



**Малахов Виктор (10 класс)**

**«Компьютерное моделирование движения материальной точки»**  
Работа прошла апробацию на Всероссийском форуме научной молодежи  
«Шаг в будущее» (г. Москва) в секции «Цифровые технологии в  
производстве»



Научный руководитель: кандидат технических наук, преподаватель  
информатики и физики Лицея № 1 «Спутник»  
**Бенькович Михаил Анатольевич**

## **Введение**

**Актуальность проекта:** Созданная в данной работе компьютерная графическая модель движения может использоваться как в процессе обучения основам механики, так и в процессе наглядного прогнозирования результатов движения физических тел. Работа выполнялась по заданию научного руководителя с перспективой её использования в качестве учебного пособия на уроках физики.

С данной программой будет проще и интереснее изучать основы механики. Эта программа отобразит любую траекторию материальной точки по заданным ранее начальным данным.

При разработке программы был выбран модульный принцип её построения. Это позволяет подключать к ней новые модули для моделирования других физических ситуаций практически без изменения ранее сделанного.

**Цель работы:** создание компьютерной графической модели для экранного представления траектории движения материальной точки.

### **Задачи:**

- изучить понятия «модель», «моделирование», «моделирование движения»;
- рассмотреть существующие виды моделей движения;
- определить состав и структуру разрабатываемой компьютерной графической модели движения; выбор языка программирования, а также разработка алгоритма для программы, реализующей графическую модель движения;
- выполнить разработку и отладку данной программы.

### **1. Изучение основных понятий и видов моделей, их описание**

**Модель** - это некоторое упрощенное подобие реального объекта, обладающее его существенными свойствами. Модели создаются, когда изучать или использовать реальный объект затруднительно или невозможно.

**Моделирование** - построение и изучение моделей реально существующих объектов, процессов или явлений с целью получения объяснений этих явлений, а также для предсказания явлений, интересующих исследователя. По природе модели делятся на материальные и информационные. Материальные модели обычно представляют собой физическое или предметное представление объекта. Информационные модели фактически представляют собой информацию о свойствах оригинала и его связях с внешним миром, которая может быть представлена в виде схем, карт, формул, чертежей, а также в виде компьютерных программ.

**Модели движения** применяются в механике для прогнозирования движения тел, а именно, изменения с течением времени взаимного расположения тел или их частей в зависимости от условий конкретных задач. Также целью моделирования движения может быть определение и прогнозирование значений других кинематических параметров, например, скорости и ускорения.

В данном проекте под телом понимается материальная точка. Т.е. целью моделирования является определение и прогнозирование во времени кинематических параметров материальной точки.

Основной математической моделью движения является совокупность кинематических уравнений, выражающих зависимость кинематических параметров тела (координаты, скорости и ускорения в проекциях на выбранные оси координат) от времени.

Оси координат	Положение	Скорость	Ускорение
X	$X(t)$	$V_x(t)$	$a_x(t)$
Y	$Y(t)$	$V_y(t)$	$a_y(t)$
Z	$Z(t)$	$V_z(t)$	$a_z(t)$

По сути кинематические уравнения или законы движения представляют собой функции времени и дают возможность на любой будущий или прошлый момент времени рассчитать все кинематические параметры материальной точки. Вид этих уравнений зависит от рассматриваемой физической ситуации.

Основной задачей создаваемой компьютерной модели движения является формирование и наглядное представление на мониторе компьютера траектории движения тела. Для заданных моментов времени выполняется расчёт кинематических параметров координат тела, в первую очередь координат. Затем производится пересчёт этих координат в координаты точки на экраны монитора, после чего точка визуализируется.

На данном этапе проекта рассматривались случаи движения тела в плоскости. Это позволяет пока рассчитывать кинематические параметры тела только по двум осям координат и, как следствие, легко пересчитывать полученные значения координат из реальных в экранные. В представленной модели рассмотрены три физические ситуации:

- движение тела, брошенного под углом к горизонту, в безвоздушном пространстве в поле силы тяжести;
- незатухающие колебания пружинного маятника;
- движение тела в гравитационном поле Земли.

## **2. Компьютерная графическая модель движения**

### **2.1. Движение тела, брошенного под углом к горизонту в поле силы тяжести**

Рассматривается движение такого тела в следующей системе координат. Ось X расположена параллельно поверхности Земли, которая, в свою очередь, считается плоской. Ось Y расположена перпендикулярно поверхности земли. Точка вылета расположена на оси Y. Начало отсчёта координат находится на земной поверхности. В этом случае результирующее движение тела является фактически суммой двух движений: вдоль оси X с постоянной скоростью и вдоль оси Y с ускорением свободного падения на поверхности Земли. Функции координат от времени имеют следующий вид:

$$X = (V_0 * \cos \alpha) * t$$
$$Y = Y_0 + (V_0 * \sin \alpha) * t - \frac{g*t^2}{2},$$

где

$\alpha$  - угол отклонения вектора начальной скорости от направления оси X;

$V_0$  - начальная скорость тела;

$Y_0$  - высота тела над поверхностью Земли на момент броска;

$g$  - ускорение свободного падения у поверхности Земли;

$t$  - временная метка.

Пересчёт реальных координат в экранные осуществляется по масштабу 1 пиксел – 1м.

### **2.2. Незатухающие колебания пружинного маятника**

Рассматривались колебания вдоль оси X, начало координат соответствовало положению равновесия пружинного маятника. Были использованы следующие функциональные зависимости координат от времени:

$$X = L * \cos (W * t)$$
$$Y = 0$$
$$W = \sqrt{\frac{K}{M}},$$

где

$L$  - максимальное отклонение маятника от положения равновесия,

$K$  - коэффициент жёсткости пружины маятника,

$M$  - масса колеблющегося тела;

$W$  - циклическая частота колебания маятника;

$t$  - временная метка.

Пересчёт реальных координат в экранные осуществляется по масштабу 1 пиксел – 1мм.

### 2.3. Движение тела в гравитационном поле Земли

Рассматривалась следующая система координат. Начало отсчёта совпадает с центром Земли, сами координатные оси расположены в плоскости баллистической траектории движения тела. Особенностью данной физической ситуации является высокая сложность представления кинематических уравнений движения в виде конкретных функций времени. Поэтому для расчёта координат тела в заданные временные метки использовались следующие формулы и уравнения.

Согласно закону Всемирного тяготения Исаака Ньютона модуль ускорения тела равен

$$a = \frac{G * M}{R^2}$$

где

$M$  – масса Земли;

$R$  – расстояние от тела до центра Земли;

$G$  – Гравитационная постоянная, равная  $6,67 \cdot 10^{-11} \text{ Н} \cdot \text{м}^2/\text{кг}^2$

В проекциях на оси выбранной системы координат

$$a_x = \frac{-a * X}{\sqrt{X^2 + Y^2}}$$

$$a_y = \frac{-a * Y}{\sqrt{X^2 + Y^2}}$$

где  $X$  и  $Y$  – текущее значение координат тела.

Допустим, что в некоторый момент времени тело имеет координаты  $X$  и  $Y$  и скорости  $V_x$  и  $V_y$ . Будем считать, что на малом интервале времени  $\Delta t$  тело совершает равноускоренное движения с проекциями ускорения на координатные оси  $a_x$  и  $a_y$ . Тогда координаты тела через интервал  $\Delta t$   $X'$  и  $Y'$  могут быть рассчитаны по формулам:

$$X' = X + V_x * \Delta t + a_x * \frac{\Delta t^2}{2}$$

$$Y' = Y + V_y * \Delta t + a_y * \frac{\Delta t^2}{2}$$

Для расчёта скоростей  $V'_x$  и  $V'_y$  можно использовать следующие зависимости:

$$V'_x = V_x + a_x * \Delta t$$

$$V'_y = Vy + a_y * \Delta t$$

Для получения ряда значений координат тела, достаточного для построения его траектории, подобные расчёты циклически повторяются необходимое количество раз.

Пересчёт реальных координат тела в экранные осуществляется по масштабу 1 пиксел = 50 км.

### 3. Разработка алгоритма и программы компьютерной модели движения

В качестве языка программирования высокого уровня была выбрана система программирования PaschalABC, в которой присутствует встроенная графическая библиотека с большим набором возможностей.

Структурно программа представляет собой четыре процедурных блока. В центральном блоке (имя процедуры main) разворачивается меню выбора конкретной модели. Остальные три блока реализуют три включённых в работу модели движения, а именно:

- движение тела, брошенного под углом к горизонту (процедура Parabola);
- свободные колебания пружинного маятника (процедура mayatnik);
- движение тела в гравитационном поле Земли (процедура okr).

#### 3.1. Главный модуль

Реализует меню выбора вида:



Движение тела, брошенного под углом к горизонту - 1  
Движение пружинного маятника - 2  
Движение тела в гравитационном поле Земли - 3  
Выход из программы - 0

**Сделайте выбор!**

Пользователь делает выбор модели нажатием определённой клавиши, после чего управление передаётся соответствующей процедуре модели.

Каждая процедура модели выполняет следующую последовательность действий:

- Запрос исходных данных для расчёта;
- формирование графического окна, установка средствами Паскаля математической системы координат вместо экранной;
- построение координатных осей;

- визуализация последовательных положений тела и рисование на экране траектории его движения.

В процессе рисования траектории последовательно с определённым интервалом задаются моменты времени, для которых и выполняются расчёты.

### 3.2. Работа модели движения тела, брошенного под углом к горизонту.

Экран ввода данных

GraphABC.NET

#### Модель движения брошенного тела

Укажите начальную скорость в метрах в секунду: 80

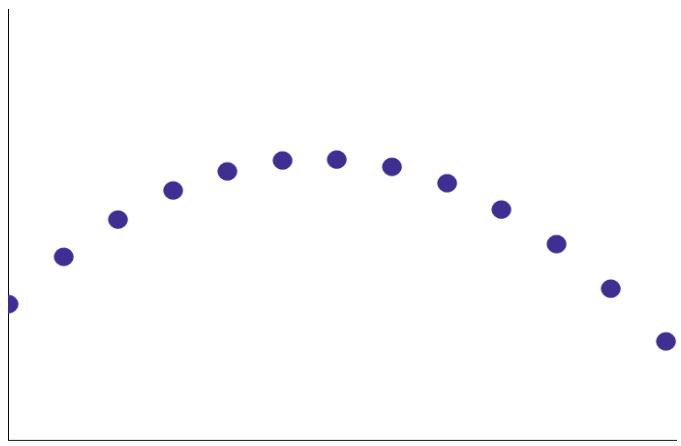
Укажите начальную высоту в метрах: 150

Укажите угол вылета в градусах: 45

Укажите начальный момент времени в секундах: 0

Укажите конечный момент времени в секундах:

#### Рабочий экран модели



### 3.3. Работа модели свободных колебаний пружинного маятника

Рабочий экран модели

#### Модель пружинного маятника

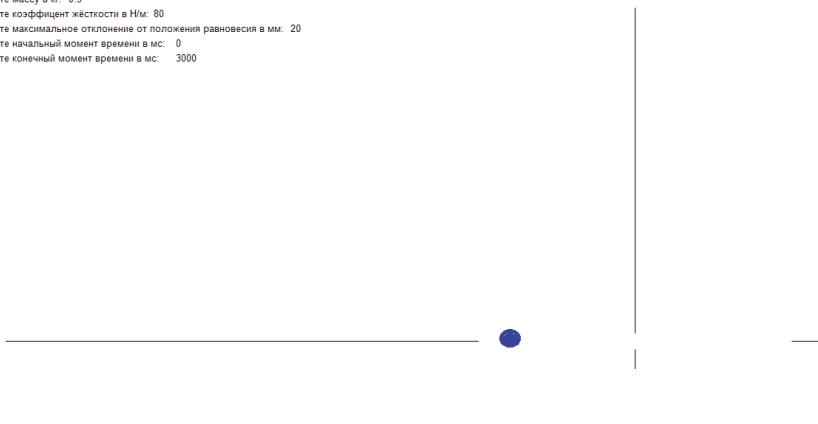
Укажите массу в кг: 0.3

Укажите коэффициент жёсткости в Н/м: 80

Укажите максимальное отклонение от положения равновесия в мм: 20

Укажите начальный момент времени в мс: 0

Укажите конечный момент времени в мс: 3000



### 3.4. Работа модели движения тела в гравитационном поле Земли.

GraphABC.NET

#### Модель движения тела в гравитационном поле Земли

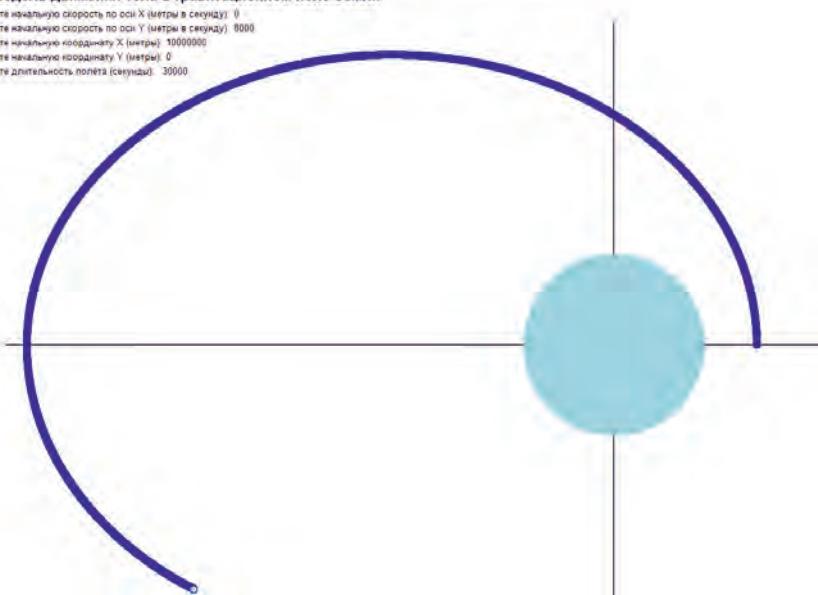
Укажите начальную скорость по оси X (метры в секунду): 0

Укажите начальную скорость по оси Y (метры в секунду): 8000

Укажите начальную координату X (метры): 10000000

Укажите начальную координату Y (метры): 0

Укажите длительность полёта (секунды): 30000



## **Заключение**

Созданная в данной работе компьютерная графическая модель уже сейчас используется на уроках физики в качестве наглядного инструмента моделирования движения физических тел.

Выбранный модульный принцип построения программы позволяет подключать к ней новые подпрограммы для моделирования других физических ситуаций практически без изменения ранее сделанного. Работы над системой продолжаются, фактически закончено моделирование движения заряженной частицы в магнитном поле под действием силы Лоренца. В планах провести моделирование движения в 3d пространстве с отображением полученной траектории движения на плоском экране.

Таким образом, поставленную в работе цель создать компьютерную графическую модель движения можно считать достигнутой, а задачи решёнными.

## **Приложение**

### **Текст программы моделирования**

```
uses graphabc;
Var V:integer;
procedure Parabola;
Var X,Y,Y0,V0,b:real;
t0,t1:integer;
Begin
  ClearWindow(CWhite);
  SetFontStyle(FsBold);
  SetFontSize(15);
  TextOut(30, 0, 'Модель пружинного маятника');
  SetFontStyle(FsNormal);
  SetFontSize(10);
  TextOut(5, 30, 'Укажите начальную скорость в метрах в секунду:');
  readln(V0); TextOut(320, 30, V0);
  TextOut(5, 50, 'Укажите начальную высоту в метрах:');
  readln(Y0); TextOut(250, 50, Y0);
  TextOut(5, 70, 'Укажите угол вылета в градусах:');
  readln(b); TextOut(225, 70, b);
  TextOut(5, 90, 'Укажите начальный момент времени в секундах:');
  readln(t0); TextOut(315, 90, t0);
  TextOut(5, 110, 'Укажите конечный момент времени в секундах:');
  readln(t1); TextOut(315, 110, t1);
  b:=(pi/180)*b;
  Coordinate.SetMathematic;
  Coordinate.SetTransform(10, 990, 0, 1, 1);
  line(0,0,0,800);
  line(0,0,1800,0);
```

```

For Var t:=t0 to t1 do Begin
  X:=(V0*cos(b))*t;
  Y:=Y0+(V0*sin(b))*t-10*t*t/2;
  SetBrushColor(ClBlue);
  FillCircle(round(X),round(Y),10);
  sleep(800);
  if Y<0 then break;
end;
  SetBrushColor(ClWhite);
  readln;
end;

procedure Mayatnik;
Var M,K,W,X,Y,L:real;
t0,t1:integer;
Begin
  ClearWindow(ClWhite);
  SetFontStyle(FsBold);
  SetFontSize(15);
  TextOut(30, 0, 'Модель пружинного маятника');
  SetFontStyle(FsNormal);
  SetFontSize(10);

  TextOut(5, 30, 'Укажите массу в кг: ');
  readln(M); TextOut(135, 30, M);
  TextOut(5, 50, 'Укажите коэффициент жёсткости в Н/м: ');
  readln(K); TextOut(250, 50, K);
  TextOut(5, 70, 'Укажите максимальное отклонение от положения равновесия
в мм: ');
  readln(L); TextOut(435, 70, L);
  TextOut(5, 90, 'Укажите начальный момент времени в мс: ');
  readln(t0); TextOut(280, 90, t0);
  TextOut(5, 110, 'Укажите конечный момент времени в мс: ');
  readln(t1); TextOut(280, 110, t1);

  W:=sqrt(K/M);
  Coordinate.SetMathematic;
  Coordinate.SetTransform(900, 500, 0, 1, 1);
  line(0,-450,0,450);
  line(-850,0,850,0);
For Var t:=t0 to t1 do Begin
  X:=L*cos(W*t/1000);
  Y:=0;
  if t mod 10 = 0 then begin
    setbrushcolor(ClBlue);

```

```

FillCircle(round(X*10),round(Y),10);
//For Var ii:=1 to 5000 do ;
sleep(1);
setbrushcolor(ClWhite);
FillCircle(round(X*10),round(Y),11);
end;
end;
readln;
end;

procedure okr;
var X,Y,V,Vx,Vy,G,a,ax,ay,td,t1,M: real;
Begin
G:=6.67E-11;
M:=5.97E24;
td:=1;
ClearWindow(ClWhite);
SetFontStyle(FsBold);
SetFontSize(15);
TextOut(30, 0, 'Модель движения тела в гравитационном поле Земли');
SetFontStyle(FsNormal);
SetFontSize(10);

TextOut(5, 30, 'Укажите начальную скорость по оси X (метры в секунду):');
readln(Vx); TextOut(370, 30, Vx);
TextOut(5, 50, 'Укажите начальную скорость по оси Y (метры в секунду):');
readln(Vy); TextOut(370, 50, Vy);
TextOut(5, 70, 'Укажите начальную координату X (метры):');
readln(X); TextOut(275, 70, X);
TextOut(5, 90, 'Укажите начальную координату Y (метры):');
readln(Y); TextOut(275, 90, Y);
TextOut(5, 110, 'Укажите длительность полёта (секунды):');
readln(t1); TextOut(270, 110, t1);

Coordinate.SetMathematic;
Coordinate.SetTransform(900, 500, 0, 1, 1);
line(0,-450,0,450);
line(-850,0,850,0);
SetBrushColor(ClLightBlue);
FillCircle(0, 0, round(6371000/50000));
setpencolor(ClBlue);
For var t:=0 to round(t1) do
begin
Circle(round(X/50000),round(Y/50000),5);
a:=G*M/(X*X + Y*Y);

```

```

ax:=-a*X/sqrt(X*X + Y*Y);
ay:=-a*Y/sqrt(X*X + Y*Y);
X:=X+Vx*td+ax*td*td/2;
Y:=Y+Vy*td+ay*td*td/2;
Vx:=Vx + ax* td;
Vy:=Vy + ay* td;
If t mod 100=0 then Begin
V:=sqrt(Vx*Vx+Vy*Vy);
end;
if sqrt(X*X + Y*Y) <6371000 then break;
end;
SetBrushColor(ClWhite);
Setpencolor(ClBlack);
readln;
end;

Begin
initwindow(5, 5, 1800, 1000, ClWhite);
While True do Begin
  ClearWindow(ClWhite);
  Coordinate.SetStandard;
  Coordinate.SetTransform(0,0,0,1,1);
  TextOut(5,0,'Движение тела, брошенного под углом к горизонту - 1');
  TextOut(5,20,'Движение пружинного маятника - 2');
  TextOut(5,40,'Движение тела в гравитационном поле Земли - 3');
  TextOut(5,60,'Выйти из программы - 0');
  SetFontSize(15);
  TextOut(5,90,'Сделайте выбор!');
  read(V);
  SetFontSize(10);

  if V = 1 Then
    Parabola
  else if V = 2 Then
    Mayatnik
  else if V = 3 Then
    okr
  else if V = 0 Then
    CloseWindow;
End;
end.

```