y2:=Funct(t); y3:=Funct(t+h);
   ПЕЧАТЬ "Первая производная ", (y2-y1)/h
   ПЕЧАТЬ "Вторая производная ", (y1-2*y2+y3)/(h*h)
   a:=1; b:=3; t:=a; S:=0;
   ПОВТОРЯТЬ { S:=S+0.5*(Funct(t)+Funct(t+h))*h; t:=t+h; }
   ПОКА НЕ t>b;
   ПЕЧАТЬ "Интеграл  ", S
КОНЕЦ ПРОГРАММЫ
```

При уменьшении шага точность вычислений обычно возрастает. Если же шаг очень мал, то для нахождения интеграла приходится суммировать слишком большое число слагаемых, что приводит к снижению точности и увеличению времени вычислений. Точные значения первых двух производных и интеграла приведены в нижней строчке таблицы.

Таблица 1.

Δt	$\dot{x}$	$\ddot{x}$	I
0,2	21,120	30,000	42,200
0,1	22,530	30,000	48,177
0,01	23,850	30,000	42,584
0,001	23,985	30,000	42,000
0,0001	23,998	30,006	42,000
0,00001	24,000	30,268	42,001
Точное значение	24	30	42

Задача 2. Точка движется по закону: $x = 3\cos(t)$, $y = 2\cos(3{,}7t + 2)$. Вычислите координаты x, y, проекции и модули скорости υ_x, υ_y, υ и ускорения a_x, a_y, a, нормальное a_n и тангенциальное a_τ ускорения в последовательные моменты времени $t = i\Delta t$, $i = 1,2,3,\ldots$.

Используется программа ПР-2. Расчет проекций скорости и ускорения на оси координат осуществляется методом численного дифференцирования: $\upsilon_x = (x - x_0)/\Delta t$, $\upsilon_y = (y - y_0)/\Delta t$,

$$a_x = (\upsilon_x - \upsilon_{x0})/\Delta t, \quad a_y = (\upsilon_y - \upsilon_{y0})/\Delta t,$$

Модули векторов $\vec{\upsilon}$, $\vec{a}$, нормальное и тангенциальное ускорения вычисляются по формулам:

$$\upsilon = \sqrt{\upsilon_x^2 + \upsilon_y^2}, \quad a = \sqrt{a_x^2 + a_y^2}, \quad a_\tau = a\cos\alpha, \quad a_n = \sqrt{a^2 - a_\tau^2},$$

где α -- угол между $\vec{\upsilon}$ и $\vec{a}$, косинус которого равен:

$$\cos\alpha = \frac{a_x\upsilon_x + a_y\upsilon_y}{a\upsilon}.$$

Результаты вычислений выводятся на экран в числовом виде.

Задача 3. Решите дифференциальное уравнение первого порядка $y'_x - \sin(x) = 0$ методом Эйлера. Получите семейство решений, соответствующих различным начальным условиям $y_0 = y(0) = C$.

Запишем уравнение в конечных разностях:

$$\frac{dy}{dx} = \sin(x), \quad \frac{y_{i+1} - y_i}{\Delta x} = \sin(x_i), \quad y_{i+1} = y_i + \sin(x_i)\Delta x.$$

Чтобы численно решить уравнение, необходимо переменной y присвоить значение $y_0 = y(0)$, а затем в цикле рассчитать последующие значения y_i при $i = 1,2,\ldots$ в соответствии с приведенной выше формулой. В программе ПР-3 решается уравнение $y_x' - \sin(x) = 0$ при трех различных

начальных условиях $y(0) = 0,\ 0{,}5,\ 1$. Соответствующий алгоритм А-3 приведен ниже. Получается семейство из трех функций, отличающихся на постоянную величину (рис. 2.1).

Алгоритм А-3

```
ДЛЯ j:=0 ДО 3 ДЕЛАТЬ {-          'Метод Эйлера
  dx:=0.01: y:=.5*j
  ДЛЯ i:=1 ДО 1000 ДЕЛАТЬ {--
    FNN:=sin(x); x:=i*dx; y:=y+FNN*dx;
    ПОСТАВИТЬ ТОЧКУ С КООРДИНАТАМИ (x, y) --}
-}
```

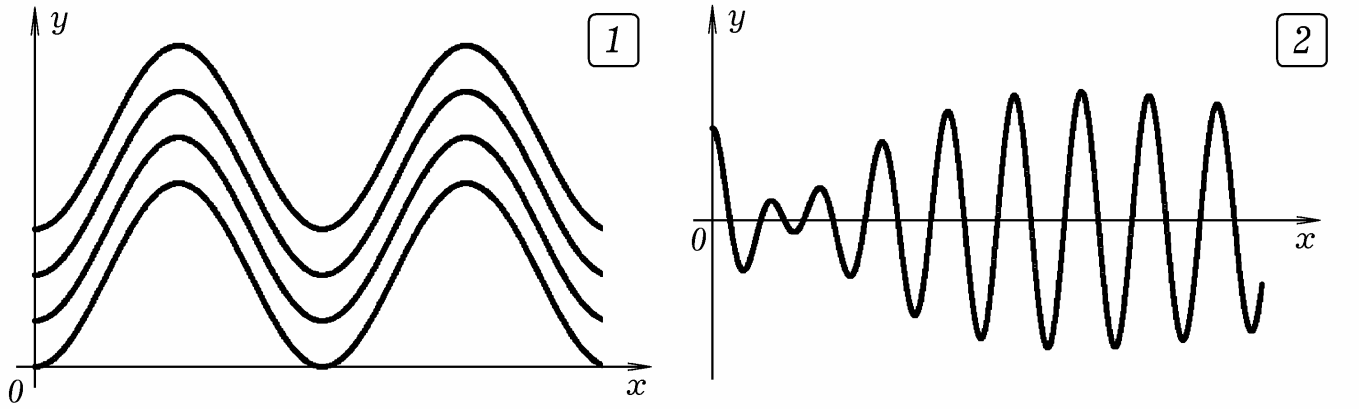

Рис. 2. Результаты решения дифференциальных уравнений (задачи 3 и 4).

Задача 4. Решите обыкновенное дифференциальное уравнение второго порядка $y_x'' - f(x, y, y_x') = 0$ методом Эйлера. Начальные условия $y(0)$, $y_x'(0)$ заданы.

Дифференциальное уравнение второго порядка представимо в виде системы двух ОДУ первого порядка:

$$y_x'' = dy_x'/dx = f(x, y, y_x') = a, \qquad y_x' = dy/dx.$$

Используем схему Эйлера:

$$(y'_{i+1} - y'_i)/\Delta x = f(x, y, y_x') = a, \qquad y'_{i+1} = (y_{i+1} - y_i)/\Delta x.$$

$$y'_{i+1} = y'_i + a\Delta x, \qquad y_{i+1} = y_i + y'_{i+1}\Delta x.$$

В программе Пр-4 создан цикл по i, в котором пересчитываются значения y_i, y_i' и решается уравнение $y_x'' + y_x' + 1{,}2y - 5\sin(x) = 0$. Используемый при этом алгоритм А-4 представлен ниже. Результат решения представлен на рис. 2.2.

Алгоритм А-4

```
y = 20: dx = 0.01                 'Метод Эйлера
ДЛЯ i = 1 ДО 5000 ДЕЛАТЬ {-
    x = x + dx
    pr2y = 5 * SIN(x) - .1 * pr1y - 1.2 * y
    pr1y = pr1y + pr2y * dx; y = y + pr1y * dx
```

```
    ПОСТАВИТЬ ТОЧКУ С КООРДИНАТАМИ (x, y)
-}
```

Задача 5. Решите обыкновенное дифференциальное уравнение вида $y_x' - f(x, y) = 0$ методом Рунге-Кутта четвертого порядка.

Сущность метода изложена в предыдущем параграфе. В качестве примера рассмотрим уравнение $y_x' - 2\sin(x) + 0{,}5/y = 0$. Для его решения используется алгоритм А-5 (программа ПР-5). Результат на рис. 3.1.

Алгоритм А-5

```
ЗАДАТЬ ФУНКЦИЮ FNN(x, y)=2*SIN(x)+.5/y
------------------------------
y = 1: dx = .01
ДЛЯ i = 1 ДО 1500 ДЕЛАТЬ {-
  x = i * dx; k1 = FNN(x, y);
  k2 = FNN(x + dx / 2, y + dx * k1 / 2)
  k3 = FNN(x + dx / 2, y + dx * k1 / 2)
  k4 = FNN(x + dx, y + k3)
  y = y + dx / 6 * (k1 + 2 * k2 + 2 * k3 + k4)
  ПОСТАВИТЬ ТОЧКУ С КООРДИНАТАМИ (x, y)
-}
```

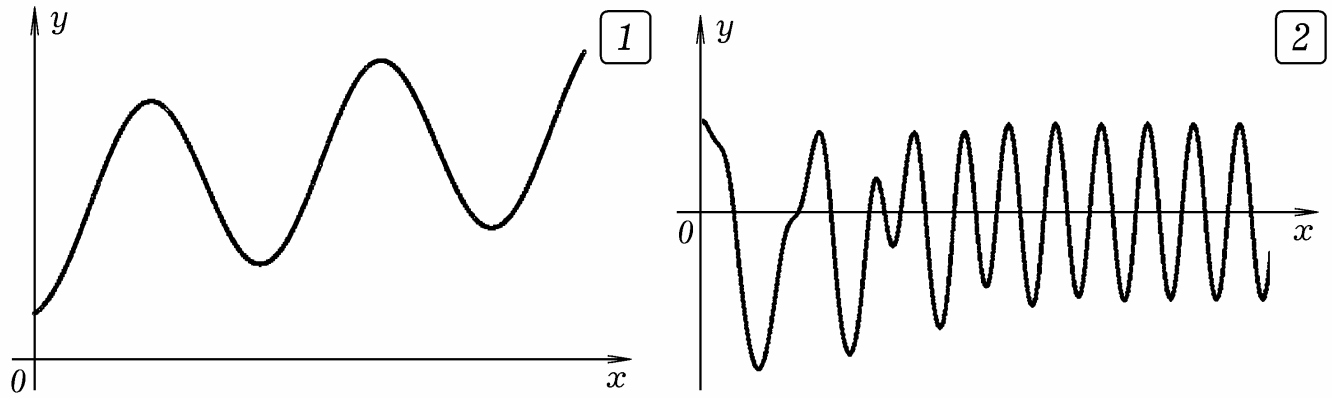

Рис. 3. Результаты решения диффуравнений (задачи 3 и 4).

Задача 6. Решите дифференциальное уравнение второго порядка $y_x'' - f(x, y, y_x') = 0$ методом Рунге-Кутта четвертого порядка. Начальные условия $y(0)$, $y_x'(0)$ заданы.

Один из способов состоит в следующем. Записывают данное уравнение в виде системы двух диффуравнений первого порядка и используют рассмотренную выше схему Рунге-Кутта. Программа ПР-6 решает уравнение $y_x'' + 0{,}25y_x' + 0{,}9\sin(y) = \sin(x)$, результат -- на рис. 3.2. Применяемый алгоритм А-6 приведен ниже.

Алгоритм А-6

```
ЗАДАТЬ ФУНКЦИЮ FNN1(x,y,pr1)= SIN(x)-.25*pr1-.9*SIN(y)
y = 2: v = 0: dx = .01
```

```
ДЛЯ i = 1 ДО 8000 ДЕЛАТЬ {-
   x = i * dx: k1 = FNN1(x, y, pr1)
   k2 = FNN1(x + dx / 2, y, pr1 + dx * k1 / 2)
   k3 = FNN1(x + dx / 2, y, pr1 + dx * k2 / 2)
   k4 = FNN1(x + dx, y, pr1 + k3)
   pr1 = pr1 + dx / 6 * (k1 + 2 * k2 + 2 * k3 + k4)
   y = y + pr1 * dx
   ПОСТАВИТЬ ТОЧКУ С КООРДИНАТАМИ (x, y)
-}
```

Задача 7. Решите дифференциальное уравнение $m\ddot{x} + r\dot{x} + kx = 0$ методом Эйлера с полушагом.

Это однородное ДУ описывает затухающие колебания. Сущность метода Эйлера с полушагом в том, что для нахождения x, $\upsilon = \dot{x}$, $a = \ddot{x}$ в момент $t + \Delta t$ сначала определяют эти величины на середине интервала $[t; t + \Delta t]$ в момент $t + \Delta t/2$, а затем используют найденные значения. Конечно-разностные уравнения выглядят так:

$$a^t = -(kx^t - r\upsilon^t)/m,\ x^{t+1/2} = x^t + \upsilon^t \Delta\tau/2,\ \upsilon^{t+1/2} = \upsilon^t + a^t\Delta\tau/2,$$

$$a^{t+1/2} = -(kx^{t+1/2} - r\upsilon^{t+1/2})/m,\ x^{t+1} = x^t + \upsilon^{t+1/2}\Delta\tau,\ \upsilon^{t+1} = \upsilon^t + a^{t+1/2}\Delta\tau.$$

В программе ПР-7 эта задача решается методом Эйлера (график -- непрерывная линия) и методом Эйлера с полушагом (график рисуется точками). Сравнивая эти решения можно обнаружить, что при увеличении шага до $0{,}03$, а также при больших t решения не совпадают.

2.4. Краевые задачи для обыкновенных дифференциальных уравнений

В некоторых случаях возникает необходимость решения краевой задачи для обыкновенного диффуравнения $F(x, y, y', ..., y^{(n)}) = 0$. Она состоит в нахождении функции $y = f(x)$ удовлетворяющей ОДУ и нескольким краевым условиям вида $y(a) = A$, $y'(b) = B$ и т.д. В общем случае ОДУ и краевые условия могут быть нелинейными.

Одним из методов решения краевой задачи является **метод стрельбы**, который состоит в следующем. Исходя из начальных условий и нулевого приближения φ_0 искомого параметра, находят состояние системы в последующие моменты времени $\tau_0 + i\Delta\tau$ (делают первый выстрел). Определяют, как сильно отличается состояние системы $S(\varphi_0, t')$ в момент t' от состояния, задаваемого краевым условием $y(t') = A$. Увеличивают па-

раметр φ на $\Delta\varphi$ и повторяют расчеты при $\varphi_1 = \varphi_0 + \Delta\varphi$ (второй выстрел), снова определяя насколько приблизилось состояние $S(\varphi_1, t')$ к $y(t') = A$. Если $|S(\varphi_1, t') - y(t')|$ уменьшилось, то продолжают увеличивать искомый параметр φ, каждый раз отслеживая величину $|S(\varphi_1, t') - y(t')|$ до тех пор, пока она не начнет увеличиваться. Значение φ_i, при котором она минимальна и есть решение задачи с точностью $\Delta\varphi$. При необходимости можно повторить эту процедуру с меньшим шагом $\Delta\varphi$. В качестве примера рассмотрим задачу о поражении цели баллистическим снарядом.

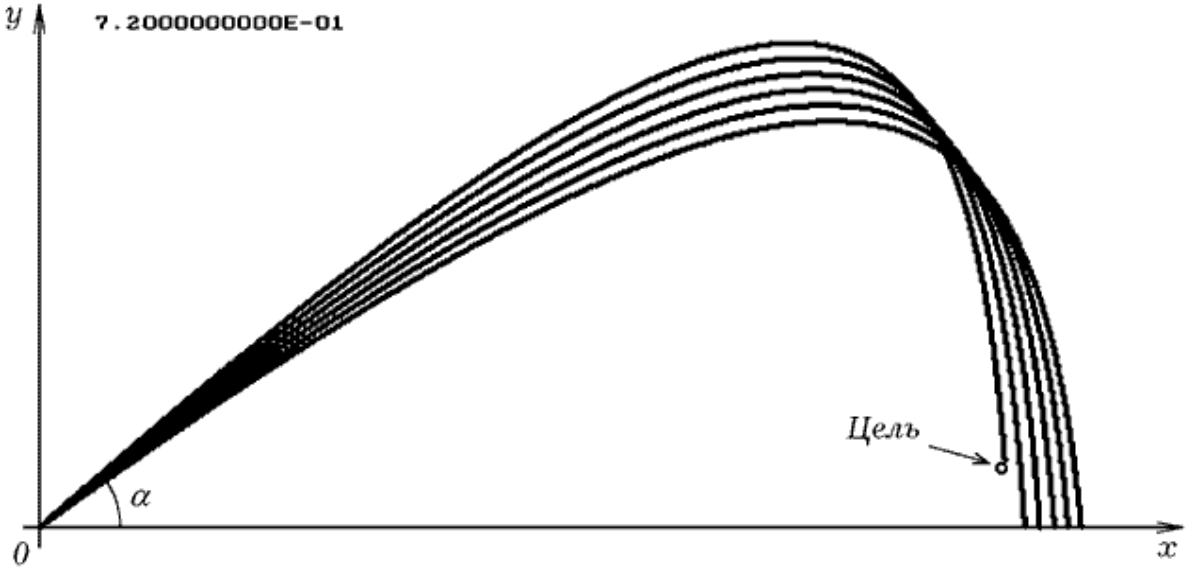

Рис. 4. Определение направления ствола методом стрельбы.

Задача 1. Из пушки вылетает снаряд с известной скоростью. Определить угол α под которым необходимо произвести выстрел для того, чтобы снаряд попал в цель с координатами x_1, y_1. Сила сопротивления, действующая на снаряд, пропорциональна его скорости.

При движении снаряда на него со стороны воздуха действует сила сопротивления $\vec{F} = -r\vec{v}$. Запишем второй закон Ньютона $m\vec{g} - r\vec{v} = m\vec{a}$ в конечных разностях, используя алгоритм Верле в скоростной форме:

$$a_x^{t+1} = -r v_x^t / m, \qquad a_y^{t+1} = -g - r v_y^t / m,$$

$$x^{t+1} = x^t + v_x^t \Delta\tau + a_x^t \Delta\tau^2 / 2, \quad v_x^{t+1} = v_x^t + (a_x^{t+1} + a_x^t)\Delta\tau / 2,$$

$$y^{t+1} = y^t + v_y^t \Delta\tau + a_y^t \Delta\tau^2 / 2, \quad v_y^{t+1} = v_y^t + (a_y^{t+1} + a_y^t)\Delta\tau / 2,$$

Для решения задачи используется метод стрельбы: в программе ПР-8 определяется траектория движения снаряда при заданной начальной скорости и произвольном угле α между стволом орудия и горизонталью. Если снаряд не поразил цель, то угол α увеличивают на некоторый шаг

$\Delta\alpha$ и повторяют расчет, затем еще раз и еще раз до тех пор, пока снаряд не попадет в цель. Искомое значение угла в радианах выводится на экран компьютера. Задача имеет два решения. На рис. 4 показан результат нахождения одного из них.

2.5. Методы численного интегрирования

Многие задачи сводятся к вычислению интеграла -- суммы бесконечно большого числа бесконечно малых величин. Он численно равен площади криволинейной трапеции, ограниченной графиком интегрируемой функции $y = f(x)$ и пределами интегрирования a и b. При использовании **метода прямоугольников** эту криволинейную трапецию разбивают на N прямоугольных полосок шириной $\Delta x = h = (b-a)/N$, длина каждой из которых равна $y_i = y(a + i\Delta x)$. Элементарная площадь полоски $\Delta S_i = y(x_i)\Delta x$. Искомая площадь S равна сумме:

$$S = \lim_{N\to\infty} \sum_{i=1}^{N} y(x_i)\Delta x = \int_a^b y(x)dx.$$

Для численного нахождения интеграла достаточно определить сумму:

$$S = \sum_{i=1}^{N} y(x_i)\Delta x.$$

Чем меньше шаг $h = \Delta x$ и больше N, тем точнее результат вычислений.

Более точное значение интеграла можно найти **методом трапеций**. Он заключается в том, что каждая i-ая полоска заменяется трапецией высотой $h = \Delta x$ с основаниями $y_i = y(a + i\Delta x)$ и $y_{i+1} = y(a + (i+1)\Delta x)$, поэтому ее площадь равна $\Delta S_i = (y_i + y_{i+1})\Delta x/2$. Интеграл равен сумме элементарных площадей всех трапецевидных полосок:

$$S = \sum_{i=1}^{N} \frac{y(x_i) + y(x_{i+1})}{2}\Delta x.$$

Задача 1. Имеется неоднородная пластина, ограниченная функцией $y = \sqrt{x}$, осью абсцисс, прямой $x = 2$, плотность которой зависит от координаты: $\rho(x, y) = 2 + 0{,}4y + 0{,}2x^2$. Найдите момент инерции пластины относительно оси Ox.

Вычисление момента инерции тела сводится к интегрированию по объему и нахождению суммы элементарных моментов инерции. Разобьем

тело на элементарные объемы $dV = h \cdot dx \cdot dy$ массами $dm = \rho \cdot dV$, имеющими моменты инерции относительно оси Ox $dI = y^2 dm$ (рис. 5). Для нахождения суммы всех элементарных моментов инерции dI организуют два вложенных цикла, в которых перебираются и суммируются все dI (программа ПР-9).

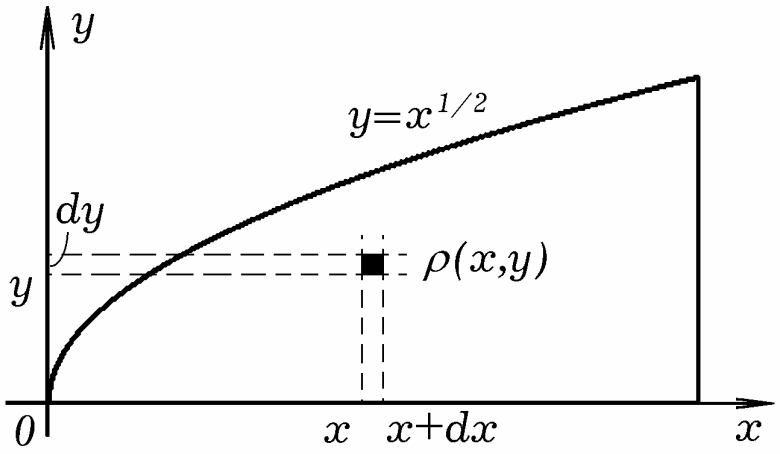

Рис. 5. Нахождение момента инерции неоднородной пластины.

Задача 2. Определите силу гравитационного притяжения, действующую со стороны шарообразного тела радиуса R на материальную точку массой m, находящуюся на расстоянии z' от центра. Плотность шара $\rho(r) = 100/r$, где r -- расстояние от его центра O (рис. 6.1).

Распределение плотности шарообразного тела, а значит и его гравитационное поле обладают центральной симметрией, поэтому задачу следует решать в сферической системе координат. Выберем элементарный объем $\Delta V = r^2 \Delta r \Delta\varphi \Delta\theta$, его масса $\Delta m = \rho(r)\Delta V$, а координаты $x = r\sin\theta\cos\varphi$, $y = r\sin\theta\sin\varphi$, $z = r\cos\theta$. Расстояние от элементарной массы Δm_i до точки m равно $d_i = \sqrt{x_i^2 + y_i^2 + (z' - z_i)^2}$, между ними действует сила притяжения: $\Delta F_i = Gm\Delta m_i / d_i^2$. Так как вектора $\Delta\vec{F}_i$ симметричны относительно оси Oz и образуют с ней угол α_i, то результирующая сила направлена противоположно оси Oz и равна:

$$F = F_z = \sum_{i=1}^{n} \Delta F_i \cos\alpha_i = \sum_{i=1}^{n} \Delta F_i \frac{l_i}{d_i} = \sum_{i=1}^{n} G\frac{m\Delta m_i}{d_i^2}\frac{l_i}{d_i}.$$

где $i = 1,2,...,n$, n -- число элементарных масс Δm_i, $l_i = z' - r\cos\theta_i$. Программа ПР-10 для расчета искомой силы F содержит вложенные друг в друга цикл по r, по φ и по θ, позволяющие перебрать все элементарные массы тела, рассчитать и просуммировать проекции сил $\Delta\vec{F}_i$ на ось Oz.

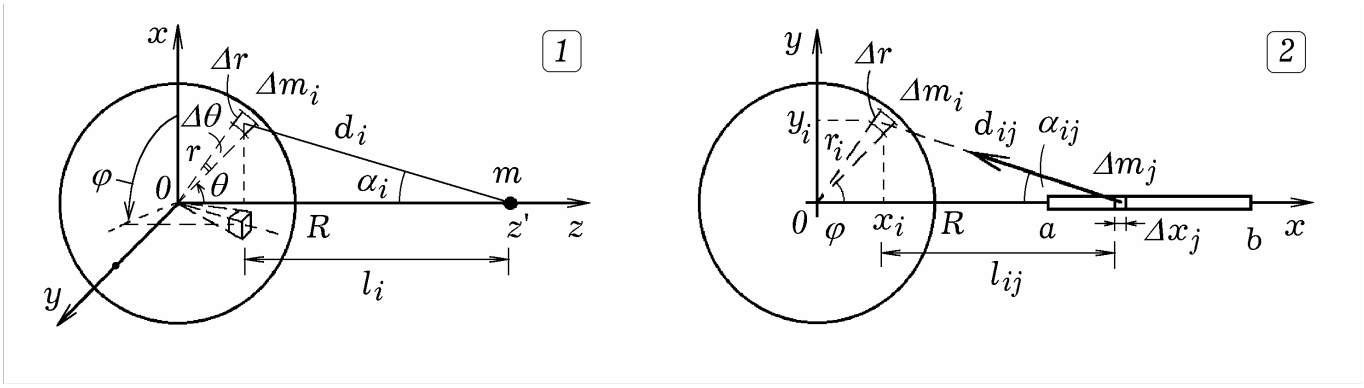

Рис. 6. Нахождение силы притяжения протяженных тел.

Задача 3. Определите силу гравитационного притяжения, действующую между неоднородным диском радиусом R, толщиной h и однородным стержнем массой m_1, концы которого имеют координаты a и $b > R$. Зависимость плотности диска от координаты: $\rho(r) = 100/r$, где r - - расстояние от его центра O.

Распределение массы обладает осевой симметрией, будем использовать полярную и декартовую системы координат (рис. 6.2). Разобьем диск на n элементарных объемов $\Delta V = rh\Delta\varphi\Delta r$ с координатами $x_i = r\cos\varphi$, $y_i = r\sin\varphi$ и массами $\Delta m = \rho(r)\Delta V = \rho(r)rh\Delta\varphi\Delta r$. Стержень рассмотрим как k элементарных масс $\Delta m_1 = m_1/k$ с координатами $x_{1j} = a + j\Delta x$, где $j = 1,2,...,k$. Расстояние между Δm_i и Δm_{1j} равно $d_{ij} = \sqrt{(x_i + x_{1j})^2 + y_i^2}$. Проекция силы $\Delta\vec{F}_{ij}$ на ось Oz равна

$$\Delta F_{ij,z} = \Delta F_{ij}\cos\alpha_{ij} = G\frac{m_i\Delta m_{1j}}{d_{ij}^2}\frac{l_{ij}}{d_{ij}},$$

так как $\cos\alpha_{ij} = l_{ij}/d_{ij}$, $l_{ij} = x_{1j} - r_i\cos\varphi_i$. В используемой программе ПР-11 перебираются все элементарные массы Δm_i и Δm_{1j} обоих тел, определяются и суммируются проекции сил притяжения на ось Oz.

Приложение

В приложении представлены тексты программ, позволяющие решить рассмотренные выше задачи. Они написанные в средах MS-DOS Qbasic 1.0, Borland Pascal 7.0, Free Pascal 1.0.10.

ПР-1.

```
uses crt; var  x,y1,y2,y3,a,b,h,S :real;
Function Funct(x:real):real;                        {Pascal}
begin  Funct:=3*x*x*x-3*x*x+4; end;
BEGIN  clrscr; x:=2; h:=0.001;
   y1:=Funct(x-h); y2:=Funct(x); y3:=Funct(x+h);
   Writeln('Первая производная  ', (y2-y1)/h:3:3);
   Writeln('Вторая производная  ', (y1-2*y2+y3)/(h*h):3:3);
   a:=1; b:=3; x:=a; S:=0;
   Repeat S:=S+0.5*(Funct(x)+Funct(x+h))*h; x:=x+h;
   until x>b;   Writeln('Интеграл  ',S:3:3);
   Repeat until KeyPressed;
END.
```

ПР-2.

```
dt = .2: FOR t = 0 TO 10 STEP dt     'QBASIC
x0 = x: y0 = y: vx0 = vx: vy0 = vy
x = 3 * COS(t): y = 2 * COS(3.7 * t + 2)
vx = (x - x0) / dt: vy = (y - y0) / dt
ax = (vx - vx0) / dt: ay = (vy - vy0) / dt
vv = SQR(vx * vx + vy * vy): aa = SQR(ax * ax + ay * ay)
cosa = (ax * vx + ay * vy) / (aa * vv): at = aa * cosa:
IF at < aa THEN an = SQR(aa * aa - (at * at))
PRINT t; x; y; vv; aa; at; an
NEXT
END
```

ПР-3.

```
SCREEN 12         'QBASIC
LINE (0, 400)-(640, 400): LINE (20, 0)-(20, 480)
'DEF FNN (x, y) = -y + 3
'DEF FNN (x, y) = -y - .5 * x + 4
DEF FNN (x, y) = SIN(x)
FOR j = 0 TO 3
dx = .01: y = .5 * j
FOR i = 1 TO 1000
x = i * dx: y = y + FNN(x, y) * dx
CIRCLE (20 + 50 * x, 400 - 100 * y), 2
NEXT: NEXT
END
```

ПР-4

```
SCREEN 12          'QBASIC
LINE (0, 240)-(640, 240): LINE (20, 0)-(20, 480)
y = 20: dx = .01
FOR i = 1 TO 5000
x = x + dx
```

```
pr2y = 5 * SIN(x) - .1 * pr1y - 1.2 * y
pr1y = pr1y + pr2y * dx : y = y + pr1y * dx
CIRCLE (20 + 12 * x, 240 - 5 * y), 2
NEXT
END
```

ПР-5.

```
SCREEN 12         'QBASIC
LINE (0, 400)-(640, 400)
LINE (20, 0)-(20, 480)
DEF FNN (x, y) = 2 * SIN(x) + .5 / y
'DEF FNN (x, y) = -y * y + .2 * x
y = 1: dx = .01
FOR i = 1 TO 1500
x = i * dx: k1 = FNN(x, y)
k2 = FNN(x + dx / 2, y + dx * k1 / 2)
k3 = FNN(x + dx / 2, y + dx * k1 / 2)
k4 = FNN(x + dx, y + k3)
y = y + dx / 6 * (k1 + 2 * k2 + 2 * k3 + k4)
CIRCLE (20 + 40 * x, 400 - 50 * y), 2
NEXT
END
```

ПР-6.

```
SCREEN 12     'QBASIC
LINE (0, 240)-(640, 240): LINE (20, 0)-(20, 480)
DEF FNN1 (x, y, pr1) = SIN(x) - .25 * pr1 - .9 * SIN(y)
y = 2: v = 0: dx = .01
FOR i = 1 TO 8000
x = i * dx: k1 = FNN1(x, y, pr1)
k2 = FNN1(x + dx / 2, y, pr1 + dx * k1 / 2)
k3 = FNN1(x + dx / 2, y, pr1 + dx * k2 / 2)
k4 = FNN1(x + dx, y, pr1 + k3)
pr1 = pr1 + dx / 6 * (k1 + 2 * k2 + 2 * k3 + k4)
y = y + pr1 * dx
CIRCLE (20 + 8 * x, 240 - 50 * y), 1
NEXT
END
```

ПР-7.

```
Uses crt, graph;
Const dt=0.01; k=40; m=0.5; r=0.05;                {Pascal}
Var x,xx,v,a,xp,vp,ap,tp,t,x1,v1,a1 : real;
    DV,MV : integer;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
x:=1; x1:=1;
Repeat xx:=x; a:=-(k*x+r*v)/m; tp:=t+dt/2;
xp:=x+v*dt/2; vp:=v+a*dt/2;
ap:=-(k*xp+r*vp)/m; v:=v+ap*dt; x:=x+vp*dt; t:=tp+dt/2;
```

```
line(round(t*40),240-round(150*x),
                    round((t-dt)*40),240-round(150*xx));
a1:=-(k*x1+r*v1)/m; v1:=v1+a1*dt; x1:=x1+v1*dt;
circle(round(t*40),240-round(150*x1),1);
until KeyPressed; CloseGraph;
END.
```

ТР-8.

```
uses crt, graph;
var x,y,vx,vy,ax,ay,axp,ayp,v,Fx,Fy,alfa : real;   {Pascal}
    Gd, Gm, i: integer; ugol: string; Label metka;
const m=100; dt=0.005; r=1.5; xc=490; yc=30;
BEGIN Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
  Line(10,0,10,450); Line(0,450,640,450);
  Circle(xc+10,450-yc,2); v:=10;
  For i:=1 to 40 do begin x:=0; y:=0;
      alfa:=0.6+0.02*i; str(alfa,ugol);
      vx:=v*cos(alfa); vy:=v*sin(alfa);
      Repeat Fy:=-3; Fx:=0; axp:=ax; ayp:=ay;
        ax:=(Fx-r*vx)/m; ay:=(Fy-r*vy)/m;
        vx:=vx+(ax+axp)*dt/2; vy:=vy+(ay+ayp)*dt/2;
        x:=x+vx*dt+axp*dt*dt/2; y:=y+vy*dt+ayp*dt*dt/2;
        Circle(round(x)+10,450-round(y),1);
        If sqr(x-xc)+sqr(y-yc)<16 then goto metka;
      until (y<0)or(KeyPressed); end;
  metka: OutTextXY(10,10,ugol);
  Repeat until KeyPressed; CloseGraph;
END.
```

ТР-9.

```
dx = .05: dy = .05: h = .02  'QBASIC
FOR x = 0 TO 2 STEP dy
WHILE y < SQR(x)
y = y + dy: rho = 2 + .4 * y + .2 * x * x
dV = dx * dy * h: I = I + rho * dV * y * y
WEND: NEXT: PRINT I
END
```

ТР-10.

```
CLS: dr = .1: dtheta = .1: dfi = .2   'QBASIC
m = 1: rho1 = 100: z1 = 20
FOR r = 0 TO 3 STEP dr
FOR theta = 0 TO 3.14 STEP dtheta
FOR fi = 0 TO 6.28 STEP dfi
IF r < 1 THEN rho = 100 ELSE rho = 100 / r
z = r * COS(theta): y = r * SIN(theta) * SIN(fi)
x = r * SIN(theta) * COS(fi)
dm = rho * r * r * dtheta * dfi * dr
rast = SQR(x * x + y * y + (z1 - z) * (z1 - z))
```

```
l = z1 - r * COS(theta)
F = F + m * dm * l/(rast * rast * rast)
NEXT: NEXT: PRINT F, r: NEXT
END
```

ПР-11.

```
CLS: dr = .1: dfi = .05: rho1 = 100   'QBASIC
dx1 = .1: dm1 = .2 * dx1: h = .1
FOR x1 = 5 TO 6 STEP dx1
FOR r = 0 TO 3 STEP dr
FOR fi = 0 TO 6.28 STEP dfi
IF r < 1 THEN rho = 100 ELSE rho = 100 / r
x = r * COS(fi): y = r * SIN(fi)
dm = rho * h * r * dr * dfi
rast = SQR((x - x1) * (x - x1) + y * y)
l = x1 - r * COS(fi)
F = F + dm1 * dm * l / (rast * rast * rast)
NEXT: NEXT: PRINT F, r, x1: NEXT
END
```

Литература

1. Булавин Л.А., Выгорницкий Н.В., Лебовка Н.И. Компьютерное моделирование физических систем. –– Долгопрудный: Издательский Дом "Интеллект", 2011. – 352 с.
2. Дульнев Г.Н., Парфенов В.Г., Сигалов А.В. Применение ЭВМ для решения задач теплообмена: Учеб. пособие для теплофизич. и теплоэнергетич. спец. вузов. –– М.: Высш. шк., 1990. –– 207 с.
3. Иванов В.В. Методы вычислений на ЭВМ: Справочное пособие. –– Киев: Наук. думка, 1986. –– 584 с.
4. Ильина В.А., Силаев П.К. Численные методы для физиков – теоретиков. II. –– Москва–Ижевск: Институт компьютерных исследований, 2004. –– 118 с.
5. Калиткин Н.Н. Численные методы. -- БХВ-Петербург, 2011. -- 592 с.
6. Кунин С. Вычислительная физика. –– М.: Мир, 1992. –– 518 с.
7. Компьютеры, модели, вычислительный эксперимент. Введение в информатику с позиций математического моделирования.–– М.: Наука, 1988. – 176 с.
8. Мудров А.Е. Численные методы для ПЭВМ на языках Бейсик, Фортран и Паскаль. –– Томск: МП "РАСКО", 1991. –– 272 с.

9. Паничев В.В., Соловьев Н.А. Компьютерное моделирование: учебное пособие. –– Оренбург: ГОУ ОГУ, 2008. -- 130 с.
10..Поттер Д. Вычислительные методы в физике. –– М.: Мир, 1975. –– 392 с.
11. Ракитин В.И., Первушин В.Е. Практическое руководство по методам вычислений с приложением программ для персональных компьютеров: Учеб. пособие. –– М.: Высшая. шк., 1998. –– 383 с.
12. Ращиков В.С., Рошаль А.С. Численные методы решения физических задач: Учебное пособие. –– СПб: Издательство "Лань", 2005. –– 208 с.
13. Рябенький В.С. Введение в вычислительную математику: Учеб. пособие. –– М.: Физматлит, 2000. –– 296 с.
14. Самарский А.А., Вабищевич П.Н., Самарская Е.А. Задачи и упражнения по численным методам: Учеб. пособие. –– М.: Эдиторал УРСС, 2000. –– 208 с.
15. Самарский А.А., Михайлов А.П. Математическое моделирование: Идеи. Методы. Примеры. –– М.: Физматлит, 2001. –– 320 с.
16. Федоренко Р.П. Введение в вычислительную физику: Учеб. пособие: Для вузов. –– М.: Изд–во Моск. физ.–техн. ин–та, 1994. –– 528 с.