

Глава 7

ДВИЖЕНИЕ СИСТЕМЫ ЧАСТИЦ В СИЛОВОМ ПОЛЕ

Механическая система представляет собой совокупность материальных точек, взаимодействующих друг с другом и окружающими телами. Основная задача механики заключается в расчете движения точек системы, исходя из их масс и действующих на них сил. Эта глава посвящена компьютерному моделированию систем, состоящих из одной или нескольких частиц, взаимодействующих друг с другом и движущихся во внешнем силовом поле [1-10].

7.1. Одномерное движение точки

Простейшей механической системой является материальная точка, которая под действием некоторой силы движется вдоль прямой (например, оси Ox). Если считать, что движение происходит в вязкой среде, а масса частицы и действующая на нее сила зависят от координат и времени, то задача становится достаточно сложной.

Пусть частица массой m движется вдоль оси Ox под действием силы $F_x(x, \tau)$. На частицу также действует сила сопротивления, которая пропорциональна скорости: $F_{cx} = -r v_x$. В начальный момент точка имеет координату x_0 и скорость v_0 . Необходимо определить координату точки $x(\tau)$, проекции ее скорости $v_x(\tau)$ и ускорения $a_x(\tau)$ в следующие моменты времени [4; 6]. Из второго закона Ньютона $ma_x = F_x(x, \tau) + F_{cx}$ получаем дифференциальное уравнение второго порядка: $m\ddot{x} + r\dot{x} = F_x(x, \tau)$. Характер движения физической системы зависит от ее параметров, начальных условий и действующей на нее внешней силы. В этом случае возможны следующие

щие ситуации: 1) внешняя сила отсутствует; 2) внешняя сила постоянна; 3) внешняя сила изменяется по некоторому закону $F_x = F_x(x, \tau)$.

Будем использовать **метод сеток**, для этого перейдем от непрерывной области $\Omega_{x,\tau}$ к дискретной области $\Omega_{\Delta x, \Delta \tau}$. В соответствии с методом Эйлера заменим бесконечно малые приращения функции $x(\tau)$ и ее первые две производные $v_x = \dot{x}(\tau)$, $a_x = \ddot{x}(\tau)$ их конечно-разностными аппроксимациями. Исходя из параметров системы (m и r), координаты $x(\tau)$ и скорости $v_x(\tau)$ частицы в момент τ , рассчитываются ее координата и скорость в следующий момент $\tau + \Delta\tau$ (дискретный момент $t+1$): $a_x^{t+1} = (F_x^t - r v_x^t) / m$, $v_x^{t+1} = v_x^t + a_x^{t+1} \Delta\tau$, $x^{t+1} = x^t + v_x^{t+1} \Delta\tau$. Это состояние рассматривается как исходное, и процедура расчета повторяется для момента времени $\tau + 2\Delta\tau$ и т. д. В одном цикле вместе с вычислениями строятся графики $x(\tau)$, $v_x(\tau)$, $a_x(\tau)$. Проанализируем несколько задач.

Задача 1. Первые $\tau_1 = 8$ с на точку массой $m = 1$ кг действовала сила $F_{1x}(\tau) = 1 \cdot \tau$ Н, до момента $\tau_2 = 30$ с - сила $F_{2x} = 0,2$ Н, затем движение происходило под действием силы $F_{3x} = -4$ Н. Постройте графики зависимостей координаты, скорости и ускорения от времени, если начальные координата и скорость точки заданы: $x_0 = -200$ м, $v_{0x} = -15$ м/с.

Программа ПР-1 (Приложение) содержит цикл по времени, в котором, исходя из известного ускорения, определяется скорость и координата точки. Результат решения задачи представлен на рис. 7.1.1. Так как $v = \dot{x}$ и $a = \dot{v}$, то в моменты времени, когда скорость (ускорение) обращается в нуль, координата (скорость) достигает экстремума. При $\tau < \tau_1$ с точка движется с ускорением, увеличивающимся пропорционально времени, скорость возрастает пропорционально квадрату τ . На втором этапе движение равноускоренное (скорость растет пропорционально τ). При $\tau > \tau_2$ движение равноускоренное, скорость уменьшается до 0, после чего точка начинает двигаться в противоположном направлении.

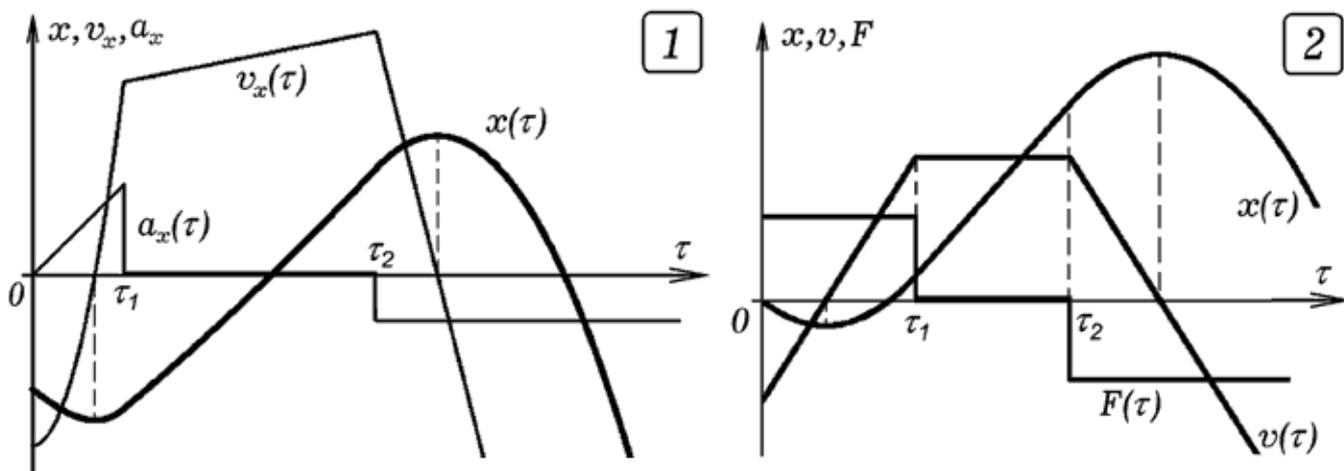


Рис. 7.1. Моделирование движения точки по прямой

На рис. 7.1.2 представлен результат моделирования движения точки массы m , на которую действует скачкообразно изменяющаяся сила: $F_x(\tau) = F$, при $0 < \tau \leq \tau_1$; $F_x = 0$, при $\tau_1 < \tau \leq \tau_2$; $F_x(\tau) = -F$, при $\tau_2 < \tau$.

Задача 2. Промоделируйте движение камня, брошенного вертикально вверх, с учетом сопротивления воздуха. Докажите, что время подъема камня меньше времени спуска. Повторите вычислительный эксперимент при других значениях коэффициента сопротивления r .

Ось Ox направим вниз, начальные условия: $x_0 = 0$, $v_{0x} < 0$. В программе ПР-2 надо правильно задать силу, коэффициент сопротивления $r = 0,5$ и начальную скорость. Получаются графики $x(\tau)$, $v_x(\tau)$, $a_x(\tau)$ (рис. 7.2.1). Программа позволяет определить время подъема (до момента τ_1 , когда скорость сменит свой знак) и общее время полета (до момента τ_2 , когда координата x сменит знак). Из рис. 7.2.1 видно, что $\tau_1 < \tau_2 - \tau_1$.

Задача 3. К шкиву массой m радиуса R , установленному на горизонтальной оси, с помощью невесомого стержня длиной l прикреплен груз массы m_1 (рис. 7.2.2). На шкив намотана нить, к концу которой привязано тело массой m_2 . При $\tau = 0$ система покоилась, $\varphi = 0$. Рассчитайте

угловое ускорение, скорость и перемещение шкива в последующие моменты времени, если тормозящий момент пропорционален скорости.

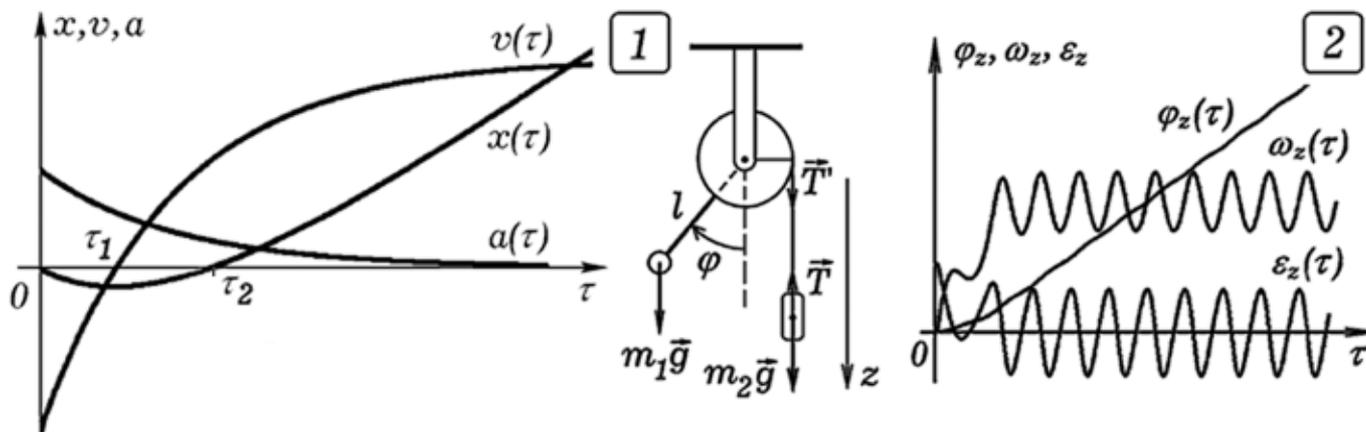


Рис. 7.2. Движение систем с одной степенью свободы

Из законов динамики: $I\varepsilon = TR - m_1gl \sin \varphi - r\omega$, $T = m_2g - m_2a$, где $I = mR^2 / 2 + m_1l^2$ - момент инерции шкива с грузом, $a = R\varepsilon$ - ускорение тела m_2 . Отсюда: $I\varepsilon = m_2gR - m_2R^2\varepsilon - m_1gl \sin \varphi - r\omega$. Угловое ускорение равно: $\varepsilon = (m_2gR - m_1gl \sin \varphi - r\omega) / (I + m_2R^2)$. Для моделирования этого движения на ПЭВМ создают цикл по времени τ , в котором рассчитываются угловое ускорение ε , скорость ω и перемещение φ . Из рис. 7.2.2 видно, что движение неравноускоренное, ускорение ε и скорость ω колеблются относительно некоторых значений, угловое перемещение φ шкива возрастает. Если масса тела m_2 мала, то шкив не совершит ни одного оборота, в системе возникнут затухающие колебания.

Задача 4. Рассчитайте движение ракеты, удаляющейся по прямой от Земли (рис. 7.3.1), если масса горючего уменьшается со скоростью $b = \Delta m / \Delta \tau = const$. Масса корпуса равна m_0 , масса заправленной ракеты $m > m_0$. Скорость вытекания газов относительно ракеты равна $v_{отн}$. Постройте графики $x(\tau)$, $v_x(\tau)$, $m(\tau)$.

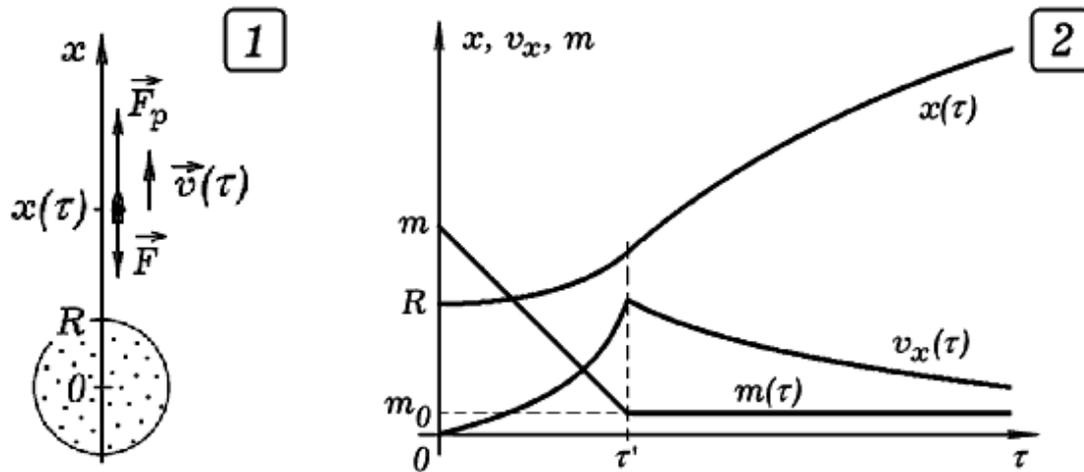


Рис. 7.3. Расчет движения ракеты, удаляющейся от Земли

Программа должна содержать цикл по времени, в котором вычисляются масса ракеты m , реактивная сила $F_p = v_{отн} \Delta m / \Delta \tau$, сила притяжения Земли F , ускорение a_x , скорость v_x , координата x :

$$m^{t+1} = m^t - b \Delta \tau, \quad F_p^{t+1} = v_{отн} (m^t - m^{t+1}) / \Delta \tau,$$

$$F^{t+1} = GMm^{t+1} / (x^{t+1})^2, \quad a_x^{t+1} = (F_p^{t+1} - F^{t+1}) / m^{t+1},$$

$$v_x^{t+1} = v_x^t + a_x^{t+1} \Delta \tau, \quad x^{t+1} = x^t + v_x^{t+1} \Delta \tau.$$

В цикле должен быть предусмотрен вывод результатов вычислений на экран. Типичные результаты приведены на рис. 7.3.2. Видно, что по мере вытекания газов масса ракеты с топливом уменьшается до значения m_0 , скорость сначала возрастает, а после остановки двигателя в момент τ' начинает убывать вследствие притяжения Земли.

7.2. Двумерное движение точки в однородном поле

Большое количество физических задач сводится к анализу движения систем, имеющих две степени свободы, в частности к двумерному движению материальной точки. Например, задача о качении шарика по искривленной поверхности, задача о движении тела, брошенного под углом к горизонту, задача о движении частицы в поле центральных сил и т. д.

Рассмотрим материальную точку массой m , движущуюся в однородном силовом поле $\vec{F} = \vec{F}(x, y, \tau)$, на которую действует сила сопротивления $\vec{F}_c = -r\vec{v}$. Необходимо, зная начальные условия x_0, y_0, v_{0x}, v_{0y} , определить координаты, проекции скорости и ускорения в последующие моменты времени, построить траекторию [2; 5].

Пусть материальная точка брошена с некоторой начальной скоростью в поле тяжести в вязкой среде (рис. 7.4). При отсутствии силы трения точка движется по параболе, а при ее наличии - по более сложной кривой. На рис. 7.4.1 показаны действующие на нее силы. По второму закону Ньютона: $m\vec{g} + \vec{F}_{mp} = m\vec{a}$, где $\vec{F}_{mp} = -r\vec{v}$. Получаем:

$$a_x^{t+1} = -rv_x^t / m, \quad a_y^{t+1} = (-mg - rv_y^t) / m, \quad v_x^{t+1} = v_x^t + a_x^{t+1} \Delta\tau,$$

$$v_y^{t+1} = v_y^t + a_y^{t+1} \Delta\tau, \quad x^{t+1} = x^t + v_x^{t+1} \Delta\tau, \quad y^{t+1} = y^t + v_y^{t+1} \Delta\tau.$$

В начале программы необходимо задать массу материальной точки m , коэффициент вязкости r , начальные координаты x_0, y_0 и проекции скорости v_{0x}, v_{0y} , силовое поле $F_x = F_x(x, y), F_y = F_y(x, y)$, а также шаг по времени $\Delta\tau$. Затем следует организовать цикл по времени t , в котором будут рассчитываться ускорение, скорость и координата точки в следующий момент времени $t + 1$ и результаты вычислений выводиться на экран в текстовом или графическом виде. Используется программа ПР-3.

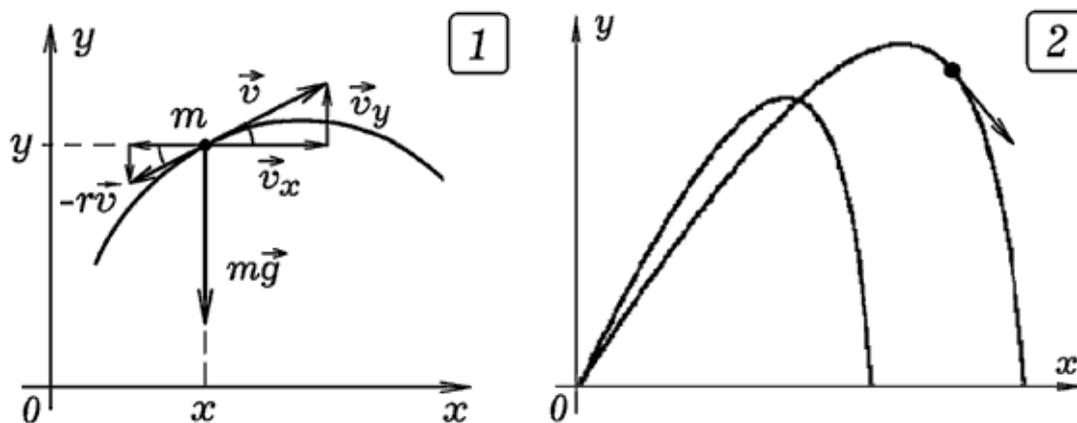


Рис. 7.4. Двумерное движение точки в однородном поле тяжести

Результаты приведены на рис. 7.4.2. Легко определить время подъема τ_1 и общее время τ движения частицы. Для этого результаты вычислений координат и времени выводят в текстовом формате. Чтобы найти время подъема, следует воспользоваться тем, что в наивысшей точке траектории проекция скорости на ось y меняет свой знак на противоположный. При вычислении времени полета расчеты производятся до тех пор, пока y не станет меньше нуля. Время подъема τ_1 оказывается меньше времени спуска $\tau_2 = \tau - \tau_1$. Внеся небольшие изменения в программу ПР-3, можно рассчитать траекторию в случае, когда вдоль оси Ox дует ветер.

7.3. Движение точки в центральном поле сил

Рассмотрим материальную точку M , движущуюся в центральном поле с потенциальной энергией $U = U(r)$. На нее действует сила притяжения $F = F(r)$, зависящая только от расстояния $r = |OM|$ и направленная вдоль прямой OM (рис. 7.5.1). Можно записать:

$$\vec{F} = F \left(-\frac{x}{r} \vec{i} - \frac{y}{r} \vec{j} \right), \quad F_x = -F \cos \alpha = -F \frac{x}{r}, \quad F_y = -F \sin \alpha = -F \frac{y}{r}.$$

Программа ПР-4 для расчета движения точки содержит цикл, в котором пересчитываются проекции ее ускорения, скорости и координаты в последовательные моменты времени t и строится траектория.

Про моделируем движение точки в поле гравитационных сил притяжения, действующих по закону обратных квадратов $F = GmM / r^2$ (рис. 7.5.2). При наличии силы вязкого трения точка движется по спирали, приближаясь к началу координат. Из рис. 7.6.1 видно, при малых скоростях точка движется по эллиптической орбите (траектории 1, 2, 3, 4), а при больших - по гиперболической (траектории 5, 6). Критическому случаю соответствует параболическая траектория. На рис. 7.6.2 представлены результаты расчетов движения частицы в центральном поле, для которого

$F = k/r^{2/3}$. Видно, что траекторией является незамкнутая кривая (розетка). Известно, что частица движется по замкнутой траектории только в поле квазиупругой силы или силы притяжения, для которой $F = k/r^2$.

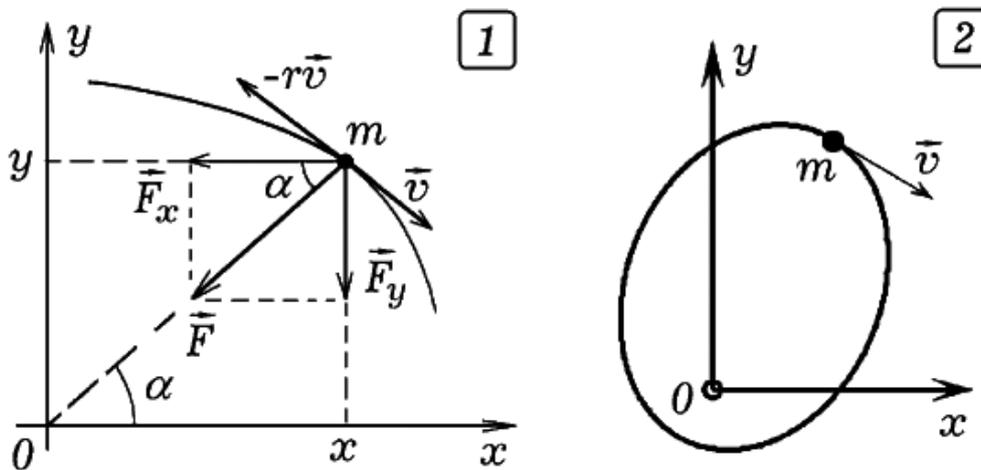


Рис. 7.5. Движение точки в поле центральной силы

Для планеты, вращающейся вокруг Солнца, построим графики зависимости расстояния, линейной скорости и секториальной скорости планеты от времени и убедимся в том, что секториальная скорость планеты остается постоянной (второй закон Кеплера). Пусть за время Δt планета перемещается из $A(x, y)$ в $B(x_1, y_1)$ (рис. 7.7.1). Длины сторон AC и BC прямоугольного треугольника ABC равны:

$$|AC| = r - r_1, \quad |BC| = \sqrt{(x - x_1)^2 + (y - y_1)^2 - (r - r_1)^2}.$$

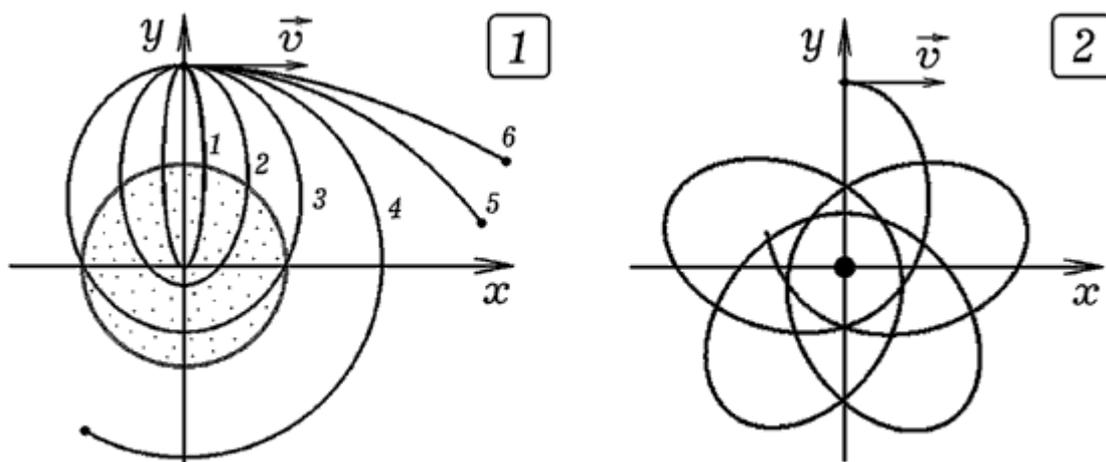


Рис. 7.6. Движение точки в гравитационном поле Земли

Радиус-вектор планеты заметает площадь $\Delta S = r |BC| / 2$, ее секториальная скорость $\omega_s = \Delta S / \Delta \tau$. Используется программа ПР-4. Из рис. 7.7.2 видно, что секториальная скорость не изменяется, что подтверждает второй закон Кеплера. Кроме расчета секториальной скорости в программе вычисляются скорость v и расстояния r от планеты до Солнца [2; 5].

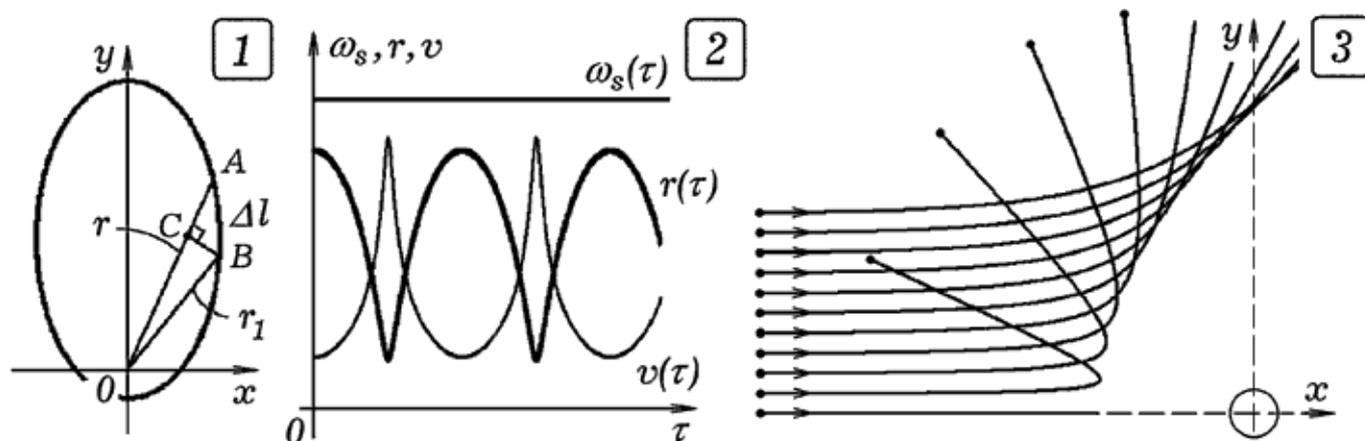


Рис. 7.7. Моделирование движения точки в центральном поле

На рис. 7.7.3 представлены результаты расчетов движения альфа-частиц в поле положительно заряженного ядра атома (опыт Резерфорда) при различных значениях прицельного параметра ρ . Действуют силы отталкивания, поэтому в программе следует изменить знак в выражении для силы F . Траекториями частиц являются гиперболы. Программа ПР-4 после небольших изменений позволяет промоделировать движение частицы в центральном поле, задаваемом уравнением: $F(r) = k_1 / r^4 - k_2 / r^2$. Коэффициенты k_1 и k_2 подбирают так, чтобы при больших r преобладали силы притяжения, а при малых - силы отталкивания.

7.4. Движение заряда в электрическом и магнитном полях

Теперь рассмотрим движение заряженной частицы в магнитном поле, если ее начальная скорость перпендикулярна линиям индукции \vec{B} . Со

стороны поля на заряд действует сила Лоренца $F = qvB$, лежащая в плоскости рисунка и направленная перпендикулярно вектору скорости (рис. 7.8.1). Проекции силы Лоренца на координатные оси равны [6]:

$$F_x = -F \sin \beta = F \frac{v_y}{v}, \quad F_y = -F \cos \beta = -F \frac{v_x}{v}.$$

Проекции ускорения, скорости и координаты частицы в момент времени $t+1$ равны:

$$a_x^{t+1} = (F_x^t - rv_x^t)/m, \quad a_y^{t+1} = (F_y^t - rv_y^t)/m,$$

$$v_x^{t+1} = v_x^t + a_x^{t+1} \Delta\tau, \quad v_y^{t+1} = v_y^t + a_y^{t+1} \Delta\tau,$$

$$x^{t+1} = x^t + v_x^{t+1} \Delta\tau, \quad y^{t+1} = y^t + v_y^{t+1} \Delta\tau.$$

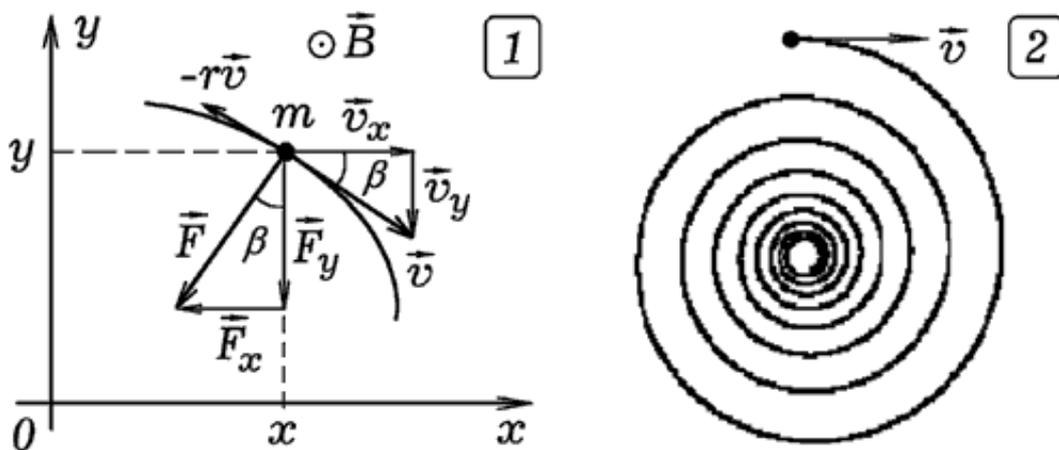


Рис. 7.8. Движение заряженной частицы в магнитном поле

Определив координаты и проекции скорости точки в момент времени $t+1$, можно повторить процедуру вычисления требуемое количество раз и построить траекторию движения точки. Заряженная частица описывает окружность, радиус которой $R = mv/Bq$. При наличии тормозящей силы скорость частицы и радиус кривизны траектории постепенно уменьшаются, частица движется по спирали (рис. 7.8.2).

Определенный интерес представляет случай, когда заряженная частица движется в **скрещенных электрическом и магнитном полях**. Пусть силовые линии электрического поля \vec{E} лежат в плоскости экрана и направлены вверх, а силовые линии магнитного поля \vec{B} направлены к нам

перпендикулярно плоскости экрана. Если заряд частицы положительный, то на него со стороны электрического поля действует постоянная сила, направленная вверх. Чтобы учесть ее влияние, необходимо к вертикальной проекции силы Лоренца прибавить постоянное слагаемое qE :

$$F_x = -F \sin \beta = F \frac{v_y}{v}, \quad F_y = qE - F \cos \beta = qE - F \frac{v_x}{v}.$$

В остальном задача решается аналогично. Результаты моделирования движения заряженной частицы в скрещенных электрическом и магнитном полях при отсутствии и наличии силы сопротивления представлены на рис. 7.9.1 и 7.9.2. Если начальная скорость частицы близка к нулю и сила сопротивления отсутствует, то траекторией является циклоида без петель.

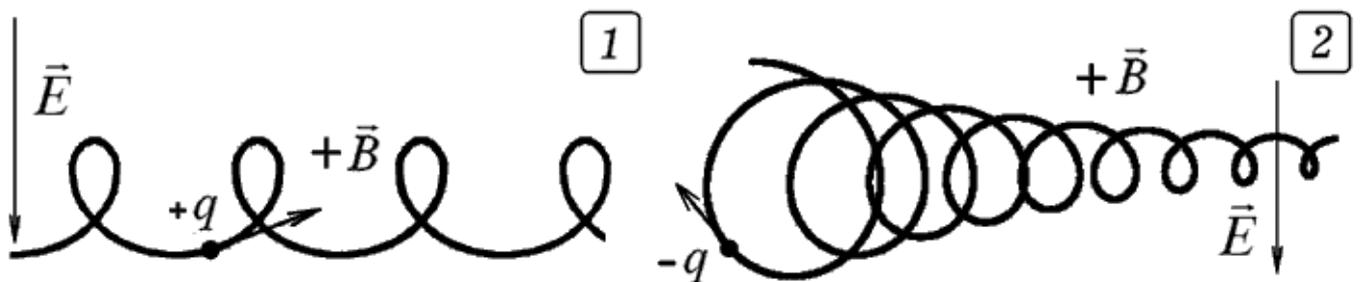


Рис. 7.9. Движение заряженной частицы в скрещенных полях

7.5. Сложные случаи движения частицы в силовом поле

Рассмотрим несколько задач, в которых рассчитывается траектория движения частиц в различных силовых полях.

Задача 5. Изучите движение заряженной частицы q массы m в электростатическом поле двух точечных зарядов Q_1 и Q_2 . Начальная скорость лежит в плоскости, содержащей заряды Q_1 , Q_2 и q .

На частицу массой m с зарядом q в электростатическом поле двух шаров с зарядами Q_1 и Q_2 действует сила:

$$\vec{F} = F_x \vec{i} + F_y \vec{j} = (-F_1 \cos \alpha + F_2 \cos \beta) \vec{i} + (-F_1 \sin \alpha - F_2 \sin \beta) \vec{j},$$

где $F_x = -F_1 \cos \alpha + F_2 \cos \beta$, $F_y = -F_1 \sin \alpha - F_2 \sin \beta$,

$$F_1 = k \frac{Q_1 q}{x^2 + y^2}, \quad F_2 = k \frac{Q_2 q}{(L-x)^2 + y^2}, \quad \cos \alpha = \frac{x}{\sqrt{x^2 + y^2}},$$

$$\sin \alpha = \frac{y}{\sqrt{x^2 + y^2}}, \quad \cos \beta = \frac{L-x}{\sqrt{(L-x)^2 + y^2}}, \quad \sin \beta = \frac{y}{\sqrt{(L-x)^2 + y^2}}.$$

Для расчета траектории необходимо написать программу, содержащую цикл, в котором вычисляются проекции действующей на точку силы и сообщаемого ей ускорения. В нем же должны определяться проекции скоростей точки и ее координаты в последующие моменты времени. Результат решения задачи - на рис. 7.10.2.

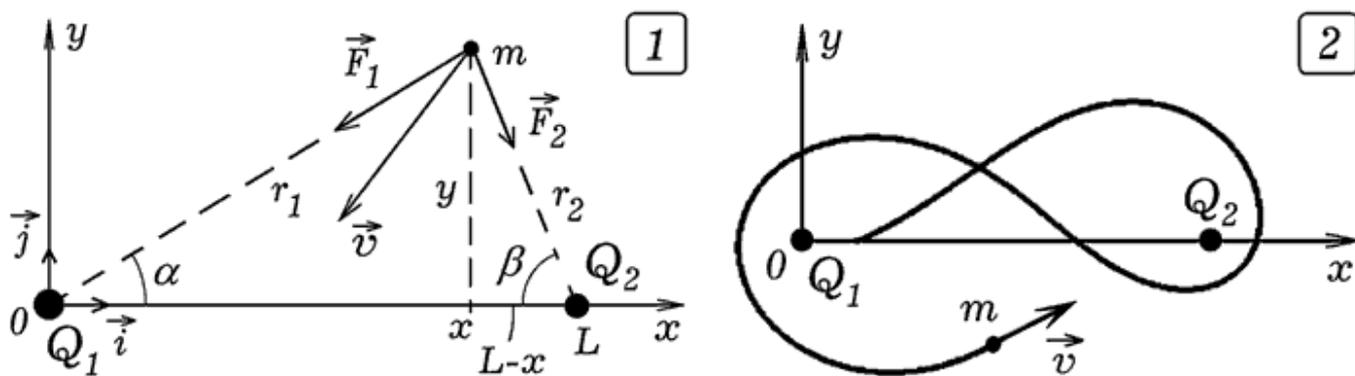


Рис. 7.10. Движение заряженной частицы в поле двух точечных зарядов

Задача 6. Пучок электронов проходит между двух параллельных стержней, заряженных отрицательно. Рассчитайте их движение.

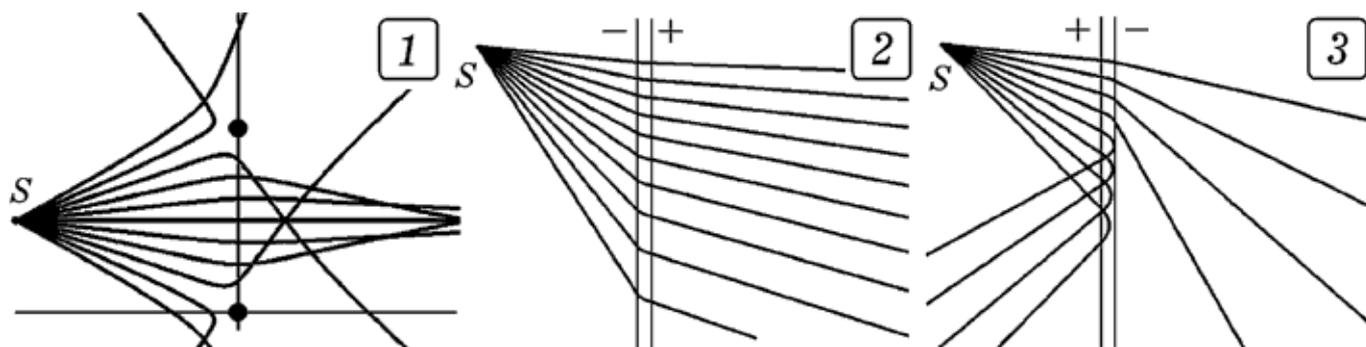


Рис. 7.11. Искривление пучка заряженных частиц в электрическом поле

Задача решается аналогично. Следует учесть, что между электронами и стержнями действует сила отталкивания. Как видно из рис. 7.11.1, это приводит к фокусировке пучка электронов.

Задача 7. В электронном микроскопе для фокусировки электронного пучка используются электростатические линзы. Преломляющая поверхность такой линзы представляет собой две параллельные металлические сетки, на которые поданы различные потенциалы. Изучите преломление и отражение электронных пучков в поле двух плоских сеток.

Пусть электронные пучки падают на двойную сетку, расположенную вертикально, причем левая сетка имеет отрицательный потенциал, а правая - положительный. Вдали от нее электрические поля обеих сеток компенсируют друг друга. Между сетками на электрон действует сила, направленная вправо (рис. 7.11.2), электронные пучки преломляются. Если поменять направление \vec{E} между сетками, то получится результат, представленный на рис. 7.11.3. Видно, что некоторые пучки отражаются.

Задача 8. Изучите движение шарика в поле тяжести по поверхности, задаваемой уравнением $z = 0,25(x^2 + y^2)^2 - 0,5(x^2 + y^2)$ и представляющей замкнутую канавку вокруг оси Oz . Постройте траекторию шарика.

Потенциальная энергия шарика массой m , движущегося по рассматриваемой поверхности равна: $U(x, y) = mg(0,25(x^2 + y^2)^2 - 0,5(x^2 + y^2))$. Горизонтальные проекции силы реакции, действующей на шарик:

$$F_x = -\frac{\partial U}{\partial x} = -mg(x^3 + y^2x - x), \quad F_y = -\frac{\partial U}{\partial y} = -mg(y^3 + x^2y - y).$$

Предлагается самостоятельно рассчитать траекторию в плоскости XOY .

7.6. Скольжение точки по поверхности

В некоторых случаях необходимо промоделировать скольжение материальной точки по заданной поверхности или кривой. В качестве примера проанализируем две задачи.

Задача 9. Изучите скольжение точки по циклоидальной горке в поле тяжести. Убедитесь в том, что циклоида является таутохроной, то есть время скольжения к основанию горки не зависит от точки запуска [5].

Исследуем скольжение точки по циклоидальной горке и вычислим время движения при различных начальных положениях точки (рис. 7.12.1). Запишем уравнения циклоиды и выразим параметр α :

$$x = R(\alpha - \sin \alpha), \quad y = R(1 - \cos \alpha), \quad c = \cos \alpha = 1 - y/R, \quad s = \sin \alpha = (1 - c^2)^{0,5},$$

$$\alpha = \begin{cases} \operatorname{arctg}(s/c), & \text{если } y \leq R, \\ \pi/2 - \operatorname{arctg}|s/c|, & \text{если } y > R. \end{cases}$$

Чтобы определить угол β , найдем координаты двух близко расположенных точек, соответствующих значениям параметра α_1, α_2 , и выразим угол β между касательной к траектории и осью Oy :

$$x_1 = R(\alpha_1 - \sin \alpha_1), \quad y_1 = R(1 - \cos \alpha_1),$$

$$x_2 = R(\alpha_2 - \sin \alpha_2), \quad y_2 = R(1 - \cos \alpha_2),$$

$$\beta = \frac{\pi}{2} - \operatorname{arctg} \frac{y_2 - y_1}{x_2 - x_1}.$$

Для расчета тангенциального ускорения точки, ее скорости и координат используются формулы: $m\vec{g} + \vec{N} - r\vec{v} = m\vec{a}$, $a_\tau^{t+1} = g \cos \beta - rv^t/m$,

$$v^{t+1} = v^t + a_\tau^{t+1} \Delta\tau, \quad x^{t+1} = x^t + v^{t+1} \sin \beta \Delta\tau, \quad y^{t+1} = y^t + v^{t+1} \cos \beta \Delta\tau.$$

Программа ПР-6, моделирующая скольжение тела по циклоидальной горке, содержит цикл по времени, в котором пересчитываются координаты и скорость частицы, строится ее изображение на экране, вычисляется время движения. Изменяя начальное значение α , можно запускать точку с

различных положений. При отсутствии трения ($r = 0$) время скольжения не зависит от точки запуска (в пределах погрешности вычислений), что и доказывает таутохронность циклоиды.

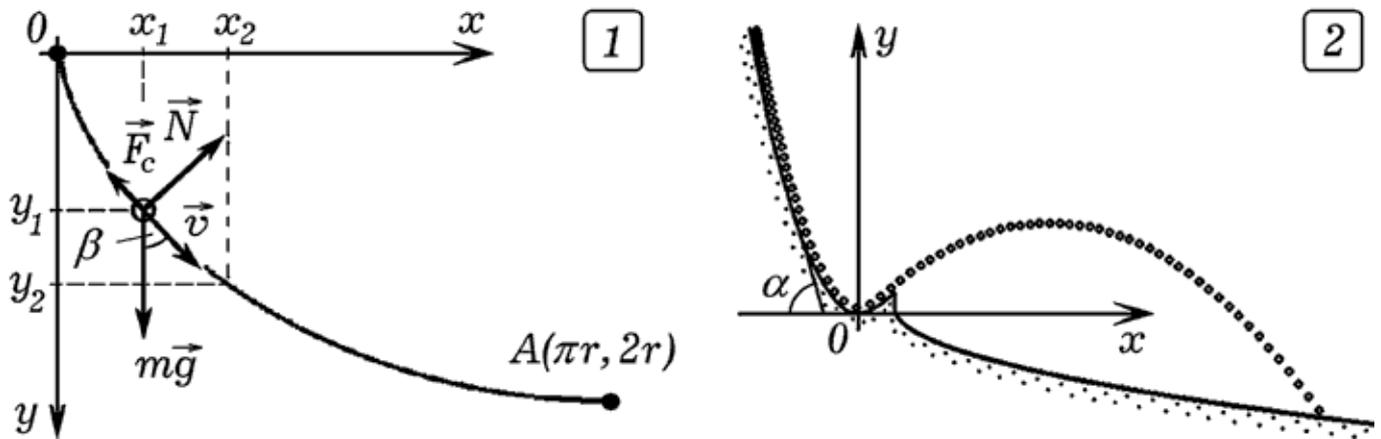


Рис. 7.12. Моделирование скольжения точки по поверхности

Задача 10. Изучите движение лыжника, скатывающегося с горы и прыгающего с трамплина на склон. Профили горки и склона, на который прыгает лыжник, задаются уравнениями:

$$y = 0,015x^2, \text{ если } -160 < x \leq 30; \quad y = -(50x - 3000)^{0,5}, \text{ если } 30 < x < 1000.$$

Для расчета движения лыжника по горке используются уравнения:

$$a_\tau = g(\sin \alpha - \mu \cos \alpha), \quad v^{t+1} = v^t + a_\tau^{t+1} \Delta \tau, \quad x^{t+1} = x^t + v^{t+1} \cos \alpha \Delta \tau, \\ y^{t+1} = y^t - v^{t+1} \sin \alpha \Delta \tau, \quad \alpha = \arctg[(y_1 - y_2)/(x_2 - x_1)].$$

После отрыва от трамплина лыжник движется так же, как тело, брошенное под углом к горизонту. Программа ПР-7 [5] содержит цикл по времени, в котором вычисляются координаты и скорость лыжника в последовательные моменты времени, а результаты выводятся на экран (рис. 7.12.2). Аналогичным методом можно исследовать скольжение частицы по цилиндрической поверхности, движение бусинки по спице и т. д.

7.7. Движение системы из двух частиц

Программа, рассчитывающая движение двух частиц, должна содержать цикл по времени, в котором пересчитываются силы взаимодействия частиц, ускорение каждой частицы, их скорости и координаты.

Задача 11. Тележка с телом движется с некоторой скоростью. В тело попадает пуля (рис. 7.13) и застревает в нем или проходит насквозь. Как изменяются скорости тележки и пули с течением времени?

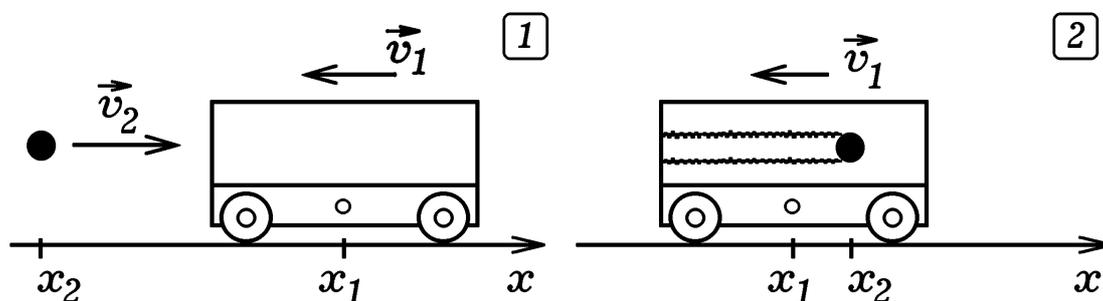


Рис. 7.13. Движение тела до и после попадания пули

Запишем формулы, позволяющие рассчитать ускорение пули и тела, их скорости и координаты во время взаимодействия:

$$F_{mp}^{t+1} = r |v_2^t - v_1^t|, \quad a_{1x}^{t+1} = F_{mp}^{t+1} / m_1, \quad a_{2x}^{t+1} = -F_{mp}^{t+1} / m_2,$$

$$v_{ix}^{t+1} = v_{ix}^t + a_{ix}^{t+1} \Delta\tau, \quad x_{ix}^{t+1} = x_{ix}^t + v_{ix}^{t+1} \Delta\tau, \quad i=1,2.$$

Когда пуля находится вне тела, она с ним не взаимодействует. Используется программа ПР-8. На рис. 7.14 приведены графики зависимостей скоростей пули и тела от времени в следующих случаях: 1) пуля и тело двигались навстречу друг другу, пуля прошла на вылет (рис. 7.14.1); 2) пуля догнала тело и застряла в нем (рис. 7.14.2).

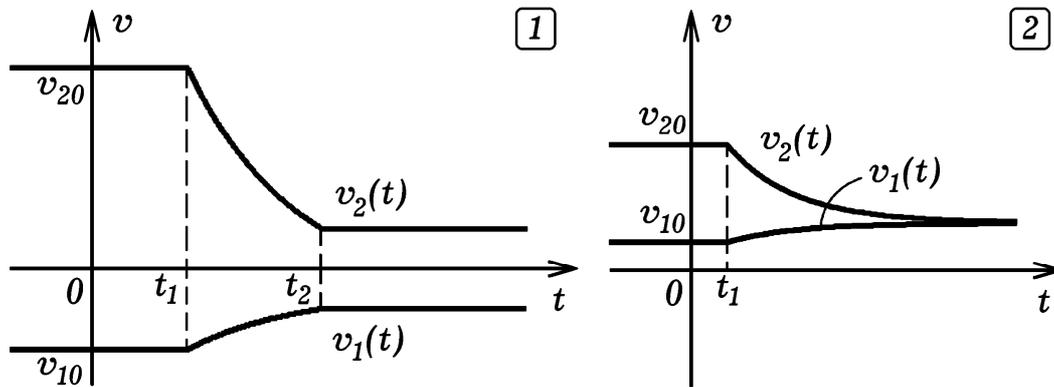


Рис. 7.14. Графики зависимостей скоростей тела и пули от времени

Задача 12. Две частицы массами m_1 и m_2 движутся навстречу друг к другу с некоторыми скоростями v_1 и v_2 . Промоделируйте абсолютно упругое взаимодействие этих частиц, если между ними действуют: а) силы отталкивания; б) силы притяжения. В обоих случаях $F = k/r^2$.

Расстояние между частицами $r = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$, где (x_1, y_1) и (x_2, y_2) - их координаты (рис. 7.15.1). Проекция силы отталкивания:

$$F_x = -F \cos \alpha, \quad F_y = -F \sin \alpha, \quad \text{где}$$

$$F = k/r^2, \quad \cos \alpha = (x_2 - x_1)/r, \quad \sin \alpha = (y_2 - y_1)/r.$$

Проекция ускорений, скоростей и координаты точек рассчитываются так:

$$a_{ix}^{t+1} = F_{ix} / m_i, \quad v_{ix}^{t+1} = v_{ix}^t + a_{ix}^{t+1} \Delta \tau, \quad x_i^{t+1} = x_i^t + v_{ix}^{t+1} \Delta \tau,$$

$$a_{iy}^{t+1} = F_{iy} / m_i, \quad v_{iy}^{t+1} = v_{iy}^t + a_{iy}^{t+1} \Delta \tau, \quad y_i^{t+1} = y_i^t + v_{iy}^{t+1} \Delta \tau, \quad i = 1, 2.$$

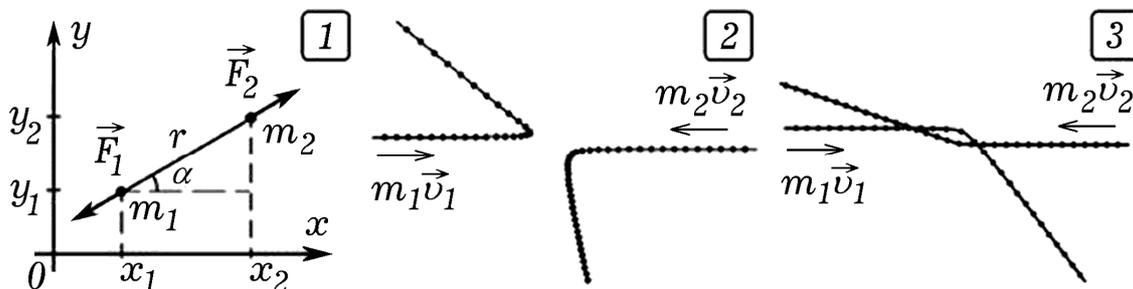


Рис. 7.15. Нецентральное взаимодействие двух частиц

Для решения проблемы двух тел используется программа ПР-9. Результаты моделирования представлены на рис. 7.15.2 и 7.15.3. При не-

центрального упругом взаимодействии двух частиц, между которыми действуют силы отталкивания, частицы движутся по гиперболам (рис. 7.15.2). На рис. 7.15.3 изображены траектории частиц в случае, когда между ними действуют силы притяжения (отношение масс $m_1 / m_2 = 1/3$). Компьютерная модель также позволяет изучить **центральный удар**.

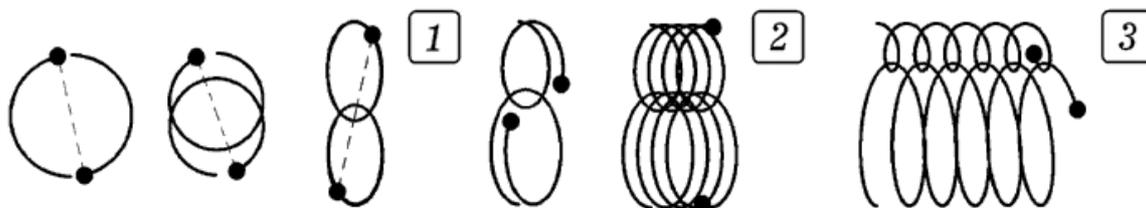


Рис. 7.16. Движение двух частиц, притягивающихся друг к другу

Задача 13. Рассчитайте движение двух материальных точек m_1 и m_2 , притягивающихся друг к другу по закону обратных квадратов. Получите траектории в различных системах отсчета [4-6].

Используется программа ПР-9. Чтобы получить силу притяжения вместо силы отталкивания, необходимо изменить знак в выражении для силы F . Результаты моделирования - на рис. 7.16. На рис. 7.16.1 показано движение частиц с равными массами ($m_1 = m_2$). Рис. 7.16.2 соответствует отношению масс $m_1 / m_2 = 3$, рис. 7.16.3 - $m_1 / m_2 = 4$. Решение задачи двух тел также приведено на рис. 7.17. Изменяя начальные скорости частиц, можно промоделировать движение спутника вокруг планеты или движение двух звезд вокруг их центра масс в различных системах отсчета.

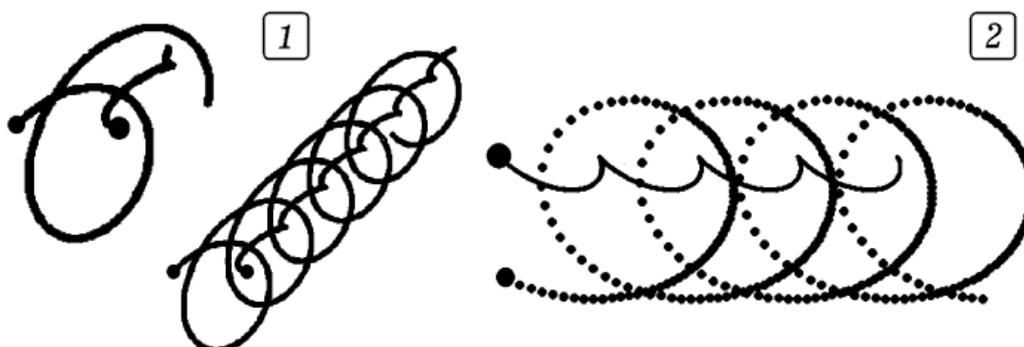


Рис. 7.17. Моделирование движения системы из двух частиц

7.8. Движение системы частиц в силовом поле

Рассмотрим систему из N частиц с массами m_i , $i = 1, 2, \dots, N$, взаимодействующих друг с другом с внутренними силами $\vec{F}'_{i,j}$. Пусть на каждую точку действует внешняя сила \vec{F}_i . Исходя из начальных координат x_{i0} , y_{i0} и скоростей \vec{v}_{i0} , необходимо определить координаты и скорости материальных точек в последующие моменты времени.

Для каждой частицы системы запишем второй закон Ньютона:

$$m_i \frac{d^2 \vec{r}_i}{d\tau^2} = \vec{F}_i + \sum_{j=1}^N \vec{F}'_{i,j}, \quad i = 1, 2, \dots, N.$$

где \vec{F}_i - равнодействующая внешних сил, действующих на i -ю материальную точку со стороны тел, не входящих в систему, $\vec{F}'_{i,j}$ - внутренняя сила, действующая на i -ю точку со стороны j -й точки.

Дифференциальное уравнение второго порядка может быть заменено двумя дифференциальными уравнениями первого порядка. Имеем:

$$\frac{d\vec{v}_i}{d\tau} = \vec{a}_i = \frac{1}{m} \left(\vec{F}_i + \sum_{j=1}^N \vec{F}'_{i,j} \right), \quad \frac{d\vec{r}_i}{d\tau} = \vec{v}_i, \quad i = 1, 2, \dots, N.$$

Зная внешние и внутренние силы, действующие на каждую материальную точку, можно рассчитать их ускорения. Исходя из координат и проекций скоростей точки в момент времени τ , можно определить состояние системы в следующий момент времени $\tau + \Delta\tau$. Рассматриваемая модель позволяет изучить движение различных механических систем.

Алгоритм, используемый для моделирования двумерного движения системы частиц, состоит в следующем. Задают число частиц N , их массы m_i , начальные координаты x_{i0} , y_{i0} , проекции скоростей v_{i0x} , v_{i0y} , силовое поле $F_x = F_x(x, y, z)$, $F_y = F_y(x, y, z)$ и шаг по времени $\Delta\tau$. Организуют цикл по целочисленной переменной t . Определяют проекции $R_{ix} = F_{ix} + F'_{ix}$, $R_{iy} = F_{iy} + F'_{iy}$ равнодействующих всех сил, действующих

на каждую i -ю частицу в момент $t+1$, и записывают их в массивы. В цикле по i от 1 до N определяют проекции ускорения, скорости и координаты для каждой i -й частицы в момент $t+1$ по формулам:

$$a_{ix}^{t+1} = R_{ix} / m_i, \quad v_{ix}^{t+1} = v_{ix}^t + a_{ix}^{t+1} \Delta\tau, \quad x_i^{t+1} = x_i^t + v_{ix}^{t+1} \Delta\tau,$$

$$a_{iy}^{t+1} = R_{iy} / m_i, \quad v_{iy}^{t+1} = v_{iy}^t + a_{iy}^{t+1} \Delta\tau, \quad y_i^{t+1} = y_i^t + v_{iy}^{t+1} \Delta\tau.$$

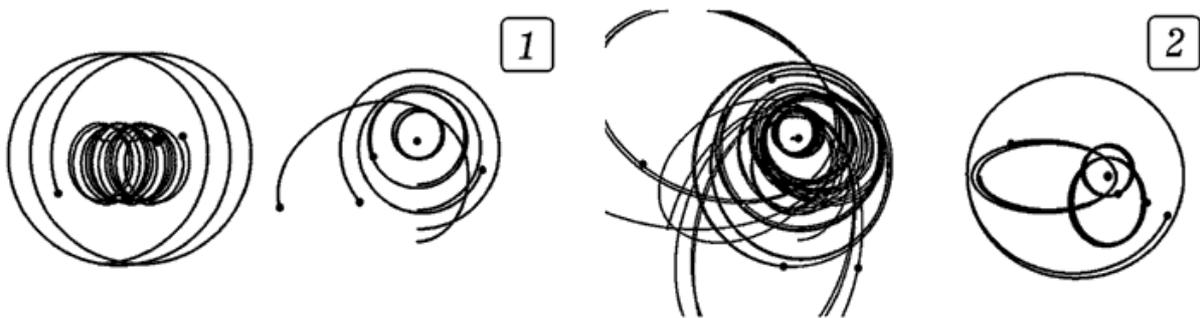


Рис. 7.18. Моделирование движения системы из нескольких частиц

Результаты записывают в массивы $x[i]$, $y[i]$, $v_x[i]$, $v_y[i]$. Стирают изображения материальных точек в предыдущий момент времени t , координаты которых сохранены в массивах $xx[i]$, $yy[i]$. На экране строят точки в следующий момент $t+1$, рисуют графики или выводят результат в числовом виде. После этого все повторяется снова [2; 4-6].

Используя программу ПР-10, можно промоделировать движение нескольких планет и комет Солнечной системы. Результат моделирования планетарной системы, состоящей из 3-5 планет, вращающихся вокруг некоторой звезды, представлен на рис. 7.18. Из-за того, что планеты взаимодействуют друг с другом силами гравитационного притяжения, они движутся по сложным незамкнутым траекториям.

В качестве еще одного примера рассмотрим разрыв снаряда на несколько осколков различной массы в поле тяжести. При взрыве возникает сила отталкивания, быстро уменьшающаяся по мере удаления осколков. Снаряд должен моделироваться несколькими частицами, имеющими равные начальные скорости и близкие начальные координаты (программа ПР-11). Через некоторое время после начала движения происходит

взрыв, - частицы начинают отталкиваться (рис. 7.19). Программа рисует отрезки, соединяющие два осколка в данный момент времени, и рассчитывает положение центра масс C (рис. 7.19.1). В поле тяжести центр масс системы продолжает двигаться по параболе (рис. 7.19.2). Задача может быть решена для трехмерного случая в отсутствие поля тяжести (рис. 7.19.3).

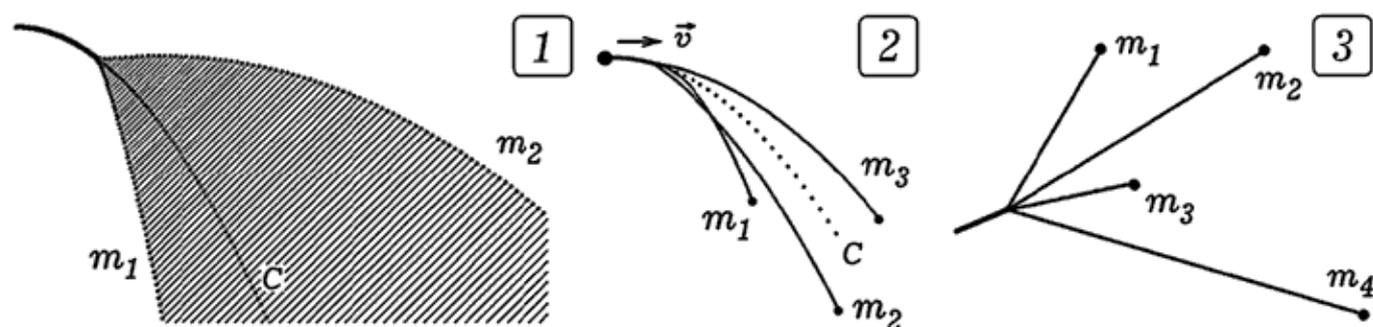


Рис. 7.19. Разрыв снаряда на несколько осколков (C - центр масс)

7.9. Моделирование движения тела

Твердое тело представимо в виде совокупности материальных точек, связанных между собой жесткими невесомыми стержнями. Если речь идет об упругом теле, то его можно рассмотреть как систему частиц, соединенных упругими стержнями. Про моделируем плоское движение тела в поле тяжести, при котором его центр масс движется по кривой, а само тело вращается вокруг центра масс. Для этого используем модель, состоящую из двух частиц, связанных невесомым упругим стержнем, и ограничимся рассмотрением движения системы в вертикальной плоскости.

Можно записать уравнения: $F = k(l - l_0)$, $l = ((x_1 - x_2)^2 + (y_1 - y_2)^2)^{0,5}$,

$$F_{1x} = F(x_2 - x_1)/l - rv_{1x}, \quad F_{1y} = F(y_2 - y_1)/l - rv_{1y} - m_1 g,$$

$$F_{2x} = -F(x_2 - x_1)/l - rv_{2x}, \quad F_{2y} = -F(y_2 - y_1)/l - rv_{2y} - m_2 g.$$

Программа ПР-12 содержит цикл по времени, в котором вычисляются действующие силы, ускорения, скорости и координаты точек. На экране

строятся положения стержня в последовательные моменты времени. Программа позволяет промоделировать: 1) отскок стержня от стены и пола; 2) движение системы в случае, когда массы точек сильно отличаются; 3) движение системы при не очень большой жесткости стержня [5].

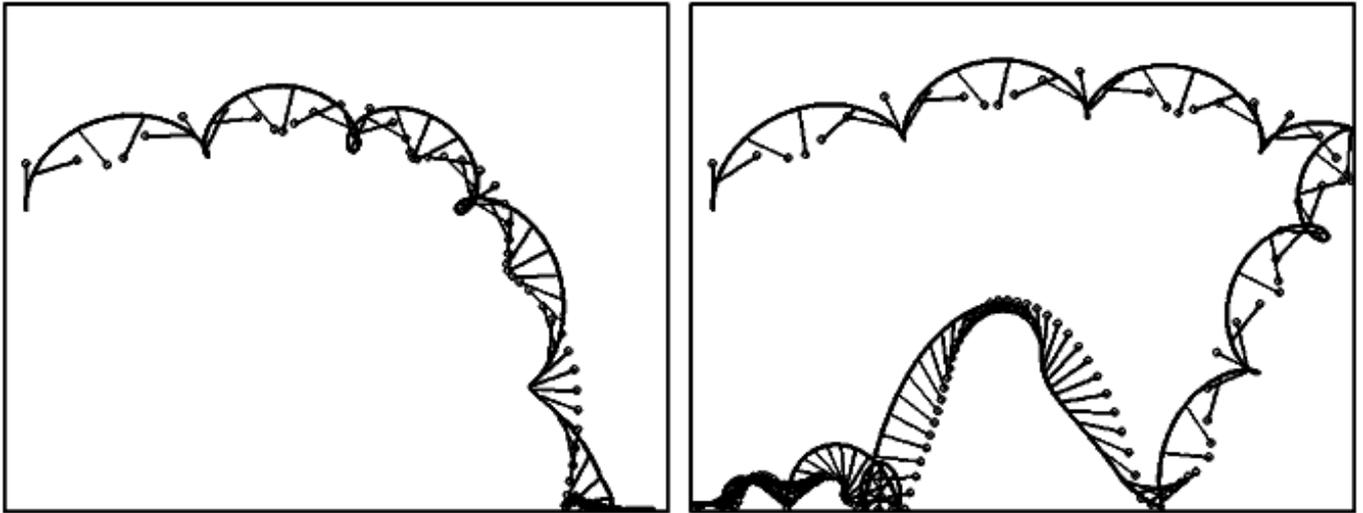


Рис. 7.20. Расчет движения стержня в поле тяжести

Аналогичным образом можно промоделировать сферическое движение гироскопа вокруг закрепленной точки в поле тяжести [5]. Мысленно заменим гироскоп, имеющий форму диска или конуса с осью (рис. 7.21.1), системой из трех материальных точек, находящихся в вершинах равнобедренного или равностороннего треугольника с длиной стороны l_0 и связанных жесткими невесомыми стержнями (рис. 7.21.2). Можно подобрать массы материальных точек и длины сторон треугольника так, чтобы получившаяся система соответствовала какому-то реальному гироскопу с вполне конкретными массой, моментом инерции и положением центра масс (ЦМ). Точка m_3 остается неподвижной, ее координаты не изменяются. Вместо жестких стержней будем рассматривать упругие стержни с большим коэффициентом жесткости k . В этом случае задача сводится к расчету движения двух частиц m_1 и m_2 .

Рассмотрим симметричную систему из трех материальных точек m_1 , m_2 , m_3 , связанных между собой упругими невесомыми стержнями, которая находится в однородном поле тяжести (рис. 7.21.2). Точка m_3 закреплена, точки m_1 и m_2 имеют равные массы и движутся в среде с коэффициентом сопротивления r . Стержни имеют жесткость k , их длины в недеформированном состоянии равны l_0 . Пусть в начальный момент $t = 0$ частицы m_1 и m_2 лежат в плоскости yOz и их координаты равны:

$$\begin{aligned} x_1 = 0, \quad y_1 = l_0 \sin(\theta_0 - \pi/6), \quad z_1 = l_0 \cos(\theta_0 - \pi/6), \\ x_2 = 0, \quad y_2 = l_0 \sin(\theta_0 + \pi/6), \quad z_2 = l_0 \cos(\theta_0 + \pi/6). \end{aligned}$$

Чтобы начальная скорость ЦМ C была равна 0, необходимо задать начальные скорости частиц m_1 и m_2 так: $v_{1x} = -v_{2x}$. Все остальные проекции скоростей в начальный момент времени равны нулю.

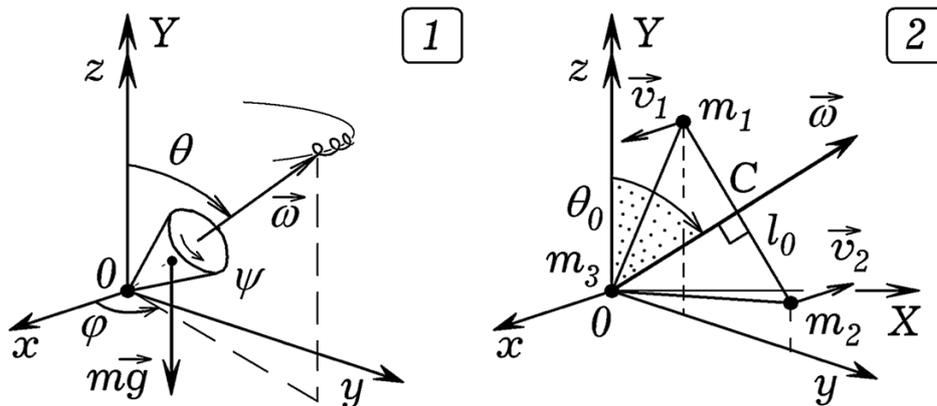


Рис. 7.21. Моделирование сферического движения твердого тела

На каждую частицу m_1 и m_2 действуют две силы упругости, сила вязкого трения и сила тяжести. Сила вязкого трения пропорциональна скорости и равна $-r\vec{v}$. Сила упругости находится так: $F_{ij} = -k(l_{ij} - l_0)$, где $l_{ij} = ((x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2)^{0,5}$ - длина деформированного стержня, соединяющего частицы m_i и m_j . Тогда равнодействующая сила, приложенная к i -й частице имеет проекции:

$$F_{ix} = \sum_{j=1, i \neq j}^3 F_{ij} \frac{x_i - x_j}{l_{ij}}, \quad F_{iy} = \sum_{j=1, i \neq j}^3 F_{ij} \frac{y_i - y_j}{l_{ij}}, \quad F_{iz} = \sum_{j=1, i \neq j}^3 F_{ij} \frac{z_i - z_j}{l_{ij}}.$$

Зная проекции силы в дискретный момент t , можно с помощью метода конечных разностей Эйлера рассчитать проекции ускорений, скоростей и координаты частиц в момент $t + 1$ по формулам:

$$a_{ix}^{t+1} = (F_{ix}^t - r v_{ix}^t) / m_i, \quad a_{iy}^{t+1} = (F_{iy}^t - r v_{iy}^t) / m_i,$$

$$a_{iz}^{t+1} = (F_{iz}^t - r v_{iz}^t) / m_i - g, \quad i = 1, 2,$$

$$v_{ix}^{t+1} = v_{ix}^t + a_{ix}^{t+1} \Delta \tau, \quad v_{iy}^{t+1} = v_{iy}^t + a_{iy}^{t+1} \Delta \tau, \quad v_{iz}^{t+1} = v_{iz}^t + a_{iz}^{t+1} \Delta \tau,$$

$$x_i^{t+1} = x_i^t + v_{ix}^{t+1} \Delta \tau, \quad y_i^{t+1} = y_i^t + v_{iy}^{t+1} \Delta \tau, \quad z_i^{t+1} = z_i^t + v_{iz}^{t+1} \Delta \tau.$$

Чтобы охарактеризовать **сферическое движение тела**, проследим за перемещениями ЦМ C и его оси вращения OC . Координаты ЦМ C определяются по формулам: $x_c = (m_1 x_1 + m_2 x_2) / (m_1 + m_2 + m_3)$,

$$y_c = (m_1 y_1 + m_2 y_2) / (m_1 + m_2 + m_3), \quad z_c = (m_1 z_1 + m_2 z_2) / (m_1 + m_2 + m_3).$$

Для получения на экране ПЭВМ трехмерного изображения траектории ЦМ C необходимо перейти от его пространственных координат x, y, z к координатам экрана X и Y (рис. 7.21.2) следующим образом:

$$X = (y_C - x_C) \cos \varepsilon, \quad Y = z_C - (x_C + y_C) \sin \varepsilon, \quad \varepsilon = \pi / 8.$$

Угол прецессии φ и угол нутации θ находят по формулам:

$$\cos \varphi = x_C / \sqrt{x_C^2 + y_C^2}, \quad \sin \varphi = \sqrt{1 - \cos^2 \varphi}, \quad \varphi = \text{arctg}(\sin \varphi / \cos \varphi),$$

$$\cos \theta = z_C / l_0 \sin(\pi / 3), \quad \sin \theta = \sqrt{1 - \cos^2 \theta}, \quad \theta = \text{arctg}(\sin \theta / \cos \theta).$$

Все это позволяет построить графики зависимостей φ и θ от времени. Используется программа ПР-13. Результаты расчетов движения центра масс C приведены на рис. 7.22. Рассмотренная модель также позволяет изучить сферическое движение твердого тела в нестационарном поле.

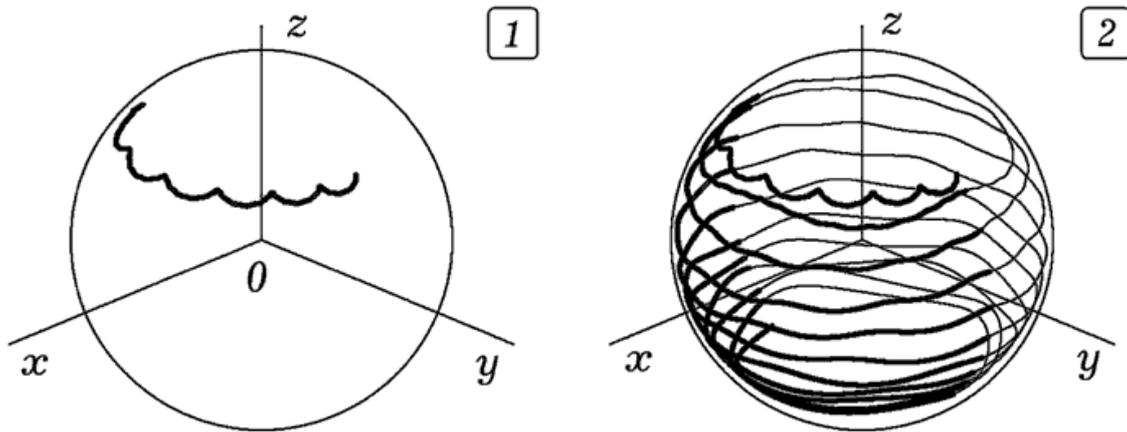


Рис. 7.22. Движение оси гироскопа в однородном поле тяжести

Приложение к главе 7

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

Программа ПР-1

```
Uses crt, graph;
Var x,v,a,dt,t : Real; Gd,Gm,i : Integer;
BEGIN Gd:= Detect; InitGraph(Gd,Gm,'c:\bp\bgi');
dt:=0.0015; x:=-200; v:=-15; line(0,300,640,300);
For i:=1 to 30000 do begin t:=t+dt;
  If t<8 then a:=1*t else If t<30 then a:=0.2 else a:=-4;
  x:=x+v*dt; v:=v+a*dt;
  circle(30+round(t*12),300-round(x/2),1);
  circle(30+round(t*12),300-round(v*10),1);
  circle(30+round(t*12),300-round(a*10),2); end;
Repeat until KeyPressed; CloseGraph;
END.
```

Программа ПР-2

```
Uses crt, graph;
Const m=2; r=0; k=0; Fm=10; dt=0.0001;
Mx=20; Mv=40; Ma=8; Mt=100;
Var x,v,a,F,t : Real; j,Gd,Gm : Integer;
BEGIN Gd:= Detect; InitGraph(Gd,Gm,'c:\bp\bgi');
```

```
t:=0; v:=0; x:=-3;
line(30,300,650,300); line(31,500,31,10);
Repeat t:=t+dt; F:=0;
  If t<1 then F:= Fm; If t>3 then F:=-Fm;
  a:=(F-r*v-k*x)/m; x:=x+v*dt; v:=v+a*dt;
  circle(30+round(t*Mt),300-round(x*Mx),1);
  circle(30+round(t*Mt),300-round(v*Mv),1);
  circle(30+round(t*Mt),300-round(a*Ma),2);
until KeyPressed; CloseGraph;
END.
```

Программа ПР-3

```
Uses crt,graph;
Var v,F,Fx,Fy,x,y,vx,vy,ax,ay: real; Gd,Gm: integer;
Const m=1; dt=0.002; r=0.05; v_vetra=4;
BEGIN Gd:= Detect; InitGraph(Gd,Gm,'c:\bp\bgi');
line(0,450,640,450); x:=0; y:=0; vx:=30; vy:=40; Fy:=-1;
Repeat ax:=(Fx-r*v_x-r*v_vetra)/m; ay:=(Fy-r*v_y)/m;
  vx:=vx+ax*dt; vy:=vy+ay*dt; x:=x+vx*dt; y:=y+vy*dt;
  circle(round(x)+10,450-round(y),2);
until KeyPressed; CloseGraph;
END.
```

Программа ПР-4

```
Uses crt,graph;
Var v,F,Fx,Fy,r,r2,dS,x,y,x1,y1,r1,vx,vy,ax,ay,t,AC: real;
Gd,Gm: integer;
Const M=1500; mm=200; dt=0.002; rr=0;
BEGIN Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
line(320,240,640,240); line(320,240,320,0);
circle(320,240,5); x:=100; y:=120; vx:=1; vy:=-2;
Repeat t:=t+dt;
  r:=sqrt(x*x+y*y); F:=M*mm/(r*r); Fx:=-F*x/r; Fy:=-F*y/r;
  ax:=(Fx-rr*v_x)/mm; ay:=(Fy-rr*v_y)/mm; vx:=vx+ax*dt;
  vy:=vy+ay*dt; x:=x+vx*dt; y:=y+vy*dt;
  If t>1 then dS:=sqrt(sqr(x-x1)+sqr(y-y1)-sqr(r-r2))*r/2;
  r2:=r; x1:=x; y1:=y; circle(round(t),430-round(r),1);
  circle(round(t),430-round(10*sqrt(vy*vy+vx*vx)),1);
  circle(round(t),430-round(dS/dt),1);
```

```

    circle(round(x)+320,240-round(y),1);
until KeyPressed; CloseGraph;
END.

```

Программа ПР-5

```

Uses crt, graph;
Const m=5; q=0.5; B=5; E=10; dt=0.005; rr=0.05;
Var v,F,Fx,Fy,r,x,y,vx,vy,ax,ay: real; Gd,Gm: integer;
BEGIN Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
x:=0; y:=120; vx:=30; vy:=0; setcolor(15);
Repeat F:=B*v*q; v:=sqrt(vx*vx+vy*vy);
    Fx:=F*vy/v; Fy:=E*q-F*v*vx/v; ax:=(Fx-rr*v*vx)/m;
    ay:=(Fy-rr*v*vy)/m; vx:=vx+ax*dt; vy:=vy+ay*dt;
    x:=x+vx*dt; y:=y+vy*dt;
    circle(round(x)+100,240-round(y),2);
until KeyPressed; CloseGraph;
END.

```

Программа ПР-6

```

uses crt, graph;
const r=100; g=10; rs=0.00; dx=0.001;
    dt=0.0005; da=0.0005; pi=3.1415926;
var c,s,a,b,x1,x2,y1,y2,atau,v,vx,vy,x,y,t: real;
    Gd,Gm,n : integer; tt : string;
BEGIN Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
a:=0.5{2.2}; x:=r*(a-sin(a)); y:=r*(1-cos(a));
line(120,200,500,200); line(120,200,120,450);
Repeat c:=1-y/r; s:=sqrt(1-c*c);
    If y<=r then a:=arctan(s/c) else a:=pi-arctan(abs(s/c));
    x1:=r*(a-sin(a)); y1:=r*(1-cos(a)); a:=a+da;
    x2:=r*(a-sin(a)); y2:=r*(1-cos(a));
    b:=pi/2-arctan((y2-y1)/(x2-x1)); atau:=g*cos(b)-rs*v;
    v:=v+atau*dt; x:=x+v*sin(b)*dt; y:=y+v*cos(b)*dt;
    If n mod 100=0 then begin
        Circle(120+round(x),200+round(y),1); end;
        Circle(120,200,2); t:=t+dt; inc(n);
until (KeyPressed) or (a>pi);
Str(round(t*1000),tt); OutTextXY(10,10,tt); Readkey;
END.

```

Программа ПР-7

```

uses crt, graph;
const r=200; m=1; mu=0.2; g=10; dt=0.001;
      dx=0.1; pi=3.1415926;
var a,b,x1,x2,y1,y2,atau,v,vx,vy,x,y,t : real;
      Gd, Gm,n : integer; tt : string;
BEGIN Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
b:=0.015; x:=-160; y:=b*x*x;
Repeat inc(n); t:=t+dt;
      If x<30 then
          begin x1:=x; y1:=b*x1*x1; x2:=x+dx; y2:=b*x2*x2;
              a:=arctan((y1-y2)/(x2-x1));
              atau:=g*(sin(a)-mu*cos(a));
              v:=v+atau*dt; x:=x+v*cos(a)*dt;
              y:=y-v*sin(a)*dt; vx:=v*cos(a); vy:=-v*sin(a);
              end else begin vy:=vy-g*dt;
                  x:=x+vx*dt; y:=y+vy*dt; end;
          If x<30 then
              circle(120+round(0.5*x),280-round(0.5*y),1);
          If n mod 300=0 then
              circle(120+round(0.5*x),280-round(0.5*y),3);
until (KeyPressed) or ((y<0) and (y<-sqrt(50*x-3000))); x:=60;
Repeat x:=x+1; y:=-sqrt(50*x-3000);
      circle(120+round(0.5*x),280-round(0.5*y),1);
until x>1000;
Str(round(t*1000),tt); OutTextXY(40,330,tt); Readkey;
END.

```

Программа ПР-8

```

uses crt, graph;
var Ftr,x1,x2,v1,v2,a1,a2,xx1,xx2,t: real;
      Gd,Gm,i: integer;
const m1=4; m2=1; dt=0.005; r=0.1{0.3};
BEGIN
      Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
      line(220,0,220,40); line(0,350,640,350);
      x1:=250; v1:=-4; x2:=0; v2:=10;
      Repeat

```

```

If abs(x2-x1)<40 then Ftr:=r*abs(v2-v1) else Ftr:=0;
a1:=Ftr/m1; v1:=v1+a1*dt; x1:=x1+v1*dt; t:=t+dt;
a2:=-Ftr/m2; v2:=v2+a2*dt; x2:=x2+v2*dt;
setcolor(8);
Rectangle(round(xx1)+220-40,20,round(xx1)+260,40);
Circle(round(xx2)+220,30,3);
setcolor(15);
Rectangle(round(x1)+220-40,20,round(x1)+260,40);
Circle(round(x2)+220,30,3);
delay(15); xx1:=x1; xx2:=x2;
Circle(round(t*10),350-round(15*v1),1);
Circle(round(t*10),350-round(15*v2),1);
until KeyPressed; CloseGraph;
END.

```

Программа ПР-9

```

Uses crt, graph; Const N=2; dt=0.0005;
Var m,x,y,vx,vy,Fx,Fy : array[0..N+1] of real;
    Gd,Gm,i,j,k: integer; xc,yc,ax,ay,F,l,t : real;
Procedure Sila;
Label mm;
begin For i:=1 to N do begin Fx[i]:=0; Fy[i]:=0; end;
    For i:=1 to N do For j:=1 to N do begin
        if i=j then goto mm;
        l:=sqrt(sqr(x[i]-x[j])+sqr(y[i]-y[j]));
        If l<1 then l:=1; F:=-5000/l/l;
        Fx[i]:=Fx[i]+F*(x[i]-x[j])/l;
        Fy[i]:=Fy[i]+F*(y[i]-y[j])/l; mm: end; end;
BEGIN Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
    m[1]:=0.2; m[2]:=0.1; x[1]:=10; y[1]:=200;
    x[2]:=600; y[2]:=230; vx[1]:=40; vx[2]:=-30;
Repeat Sila; t:=t+dt; inc(k);
    For i:=1 to N do begin ax:=Fx[i]/m[i]; ay:=Fy[i]/m[i];
        vx[i]:=vx[i]+ax*dt;    vy[i]:=vy[i]+ay*dt;
        x[i]:=x[i]+vx[i]*dt;    y[i]:=y[i]+vy[i]*dt; end;
    If k mod 1000=0 then begin k:=0;
        circle(round(x[1]),round(y[1]),1);
        circle(round(x[2]),round(y[2]),2); end;
until (KeyPressed) or (t>50);

```

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

Программа ПР-10

```
Uses crt,graph; Const N=4; dt=0.001;
Var m,Fx,Fy,x,y,vx,vy,xx,yy: array[1..N] of real;
    Gd,Gm,i,j: integer; ax,ay,F,l: real;
Procedure Sila;
label Metka;
begin For i:=1 to N do begin Fx[i]:=0; Fy[i]:=0; end;
For i:=1 to N do For j:=1 to N do begin
    If j=i then goto Metka;
    l:=sqrt(sqr(x[i]-x[j])+sqr(y[i]-y[j])); If l<1 then l:=1;
    F:=-200*m[i]*m[j]/l/l; Fx[i]:=Fx[i]+F*(x[i]-x[j])/l;
    Fy[i]:=Fy[i]+F*(y[i]-y[j])/l; Metka: end; end;
BEGIN Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
m[1]:=100; x[1]:=320; y[1]:=240; vx[1]:=-0.4;
m[2]:=1; x[2]:=320; y[2]:=200; vx[2]:=22;
m[3]:=0.6; x[3]:=320; y[3]:=150; vx[3]:=14;
m[4]:=0.5; x[4]:=320; y[4]:=120; vx[4]:=13;
Repeat Sila; For i:=1 to N do begin xx[i]:=x[i];
    yy[i]:=y[i]; ax:=Fx[i]/m[i]; ay:=Fy[i]/m[i];
    vx[i]:=vx[i]+ax*dt; vy[i]:=vy[i]+ay*dt;
    x[i]:=x[i]+vx[i]*dt; y[i]:=y[i]+vy[i]*dt; end;
    setcolor(8); For i:=1 to N do
        circle(round(xx[i]),round(yy[i]),2);
    setcolor(15); For i:=1 to N do
        circle(round(x[i]),round(y[i]),2);
until KeyPressed; CloseGraph;
END.
```

Программа ПР-11

```
Uses crt, graph; Const N=2; dt=0.0005;
Var m,x,y,vx,vy,Fx,Fy : array[0..N+1] of real;
    Gd,Gm,i,j,k: integer; xc,yc,ax,ay,F,l,t : real;
Procedure Sila;
Label mm;
begin For i:=1 to N do begin Fx[i]:=0; Fy[i]:=0; end;
    For i:=1 to N do For j:=1 to N do begin
```

```

if i=j then goto mm;
l:=sqrt(sqr(x[i]-x[j])+sqr(y[i]-y[j])); If l<1 then l:=1;
If (t>3)and(t<3.6) then F:=50/l else F:=0;
Fx[i]:=Fx[i]+F*(x[i]-x[j])/l;
Fy[i]:=Fy[i]+F*(y[i]-y[j])/l+m[i]*10; mm: end; end;
BEGIN Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
m[1]:=0.05; m[2]:=0.1; x[1]:=12; y[1]:=100;
x[2]:=10; y[2]:=102; vx[1]:=40; vx[2]:=40;
Repeat Sila; t:=t+dt; inc(k);
For i:=1 to N do begin ax:=Fx[i]/m[i]; ay:=Fy[i]/m[i];
vx[i]:=vx[i]+ax*dt; vy[i]:=vy[i]+ay*dt;
x[i]:=x[i]+vx[i]*dt; y[i]:=y[i]+vy[i]*dt; end;
If k mod 200=0 then For i:=1 to N do begin k:=0;
circle(round(x[i]),round(y[i]),2);
xc:=(m[1]*x[1]+m[2]*x[2])/(m[1]+m[2]);
yc:=(m[1]*y[1]+m[2]*y[2])/(m[1]+m[2]);
circle(round(xc),round(yc),2);
line(round(x[1]),round(y[1]),round(x[2]),round(y[2]));
end;
until (KeyPressed)or(t>50);
Repeat until KeyPressed; CloseGraph;
END.

```

Программа ПР-12

```

uses crt, graph;
const m1=0.2; m2=0.1; k=1000; r=0.005;
b=0.9; l0=40; dt=0.001;
var l,x1,x2,y1,y2,vx1,vy1,vx2,vy2,F,Fx1,Fy1,Fx2,Fy2 : real;
Gd,Gm: integer; n: Longint;
BEGIN Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
x1:=0; y1:=300; x2:=0; y2:=260; vx1:=50; vy1:=20;
rectangle(2,2,572,442);
Repeat l:=sqrt(sqr(x1-x2)+sqr(y1-y2)); F:=k*(l-l0);
Fx1:=F*(x2-x1)/l-r*vx1; Fy1:=F*(y2-y1)/l-r*vy1-1*m1;
Fx2:=-F*(x2-x1)/l-r*vx2; Fy2:=-F*(y2-y1)/l-r*vy2-1*m2;
vx1:=vx1+Fx1/m1*dt; vy1:=vy1+Fy1/m1*dt;
vx2:=vx2+Fx2/m2*dt; vy2:=vy2+Fy2/m2*dt;
x1:=x1+vx1*dt; y1:=y1+vy1*dt;
x2:=x2+vx2*dt; y2:=y2+vy2*dt;

```

```

If x1>550 then vx1:=-b*vx1; If x2>550 then vx2:=-b*vx2;
If y1<-0 then vy1:=-b*vy1; If y2<-0 then vy2:=-b*vy2;
circle(20+round(x2),440-round(y2),1);
If n mod 1000=0 then begin
    line(20+round(x1),440-round(y1),
        20+round(x2),440-round(y2));
    circle(20+round(x1),440-round(y1),3); end; inc(n);
until (KeyPressed)or(n>100000); Repeat until KeyPressed;
END.

```

Программа ПР-13

```

uses crt, graph;
const N=3; dt=0.001; Ms=40; pi=3.1415926;
    r=0.006; teta0=0.8; l0=5;
var m,Fx,Fy,Fz,x,y,z,vx,vy,vz,xx,yy: array[1..N] of real;
    Gd,Gm,i,j,k: integer; xc,yc,zc,ax,ay,az,F,l,t: real;
    teta,cost,sint,fi,cosfi,sinfi,Mx,My: real;
Procedure Sila;
begin
For i:=1 to N do begin Fx[i]:=0; Fy[i]:=0; Fz[i]:=0; end;
For i:=1 to N do for j:=1 to N do begin
    l:=sqrt(sqr(x[i]-x[j])+sqr(y[i]-y[j])+sqr(z[i]-z[j]));
    If l<>0 then begin F:=25000*(l0-l);
        Fx[i]:=Fx[i]+F*(x[i]-x[j])/l;
        Fy[i]:=Fy[i]+F*(y[i]-y[j])/l;
        Fz[i]:=Fz[i]+F*(z[i]-z[j])/l; end; end; end;
Procedure Nach_uslov;
begin m[1]:=2; m[2]:=2.1; m[3]:=0.01;
x[1]:=0; y[1]:=l0*sin(teta0-pi/6); z[1]:=l0*cos(teta0-pi/6);
x[2]:=0; y[2]:=l0*sin(teta0+pi/6); z[2]:=l0*cos(teta0+pi/6);
vx[1]:=-4; vx[2]:=3.5; end;
BEGIN Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
line(320,0,320,240); line(320,240,round(320+300*cos(pi/8)),
round(240+300*sin(pi/8))); line(320,240,
round(320-300*cos(pi/8)),round(240+300*sin(pi/8)));
circle(320,240,210); Nach_uslov;
Repeat Sila; For i:=1 to 2 do begin
    ax:=(Fx[i]-r*vxi)/m[i]; ay:=(Fy[i]-r*vyi)/m[i];
    az:=(Fz[i]-r*vzi)/m[i]-0.05; vxi:=vxi+ax*dt;

```

```

vy[i]:=vy[i]+ay*dt; vz[i]:=vz[i]+az*dt;
x[i]:=x[i]+vx[i]*dt; y[i]:=y[i]+vy[i]*dt;
z[i]:=z[i]+vz[i]*dt;
end; inc(k); t:=t+dt;
If k mod 30=0 then begin k:=0;
xc:=(m[1]*x[1]+m[2]*x[2])/(m[1]+m[2]+m[3]);
yc:=(m[1]*y[1]+m[2]*y[2])/(m[1]+m[2]+m[3]);
zc:=(m[1]*z[1]+m[2]*z[2])/(m[1]+m[2]+m[3]);
Mx:=cos(pi/8); My:=sin(pi/8);
If xc>0 then circle(320+round(Ms*(yc-xc)*Mx*0.9),
                    240-round(Ms*(zc-(xc+yc)*My)),2);
If xc<0 then circle(320+round(Ms*(yc-xc)*Mx*0.9),
                    240-round(Ms*(zc-(xc+yc)*My)),1);
cost:=zc/10/0.87; sint:=sqrt(1-cost*cost);
teta:=arctan(sint/cost);
circle(round(t/1),440-round(teta*200),1);
circle(round(t/1),441-round(teta*200),1);
cosfi:=xc/sqrt(xc*xc+yc*yc); sinfi:=sqrt(1-cosfi*cosfi);
fi:=arctan(sinfi/cosfi);
circle(round(t/1),240-round(fi*100),1); end;
until KeyPressed; CloseGraph;
END.

```

Список литературы

1. Булавин Л. А., Выгорницкий Н. В., Лебовка Н. И. Компьютерное моделирование физических систем. Долгопрудный: Издательский Дом "Интеллект", 2011. 352 с.
2. Гулд Х., Тобочник Я. Компьютерное моделирование в физике. В 2 ч. Ч. 1. М.: Мир, 1990. 350 с.
3. Ильина В. А., Силаев П. К. Численные методы для физиков-теоретиков. II. Москва-Ижевск: Институт компьютерных исследований, 2004. 118 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. Поттер Д. Вычислительные методы в физике. М.: Мир, 1975. 392 с.
8. Самарский А. А., Михайлов А. П. Математическое моделирование: Идеи. Методы. Примеры. М.: Физматлит, 2001. 320 с.
9. Тихонов А. Н., Самарский А. А. Уравнения математической физики. М.: Наука, 1966. 724 с.
10. Федоренко Р. П. Введение в вычислительную физику: учеб. пособие для вузов. М.: Изд-во Моск. физ.-техн. ин-та, 1994. 528 с.

[ВВЕРХ](#)