МОДЕЛИРОВАНИЕ ФИЗИЧЕСКИХ ПРОЦЕССОВ

Лекция 2. Часть 1

СОДЕРЖАНИЕ

ДЕТЕРМИНИРОВАННЫЕ МОДЕЛИ

Физика – это наука, в которой компьютерное моделирование является весьма важным методом исследования.

Исторически так сложилось что моделирование начиналось именно с построения моделей физических процессов.

Детерминированные модели

При построении моделей используются два принципа:

- дедукции (от общего к частному);
- индукции (от частного к общему).

В зависимости от способа построения различают дедуктивные или детерминированные и индуктивные (недетерминированные) модели.

ДЕТЕРМИНИРОВАННЫЕ МОДЕЛИ

Построение детерминированных моделей основано на использовании фундаментальных законов – именно такие модели строятся при моделировании физических процессов.

Если же фундаментальные законы, управляющие моделируемым явлением неизвестны, как это часто бывает при моделировании в биологии, социологии, экономике - то используются гипотезы.

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

В рассматриваемых ниже физических задачах фундаментальную роль играет второй закон Ньютона.

Он гласит, что ускорение, с которым движется тело, прямо пропорционально действующей на него силе (если их несколько, то равнодействующей, т.е. векторной сумме сил) и обратно пропорционально его массе:

$$\vec{a} = \frac{\vec{F}}{m}.$$

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

В ситуации, когда сила или масса не являются величинами постоянными, необходимо записать этот закон в более общей математической форме.

Поскольку ускорение есть приращение скорости $\vec{v}(t)$, а скорость – приращение перемещения $\vec{s}(t)$, то

$$\frac{d\vec{s}}{dt} = \vec{v}(t), \quad \frac{d\vec{v}}{dt} = \frac{\vec{F}(t, \vec{v}, \vec{s})}{m(t, \vec{v}, \vec{s})}$$

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Примем, что тело массой m падает с высоты h с начальной скоростью v_0.

На тело действует сила тяжести $F=mg$, направленная вниз и сила сопротивления среды

$$F_c = k_1 v + k_2 v^2$$

Падение тела описывается 2 законом Ньютона:

$$ma=mg - Fc$$

в одномерной системе координат с осью x, направленной вниз, и с началом в точке начального падения тела.

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Рассматриваемое движение является одномерным. Проецируя силу $\vec{F}$, скорость $\vec{v}$ и перемещение h на ось, направленную вертикально вниз, получаем

$$\frac{dh}{dt} = v, \qquad (1)$$
$$\frac{dv}{dt} = \frac{mg - k_1 v - k_2 v^2}{m}$$

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Сила сопротивления среды $F_c = k_1 v + k_2 v^2$ зависит от скорости тела и его сечения,

k_1 – коэффициент Стокса, зависит от вязкости среды, большая величина;

k_2 – коэффициент лобового сопротивления, зависит от площади сечения тела, маленькая величина.

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Если скорость не очень большая, то доминирует линейная составляющая, квадратичной составляющей $k_2 v^2$ можно пренебречь.

При более высоких скоростях напротив, резко возрастает квадратичная составляющая, а линейной составляющей $k_1 v$ можно пренебречь.

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Что подразумевается под моделированием движения какого-либо тела?

Это означает, что в каждый момент времени t_i мы должны знать положение тела в пространстве или пройденный им путь $x = x(t)$, его скорость $v = v(t)$ и ускорение $a = a(t)$, которые будут являться функциями от времени.

В начальный момент времени $t_0 = 0, x_0 = 0$, $v_0 = 0, a_0 =$g.

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Для построения расчетной модели предположим, что в течение малого промежутка времени $\Delta t = \tau$ движение равноускорено, тогда можно использовать известные законы прямолинейного равноускоренного движения.

$$x = \frac{x_o + v_0\tau + a\tau^2}{2}$$

$$v = v_0 + a\tau$$

$$a = const$$

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Теперь можно построить такой вычислительный процесс:

$t_0=0,$	$x_0=0,$	$v_0=0,$	$a_0=g$
$t_1=t_0+\tau$	$x_1=x_0+v_0+a_0\tau^2/2$	$v_1=v_0+a_0\tau$	$a_1=(mg-k_1v_1-k_2v_1^2)\ /\ 2$
и т.д., далее пошли итерации, в i момент времени			
$t_i=t_0+i\tau,$	$x_i=x_{i-1}+v_{i-1}+a_{i-1}\tau^2/2$	$v_i=v_{i-1}+a_{i-1}\tau$	$a_i=(mg-k_1v_i-k_2v_i^2)\ /\ 2$
Процесс закончен, когда $x_i=h$			

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Пример 1. Свободное падение при затяжном прыжке.

Парашютист совершает затяжной прыжок. Считая массу парашютиста заданной (m=80 кг), определить, начиная с какого времени, после начала полета скорость человека в затяжном прыжке становится постоянной.

Построить график зависимости скорости падения «безпарашютиста» от времени.

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Решение.

Нужно определить характер изменения скорости со временем, если все параметры, входящие в уравнения системы (1), заданы.

При такой постановке модель носит дескриптивный характер.

Ясно, что при наличии сопротивления, растущего со скоростью, в какой-то момент сила сопротивления сравняется с силой тяжести, после чего скорость больше возрастать не будет.

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Выберем путь численного моделирования.

Итак, математическая модель выражается системой дифференциальных уравнений (1), однако, поскольку нужен график изменения скорости, то будем рассматривать только второе уравнение системы (1).

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Скорость движения достаточно большая, поэтому вкладом линейной составляющей силы сопротивления $k_1 v$ можно пренебречь.

Тогда получим систему дифференциальных уравнений:

$$\frac{dh}{dt} = v,$$
$$\frac{dv}{dt} = \frac{mg - k_2 v^2}{m} \qquad (2)$$

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Из данных уравнений будем рассматривать только второе уравнение.

Здесь

v – скорость,

t – время,

h – высота,

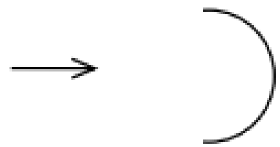
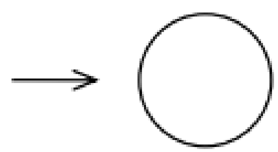
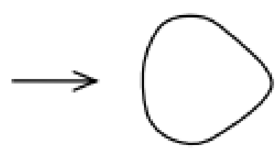
m – масса,

g – ускорение свободного падения,

k_2 – коэффициент лобового сопротивления.

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Таблица 1. Коэффициенты лобового сопротивления

	Полусфера	$c = 1{,}33$
	Шар	$c = 0{,}4$
	Каплевидное тело	$c = 0{,}045$

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Величина k_2 пропорциональна площади сечения тела S, поперечного по отношению к потоку, и плотности среды $\rho_{\text{среды}}$ и зависит от формы тела.

Обычно представляют k_2=0,5$cS\rho_{\text{среды}}$,

где c – коэффициент лобового сопротивления, безразмерен.

Некоторые значения с приведены в таблице 1.

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Вычислим значение коэффициента k_2 для данной задачи.

Средний рост человека возьмем средний – 1,7 м, а полу обхват грудной клетки – характерный размер – это приблизительно 0,4 м.

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Выберем число c = 1,22 как среднее между коэффициентами для диска и для полусферы (выбор для качественной оценки правдоподобен).

Оценим площадь поперечного сечения:

$$S = 1{,}7 \cdot 0{,}4 = 0{,}7\ [м^2].$$

$$\rho_{среды} = 1{,}29\ [кг/м^3].$$

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Тогда

$$k_2 = 0{,}5cS\,\rho_{среды} = 0{,}5 \cdot 1{,}22 \cdot 0{,}7 \cdot 1{,}29 = 0{,}55083\ [кг/м].$$

m – масса парашютиста, m = 80 кг.

g – ускорение свободного падения, g=9,8 [$кг$/м^2]

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Теперь можно приступить к численному решению задачи.

При этом следует воспользоваться одним из известных численных методов интегрирования систем обыкновенных дифференциальных уравнений.

Решение находится с помощью, так называемого, модифицированного метода Эйлера – метода Эйлера-Коши.

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Метод Эйлера-Коши. В основе метода ломаных Эйлера лежит идея графического построения решения дифференциального уравнения, однако этот метод дает одновременно и способ нахождения искомой функции в численной форме.

Пусть дано уравнение

$$y' = f(x, y) \tag{3}$$

с начальным условием

$$y(x_0) = y_0 \tag{4}$$

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Получение значений искомой функции *y(x)* по методу Эйлера заключается в циклическом применении пары формул:

$$\Delta y_i = hf(x_i, y_i)$$
$$y_{i+1} = y_i + \Delta y_i \qquad (5)$$

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Существуют различные уточнения метода Эйлера, повышающие его точность.

Модификации метода обычно направлены на то, чтобы более точно определить направление перехода из точки (x_i, y_i) в точку (x_{i+1}, y_{i+1}).

В методе Эйлера-Коши, например, предлагается следующий прядок вычислений:

$$\begin{aligned} y^*_{i+1} &= y_i + h \cdot f(x_i, y_i) \\ y_{i+1} &= y_i + h \cdot \frac{f(x_i, y_i) + f(x_{i+1}, y^*_{i+1})}{2} \end{aligned} \tag{6}$$

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Геометрически это означает, что определяется направление интегральной кривой в исходной точке $(x_i,\ y_i)$ и во вспомогательной точке $(x_{i+1},\ y^*_{i+1})$, а в качестве окончательного берем среднее этих направлений.

В соответствии с методом Эйлера-Коши запишем итерационные уравнения нахождения значения скорости v_{i+1} в следующий момент времени из предыдущего значения v_i .

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Обозначим через τ – шаг по времени: $\tau = t_{i+1} - t_i$.

Обозначим правую часть второго уравнения системы (1) через:

$$f(t_i, v_i) = \frac{mg - k_2 v_i^2}{m} \tag{7}$$

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Тогда в момент времени t_{i+1} согласно методу Эйлера-Коши запишем формулы:

$$v_{i+1}^{*} = v_i + \tau \cdot f(t_i, v_i),$$
$$v_{i+1} = v_i + \tau \cdot \frac{f(t_i, v_i) + f(t_{i+1}, v_{i+1}^{*})}{2}. \qquad (8)$$

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

В итоге, подставляя (7) в формулы (8), в итоге получим:

$$v_{i+1} = v_i + \frac{\tau}{2}[(mg - k_2 v_i^{\,2}) / m + (mg - k_2 \cdot (v_i + \tau(mg - k_2 v_i^{\,2}) / m)^2) / m]$$

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Параметры модели: *статические*

- h – высота, с которой падает тело;
- v_0 – начальная скорость падения, в частности, v_0=0;
- m – масса тела;
- g – ускорение свободного падения;

Параметры модели: *динамические*

- τ – шаг по времени,
- k – коэффициент сопротивления.

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Замечание. Быстрее всего протекает процесс без сопротивления (нижняя оценка), и наоборот, самый медленный процесс, когда ускорение равно нулю, т.е. движение установившееся и происходит с постоянной скоростью.

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Таблица 2. Значения вязкости и плотности жидкостей

Жидкость	***μ***	***ρ, кг/м³***
Бензин	0,65	$0{,}75 \cdot 10^3$
Глицерин	1499	$1{,}26 \cdot 10^3$
Керосин	2,17	$0{,}78 \cdot 10^3$
Мазут	3870	$0{,}8 \cdot 10^3$

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Таблица 3. Значения плотности разных веществ

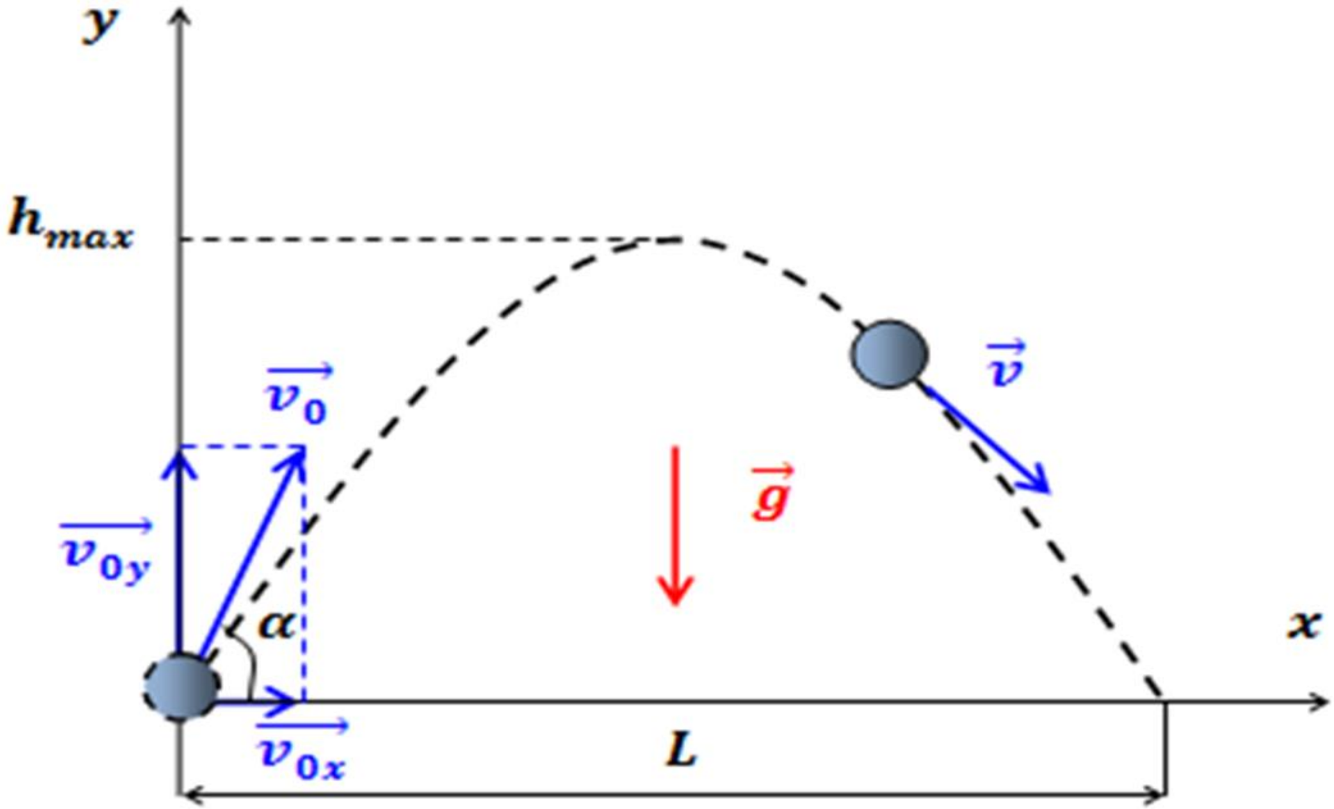
Вещество	***ρ, кг/м³***	***Вещество***	***ρ, кг/м³***
Воздух	1,29	Олово	$7{,}3\cdot10^{3}$
Вода	$1\cdot10^{3}$	Титан	$4{,}5\cdot10^{3}$
Глицерин	$1{,}26\cdot10^{3}$	Резина	$1{,}2\cdot10^{3}$
Алюминий	$2{,}7\cdot10^{3}$	Сталь	$7{,}7\cdot10^{3}$
Серебро	$10{,}5\cdot10^{3}$	Никель	$8{,}8\cdot10^{3}$
Чугун	$7{,}8\cdot10^{3}$	Золото	$19{,}3\cdot10^{3}$
Кварцевое стекло	$2{,}21\cdot10^{3}$	Медь	$8{,}89\cdot10^{3}$

МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ТЕЛА, БРОШЕННОГО ПОД УГЛОМ К ГОРИЗОНТУ

Будучи брошенным, под углом α к горизонту с высоты h с начальной скоростью v_0, тело летит без учета сопротивления воздуха по параболе и через некоторое время падает на землю.

МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ТЕЛА, БРОШЕННОГО ПОД УГЛОМ К ГОРИЗОНТУ

Рисунок 1. Графическая модель



МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ТЕЛА, БРОШЕННОГО ПОД УГЛОМ К ГОРИЗОНТУ

Модель движения тела, брошенного под углом к горизонту строиться аналогично предыдущей модели.

Данная модель будет двумерной, то есть вектора скорости и ускорения необходимо будет разложить по двум осям и учитывать составляющие вектора скорости v по x и по y:

$$v_x \quad v_y$$

и вектора ускорения a по x и по y:

$$a_x \quad a_y$$

МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ТЕЛА, БРОШЕННОГО ПОД УГЛОМ К ГОРИЗОНТУ

Сила сопротивления будет направлена обратно движению, сила тяжести – по-прежнему направлена вниз.

$$X\colon ma_x = -k_1 v_x - k_2 v_x^2$$

$$\mathbf{Y}\colon ma_y = -(mg - k_1 v_y - k_2 v_y^2)$$

Начальные условия в этой модели:

$T_0=0$	$x_0=0$	$y_0=h$	$v_0^x=v_0\cos\alpha$	$v_0^y=v_0\sin\alpha$	$a_0^x=0$	$a_0^y=-g$

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Построим расчетную модель

$x = \dfrac{x + v0_x\tau + a0_x\tau^2}{2}$ $y = \dfrac{x + v0_y\tau + a0_y\tau^2}{2}$	$v_x = v0_x + a0_x\tau$ $v_y = v0_y + a0_y\tau$	$a_x = -(k_1 v_x - k_2 v_x^2)/m$ $a_y = -(mg - k_1 v_y - k_2 v_y^2)/\mathbf{m}$

МОДЕЛИРОВАНИЕ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Параметры модели:

- h – начальная высота бросания, в частности, h=0;
- v_0 – начальная скорость бросания тела;
- α – угол бросания;
- m – масса тела;
- k_1 – коэффициент сопротивления среды;
- k_2 – коэффициент лобового сопротивления;
- g – ускорение свободного падения;
- τ – шаг по времени.

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Задача 1. Тело брошено с некоторой высоты с начальной скоростью, направленной под углом к горизонту. Определить угол при котором дальность полета будет максимальной.

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Этап 1. Построение информационной модели

В данном случае с использованием физических понятий создается идеализированная модель движения объекта.

Из условия задачи можно сформулировать следующие основные предположения:

1) тело мало по сравнению с Землей, поэтому его можно считать материальной точкой;

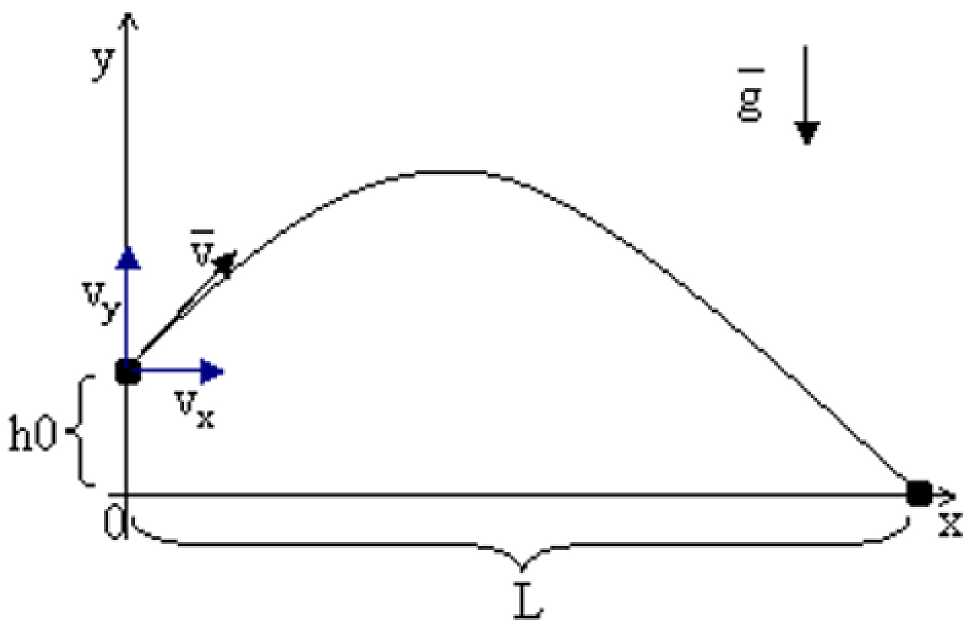
2) скорость бросания тела мала, поэтому:

- ускорение свободного падения считать постоянной величиной;
- сопротивлением воздуха можно пренебречь.

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Этап 2. Построение формализованной модели

Полученная информационная модель записывается с помощью какого-либо формального языка



ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Из курса физики известно, что описанное выше движение является равноускоренным. При заданных начальной скорости (v_0), начальной высоте (h_0), угле бросания зависимость дальности полета (L) можно описать следующими математическими формулами:

Зависимость координат от времени:

$$x = v_0 t \cos \alpha$$

$$y = y_0 + v_0 t \sin \alpha - \frac{gt^2}{2}$$

Отсюда

$$0 = h_0 + v_0 t \sin \alpha - \frac{gt^2}{2}$$

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Получаем квадратное уравнение:

$$gt^2 - 2v_0 t \sin \alpha - 2h_0 = 0$$

Решаем данное уравнение

$$D = (2v_0 \sin \alpha)^2 + 4g\,2h_0$$

$$t_{1,2} = \frac{2v_0 \sin \alpha \pm \sqrt{(2v_0 \sin \alpha)^2 - 8gh}}{2g}$$

Дальность полета:

$$L = v_0 t \cos \alpha$$

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Этап 3. Построение компьютерной модели

Существуют два принципиально различных пути построения компьютерной модели:

1. Создание алгоритма решения задачи и его кодирование на одном из языков программирования;
2. Формирование компьютерной модели с использованием одного из приложений (электронных таблиц, СУБД и т. д.).

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Для реализации первого пути надо построить алгоритм и закодировать его на одном из языков программирования.

В программе используются две константы:

Const PI = 3.14159265

Const g = 9.8

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Дальность полета, если известны начальные данные, можно определить так:

D = (2 * v0 * Sin(a * (PI / 180))) ^ 2 + 4 * g * 2 * h0
t1 = (2 * v0 * Sin(a * (PI / 180)) + Sqr(D)) / (2 * g)
t2 = (2 * v0 * Sin(a * (PI / 180)) - Sqr(D)) / (2 * g)

Корень t_2 не удовлетворяет нашей задаче. Значит время полета – это t_1.

L = v0 * Cos(a * (PI / 180)) * t1

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Определить искомый угол можно используя следующий алгоритм:

```
aa = 0 'искомый угол
maxL = 0 'максимальная дальность полета
For a = 0 To 90
D = (2 * v0 * Sin(a * (PI / 180))) ^ 2 + 4 * g * 2 * h0
t1 = (2 * v0 * Sin(a * (PI / 180)) + Sqr(D)) / (2 * g)
L = v0 * Cos(a * (PI / 180)) * t1
If L >= maxL Then
maxL = L
aa = a
End If
Next
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Проект, определяющий искомый угол, и график движения тела:

Объект	***Свойство***	***Значение***
Форма	имя	frmFlight
	Caption	Движение тела
Frame	Caption	Начальные данные:
Frame	Caption	Масштаб:
Label (надпись)	Caption	Высота (м):
Label (надпись)	Caption	Скорость (м/с):
Label (надпись)	Caption	Угол (град):
Label (надпись)	Caption	Макс. дальность достигается при угле:
Label (надпись)	Caption	Максимальная дальность:
Label (надпись)	Caption	Минимальное значение X:
Label (надпись)	Caption	Минимальное значение Y:
Label (надпись)	Caption	Максимальное значение X:
Label (надпись)	Caption	Максимальное значение Y:

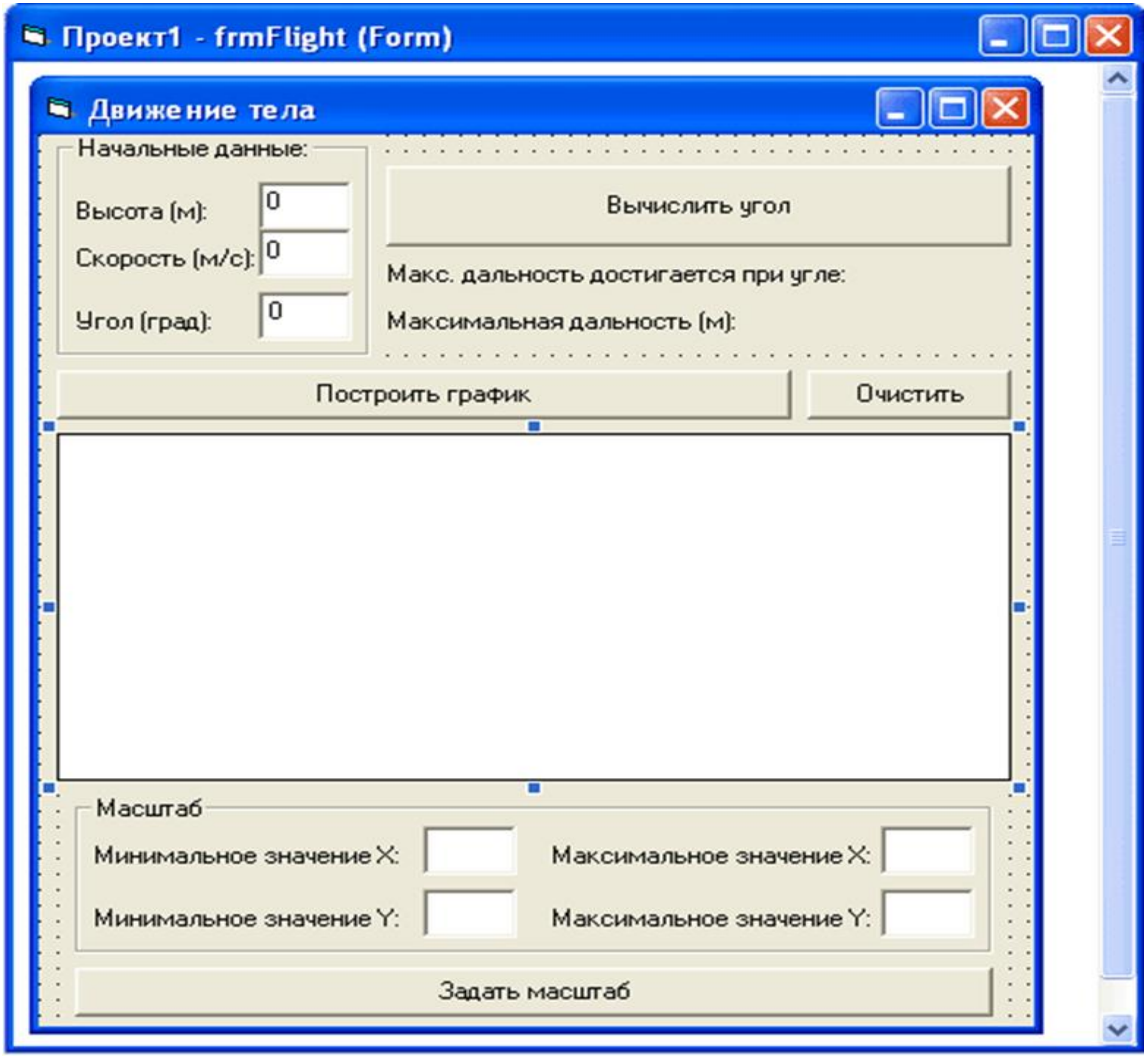
ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Объект	***Свойство***	***Значение***
Текстовое поле	имя	txtH0
	Text	0
Текстовое поле	имя	txtV0
	Text	0
Текстовое поле	имя	txtA0
	Text	0
Текстовое поле	имя	txtMinX
	Text	*(пустая строка)*
Текстовое поле	имя	txtMaxX
	Text	*(пустая строка)*
Текстовое поле	имя	txtMinY
	Text	*(пустая строка)*
Текстовое поле	имя	txtMaxY
	Text	*(пустая строка)*

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Объект	***Свойство***	***Значение***
Кнопка	имя	cmdCalculate
	Caption	Вычислить угол
Кнопка	имя	cmdPaint
	Caption	Построить график
Кнопка	имя	cmdClear
	Caption	Очистить
Кнопка	имя	cmdNewScale
	Caption	Задать масштаб
Графическом поле (PictureBox)	имя	picGrafic
	Appearance	0-плоская
	AutoRedraw	true

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ



ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Код проекта:

В разделе "General" опишем некоторые переменные и константы:

```
Dim v0, h0, D, a, L As Double
Dim XMin, YMin, YMax, XMax As Integer
Const PI = 3.14159265
Const g = 9.8
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Координатная сетка:

```
Dim v0, h0, D, a, L As Double
Dim XMin, YMin, YMax, XMax As Integer
Const PI = 3.14159265
Const g = 9.8
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Координатная сетка:

```
Private Sub cmdNewScale_Click()
XMin = txtMinX
YMin = txtMinY
XMax = txtMaxX
YMax = txtMaxY
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Координатная сетка:

picGrafic.Cls

SetScale

End Sub

Private Sub Form_Load()

XMin = -1

YMin = -1

YMax = 10

XMax = 10

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Координатная сетка:

```
txtMinX = XMin
txtMinY = YMin
txtMaxX = XMax
txtMaxY = YMax

SetScale
End Sub
Private Sub SetScale()
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Координатная сетка:

```
picGrafic.Scale (XMin, YMax)-(XMax, YMin)
'координатная сетка
For x = XMin To XMax
picGrafic.Line (x, YMax)-(x, YMin), RGB(240, 240, 240)
Next x

For y = YMin To YMax
picGrafic.Line (XMin, y)-(XMax, y), RGB(240, 240, 240)
Next y
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Координатная сетка:

'оси координат

picGrafic.Line (0, YMin)-(0, YMax), RGB(150, 150, 150)

picGrafic.Line (XMin, 0)-(XMax, 0), RGB(150, 150, 150)

'подпись осей

picGrafic.ForeColor = RGB(150, 150, 150)

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Координатная сетка:

```
For x = XMix To XMax 'Step Int((XMax - XMin) / 5)
picGrafic.CurrentX = x
picGrafic.CurrentY = 0
picGrafic.Print x
Next x
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Координатная сетка:

```
For y = YMix To YMax 'Step Int((YMax - YMin) / 5)
picGrafic.CurrentX = 0
picGrafic.CurrentY = y
picGrafic.Print y
Next y
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Координатная сетка:

```
picGrafic.ForeColor = RGB(100, 0, 0)
picGrafic.CurrentX = XMax - picGrafic.TextWidth("X, м")
picGrafic.CurrentY = 0 - picGrafic.TextHeight("X, м")
picGrafic.Print "X, м"
picGrafic.CurrentX = 0 - picGrafic.TextWidth("Y, м")
picGrafic.CurrentY = YMax
picGrafic.Print "Y, м"
End Sub
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Вычисление угла и дальности полета проведем по рассмотренным выше формулам:

```
Private Sub cmdCalculate_Click()
h0 = Val(txtH0)
v0 = Val(txtV0)
aa = 0 'искомый угол
maxL = 0 'максимальная дальность полета
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Вычисление угла и дальности полета проведем по рассмотренным выше формулам:

```
For a = 0 To 90
D = (2 * v0 * Sin(a * (PI / 180))) ^ 2 + 4 * g * 2 * h0
t1 = (2 * v0 * Sin(a * (PI / 180)) + Sqr(D)) / (2 * g)
L = v0 * Cos(a * (PI / 180)) * t1
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Вычисление угла и дальности полета проведем по рассмотренным выше формулам:

```
If L >= maxL Then
maxL = L
aa = a
End If
Next
lblResultA.Caption = aa
lblResultL.Caption = maxL
End Sub
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Код рисования графика движения тела:

```
Private Sub cmdPaint_Click()
h0 = Val(txtH0)
v0 = Val(txtV0)
a = Val(txtA0)
D = (2 * v0 * Sin(a * (PI / 180))) ^ 2 + 4 * g * 2 * h0
t1 = (2 * v0 * Sin(a * (PI / 180)) + Sqr(D)) / (2 * g)
For t = 0 To t1 Step 0.001
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Код рисования графика движения тела:

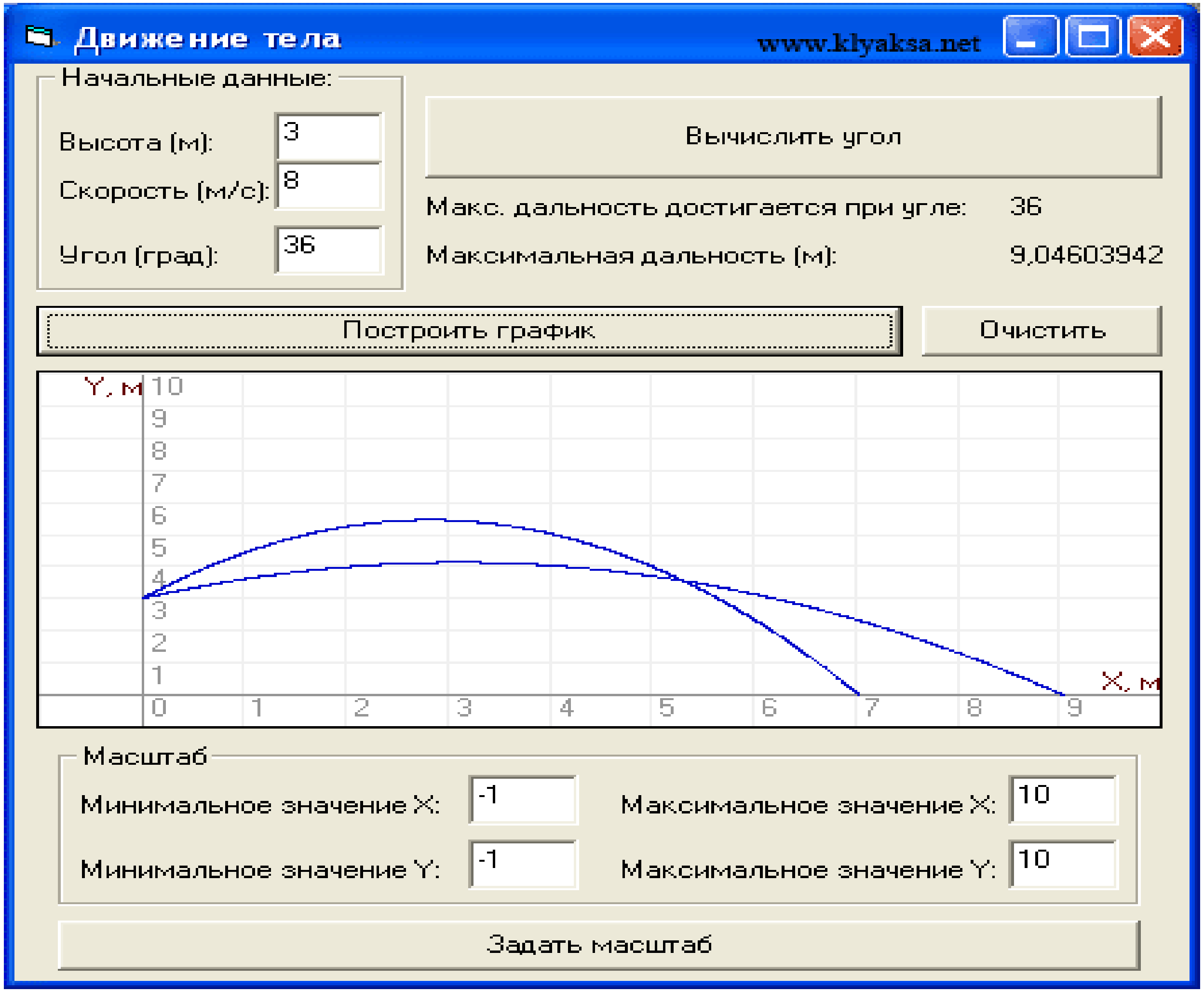
```
x = v0 * Cos(a * (PI / 180)) * t
y = h0 + v0 * Sin(a * (PI / 180)) * t - g * t * t / 2
picGrafic.PSet (x, y), RGB(0, 0, 200)
Next t
End Sub
Private Sub cmdClear_Click()
picGrafic.Cls
SetScale
End Sub
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Этап 4. Компьютерный эксперимент

На основе полученной программы необходимо провести эксперимент и получить результаты

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ



ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Этап 5. Анализ модели

В данной модели необходимо учесть, что не имеет физического смысла вычисление координаты тела после его падения на поверхность Земли.

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Задача 2. Рассчитать время падения шара в воде с точностью до 0,01 секунды.

Рекомендации

- приближенные численные вычисления с точностью 0,001 (dt)
- число шагов вычислений $n = 100$
- результат округлить до 0,01

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Код программы:

```
Const Ro_shar=7800;
         Ro_sreda=1000;
         Mju=1.02; h=10;  v0=0;  r=0.05; g=9.8;
Var i,n: integer;
     t, y, dt, m, V, k1, k2:real;
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Код программы:

```
Begin
k1:=6*Pi*Mju*r; k2:=0.2*Pi*r*r*Ro_sreda;
m:=4/3*Pi*r*r*r*Ro_shar;
Write ('шаг по времени:');readln (dt);
Write ('Число шагов');readln (n);
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Код программы:

```
i:=0; t:=0; v:=v0; y:=h+v*dt;
While y>0 do
Begin
i:=i+1; t:=t+dt;
v:=v+(k1*V+k2*v*v-m*g)/m*dt;
If I mad n=0 then writeln (t:7:4,abs(V):7:4,y:7:4);
y:=y+V*dt;
End;
writeln (‘Tmax=‘,t:7:4,’Vmax=‘,abs(V):7:4)
End.
```

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ

Тестирование программы:

При

$\Delta t = 0{,}001$

T_{max} = 2,23 сек

V_{max} = 5,355 м\с

СПАСИБО ЗА ВНИМАНИЕ!!!