

Майер Р.В., Глазовский пединститут  
**КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ:**  
**8. КОЛЕБАТЕЛЬНОЕ И ВОЛНОВОЕ ДВИЖЕНИЕ**

Настоящая глава посвящена изучению методов моделирования колебательного и волнового движения. В ней рассмотрены компьютерные модели свободных и вынужденных колебаний, автоколебаний, распространение возмущений в упругой среде [1-9]. Кроме того, проанализированы решения волнового уравнения для одномерной и двумерной сред, а также уравнения синус-Гордона.

**8.1. Свободные колебания**

Механические колебания -- это периодические движения тела, при котором его координата через равные промежутки времени принимает примерно одинаковые значения. Допустим имеется механическая система с одной степенью свободы, состоящая из груза (инерционного элемента) массой  $m$ , пружины (упругого элемента) жесткостью  $k$  и вязкой среды (диссипативного элемента) с коэффициентом сопротивления  $r$ . Сила сопротивления пропорциональна скорости:  $\vec{F}_c = -r\vec{v}$ . Известны начальные координата и скорость груза, а также действующая на него внешняя сила  $F(\tau)$ . Необходимо определить координату  $x(\tau)$ , скорость  $v(\tau)$  и ускорение  $a(\tau)$  груза в последующие моменты времени [6].

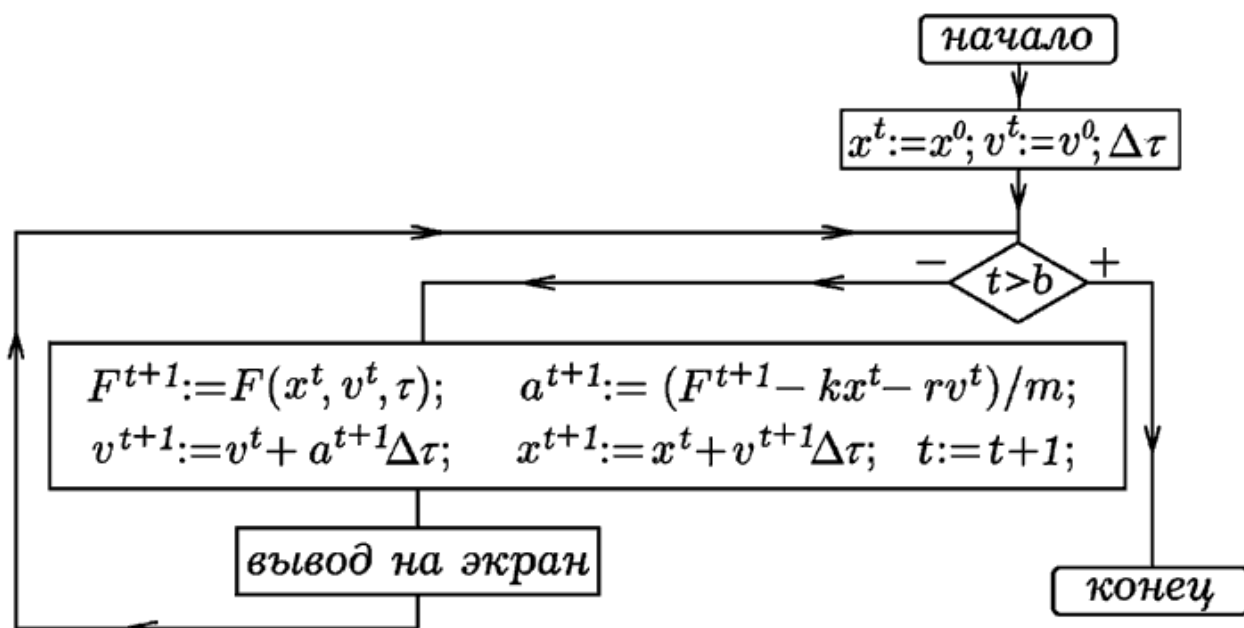


Рис. 1. Моделирование колебательной системы.

Свободные затухающие колебания в линейной системе описываются однородным дифференциальным уравнением второго порядка:  $m\ddot{x} + r\dot{x} + kx = 0$ . Их частота не зависит от амплитуды. Дифференциальное уравнение вынужденных колебаний является неоднородным и имеет вид:  $m\ddot{x} + r\dot{x} + kx = F \sin \omega\tau$ . Оно охватывает широкий класс физических явлений различной природы, например, процессы, происходящие в колебательном контуре, состоящем из резистора, конденсатора и катушки индуктивности.

Про моделируем колебания пружинного маятника. По второму закону Ньютона  $ma_x = -kx - rv_x$ . Отсюда следует  $a_x = (-kx - rv_x)/m$ . Перейдем от непрерывной области  $\Omega_{x,\tau}$  изменения аргумента  $\tau$  к дискретной области  $\Omega_{\Delta x, \Delta\tau}$ . Для координаты и скорости в конечных разностях получаем:  $v_x^{t+1} = v_x^t + a_x^{t+1} \Delta\tau$ ,  $x^{t+1} = x^t + v_x^{t+1} \Delta\tau$ . Алгоритм содержит цикл по  $t$  (рис. 1), в котором будут вычисляться координата, скорость и ускорение в дискретные моменты  $t$  и строиться графики  $x(\tau)$ ,  $v_x(\tau)$ ,  $a_x(\tau)$ , а также фазовая кривая. Используется программа ТР-1. В случае слабого затухания ( $r/2m < \omega_0 = \sqrt{k/m}$ ), система совершает почти гармонические колебания, амплитуда которых уменьшается по экспоненциальному закону (рис. 2.1). В случае сильного затухания ( $r/2m > \omega_0 = \sqrt{k/m}$ ) движение системы аperiodическое (рис. 2.2).

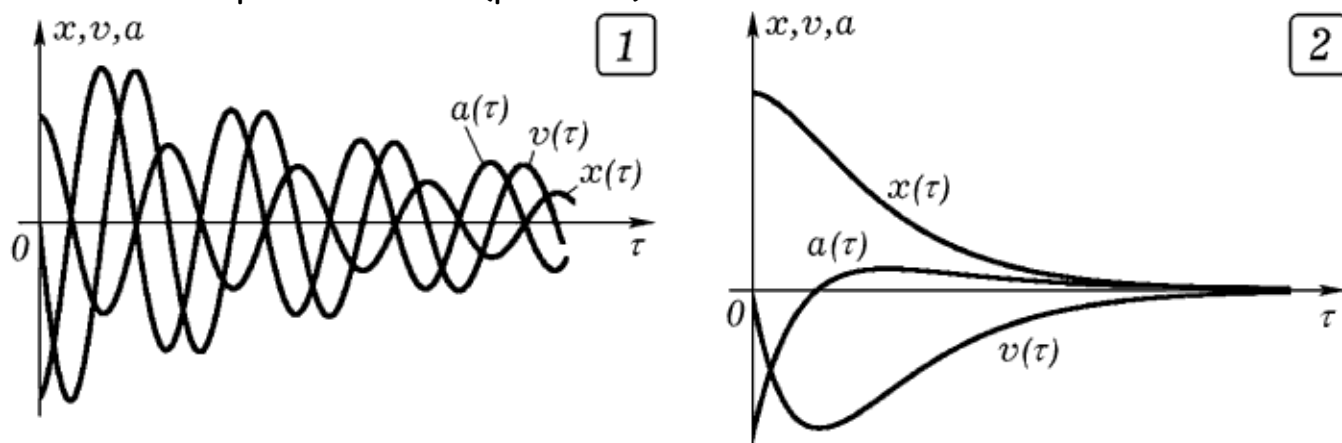


Рис. 2. Затухающие колебания и аperiodическое движение.

Движение колебательной системы удобно анализировать на фазовой плоскости в осях  $x, v$  (или  $x, p$ ). Чтобы построить фазовую кривую, необходимо активизировать закомментированный оператор. На рис. 3 показаны фазовые кривые для случаев: 1) незатухающие колебания (рис. 3.1); 2) линейные и нелинейные колебания математического маятника (рис. 3.2); 3) затухающие колебания с сильным затуханием (рис. 3.3); 4) затухающие колебания со слабым затуханием (рис. 3.4).

Примером нелинейной колебательной системы является математический маятник. Его движение описывается уравнениями:

$$I\epsilon = -mgl \sin \varphi - r\dot{\varphi}, \quad \epsilon^{t+1} = (-mgl \sin \varphi^t - r\omega^t) / I,$$

$$\omega^{t+1} = \omega^t + \epsilon^{t+1} \Delta\tau, \quad \varphi^{t+1} = \varphi^t + \omega^{t+1} \Delta\tau.$$

После внесения небольших изменений в программу ПР-1, можно получить фазовые кривые (рис. 3.2) и определить периоды колебаний при различных значениях амплитуды.

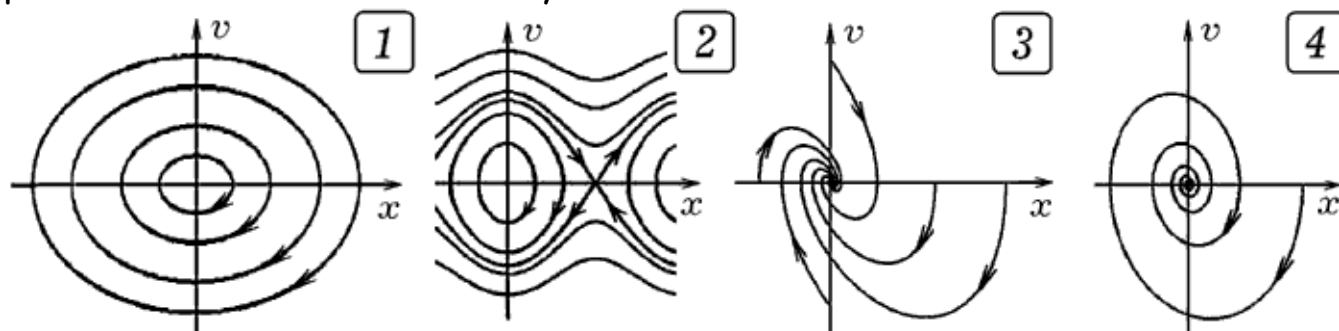


Рис. 3. Фазовые кривые различных колебательных систем.

## 8.2. Вынужденные колебания. Автоколебания

Про моделируем колебания пружинного маятника, происходящие под действием внешней гармонической силы. Из второго закона Ньютона:  $ma_x = -kx - rv_x + F_m \sin \omega\tau$ ,  $a_x = (-kx - rv_x + F_m \sin \omega\tau) / m$ . Во остальном задача решается аналогично, используется программа ПР-1.

Эта модель позволяет исследовать зависимость амплитуды вынужденных колебаний от частоты при различных коэффициентах затухания и получить **резонансные кривые**. Для этого необходимо с некоторым шагом изменять частоту  $\omega$  вынуждающей силы вблизи собственной частоты колебательной системы  $\omega_0 = \sqrt{k/m}$ , и определять соответствующие ей значения амплитуды установившихся колебаний. Это позволит получить амплитудно-частотную характеристику колебательной системы. Если до начала действия вынуждающей силы система покоилась, то какое-то время будет происходить **релаксационный процесс**, в ходе которого система из состояния покоя перейдет в установившийся колебательный режим с некоторой постоянной амплитудой. Графики вынужденных колебаний  $x(\tau)$  для системы с параметрами  $k = 100$  Н/м,  $m = 1$  кг (резонансная частота  $\omega_0 = 10$  рад/с) и  $r = 0,2$  Нс/м или  $r = 0,15$  Нс/м, при частоте вынуждающей силы  $\omega = 11$  рад/с изображены на рис. 4.1. Видно, что в про-

цессе перехода системы в установившийся режим, амплитуда периодически изменяется, причем ее колебания затухают [6].

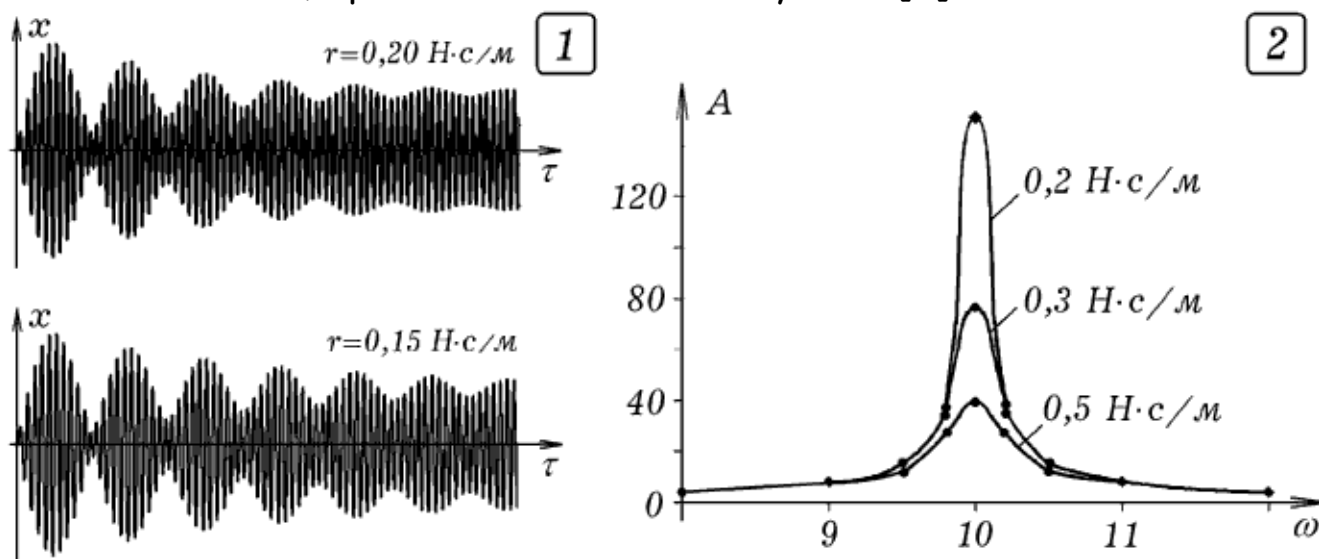


Рис. 4. Моделирование вынужденных колебаний.

Резонансные кривые при различных  $r$  представлены на рис. 4.2. Как это следует из теории, на резонансной частоте амплитуда колебаний достигает максимального значения, а по мере удаления от резонанса она довольно быстро уменьшается. Если коэффициент сопротивления мал, то добротность велика, резонансная кривая имеет ярко выраженный максимум. При низкой добротности резонанс выражен слабо. Аналогично можно промоделировать вынужденные колебания, происходящие под действием негармонической силы, а также переходные процессы.

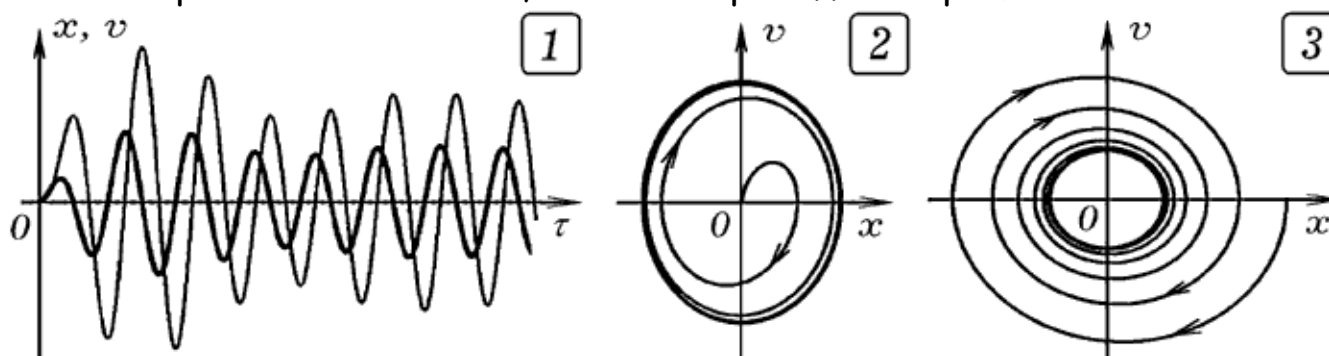


Рис. 5. График вынужденных колебаний и фазовые кривые.

Для вынужденных колебаний и автоколебаний характерно, что амплитуда установившихся колебаний не зависит от начальных условий. Независимо от начальных условий (координаты и скорости) фазовая кривая вынужденных колебаний при  $\tau \rightarrow \infty$  стремится к некоторому **предельному циклу**. Чтобы убедиться в этом, достаточно построить фазовую кривую вынужденных колебаний при различных начальных координате и

скорости. Видно, что фазовая кривая системы стремится к **аттрактору**, -- некоторому предельному циклу, соответствующему колебаниям с установившейся амплитудой (рис. 5.2 и 5.3).

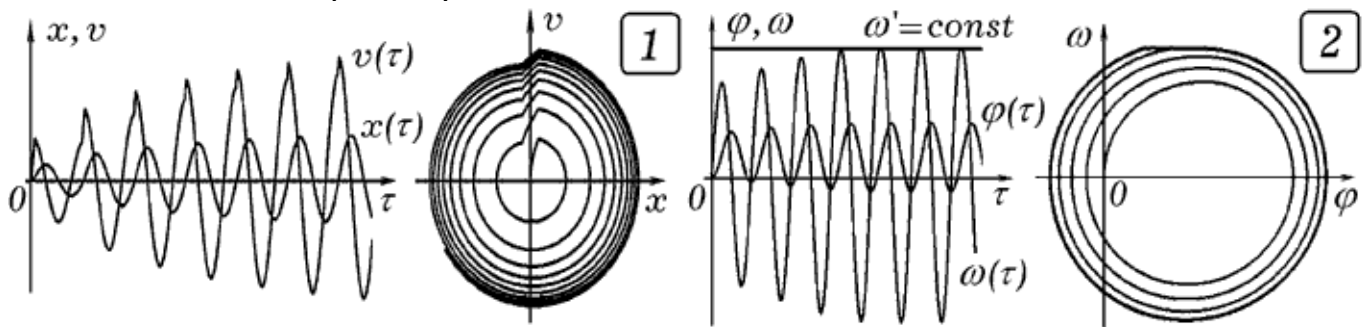


Рис. 6. Моделирование автоколебательной системы.

Изучим **автоколебательную систему**, которая посредством положительной обратной связи сама регулирует поступление энергии от источника. Пусть вблизи положения равновесия ( $-b < x < b$ ) когда тело движется в направлении оси  $Ox$ , на него действует постоянная сила  $F_m$ , увеличивающая его скорость. Используется программа ПР-1; в цикл по времени  $t$  надо включить условный оператор, присваивающий переменной  $F$  некоторое значение  $F_m$ , когда  $|x| < b$  и  $v_x > 0$ . На графике и фазовой кривой (рис. 6.1) хорошо видно, как один раз за период система получает порцию энергии и скорость тела увеличивается. Колебания незатухающие, фазовая кривая стремится к замкнутому аттрактору.

Теперь промоделируем автоколебания **фрикционного маятника Фроуда**, состоящего из физического маятника, расположенного на вращающемся валу. Пусть момент силы трения между валом и маятником убывает по экспоненциальному закону  $M_{mp} = M_0 \exp(-\beta |\omega' - \dot{\phi}|)$ , где  $\omega'$  и  $\omega = \dot{\phi}$  -- скорости вала и маятника. Автоколебания объясняются тем, что при движении в направлении вращения на маятник со стороны вала действует сила трения, превышающая силу трения, действующую при обратном движении. Средний подталкивающий момент сил трения превышает тормозящий, в систему периодически поступает энергия. Графики  $\varphi(\tau)$ ,  $\omega(\tau)$  и фазовый портрет показаны на рис. 6.2. Программа ПР-2 отличается тем, что содержит условный оператор: если скорость точек на ободке вала  $v'$  больше скорости маятника  $\omega = \dot{\phi}$ , то  $M_{mp} = M_0 \exp(-\beta |\omega' - \dot{\phi}|)$ , а иначе  $M_{mp} = -M_0 \exp(-\beta |\omega' - \dot{\phi}|)$ .

### 8.3. Моделирование колебаний сложных систем

Рассмотренный выше подход позволяет исследовать колебания более сложных систем. Проанализируем несколько задач.

**Задача 1.** На тележке массой  $m_1$  подвешен маятник, состоящий из тела массой  $m_2$  и нити длиной  $l$  (рис. 7). Маятник выводят из положения равновесия и отпускают. На систему действует сила вязкого трения. Напишите программу, моделирующую затухающие колебания системы.

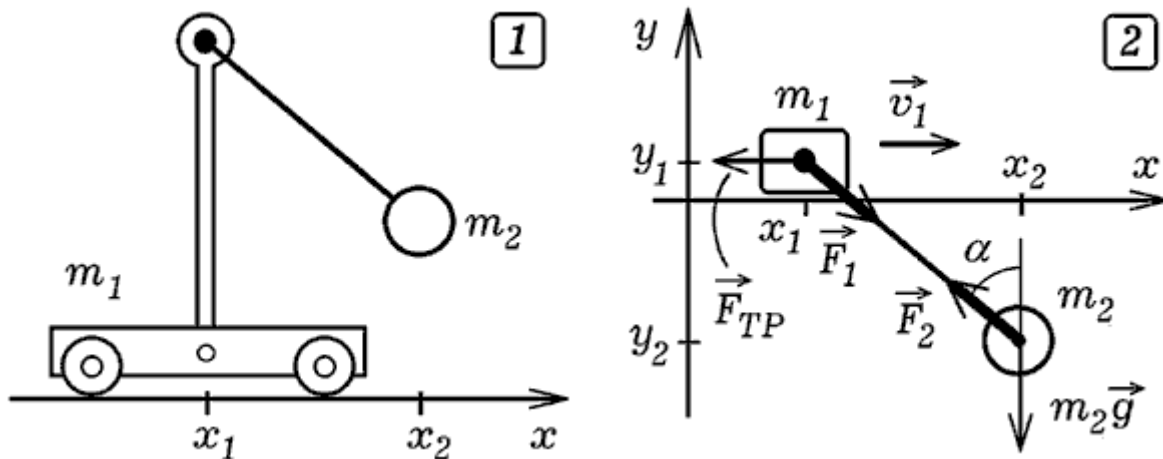


Рис. 7. Движение тележки с маятником.

Заменяем систему "маятник-тележка" системой, состоящей из двух материальных точек  $m_1$  и  $m_2$ , связанных упругим стержнем жесткостью  $k$  и длиной  $l_0$ . Материальная точка  $m_1$  способна скользить по горизонтальной линии так, что ее координата  $y$  остается постоянной. При этом на нее действует сила вязкого трения, направленная противоположно скорости и пропорциональная ее величине. Проекции сил, действующих на точки системы, вычисляются по формулам [5]:  $F_1 = F_2 = F = k(l - l_0)$ ,

$$F_{1x} = F \sin \alpha - r v_{1x} = F(x_2 - x_1)/l - r v_{1x}, \quad F_{2x} = -F \sin \alpha = -F(x_2 - x_1)/l,$$

$$F_{2y} = F \cos \alpha - m_2 g = -F(y_2 - y_1)/l - m_2 g.$$

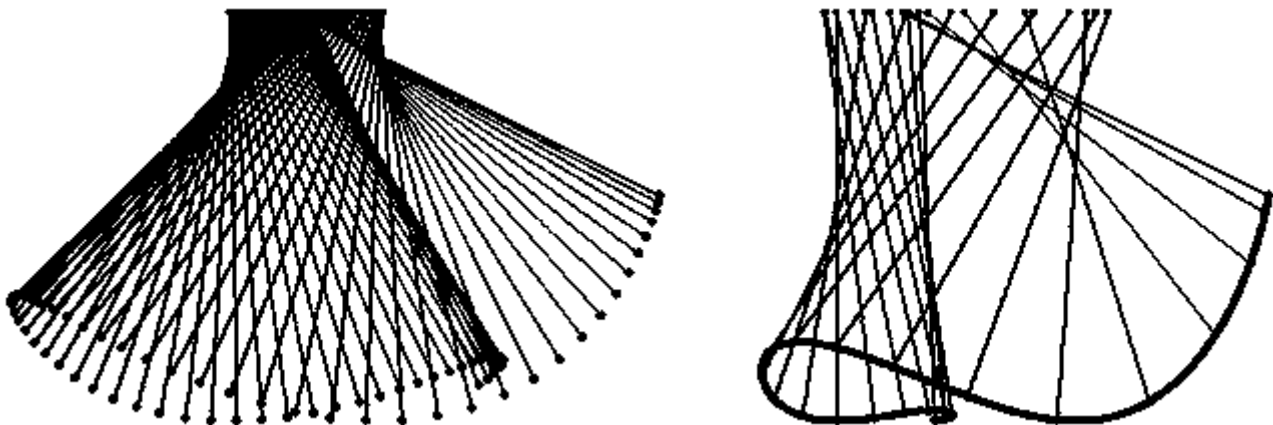


Рис. 8. Колебания маятника на тележке.

В программе ПР-3 осуществляется расчет действующих на точки сил, а также проекций их ускорений, скоростей и координат в дискретные моменты времени. Результаты моделирования представлены на рис. 8.

**Задача 2.** Про моделируйте движение физического маятника, точка подвеса которого колеблется в вертикальном направлении. Изучите колебания маятника вблизи верхнего положения равновесия [5].

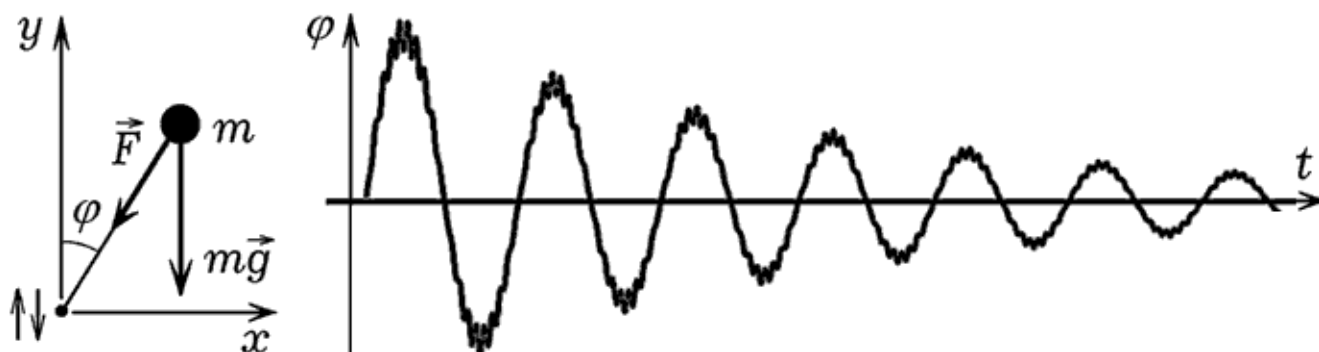


Рис. 9. Моделирование колебаний маятника Капицы.

Речь идет о маятнике Капицы, который представляет собой стержень с грузиком, подвешенный за один конец к вибратору. Если точка подвеса колеблется по вертикали с достаточно большой частотой и амплитудой, то маятник приобретает способность совершать колебания относительно вертикального положения равновесия. Вместо массивного стержня будем рассматривать две материальные точки, связанные невесомым упругим стержнем с большой жесткостью  $k$  (рис. 9). Одна материальная точка колеблется вдоль вертикальной оси  $OY$  с заданной частотой, от ее массы ничего не зависит. Расчет движения анализируемого маятника Капицы сводится к определению координат  $x$ ,  $y$  и скоростей  $v_x$ ,  $v_y$  второй точки. В программе ПР-4 осуществляется моделирование движения маятника длиной  $L=20$  см, массой 10 г, с частотой колебаний точки подвеса  $\omega=30$  Гц, амплитудой колебаний  $A=2$  см. Из теории следует, что частота колебаний маятника вблизи верхнего положения равновесия равна:  $\omega' = \sqrt{A^2 \omega^2 / 2L^2 - \omega_0^2}$ , где  $\omega_0$  -- его собственная частота. Компьютерная модель на качественном уровне подтверждает эту формулу. Получающийся график  $\varphi(t)$  показан на рис. 9.

**Задача 3.** Рассмотрим систему из  $N$  материальных точек, связанных друг с другом жесткими невесомыми стержнями длиной  $b$ , находящуюся в поле тяжести. Необходимо рассчитать ее движение в воздухе, если дует

ветер со скоростью  $v_в$ , а материальная точка  $m_1$  совершает колебания вдоль оси  $Ox$  или  $Oy$  по заданному закону.

Из второго закона Ньютона  $m_i \vec{a}_i = \vec{F}_{i1} + \vec{F}_{i2} + \vec{F}_{ic} + \vec{F}_{iв} + m_i \vec{g}$ . Будем считать стержни упругими с жесткостью  $k$ , тогда проекции сил, действующих на  $i$ -ую частицу, равны:

$$F_{ix} = k \left( (l_{i,i+1} - b) \frac{x_{i+1} - x_i}{l_1} - (l_{i-1,i} - b) \frac{x_i - x_{i-1}}{l_2} \right) - r v_{ix} + r v_{вx},$$

$$F_{iy} = k \left( (l_{i,i+1} - b) \frac{y_{i+1} - y_i}{l_1} - (l_{i-1,i} - b) \frac{y_i - y_{i-1}}{l_2} \right) - r v_{iy} + m_i g.$$

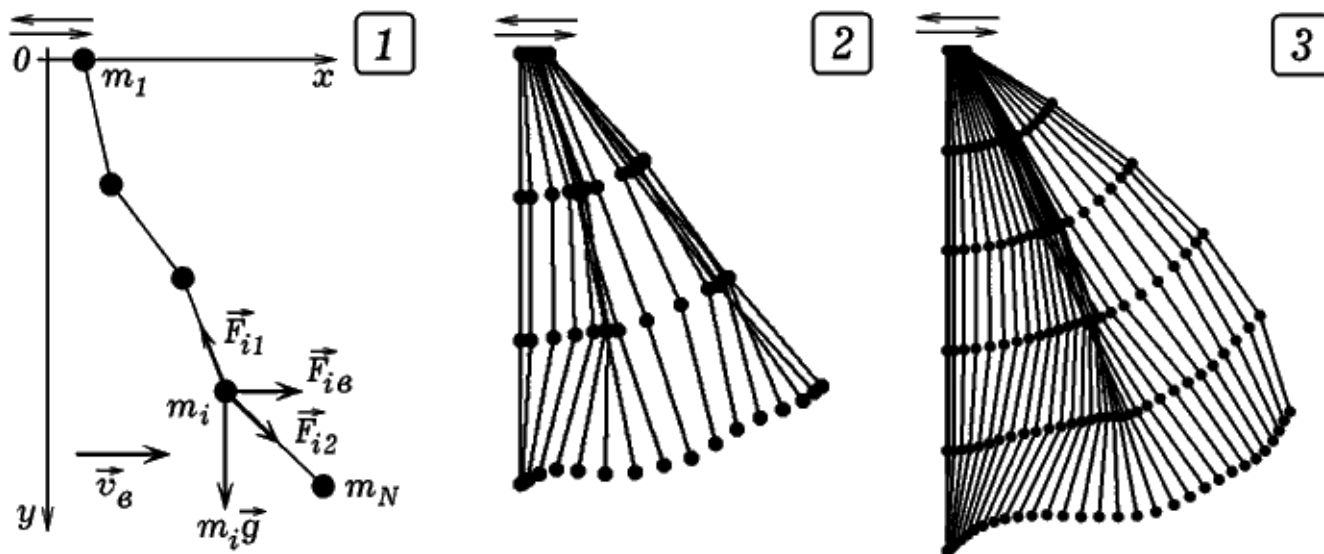


Рис. 10. Вынужденные колебания сложного маятника.

Используется программа ТР-5, содержащая цикл по времени  $t$ , в котором вычисляются проекции силы, действующей на  $i$ -ую материальную точку, ее ускорения, скорости и координаты в момент  $\tau + \Delta\tau$ . Типичные результаты моделирования представлены на рис. 10.2 и 10.3. Аналогичным образом рассчитываются колебания системы в случае, когда  $N$ -ая частица соединена с некоторой точкой  $O'$  упругой нитью (рис. 11).

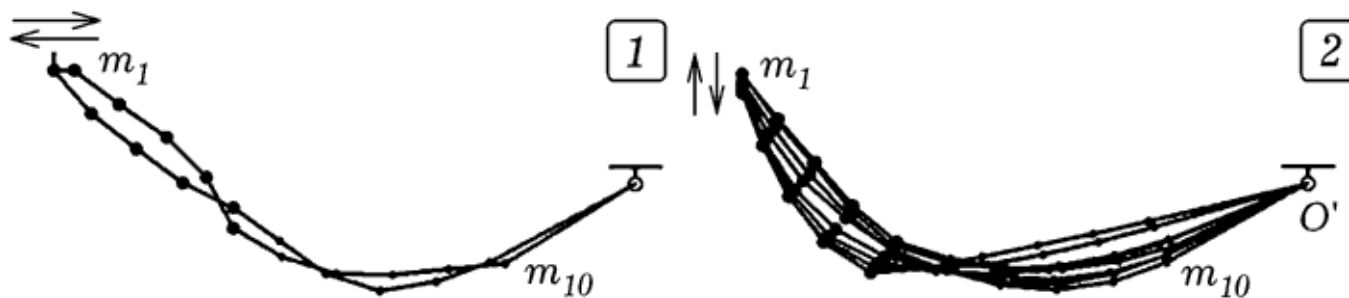


Рис. 11. Колебания системы связанных между собой частиц.



**Задача 4.** На горизонтальной поверхности покоится кольцо массой  $m_k$  и радиуса  $R$ , к внутренней стороне которого прикреплен груз на расстоянии  $l$  от центра  $O$ . Требуется изучить собственные колебания кольца.

Необходимо рассчитать расстояние  $l_c = |OC|$  от центра кольца  $O$  до центра масс  $C$ , момент  $M$  силы тяжести, момент инерции  $I$  относительно мгновенной оси вращения  $A$ . Для этого используются формулы:

$$l_c = m_2 l / (m_2 + m_k), \quad d^2 = R^2 + l^2 - 2Rl \cos \varphi, \quad M_z = -(m_2 + m_k) g l_c \sin \varphi,$$

$$I = 2m_k R^2 + m_2 d^2, \quad \varepsilon_z = M_z / I = -(m_2 + m_k) g l_c \sin \varphi / (2m_k R^2 + m_2 d^2).$$

Зная угловое ускорение тела, можно определить угловую скорость и координату:  $\omega^{t+1} = \omega^t + \varepsilon^{t+1} \Delta \tau$ ,  $\varphi^{t+1} = \varphi^t + \omega^{t+1} \Delta \tau$ . Программа приведена в [5], результаты -- на рис. 12. Если задать достаточно большую начальную скорость, то можно промоделировать неравномерное качение кольца.

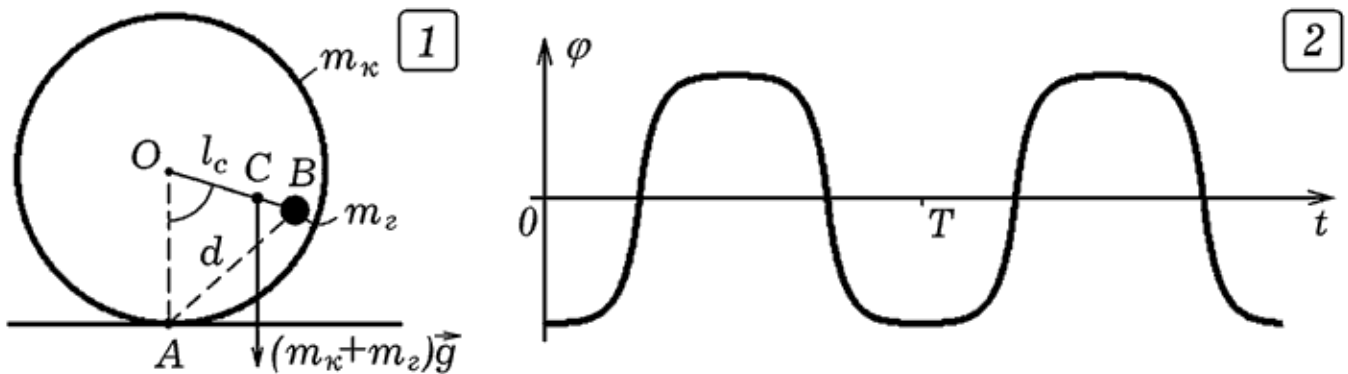


Рис. 12. Нелинейные колебания кольца со смещенным центром тяжести.

## 8.4. Колебания системы связанных осцилляторов

Рассмотрим колебания двух маятников, связанных между собой упругой связью жесткостью  $q$ . Это могут быть физические, математические маятники (рис. 13.1) или пружинные маятники (рис. 13.2). Система из связанных пружинных маятников (рис. 13.2) имеет две степени свободы и описывается уравнениями:

$$m \ddot{\xi}_1 + r \dot{\xi}_1 + k \xi_1 + q(\xi_1 - \xi_2) = 0, \quad m \ddot{\xi}_2 + r \dot{\xi}_2 + k \xi_2 + q(\xi_2 - \xi_1) = 0.$$

Для их решения необходимо создать цикл по  $t$ , в котором вычисляются ускорения, скорости и координаты маятников по формулам:

$$\theta_1^{t+1} = (-F^t - r \eta_1^t - k \xi_1^t) / m_1, \quad \eta_1^{t+1} = \eta_1^t + \theta_1^{t+1} \Delta \tau, \quad \xi_1^{t+1} = \xi_1^t + \eta_1^{t+1} \Delta \tau,$$

$$\theta_2^{t+1} = (F^t - r \eta_2^t - k \xi_2^t) / m_2, \quad \eta_2^{t+1} = \eta_2^t + \theta_2^{t+1} \Delta \tau, \quad \xi_2^{t+1} = \xi_2^t + \eta_2^{t+1} \Delta \tau,$$

где  $F = q(\xi_1 - \xi_2)$ . Результаты выводятся на экран. Используется программа ПР-6. Получающиеся графики изображены на рис. 13.3. При запуске программы первый осциллятор начинает совершать колебания, в процессе которых его энергия передается второму осциллятору. Наступает момент, когда колебания первого осциллятора практически прекращаются, а второй осциллятор колеблется с максимальной амплитудой. После этого снова энергия передается от второго осциллятора к первому и т.д.

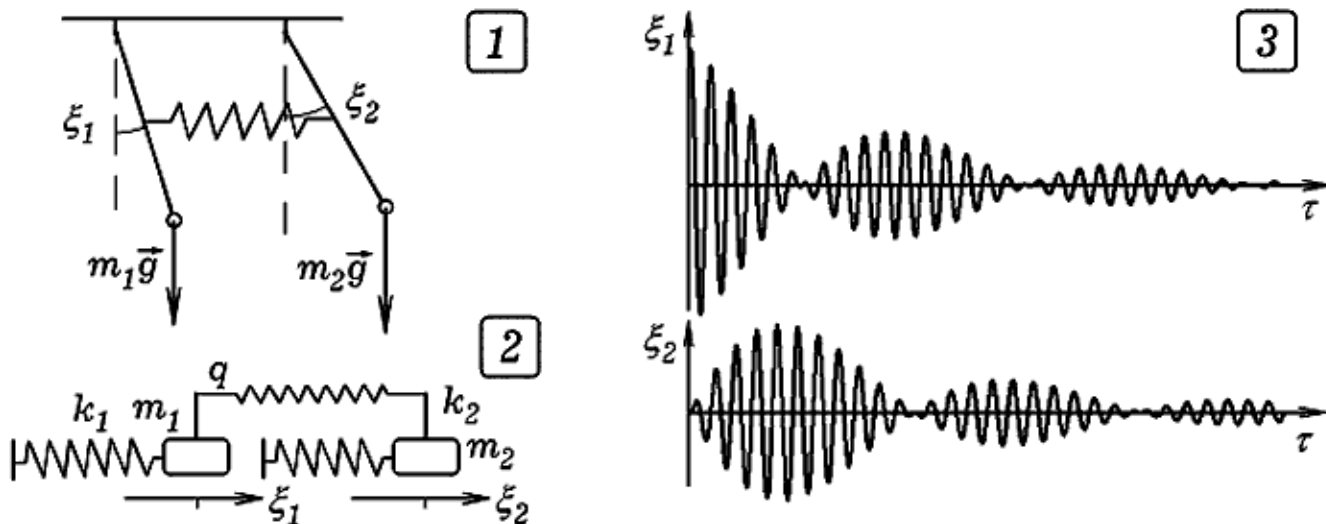


Рис. 13. Движение двух связанных осцилляторов.

Для изучения распространения упругих волн в непрерывных средах может быть использована **модель одномерной упругой среды**, состоящая из совокупности осцилляторов, расположенных вдоль прямой и связанных между собой упругими связями. Она, не смотря на свою простоту, позволяет исследовать целый ряд волновых явлений: распространение возмущения в упругой среде, отражение и прохождение импульса через границу раздела двух сред, интерференцию волн и т.д.

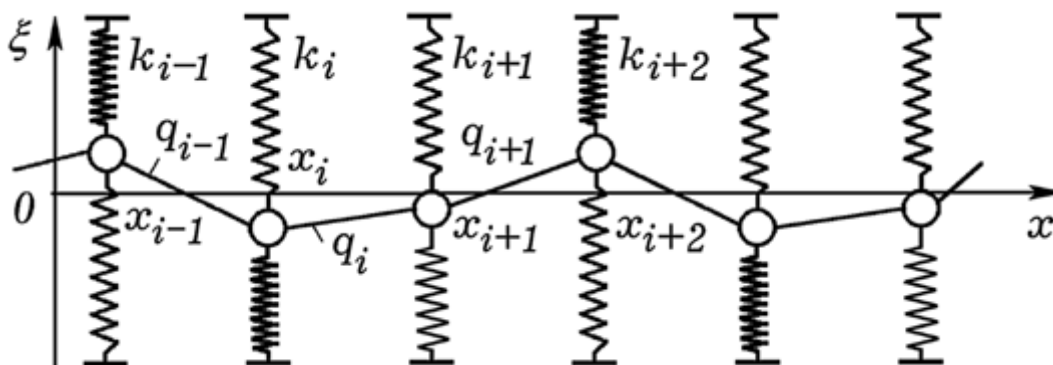


Рис. 14. Система связанных осцилляторов.

Рассмотрим систему из  $N$  пружинных маятников (осцилляторов) с массами  $m_i$  и жесткостью пружин  $k_i$ , которые расположены в ряд и связа-

ны между собой упругими связями с жесткостью  $q_i$  (рис. 14). На каждый осциллятор действует сила вязкого трения  $-r\eta_i$ , где  $\eta_i$  -- скорость  $i$ -ого осциллятора. Заданы начальные смещения  $\xi_{i0}$  и скорости  $\eta_{i0}$  всех осцилляторов. На отдельные осцилляторы действует вынуждающая сила  $F_i\xi(\tau)$ ; некоторые осцилляторы колеблются по заданному закону  $\xi_i(\tau)$ . Известно, что крайние осцилляторы закреплены или свободны. Необходимо рассчитать координаты, скорости и ускорения колеблющихся осцилляторов, изучить колебания системы [2, 4, 6].

Анализируемая система имеет  $N$  степеней свободы. На каждый  $i$ -ый осциллятор со стороны соседних  $(i-1)$ -ого и  $(i+1)$ -ого осцилляторов действует сила упругости  $F_{i\xi} = -q_{i-1}(\xi_i - \xi_{i-1}) - q_i(\xi_i - \xi_{i+1})$ . Кроме нее действуют сила упругости пружины  $-k_i\xi_i$  и сила вязкого трения  $-r\eta_i$ , где  $\eta_i$  -- скорость  $i$ -ого осциллятора. Колебания  $N$  осцилляторов описываются системой уравнений:

$$m_i\theta_i = F_{i\xi} - r\eta_i - k_i\xi_i, \quad \theta_i = d\eta_i / d\tau, \quad \eta_i = d\xi_i / dt, \quad i=1,2,\dots,N.$$

Для дискретных моментов времени ускорение  $i$ -ого осциллятора равно:  $\theta_i^{t+1} = (F_{i\xi}^t - r\eta_i^t - k_i\xi_i^t) / m_i$ . Определив  $\theta_i$ , можно найти скорость и смещение  $i$ -ого осциллятора в следующий момент времени  $t+1$ :  $\eta_i^{t+1} = \eta_i^t + \theta_i^{t+1}\Delta\tau$ ,  $\xi_i^{t+1} = \xi_i^t + \eta_i^{t+1}\Delta\tau$ .

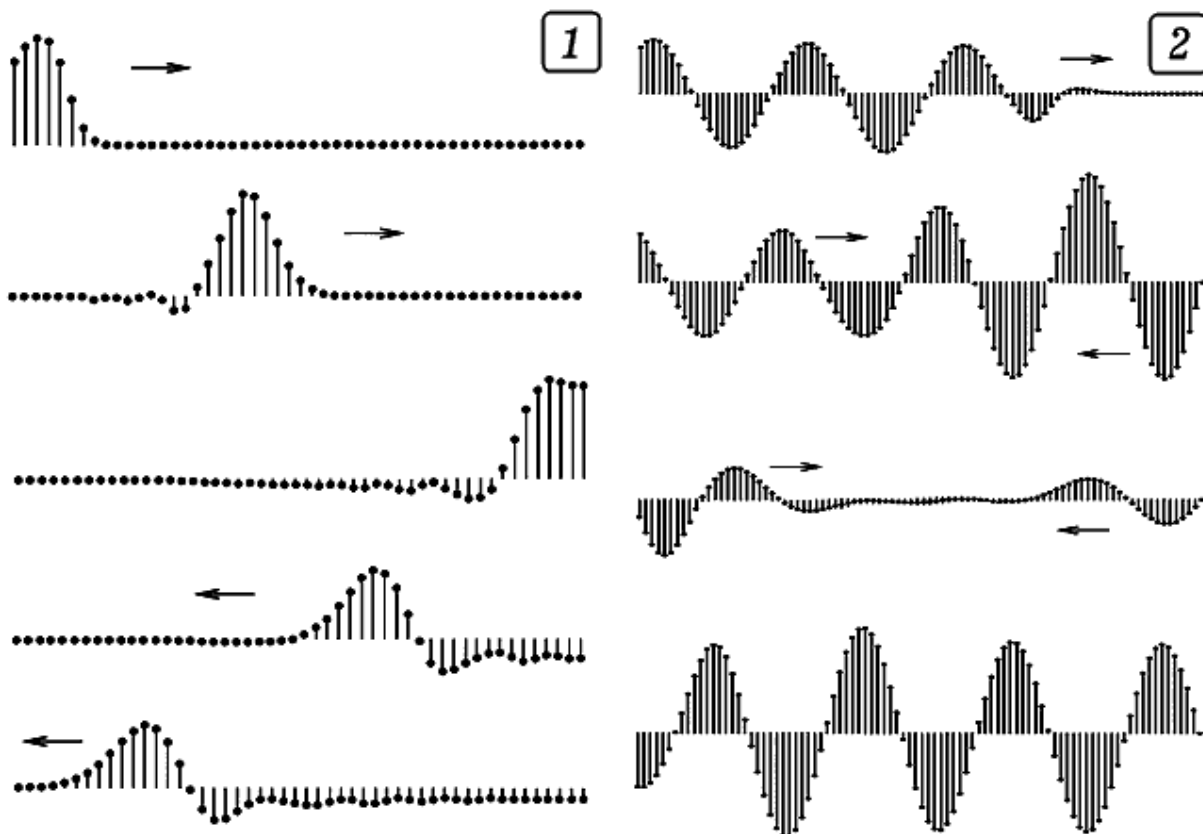


Рис. 15. Исследование колебаний одномерной упругой среды.

Программа Пр-7 моделирует распространение импульса в одномерной упругой среде в случае, когда левый элемент среды совершает полколебания. Компьютерная модель ведет себя подобно реальному физическому объекту: мы только определенным образом смещаем крайний осциллятор и наблюдаем за тем, как по цепочке осцилляторов распространяется импульс (рис. 15.1).

Представим, что лежит длинный стержень. Резко ударим его по левому концу. Как скоро "почувствует" это правый конец стержня? Предположение, что правый конец сразу же придет в движение, является ошибочным. На самом деле по стержню побежит импульс сжатия, скорость которого равна скорости звука в данном материале. Дойдя до правого конца, он отразится. При отражении от закрепленного конца фаза импульса изменится на противоположную. Программа Пр-7 позволяет промоделировать ситуацию, когда левый крайний осциллятор резко сместился из положения равновесия и не вернулся обратно. При отражении от незакрепленного конца упругой среды фаза импульса не изменяется: первые полуволны у падающего и отраженного импульсов положительные. При отражении от закрепленного конца происходит изменение фазы на противоположную.

## 8.5. Моделирование других явлений физики волн

Теперь промоделируем распространение гармонической волны и ее отражение от закрепленного или незакрепленного конца струны (одномерной упругой среды), которая моделируется 50 связанными осцилляторами (программа Пр-7). Для этого необходимо левый осциллятор привести в колебательное движение. Зададим гармонические колебания первого осциллятора упругой среды; для этого достаточно в цикл по времени вставить оператор  $x_1[1]=10*\sin(5*t)$ . При запуске программы видно, как образовавшаяся волна распространяется вдоль упругой среды, отражается от ее правого конца и интерферирует с падающей (рис. 15.2). В узлах (минимумах) образующейся **стоячей волны** колебания практически отсутствуют, а в пучностях (максимумах) -- происходят с максимальной амплитудой. Чтобы промоделировать две **когерентные волны**, распространяющиеся навстречу друг другу, достаточно заставить колебаться левый и правый концы струны с равными частотами. Если частоты колебаний будут отличаться, то интерференции в области наложения волн не получит-

ся: все точки одномерной упругой среды будут совершать сложные колебания, стоячей волны не возникнет.

Аналогично может быть промоделирована **интерференция двух импульсов**. Для того, чтобы получить импульс достаточно заставить один из концов одномерной среды совершить одно или два колебания. Если оба конца совершат два колебания с равными частотами, то возникнут два одинаковых импульса (цуга), которые будут распространяться навстречу друг другу. В области наложения цуги, в зависимости от разности хода, будут частично или полностью усиливать либо ослаблять друг друга.

Чтобы изучить **отражение импульса** от границы раздела двух сред, можно промоделировать распространение импульса вдоль цепочки связанных осцилляторов, в случае, когда их масса или жесткость пружин, начиная с некоторого осциллятора, резко изменяется. Типичные результаты представлены на рис. 16. Если вторая среда имеет большую плотность (массы осцилляторов больше), то скорость распространения импульса оказывается меньше ( $v_1 > v_2$ ). Это проявляется в том, что во второй среде импульс имеет меньшую пространственную протяженность и за то же время проходит меньшее расстояние, чем отраженный импульс (рис. 16.1). При отражении от границы происходит **потеря полуволны**, то есть фаза изменяется на противоположную. Видно, что первая полуволна у падающего импульса положительная, а у отраженного -- отрицательная. При отражении от менее плотной среды ( $v_1 < v_2$ ) потери полуволны не происходит (рис. 16.2). Все это подтверждается экспериментально.

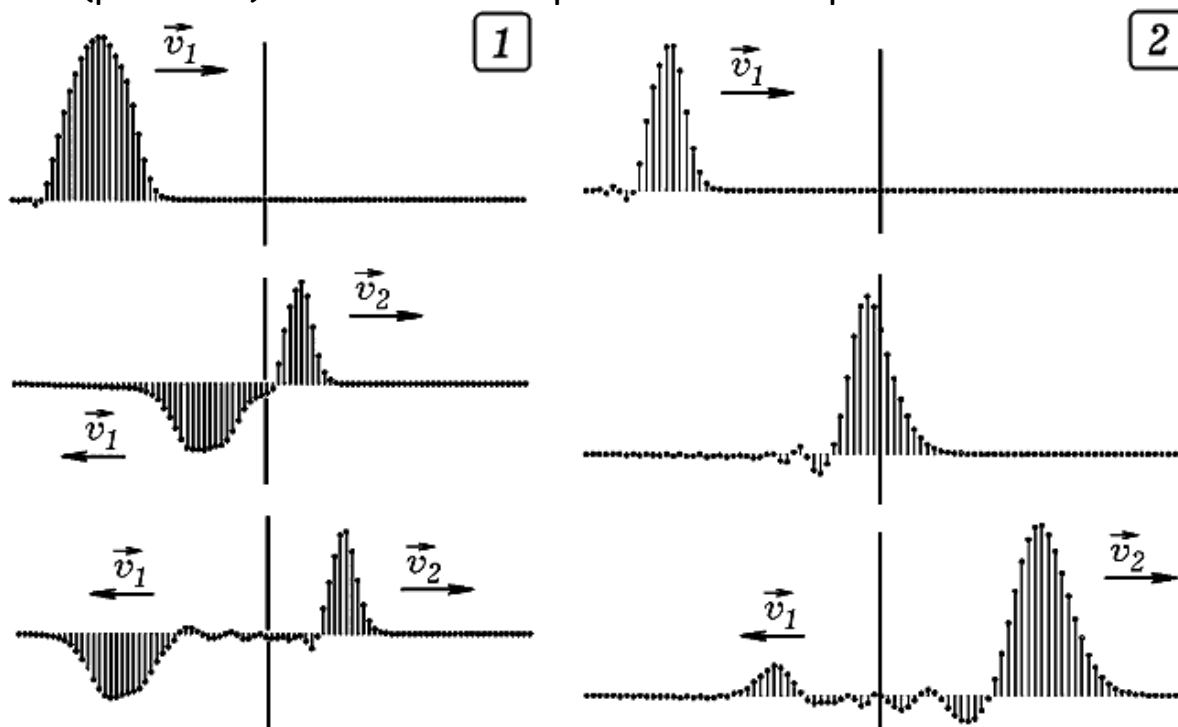


Рис. 16. Отражение и прохождение импульса через границу двух сред.

Известно, что при распространении волны в среде с осцилляторами, собственная частота  $\omega_0$  которых близка к частоте волны  $\omega$ , имеет место **дисперсия**, -- зависимость скорости переноса колебаний (фазовой скорости) от частоты. В результате импульс растягивается, фазовая скорость оказывается не равна групповой скорости переноса энергии. Чтобы визуализировать пространственное распределение энергии рассчитывают кинетическую и потенциальную энергию каждого осциллятора и упругой связи по формуле:  $E_i = 0,5(q(\xi_{i-1} - \xi_i)^2 + k\xi_i^2 + m_i\eta_i^2)$ . Это позволяет построить график зависимости  $E(x)$ , являющийся "мгновенной фотографией" распределения энергии вдоль луча (рис. 17). Скорость перемещения максимума функции  $E(x)$  равна **групповой скорости**. Если жесткость  $k$  и массу  $m$  осцилляторов подобрать так, чтобы частота волны  $\omega$  была бы близка к его собственной частоте колебаний  $\omega_0 = \sqrt{k/m}$ , то **фазовая скорость** (скорость максимального или нулевого смещения) заметно превышает групповую. Используется программа ПР-8. При шаге  $\Delta t = 10^{-4}$  энергия импульса остается постоянной. На рис. 17 представлены "моментальные фотографии", соответствующие моментам времени 0,6, 1,0, 1,4, 1,8; гребни волны пронумерованы.

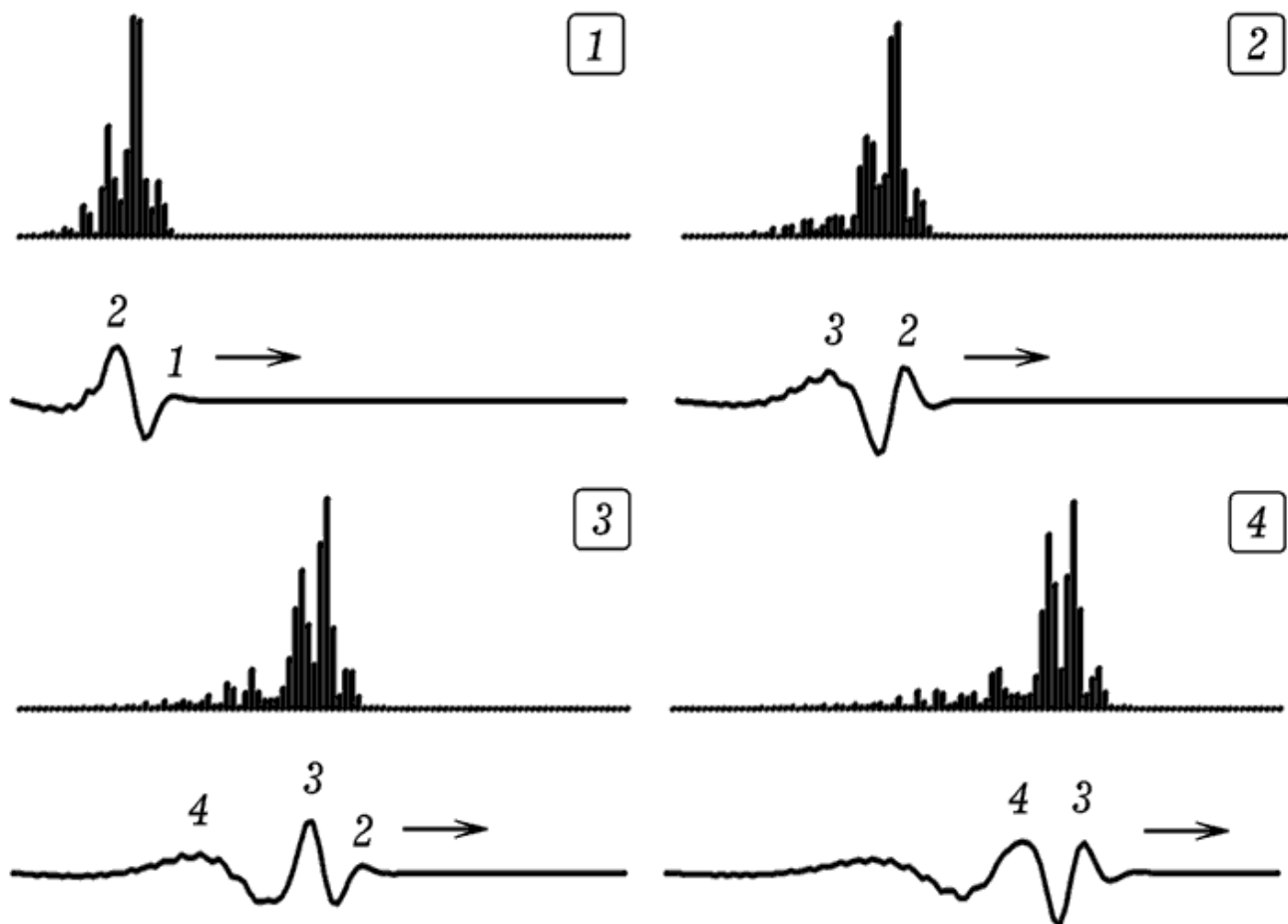


Рис. 17. Скорость переноса фазы превышает скорость переноса энергии.

## 8.6. Моделирование волны в одномерной среде

Создадим компьютерную модель волны, основанную на численном решении волнового уравнения. Сформулируем **краевую задачу для колеблющейся струны**. Имеется одномерная упругая среда (струна), на ее отдельные точки действуют вынуждающие силы  $F(x, y, \tau)$ . Необходимо рассчитать движение струны, то есть найти такую функцию  $\xi(x, \tau)$ , которая удовлетворяет заданным краевым условиям и волновому уравнению

$$\frac{\partial^2 \xi}{\partial x^2} = \frac{1}{v^2} \frac{\partial^2 \xi}{\partial \tau^2} + \beta \frac{\partial \xi}{\partial \tau} - F(x, \tau),$$

где  $\beta$  -- коэффициент затухания,  $F(x, \tau)$  -- внешняя сила, действующая на отдельные элементы среды. Это уравнение также называют **уравнением вынужденных колебаний струны с затуханием**. К краевым условиям относятся начальные условия  $\xi|_{\tau=0} = \varphi(x)$  и  $\partial \xi / \partial \tau|_{\tau=0} = \psi(x)$ , характеризующие смещение и скорость различных точек струны в момент времени  $\tau = 0$ , а также граничные условия  $\xi_G = \varphi'(x, \tau)$  и  $\partial \xi / \partial \tau_G = \psi'(x, \tau)$ , определяющие смещение и скорость точек концов струны в различные моменты времени [8]. Запишем это уравнение в конечных разностях:

$$\xi_i^{t+1} = 2\xi_i^t - \xi_i^{t-1} + v^2 \frac{\xi_{i-1}^t - 2\xi_i^t + \xi_{i+1}^t}{\Delta x^2} \Delta \tau^2 + F_i^t \Delta \tau^2.$$

Используется программа ПР-9. Для получения движущегося профиля волны необходимо организовать цикл по  $i$ , в котором перебираются все элементы среды и вычисляются их смещения  $\xi_i^{t+1}$ . После этого стирается предыдущая моментальная фотография волны и строится новая. Все это должно находиться внутри цикла по времени.

Рассмотренная компьютерная модель позволяет выполнить серию численных экспериментов и изучить следующие явления: 1) распространение и отражение волны (одиночного импульса, цуга) от закрепленного и незакрепленного конца струны; 2) интерференция волн (одиночных импульсов, цугов), возникающая в результате отражения падающей волны либо излучения двух когерентных волн; 3) отражение и прохождение волны (одиночного импульса, цуга) через границу раздела двух сред; 5) изменение фазы отраженной волны на  $\pi$  при отражении от среды, в которой скорость волны меньше.

Промоделируем **распространение импульса** в одномерной упругой среде, его отражение и прохождение через границу раздела двух сред. С помощью условного оператора необходимо задать резкое изменение скорости волны в середине струны. Если левый конец струны совершит одно колебание, то вдоль струны побежит волна, которая отразится от границы раздела сред с различными скоростями  $v$  распространения волны (рис. 18). Можно убедиться в том, что при отражении от среды с меньшей скоростью происходит потеря полуволны.

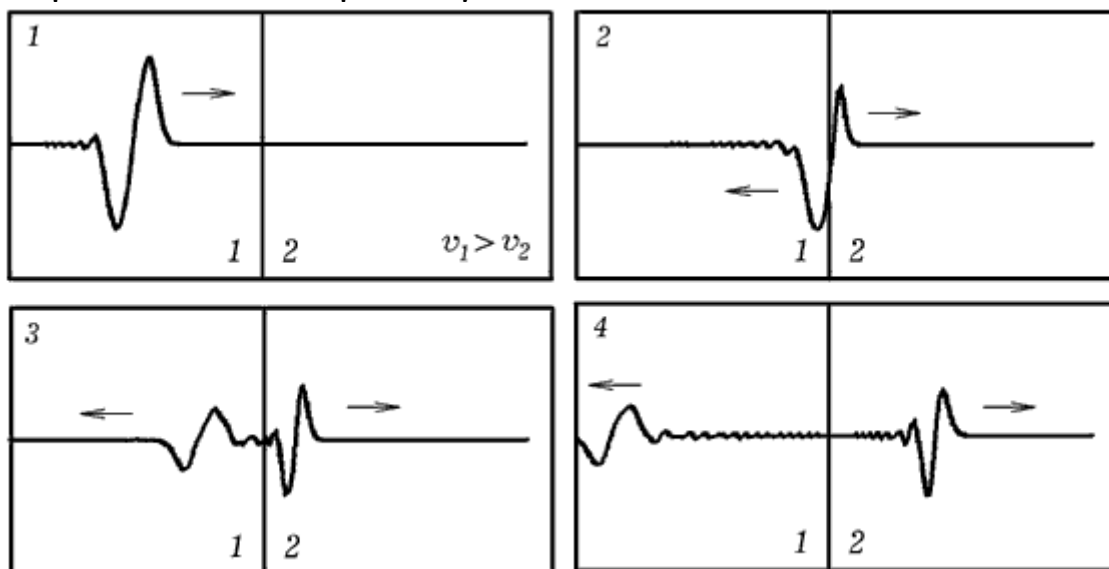


Рис. 18. Прохождение импульса через границу двух сред ( $v_1 > v_2$ ).

Обсуждаемая модель одномерной волны позволяет исследовать **интерференцию падающей и отраженной волн**. При длительных колебаниях левого конца струны возникает гармоническая волна, отражающаяся от правого незакрепленного (или закрепленного) конца струны, которая интерферирует с падающей. Возникает стоячая волна (рис. 19). В области наложения падающей и отраженной волн некоторые точки струны колеблются с максимальной амплитудой (пучности, показаны треугольниками), а некоторые не колеблются вообще (узлы, показаны точками). Аналогичным образом можно изучить **интерференцию двух цугов**, распространяющихся навстречу друг другу.

Теперь промоделируем **вынужденные колебания струны**, на одну из точек которой действует периодическая вынуждающая сила. Запишем уравнение вынужденных колебаний струны с затуханием:

$$\frac{\partial^2 \xi}{\partial \tau^2} = v^2 \left( \frac{\partial^2 \xi}{\partial x^2} - \beta \frac{\partial \xi}{\partial \tau} + F(x, \tau) \right).$$

В конечных разностях оно выглядит так:



$$\frac{\xi_i^{t-1} - 2\xi_i^t + \xi_i^{t+1}}{\Delta\tau^2} = v^2 \left( \frac{\xi_{i-1}^t - 2\xi_i^t + \xi_{i+1}^t}{\Delta x^2} - \beta \frac{\xi_i^{t+1} - \xi_i^t}{\Delta\tau} + F_i^t \right),$$

$$\xi_i^{t+1} = (2\xi_i^t - \xi_i^{t-1} + v^2 \frac{\xi_{i-1}^t - 2\xi_i^t + \xi_{i+1}^t}{\Delta x^2} \Delta\tau^2 + v^2 \beta \Delta\tau \xi_i^t + F v^2 \Delta\tau^2) \frac{1}{1 + v^2 \beta \Delta\tau}.$$

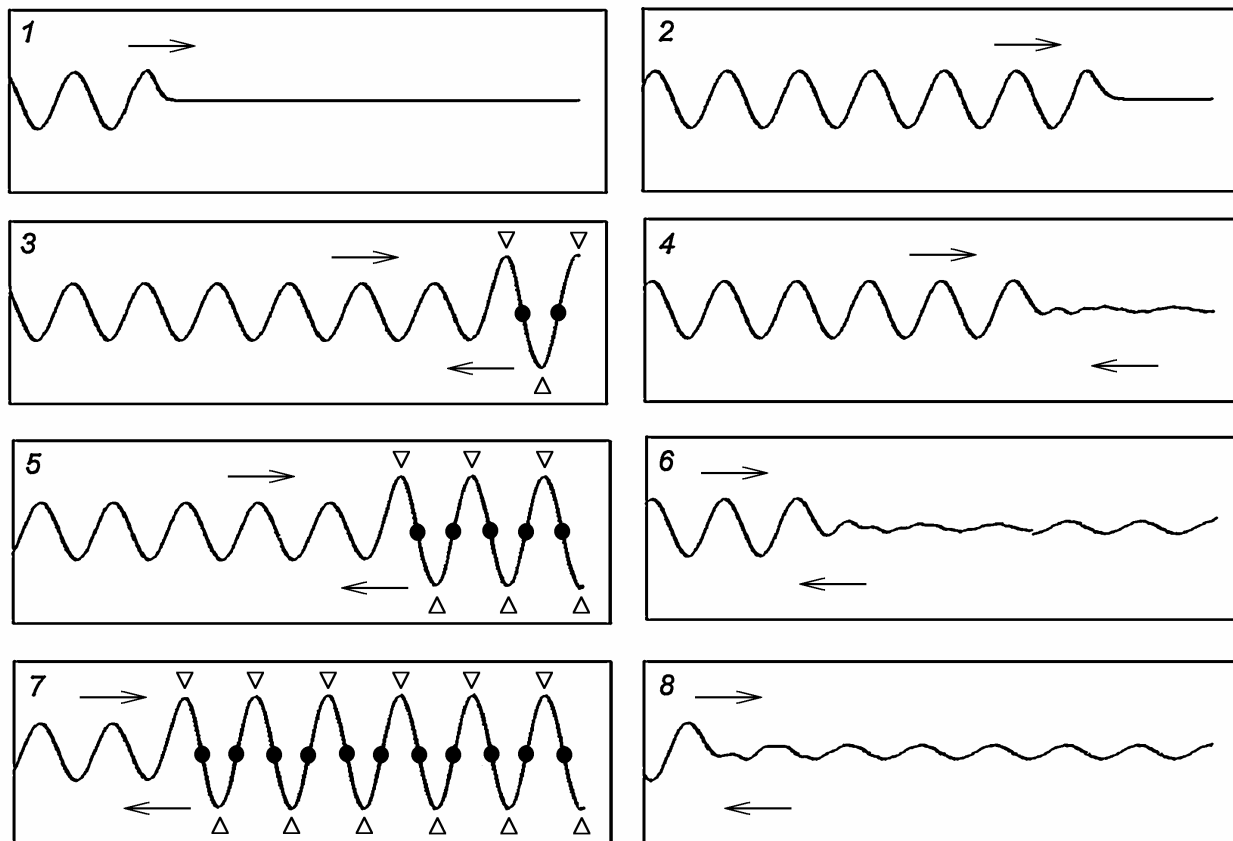


Рис. 19. Процесс образования стоячей волны.

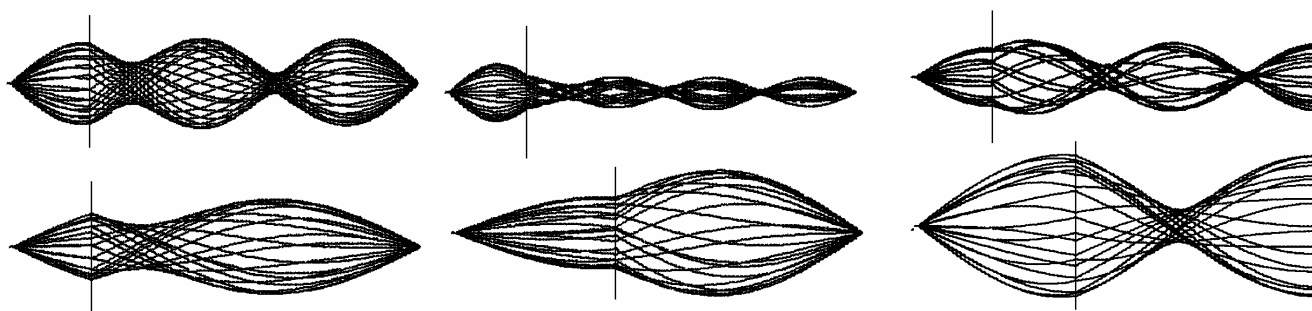


Рис. 20. Вынужденные колебания струны (одномерной среды).

Результаты использования программы Пр-10 приведены на рис. 20. Точка приложения вынуждающей силы показана вертикальным отрезком. Компьютерная модель позволяет изучить зависимость колебаний струны от точки приложения вынуждающей силы, скорости распространения возмущения, коэффициента затухания.

Теперь рассмотрим прохождение плоской гармонической волны сквозь слой поролона, имеющий высокий коэффициент поглощения. Для этого следует решить одномерное волновое уравнение с затуханием:

$$\frac{\partial^2 \xi}{\partial x^2} = \frac{1}{v^2} \frac{\partial^2 \xi}{\partial \tau^2} + \beta(x) \frac{\partial \xi}{\partial \tau}.$$

Заменим производные их конечно-разностными аппроксимациями и решим получающееся алгебраическое уравнение относительно  $\xi_i^{t+1}$ :

$$\xi_i^{t+1} = (2\xi_i^t - \xi_i^{t-1} + v^2 \frac{\xi_{i-1}^t - 2\xi_i^t + \xi_{i+1}^t}{\Delta x^2} \Delta \tau^2 + v^2 \beta_i \Delta \tau \xi_i^t) \frac{1}{1 + v^2 \beta_i \Delta \tau}.$$

В остальном задача решается традиционным способом. Результат приведен на рис. 21.1. При прохождении среды с высоким коэффициентом поглощения амплитуда волны уменьшается по экспоненциальному закону.

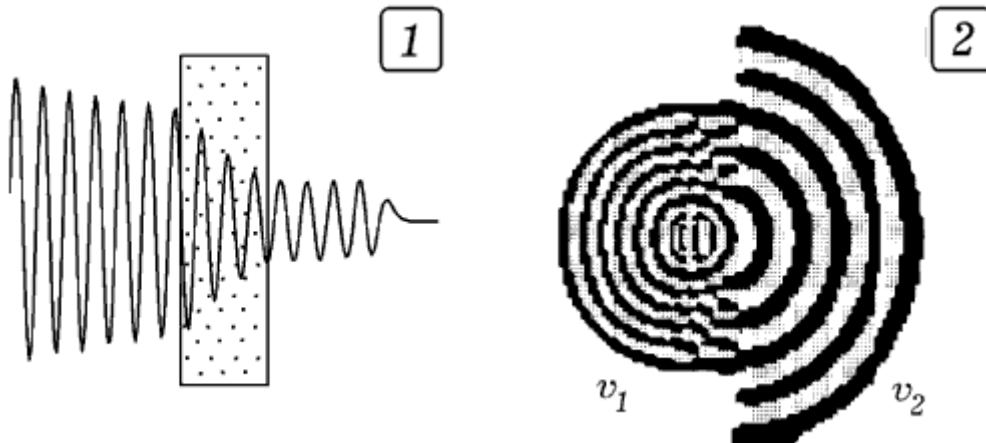


Рис. 21. Результаты моделирования одномерной и двумерной волны.

## 8.7. Волна в двумерной среде

Сформулируем краевую задачу для колеблющейся мембраны. Пусть в положении покоя мембрана занимает прямоугольную область  $\Omega$ , ограниченную контуром  $G$ , являющимся краем мембраны. На различные точки мембраны действуют вынуждающие силы, описываемые функцией  $F(x, y, \tau)$ . Необходимо рассчитать движение мембраны, то есть найти такую функцию  $\xi(x, y, \tau)$ , которая удовлетворяет заданным краевым условиям и волновому уравнению [6, 8]:

$$\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} = \frac{1}{v^2} \frac{\partial^2 \xi}{\partial \tau^2} - F(x, y, \tau).$$

К краевым условиям относятся начальные условия  $\xi|_{\tau=0} = \varphi(x, y)$  и  $\partial\xi/\partial\tau|_{\tau=0} = \psi(x, y)$ , характеризующие смещение и скорость различных точек мембраны в момент времени  $\tau=0$ , а также граничные условия  $\xi_G = \varphi'(x, y, \tau)$  и  $\partial\xi/\partial\tau_G = \psi'(x, y, \tau)$ , определяющие смещение и скорость точек края мембраны в различные моменты времени.

Дискретизируем задачу, перейдем от непрерывной функции  $\xi(x, y, \tau)$  к функции дискретного аргумента. Запишем двумерное волновое уравнение в конечных разностях:

$$\frac{\xi_{i,j}^{t-1} - 2\xi_{i,j}^t + \xi_{i,j}^{t+1}}{v^2 \Delta\tau^2} = \frac{\xi_{i-1,j}^t - 2\xi_{i,j}^t + \xi_{i+1,j}^t}{\Delta x^2} + \frac{\xi_{i,j-1}^t - 2\xi_{i,j}^t + \xi_{i,j+1}^t}{\Delta y^2} + F_{i,j}^t.$$

Отсюда смещение элемента с координатами  $i, j$  в момент  $t+1$  равно:

$$\xi_{i,j}^{t+1} = 2\xi_{i,j}^t - \xi_{i,j}^{t-1} + v^2 \left( \frac{\xi_{i-1,j}^t - 2\xi_{i,j}^t + \xi_{i+1,j}^t}{\Delta x^2} + \frac{\xi_{i,j-1}^t - 2\xi_{i,j}^t + \xi_{i,j+1}^t}{\Delta y^2} + F_{i,j}^t \right) \Delta\tau^2.$$

Для нахождения смещения элементов пластины на временном слое  $t+1$  необходимо знать смещение элементов на слоях  $t$  и  $t-1$ . То есть компьютер должен работать с тремя массивами, в которых записаны  $\xi_{i,j}^{t-1}$ ,  $\xi_{i,j}^t$ ,  $\xi_{i,j}^{t+1}$ . Для решения задачи создается цикл, в котором перебираются все узлы пространственной сетки и рассчитываются значения  $\xi_{i,j}^t$  при  $t=1,2,3\dots$  Отдельно следует предусмотреть цикл для переобозначения переменных  $\xi_{i,j}^{t-1}$ ,  $\xi_{i,j}^t$ ,  $\xi_{i,j}^{t+1}$ . Для решения этой задачи необходим большой объем памяти для хранения смещений точек в узлах сетки. Выход состоит в использовании динамической памяти.

Программа Пр-11 позволяет изучить **распространение, интерференцию и дифракцию волн в двумерной среде**. Пусть точечный источник двумерной волны расположен вблизи границы раздела двух сред с различными скоростями распространения волны. Исследуем прохождение волны из одной среды в другую. Чтобы получить границу раздела двух сред, необходимо задать в узлах сетки, для которых  $x < b$ , скорость волны  $v_1$ , а во всех остальных узлах --  $v_2$ . Результат -- на рис. 21.2.

На рис. 22 показаны результаты моделирования распространения, отражения, интерференции и дифракции двумерной волны. Источник И моделируется совокупностью элементов, колеблющихся по гармониче-

скому закону, экран Э состоит из неподвижных элементов. Видно, что имеет место отражение волны от края пластины, и в области наложения падающей и отраженной волн наблюдается **интерференция**. На остальных рисунках приведены результаты моделирования **дифракции волн** на отверстиях и экране. Так как длина волны сравнима с размером препятствия, то волны их огибают и заходят в область геометрической тени.



Рис. 22. Моделирование волновых явлений в двумерной среде.

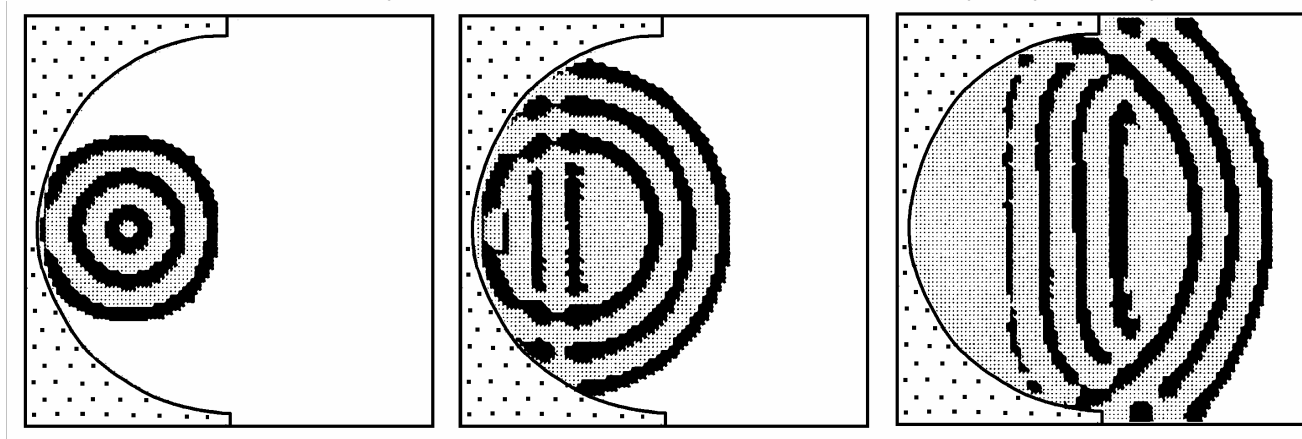


Рис. 23. Модель отражения волны от цилиндрического зеркала.

Аналогично моделируется **отражение двумерной волны** от вогнутого зеркала. Вблизи центра пластины создается источник гармонических волн (четыре соседних элемента совершают несколько гармонических колебаний). Вогнутая поверхность моделируется путем задания граничных условий: точки этой поверхности должны оставаться неподвижными. То есть при выполнении вычислений смещения  $\xi_{i,j}$  для всех узлов сетки, удовлетворяющих двум неравенствам  $(50-i)^2 + (50-j)^2 > 2000^2$  и

$i < 50$ , должны приравняться к нулю. Результат -- на рис. 23. Видно, как образовавшаяся сферическая волна дошла до отражающей поверхности и, отразившись от нее, превратилась в плоскую волну.

Определенный интерес представляет собой моделирование вынужденных колебаний упругой пластины квадратной формы. Источник гармонических колебаний находится в центре, края мембраны закреплены. Задача решается с помощью программы ТР-11. Частота колебаний источника должна быть близка к собственной частоте пластины, в этом случае длина ее стороны будет примерно равна длине волны. Через некоторое время после начала колебаний источника, когда излучаемая волна многократно отразится от края пластины, на экране монитора наблюдаются картины, подобные изображенным на рис. 24. На каждом из рисунков изображена "моментальная фотография" пластины; будем считать, что темные участки смещены к нам, а светлые -- от нас. Можно промоделировать колебания мембраны, когда источник смещен из центра.

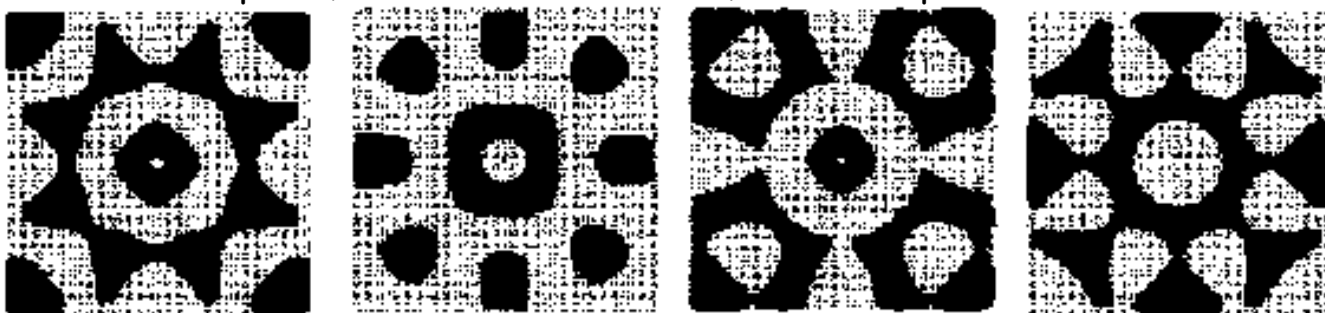


Рис. 24. Результаты расчетов колебания пластины.

## 8.8. Решение уравнения синус-Гордона

Определенный интерес представляет собой проблема распространения возмущения в нелинейной среде. Известно, что в таких средах при определенных условиях могут возникать **солитоны** -- уединенные волны, сохраняющие свою форму. Примером такой дискретной среды является система из одинаковых математических маятников, соединенных пружинками (рис. 25.1). На  $i$ -ый осциллятор со стороны соседних маятников действует момент сил  $M_i = -k_{i+1}(\varphi_i - \varphi_{i+1}) - k_i(\varphi_i - \varphi_{i-1})$ . Из законов механики:  $I\varepsilon_i = -mg \sin \varphi_i - r\omega_i + M_i$ ,  $\varepsilon_i = d\omega_i / d\tau$ ,  $\omega_i = d\varphi_i / dt$ ,  $i = 1, 2, \dots, N$ . Ускорение  $i$ -ого осциллятора  $\varepsilon_i^{t+1} = (M_i^t - r\omega_i^t - mg \sin \varphi_i^t) / I_i$ . Определив  $\varepsilon_i$ , можно найти скорость и смещение  $i$ -ого осциллятора в следующий момент времени  $t+1$ :  $\omega_i^{t+1} = \omega_i^t + \varepsilon_i^{t+1} \Delta\tau$ ,  $\varphi_i^{t+1} = \varphi_i^t + \omega_i^{t+1} \Delta\tau$ .

Используется программа ПР-12, результаты моделирования представлены на рис. 25.2, 26 и 27. Солитонам типа **кинк** и **антикинк** отвечают переходы от стационарного решения  $\varphi = 0$  к стационарному решению  $\varphi = 2\pi$  и наоборот от  $\varphi = 2\pi$  до  $\varphi = 0$  соответственно. Кинк и антикинк, следующие друг за другом, образуют **бризер** -- одиночную полуволну, сохраняющую свою форму (рис. 25.2 и 26). Речь идет о дискретной модели нелинейной среды.

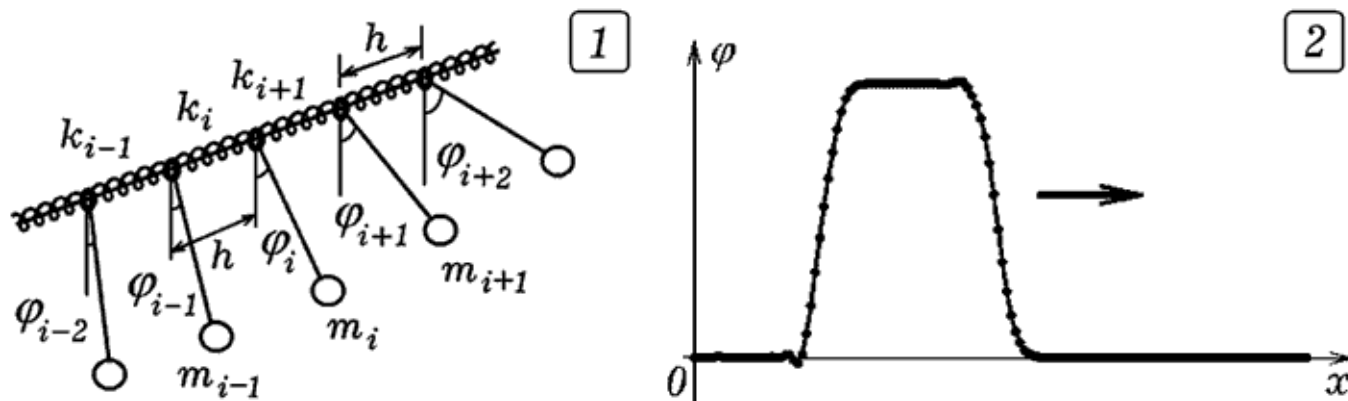


Рис. 25. Колебания системы связанных математических маятников.

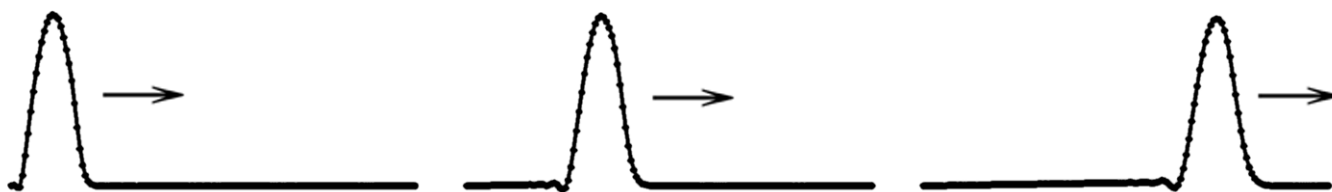


Рис. 26. Распространение бризера по цепочке связанных маятников.

Если левый конец среды сместить от  $0$  до  $2\pi$ , а правый конец -- от  $0$  до  $-2\pi$ , то возникнут два кинка, распространяющиеся навстречу друг другу. Кинки с одинаковыми зарядами отталкиваются (рис. 27.1). Программа ПР-12 позволяет промоделировать **столкновение кинка с бризером** (рис. 27.2), **кинка и антикинка**, двух бризеров и т.д.

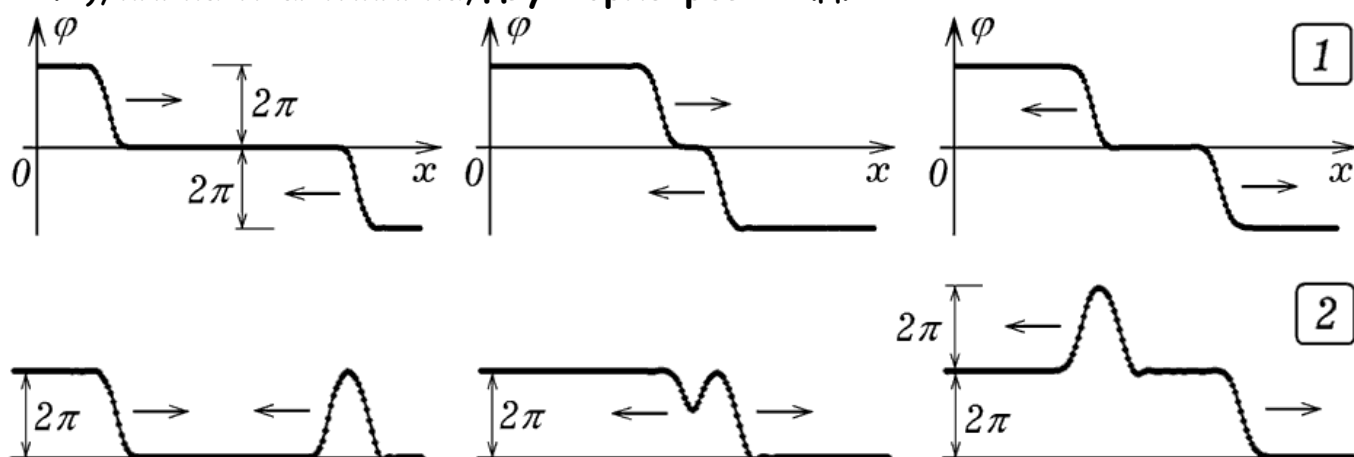


Рис. 27. Столкновение двух кинков (1) и кинка с бризером (2).

Распространение волны по цепочке математических маятников, связанных упругими связями (рис. 25.1) также может быть описано уравнением синус-Гордона (непрерывная модель):

$$\frac{\partial^2 \varphi}{\partial t^2} - c^2 \frac{\partial^2 \varphi}{\partial x^2} + \omega_0^2 \sin \varphi = 0.$$

Это нелинейное уравнение. В конечных разностях получаем:

$$\varphi_i^{t+1} = 2\varphi_i^t - \varphi_i^{t-1} + c^2 (\varphi_{i-1}^t - 2\varphi_i^t + \varphi_{i+1}^t) \frac{\Delta \tau^2}{\Delta x^2} - \omega_0^2 \sin \varphi_i^t \Delta \tau^2.$$

С помощью программы ТР-13 можно промоделировать **распространение бризера, прохождение бризеров друг сквозь друга** (рис. 28) и другие явления, получив при этом двух-, трех- и четырехсолитонные решения.

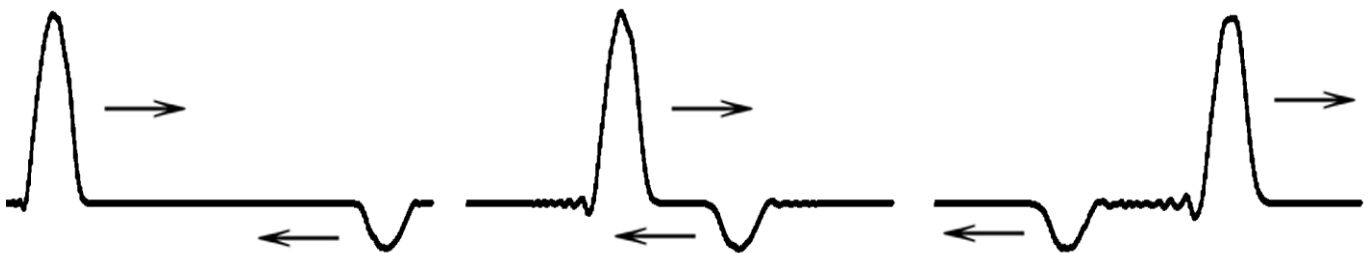


Рис. 28. Прохождение солитонов (бризеров) друг сквозь друга.

## Приложение

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

ТР-1.

```
Uses crt, graph;
Const m=1; r=1.2; k=100; Mx=100; Mv=4; Ma=1; Mt=100;
dt=0.0001; Fm=15; w=12;
Var x,v,a,F,t: Real; j,Gd,Gm: Integer;
BEGIN Gd:= Detect; InitGraph(Gd,Gm,'c:\bp\bgi');
x:=2; line(30,300,650,300); line(31,500,31,10);
Repeat t:=t+dt; {F:=Fm*sin(w*t);}
  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);
  { circle(300+round(x*Mx),300-round(v*15),1); }
until KeyPressed; CloseGraph;
END.
```

ТПР-2.

```

uses crt, graph;
const I=1; k=1; r=0.05; dt=0.005;
var Gd,Gm,z: integer; fi,w,w_vala,eps,M,t: real;
BEGIN Gd:=Detect; InitGraph(Gd,Gm,'c:\bp\bgi');
fi:=0; w_vala:=2; w:=0.01; line(0,300,700,300);
Repeat t:=t+dt;
  If ((w_vala-w)>0) then z:=1 else z:=-1;
  M:=1.2*z*exp(-0.1*abs(w-w_vala));
  eps:=(M-k*fi-r*w)/I; w:=w+eps*dt; fi:=fi+w*dt;
  circle(10+round(10*t),300-round(20*fi),1);
  circle(10+round(10*t),300-round(100*w),1);
  circle(10+round(10*t),300-round(100*w_vala),1);
  circle(200+round(100*fi),220-round(80*w),1);
until KeyPressed; CloseGraph;
END.

```

ТПР-3.

```

uses crt, graph;
const m1=0.1; m2=0.2; k=100; r=0.01; dt=0.001;
var Gd, Gm : integer; n: longint;
l0,l,x1,x2,y1,y2,vx1,vy1,vx2,vy2,F,Fx1,Fy1,Fx2,Fy2: real;
BEGIN Gd:=Detect; InitGraph(Gd,Gm,'c:\bp\bgi');
x1:=0; y1:=0; x2:=300; y2:=-150;
l0:=sqrt(sqr(x1-x2)+sqr(y1-y2));
Repeat l:=sqrt(sqr(x1-x2)+sqr(y1-y2)); F:=k*(l-l0);
  Fx1:=F*(x2-x1)/l-r*v1; Fx2:=-F*(x2-x1)/l;
  Fy2:=-F*(y2-y1)/l-0.5*m2;
  vx1:=vx1+Fx1/m1*dt; vx2:=vx2+Fx2/m2*dt;
  vy2:=vy2+Fy2/m2*dt; x1:=x1+vx1*dt;
  x2:=x2+vx2*dt; y2:=y2+vy2*dt;
  circle(250+round(x2),50-round(y2),1);
  If n mod 8000=0 then begin
    line(250+round(x1),50-round(y1),250+round(x2),
      50-round(y2));
    line(251+round(x1),51-round(y1),251+round(x2),
      51-round(y2));
    circle(250+round(x1),50-round(y1),2);
    circle(250+round(x2),50-round(y2),2);
    circle(250+round(x2),50-round(y2),3); end; inc(n);
until KeyPressed;
END.

```

ТПР-4.

```

uses dos, crt, graph;
const dt=0.00001; rs=0.005; chastota=30;
  Dlina=20; Am=2; P=6;
var m,F,Fx,Fy,x,y,x0,y0,vx,vy,ax,ay,bx,by,t,l,fi : real;

```



```

    Gd,Gm,k : integer;
BEGIN Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
m:=0.01; x:=0; y:=Dlina; vx:=150;
Repeat inc(k);
l:=sqrt(sqr(x0-x)+sqr(y0-y)); F:=500000*(Dlina-1);
Fx:=F*(x-x0)/l-rs*vx; Fy:=F*(y-y0)/l-rs*vy-981*m;
y0:=Am*sin(2*3.14*chastota*t); t:=t+dt;
bx:=ax; by:=ay; ax:=Fx/m; ay:=Fy/m;
vx:=vx+(ax+bx)*dt/2;   vy:=vy+(ay+by)*dt/2;
x:=x+vx*dt+ax*dt*dt/2; y:=y+vy*dt+ay*dt*dt/2;
If k mod 100=0 then begin {cleardevice;} k:=0;
    line(320+round(P*x0),240-round(P*y0),
        320+round(P*x),240-round(P*y));
    circle(320+round(P*x),240-round(P*y),2); end;
fi:=arctan(x/(y-y0)); circle(10+round(200*t),240,1);
circle(10+round(200*t),240-round(fi*200),1);
until KeyPressed; CloseGraph;
END.

```

ПР-5.

```

Uses crt, graph;
Const r=0.025; dt=0.001; N=5; b=100; k=1000;
Var l1,l2,F,t,VV : real; i,s, Gd, Gm : integer;
Fx,Fy,vx,vy,ax,ay,x,y,m : array [0..N+1] of real;
Procedure Sila;
begin For i:=1 to N do begin
L1:=sqrt(sqr(x[i]-x[i+1])+sqr(y[i]-y[i+1]));
L2:=sqrt(sqr(x[i]-x[i-1])+sqr(y[i]-y[i-1]));
If i=N then L1:=b; If i=1 then L2:=b;
Fx[i]:=k*((L1-b)*(x[i+1]-x[i])/L1-
    (L2-b)*(x[i]-x[i-1])/L2)-r*vx[i]+r*VV;
Fy[i]:=k*((L1-b)*(y[i+1]-y[i])/L1+
    (L2-b)*(y[i-1]-y[i])/L2)-r*vy[i]+m[i]*9.8; end; end;
BEGIN Gd:=Detect; InitGraph(Gd,Gm,'c:\bp\bgi');
For i:=1 to N do begin y[i]:=b*(i-1); m[i]:=0.2; end;
m[N-1]:=1.5; VV:=50; x[0]:=0; vx[N]:=40;
Repeat t:=t+dt; Sila; inc(s); x[1]:=10*sin(1*t);
    For i:=2 to N do begin
        ax[i]:=Fx[i]/m[i]; ay[i]:=Fy[i]/m[i];
        vx[i]:=vx[i]+ax[i]*dt; vy[i]:=vy[i]+ay[i]*dt;
        x[i]:=x[i]+vx[i]*dt;   y[i]:=y[i]+vy[i]*dt; end;
If (t>25)and(s mod 1000=1) then begin cleardevice; s:=1;
For i:=1 to N do begin
    circle(200+round(x[i]),15+round(y[i]),2);
    circle(200+round(x[i]),15+round(y[i]),4);
    line(200+round(x[i-1]),15+round(y[i-1]),
        200+round(x[i]),15+round(y[i])); end; end;

```

```
until (KeyPressed){or(t>39)};
Repeat until KeyPressed; CloseGraph;
END.
```

ПР-6.

```
Uses crt, graph;
Const m=1; r=0.2; k=100; q=8; dt=0.001;
Var x1,x2,v1,v2,a1,a2,t,F: real; Gd,Gm: integer;
BEGIN Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi'); x1:=1;
Repeat t:=t+dt; F:=-q*(x1-x2);
  a1:=(-F-r*v1-k*x1)/m; v1:=v1+a1*dt; x1:=x1+v1*dt;
  a2:=(F-r*v2-k*x2)/m; v2:=v2+a2*dt; x2:=x2+v2*dt;
  circle(10+round(20*t),150-round(100*x1),1);
  circle(10+round(20*t),350-round(100*x2),1);
until KeyPressed; CloseGraph;
END.
```

ПР-7.

```
Uses crt, graph;
Const r=0.1; k=0.01; dt=0.001; q=100; N=60;
Var teta,F,t,m: real; i,Gd,Gm: integer;
eta,ksi: array [0..N+1] of real;
Procedure Oscillator;
  begin F:=q*(ksi[i-1]-ksi[i])+q*(ksi[i+1]-ksi[i]);
  If i>N/2 then m:=1 else m:=0.3;
  teta:=(F-r*eta[i]-k*ksi[i])/m;
  eta[i]:=eta[i]+teta*dt; ksi[i]:=ksi[i]+eta[i]*dt; end;
BEGIN Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
Repeat t:=t+dt;
  For i:=1 to N do begin Oscillator;
    If 5*t<3.142 then ksi[1]:=10*sin(5*t) else ksi[1]:=0;
    ksi[N]:=0; {ksi[N]:=ksi[N-1]; } end; cleardevice;
  For i:=1 to N do circle(8*i,240-round(ksi[i]*10),2);
until KeyPressed; CloseGraph;
END.
```

ПР-8.

```
uses crt, graph;
const m=0.05; r=0.01; k=12; dt=0.0001;
  q=100; N=100; pi=3.1415; w=20;
var F,teta,t,Sum_E: Real; i,j,Gd,Gm,p: integer;
  eta,ksi,E: array [1..N] of real;
BEGIN Gd:= Detect; InitGraph(Gd,Gm,'c:\bp\bgi');
Repeat t:=t+dt; Sum_E:=0; inc(p);
  For i:=2 to N-1 do begin ksi[N]:=0; Sum_E:=Sum_E+E[i];
    If w*t<pi then ksi[1]:=1*sin(w*t) else ksi[1]:=0;
    F:=q*(ksi[i-1]-ksi[i])+q*(ksi[i+1]-ksi[i]);
    teta:=(F-r*eta[i]-k*ksi[i])/m;
```

```

ksi[i]:=ksi[i]+eta[i]*dt+teta*dt*dt/2;
eta[i]:=eta[i]+teta*dt; E[i]:=q*sqr(ksi[i-1]-ksi[i])/2
+k*ksi[i]*ksi[i]/2+m*eta[i]*eta[i]/2; end;
If p mod 20=0 then begin cleardevice; p:=0;
For i:=2 to N-1 do begin
{rectangle(6*i,400,5*i+2,400-round(Sum_E));}
line(6*(i-1),400-round(ksi[i-1]*100),
      6*i,400-round(ksi[i]*100));
rectangle(6*i,250,6*i+3,250-round(e[i]*20)); end; end;
until (KeyPressed); CloseGraph;
END.

```

ТПР-9.

```

Uses crt, graph;
Const n=200; h=1; dt=0.02; vv=8;
Var i,j,DV,MV,EC: integer; t,b: real;
    eta,xi,xil,xi2: array[0..N+1] of real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat t:=t+dt;
If t<6.28 then xil[1]:=2*sin(t) else xil[1]:=0;
For i:=2 to N-1 do
xi2[i]:=2*xil[i]-xi[i]+vv*(xil[i-1]-2*xil[i]
+xil[i+1])/h/h*dt*dt;
For i:=2 to N-1 do begin xi[i]:=xil[i];
xil[i]:=xi2[i]; end;
For i:=1 to N do begin
setcolor(black); circle(i*3-3,240-round(xi[i-1]*50),2);
setcolor(white); circle(i*3-3,240-round(xil[i-1]*50),2);
end;
until KeyPressed; CloseGraph;
END.

```

ТПР-10.

```

Uses crt, graph; Const n=100; h=1; dt=0.1;
Var k,i,j,DV,MV,EC: integer; f,t,vv,b: real;
    eta,xi,xil,xi2: array[0..N+1] of real;
Procedure Raschet;
begin vv:=5; For i:=2 to N-1 do begin b:=0.03;
If i=20 then f:=0.01*sin(0.2*t) else f:=0;
xi2[i]:=(2*xil[i]-xi[i]+vv*(xil[i-1]-2*xil[i]+xil[i+1])
/h/h*dt*dt+b*xil[i]*dt+F*dt)/(1+b*dt); end;
For i:=2 to N-1 do begin xi[i]:=xil[i];
xil[i]:=xi2[i]; end; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
EC:=GraphResult; If EC<>grOK then Halt(1);
Repeat t:=t+dt; xil[N]:=xil[N-1]; Raschet; k:=k+1;
If (k>1000)and(k mod 1500=0) then begin k:=0;

```

```

    For i:=1 to N do line(i*5-3,240-round(xi1[i-1]*200),
        i*5,240-round(xi1[i]*200)); end;
until KeyPressed; CloseGraph;
END.

```

ТТР-11.

```

{$N+} Uses crt, graph;      {Borland Pascal}
Const N=100; M=100; NN=200; h=1; dt=0.2; SizeOfReal=6;
type RealPoint= ^real;
Var t: single; r,s,i,j,ii,jj,DV,MV,EC: integer;
    xi: array[1..NN] of pointer; vv: real;
Function Ad(i,j: word) : RealPoint;
    begin Ad:=Ptr(Seg(xi[i]^),ofs(xi[i]^)+(j-i)*SizeOfReal);
end;
Procedure Raschet; begin
For i:=2 to N-1 do For j:=2 to M-1 do begin
If i<50 then vv:=4 else vv:=4;
Ad(i+n,j)^:=Ad(i+n,j)^+vv*(Ad(i,j+1)^-2*Ad(i,j)^+Ad(i,j-1)^)
/h*dt+vv*(Ad(i+1,j)^-2*Ad(i,j)^+Ad(i-1,j)^)/h*dt; end;
    For i:=2 to N-1 do For j:=2 to M-1 do
Ad(i,j)^:=Ad(i,j)^+Ad(i+n,j)^*dt;
    For ii:=2 to N-1 do For jj:=2 to M-1 do
        begin If i<50 then vv:=4 else vv:=4;
j:=M+1-jj; i:=N+1-ii;
Ad(i+n,j)^:=Ad(i+n,j)^+vv*(Ad(i,j+1)^-2*Ad(i,j)^+Ad
(i,j-1)^)/h*dt+vv*(Ad(i+1,j)^-2*Ad(i,j)^+
Ad(i-1,j)^)/h*dt; end;
    For j:=2 to M-1 do For i:=2 to N-1 do
Ad(i,j)^:=Ad(i,j)^+Ad(i+n,j)^*dt;
end;
Procedure Draw;
    begin For i:=2 to N-1 do For j:=2 to M-1 do begin
setcolor(round(Ad(i,j)^/2)+5);
rectangle(10+i*4,450-j*4,13+i*4,453-j*4);
rectangle(11+i*4,451-j*4,12+i*4,452-j*4); end;
end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
    setbkcolor(15); setcolor(8);
    For i:=1 to NN do begin GetMem(xi[i], M*SizeOfReal);
    For j:=1 to M do Ad(i,j)^:=0; end;
    Repeat t:=t+dt; if 4*t>0*6.28 then begin
Ad(50,50)^:=30*sin(t*1); Ad(50,51)^:=30*sin(t*1);
Ad(21,50)^:=30*sin(t*1); Ad(21,51)^:=30*sin(t*1); end;
    Raschet; Draw; until KeyPressed; Release(HeapOrg);
CloseGraph;
END.

```

ТПР-12.

```

uses crt, graph;
Const m=0.03; r=0.15; dt=0.001; q=150; N=150; w1=10;
Var F,eps,t : real; i,x,Gd,Gm : integer;
fi,w : array [0..N] of real;
BEGIN Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
  Repeat t:=t+dt; delay(2); cleardevice;
  For i:=1 to N-1 do begin x:=10+4*i;
    F:=q*(fi[i-1]-fi[i])+q*(fi[i+1]-fi[i]);
    eps:=(F-r*w[i]+0.01*sin(fi[i]))/m;
    w[i]:=w[i]+eps*dt; fi[i]:=fi[i]+w[i]*dt;
    If w1*t<3.142{/2} then fi[1]:=2.5*sin(w1*t) else
      begin fi[1]:=0{2.5}; w[1]:=0; end; end;
  For i:=1 to N-1 do begin x:=10+4*i;
    circle(x,370-round(fi[i]*100),2);
      circle(x,370-round(fi[i]*100),3);
    line(10+4*(i-1),369-round(fi[i-1]*100),x,
      369-round(fi[i]*100)); end;
until KeyPressed; CloseGraph;
END.

```

ТПР-13.

```

Uses crt, graph;
Const n=200; h=1; dt=0.004; vv=16;
Var i,j,DV,MV,EC: integer;
  f,f1,f2: array[1..N] of real; t: real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
  Repeat t:=t+dt;
  If t/2<6.28/2 then f1[1]:=0.7*sin(t/2) else f1[1]:=0;
  For i:=2 to N-1 do f2[i]:=2*f1[i]-f[i]+vv*(f1[i-1]-
  2*f1[i]+f1[i+1])/h/h*dt*dt+0.0005*sin(f1[i])*dt*dt;
  For i:=2 to N-1 do begin f[i]:=f1[i]; f1[i]:=f2[i]; end;
  f[N]:=f1[N-1]; f1[N]:=f2[N-1];
  For i:=1 to N-1 do begin setcolor(black);
  line(i*3-3,240-round(f[i]*250),(i+1)*3-3,
    240-round(f[i+1]*250));
  line(i*3-2,241-round(f[i]*250),(i+1)*3-2,
    241-round(f[i+1]*250));
  setcolor(white);
  line(i*3-3,240-round(f1[i]*250),(i+1)*3-3,
    240-round(f1[i+1]*250));
  line(i*3-2,241-round(f1[i]*250),(i+1)*3-2,
    241-round(f1[i+1]*250)); end;
  until KeyPressed; CloseGraph;
END.

```

## ЛИТЕРАТУРА

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-х частях. Часть 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. [Электронный ресурс] /URL: <http://maier-rv.glazov.net>, <http://komp-model.narod.ru>.
6. Майер Р.В. Компьютерное моделирование физических явлений. -- Глазов, ГГПИ: 2009. -- 112 с.
7. Поттер Д. Вычислительные методы в физике.-- М.: Мир, 1975.-- 392 с.
8. Тихонов А.Н., Самарский А.А. Уравнения математической физики. -М.: Наука, 1966. -- 724 с.
9. Федоренко Р.П. Введение в вычислительную физику: Учеб. пособие для вузов. -- М.: Изд-во Моск. физ.-техн. ин-та, 1994. -- 528 с.