Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего профессионального образования

«Казанский (Приволжский) федеральный университет»

ИНСТИТУТ МАТЕМАТИКИ И МЕХАНИКИ ИМ. ЛОБАЧЕВСКОГО

КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ

Направление: 050100.68: Педагогическое образование

Профиль: Информационные технологии в физико-математическом образовании

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Практикум по введению в компьютерное моделирование

Работа завершена:

26 мая 2015 (Шарафутдинова Р.Р.)

Работа допущена к защите:

Научный руководитель, к.ф.-м.н., доцент

1 июня 2015 (Попов А.А.)

Рецензент, д.ф.-м.н.

3 июня 2015 (Щербакова Н.К.)

Дата защиты:

8 июня 2015 Оценка

Заведующий кафедрой, д.ф.-м.н., профессор

1 июня 2015 (Игнатьев Ю.Г.)

Казань 2015

Содержание

1 Введение

Многие полагают, что системы компьютерной математики хорошо работают на простых примерах, но от них мало толку при решении реальных задач математики и физики. Это, конечно, заблуждение. Дело просто в том, что при решении таких задач руководящая роль пользователя сильно возрастает. Вы должны понимать, что не компьютер решает вашу задачу, а вы! А компьютер лишь помогает в этом трудном деле. Так что при неудачах в решении своих специфических задач следует прежде всего пенять на себя и на свое незнание возможностей программного обеспечения, в котором задача решается. В работе в качестве такого программного обеспечения используется система компьютерной математики Maple. В том, что Maple можно успешно использовать при решении вполне конкретных научных и практических задач, призваны убедить примеры, приведенные ниже.

Целью квалификационной работы является:

- создание элективного курса введения в компьютерное моделирование,
- написание ряда программ в системе компьютерной математики Maple, моделирующих столкновение двух тел, двигающихся либо равномерно и прямолинейно, либо равноускоренно на плоскости.

Задачи демонстрируют возможность создания элективного курса компьютерного моделирования в средней школе.

Для разработки программ, позволяющих моделировать тот или иной процесс, от обучающихся потребуется не только знание конкретных языков программирования, но и владение методами вычислительной математики,

знания законов физики. В школе с этим большая проблема и мы ее обходим используя изучаемые с 8-го класса законы прямолинейного равномерного и равноускоренного движений:
$x(t) = x_0 + v(t - t_0)$ - закон равномерного прямолинейного движения,
$x(t) = x_0 + v(t - t_0) + \dfrac{a(t-t_0)^2}{2}$ - закон равноускоренного движения.
Это порождает возможность моделировать столкновение тел двигающихся по таким законам в плоскости.
Равномерное прямолинейное движение по обеим осям:

$$\begin{cases} x(t) = x_0 + v^x(t - t_0), \\ y(t) = y_0 + v^y(t - t_0). \end{cases} \tag{1}$$

Равномерное прямолинейное движение по одной оси и равноускоренное движение по другой оси:

$$\begin{cases} x(t) = x_0 + v^x(t - t_0), \\ y(t) = y_0 + v^y(t - t_0) + \dfrac{a^y(t-t_0)^2}{2}. \end{cases} \tag{2}$$

Равноускоренное движение по обеим осям:

$$\begin{cases} x(t) = x_0 + v^x(t - t_0) + \dfrac{a^x(t-t_0)^2}{2}, \\ y(t) = y_0 + v^y(t - t_0) + \dfrac{a^y(t-t_0)^2}{2}. \end{cases} \tag{3}$$

В работе представлены такие задачи, как

- моделирование попадания прямолинейно и равномерно движущейся торпеды в прямолинейно и равномерно движущийся корабль,

- моделирование попадания снаряда в вертикально и свободно падающую мишень,
- моделирование попадания снаряда, выпущенного под углом к горизонту, в прямолинейно и равномерно движущуюся горизонтально мишень,
- моделирование попадания снаряда в свободно падающую (не вертикально) мишень,
- моделирование движения точки на ободе прямолинейно и равномерно движущегося колеса.

2 Моделирование физических явлений

2.1 Моделирование попадания прямолинейно и равномерно движущейся торпеды в прямолинейно и равномерно движущийся корабль

Торпеда, пущенная с подводной лодки со скоростью v_1, попадает в движущийся со скоростью v_2 по поверхности воды корабль. Найти угол, под которым произведен выстрел, если известны начальные координаты подводной лодки в момент выстрела t_0 и координаты корабля в этот же момент времени. Сопротивлением воздуха пренебречь.

Решение:

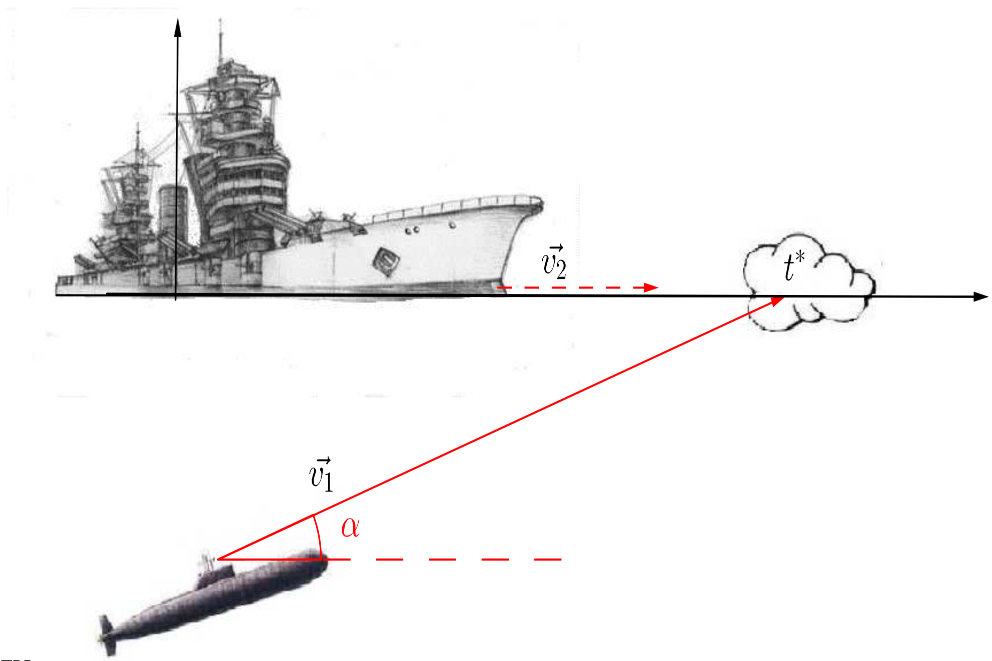
Дано:

$v_{\text{торпеды}} = v_1$

$v_{\text{коробля}} = v_2$

$t_0 = 0, (a, b)$

Найти: α



Закон движения торпеды

$$\begin{cases} x_1 = a + v_1 \cos(\alpha)(t - t_0), \\ y_1 = b + v_1 \sin(\alpha)(t - t_0). \end{cases}$$

Закон движения корабля

$$\begin{cases} x_1 = a + v_2 \cos(\alpha)(t - t_0), \\ y_1 = b + v_2 \sin(\alpha)(t - t_0). \end{cases}$$

Мы выбираем систему координат так, чтобы в начальный момент времени корабль находился в начале координат, а направление его движения совпадало с направлением оси OX. Ось OY направим вертикально вверх.

Момент начала движения торпеды и корабля выберем равным нулю.

Предположим, что торпеда тоже движется равномерно прямолинейно и что в начальный момент времени координаты её были a и b. Угол, под которым был произведен выстрел, обозначим α.

Движение торпеды, запущенной под углом α к горизонту со скоростью v_1 из точки (a, b) в пренебрежении сопротивления воздуха можно описать следующим образом:

$$\begin{cases} x_1 = a + v_1 \cos(\alpha)(t - t_0), \\ y_1 = b + v_1 \sin(\alpha)(t - t_0), \end{cases} \tag{4}$$

где

$v_1 \cos(\alpha)$ - горизонтальная составляющая начальной скорости торпеды,

$v_1 \sin(\alpha)$ - вертикальная составляющая начальной скорости торпеды.

Запишем закон движения корабля:

$$\begin{cases} x_2 = v_2 t, \\ y_2 = 0. \end{cases} \tag{5}$$

Если тела встретились, то очевидно, что их координаты в этот момент времени совпадают. Приравняем x_1 к x_2 и y_1 к y_2 и найдем угол, под которым производится выстрел α^*

$$\begin{cases} a + v_1 \cos(\alpha)(t - t_0) = v_2 t, \\ b + v_1 \sin(\alpha)(t - t_0) = 0. \end{cases} \tag{6}$$

$$\begin{cases} v_1 \cos(\alpha)(t - t_0) = v_2 t - a, \\ v_1 \sin(\alpha)(t - t_0) = -b. \end{cases} \tag{7}$$

$$\frac{v_2 t - a}{-b} = \frac{\cos(\alpha)}{\sin(\alpha)} = \mathrm{ctg}(\alpha) \;\; \Rightarrow \;\; t = \frac{a - b\mathrm{ctg}(\alpha)}{v_2} \tag{8}$$

$$b + v_1 \sin(\alpha) \left(\frac{a - b\,\mathrm{ctg}(\alpha)}{v_2} - t_0 \right) \;=\; 0 \tag{9}$$

$$b - v_1 t_0 \sin(\alpha) + \frac{a v_1 \sin(\alpha)}{v_2} - \frac{b v_1 \cos(\alpha)}{v_2} \;=\; 0 \tag{10}$$

$$\frac{b v_1}{v_2} \cos(\alpha) + \left(v_1 t_0 - \frac{a v_1}{v_2} \right) \sin(\alpha) \;=\; b \tag{11}$$

Это выражение можно переписать так:

$$\sqrt{\left(\frac{b v_1}{v_2}\right)^2 + \left(v_1 t_0 - \frac{a v_1}{v_2}\right)^2} \sin\left[\alpha + \mathrm{arctg}\left(\frac{v_2 t_0 - a}{b}\right)\right] \;=\; b \tag{12}$$

Отсюда,

$$\alpha^* \;=\; \arcsin\left(\frac{b}{\sqrt{\left(\frac{b v_1}{v_2}\right)^2 + \left(\frac{a v_1}{v_2} - v_1 t_0\right)^2}} \right) - \mathrm{arctg}\left(\frac{v_2 t_0 - a}{b}\right) \tag{13}$$

Решение этой задачи в СКМ Maple имеет вид:

```
> restart:
Нужные библиотеки.
>with(plots): with(plottools):
Процедура построения шарика
>kr:=proc(x,y,r,clr)
display(disk([x,y],r,color=clr))
end proc:
Процедура создания анимации движения торпеды в поле
тяготения Земли из начальной точки (a,b) с начальной
скоростью V1, направленной под углом alpha к горизонту
```

```
>torpeda:=proc(a,b,V1,t0,alpha,pred,clr,r)
(нач.x-координата; нач.y-координата; модуль начальной
скорости; нач.момент времени угол; время полета; цвет
линии; радиус круга)
local V1x, V1y, xт, yт, t, T; # локальные переменные
V1x:=V1*cos(alpha); # проекция нач. скорости на ось x
V1y:=V1*sin(alpha); # проекция нач. скорости на ось y
xт(t):=a+V1x*(t-t0); # закон движения вдоль оси x
yo(t):=b+V1y*(t-t0); # закон движения вдоль оси y
animate(plot,[[xт(t),yт(t),t=0..T],color=clr],T=0..pred,
scaling=constrained,thickness=2, frames=50);
animate(kr,[xт(t),yт(t),r,clr],t=0..pred,
scaling=constrained,frames=50);
display(%,%%);
end proc:
```

Процедура создания анимации движения мишени в поле
тяготения Земли из начальной точки (x0,y0)
с начальной скоростью V2.

```
>warship:=proc(V2,pred,clr,r)
(модуль начальной скорости; угол; время полета;
цвет линии; радиус круга)
local xk, yk, t, T; # локальные переменные
xk(t):=V2*t; # закон движения вдоль оси x
yk(t):=0;#закон движения вдоль оси y
```

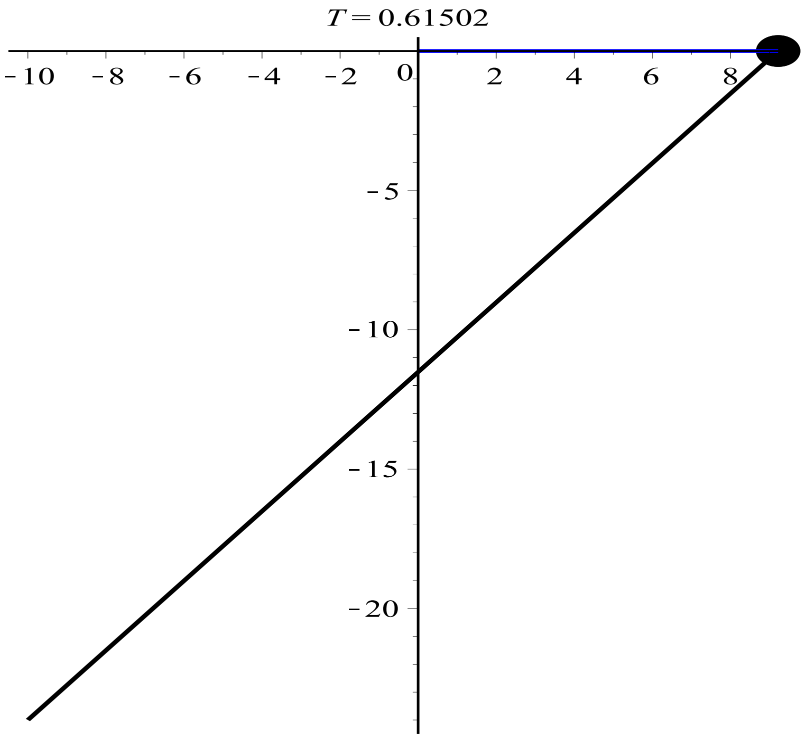
```
animate(plot,[[xm(t),ym(t),t=0..T],color=clr],T=0..pred,
scaling=constrained,thickness=2, frames=50);
animate(kr,[xk(t),yk(t),r,clr],t=0..pred,
scaling=constrained,frames=50);
display(%,%%);
end proc:
```

Процедура создания анимации полета снаряда из точки (a,b) с начальной скоростью V1, направленной под углом alpha к горизонту и движение мишени из точки (x0,y0) со скоростью V2

```
>blast:=proc(a,b,V1,V2,t0,clr1,clr2,r1,r2)
local alpha,pred,direct,anim1,anim2;
alpha:=arcsin(b/sqrt((b*V1/V2)^2+(a*V1/V2-V1*
t0)^2))-arccot((V2*t0-a)/b)+evalf(Pi);
# вычисление угла наклона начальной скорости снаряда
pred:=(a-b*cot(alpha))/V2: # вычисление времени
столкновения тел
anim1:=torpeda(a,b,V1,t0,alpha,pred,clr1,r1);
anim2:=warship(V2,pred,clr2,r2);
plots[display](anim1,anim2);
end proc:
```

Задаем начальное положение снаряда, модули начальной скорости снаряда и мишени, нач.момент времени, цвет снаряда и мишени, радиусы снаряда и мишени.

```
>blast(-10,-24,50,15,0,black,blue,0.5,0.5);
```

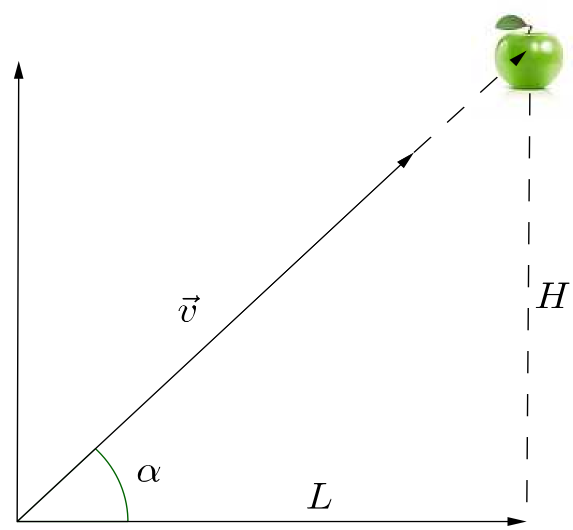


2.2 Моделирование попадания снаряда в вертикально и свободно падающую мишень

Мишень, находящаяся на высоте H, падает вертикально вниз. В нее стреляют снарядом, из пушки находящейся на поверхности земли на расстоянии L от вертикали вдоль которой падает мишень. Снаряд вылетает со скоростью v под углом α к горизонту. Найти угол, при котором произойдет столкновение снаряда с мишенью, если известно, что снаряд и мишень начинают двигаться в один и тот же начальный момент времени. Сопротивлением воздуха пренебречь.

Решение:

Дано:
$x_m = L, y_m = H, t_0 = 0,$
a— ускорение свободного падения
$x_c = 0, y_c = 0,$
$\vec{v}$— скорость снаряда
Найти: α



Движение снаряда

$$\begin{cases} x_c = v\cos(\alpha)t, \\ y_c = v\sin(\alpha)t - \dfrac{at^2}{2}. \end{cases}$$

Движение мишени

$$\begin{cases} x_m = L, \\ y_m = H - \dfrac{at^2}{2}. \end{cases}$$

Выберем декартову систему координат так, чтобы пушка находилась в начале координат, ось OX совпадала с горизонталью и была направлена в сторону вертикали по которой падает мишень. Ось OY направим вертикально вверх. Момент начала движения снаряда и мишени выберем равным нулю.

Движение снаряда, запущенного под углом α к горизонту со скоростью v из точки O в пренебрежении сопротивления воздуха можно описать следующим образом:

$$\begin{cases} x_c = v\cos(\alpha)t, \\ y_c = v\sin(\alpha)t - \dfrac{at^2}{2}, \end{cases} \tag{14}$$

где

$v\cos(\alpha)$ - горизонтальная составляющая начальной скорости снаряда,

$v\sin(\alpha)$ - вертикальная составляющая начальной скорости снаряда,

a - ускорение свободного падения.

Запишем закон движения мишени:

$$\begin{cases} x_m = L, \\ y_m = H - \frac{at^2}{2}. \end{cases} \tag{15}$$

Если тела встретились, то очевидно, что их координаты в этот момент времени совпадают. Приравняем x_c к x_m и y_c к y_m и и найдем угол, под которым производится выстрел α^*:

$$\begin{cases} v\cos(\alpha)t = L, \\ v\sin(\alpha)t - \dfrac{at^2}{2} = H - \frac{at^2}{2}. \end{cases} \tag{16}$$

$$v\cos(\alpha)t = L \;\Rightarrow\; t = \frac{L}{v\cos(\alpha)} \tag{17}$$

$$v\sin(\alpha)t-\frac{at^2}{2}=H-\frac{at^2}{2} \ \Rightarrow \ v\sin(\alpha)t=H\Rightarrow\frac{v\sin(\alpha)}{v\cos(\alpha)}L=H\Rightarrow \tag{18}$$

$$\mathrm{tg}(\alpha)L=H \ \Rightarrow \ \mathrm{tg}(\alpha)=\frac{H}{L} \tag{19}$$

$$\alpha^*=\mathrm{arctg}\left(\frac{H}{L}\right) \tag{20}$$

Выполнив необходимые алгебраические операции, мы получим, что α^* равен отношению высоты мишени к расстоянию между мишенью и пушкой, из которой был выпущен снаряд, т.е. угол бросания не зависит от начальной скорости.

Решение этой задачи в СКМ Maple имеет вид:

```
> restart:
Нужные библиотеки.
>with(plots): with(plottools):
Процедура построения шарика
>kr:=proc(x,y,r,clr)
display(disk([x,y],r,color=clr))
end proc:
Процедура создания анимации движения в поле тяготения Земли
из начальной точки (x0,y0) с начальной скоростью V,
направленной под углом alpha к горизонту
>fall:=proc(x0,y0,V,alpha,pred,clr,r)
(нач.x-координата; нач.y-координата; модуль начальной
скорости; угол; время полета; цвет линии; радиус круга)
local Vx, Vy, a, xc, yc, t, T; # локальные переменные
```

```
a:=10; # ускорение свободного падения
Vx:=V*cos(alpha); # проекция нач. скорости на ось x
Vy:=V*sin(alpha); # проекция нач. скорости на ось y
xc(t):=x0+Vx*t; # закон движения вдоль оси x
yc(t):=y0+Vy*t-a*t^2/2; # закон движения вдоль оси y
animate(plot,[[xc(t),yc(t),t=0..T],color=clr],T=0..pred,
scaling=constrained,thickness=2, frames=50);
animate(kr,[xc(t),yc(t),r,clr],t=0..pred,
scaling=constrained,frames=50);
display(%,%%);
end proc:
```

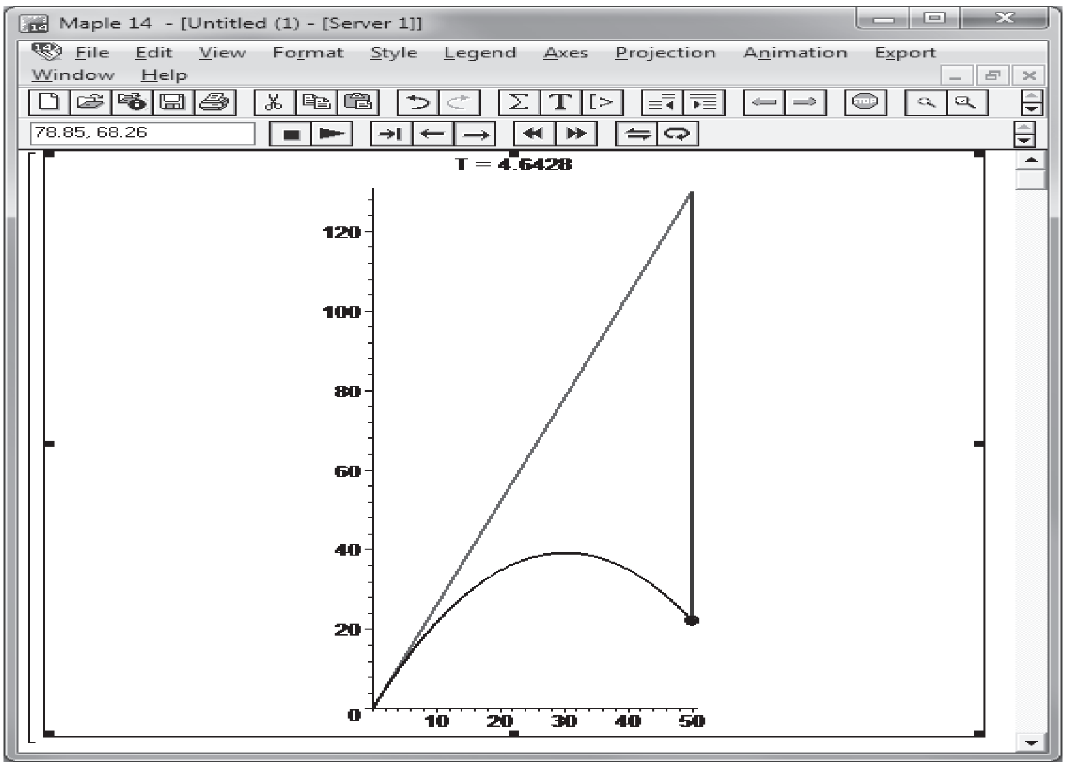
Процедура создания анимации падения тела из точки (L,H) и полет снаряда из точки(x0,y0) с начальной скоростью V.

```
>monkey:=proc(x0,y0,L,H,V,clr1,clr2,r1,r2)
local alpha,pred,direct,anim1,anim2; # локальные переменные
alpha:=arctan(H/L); # вычисление угла наклона начальной
скорости снаряда
pred:=solve(L=x0+V*cos(alpha)*t,t); # вычисление времени
столкновения тел
direct:=plottools[line]([x0,y0], [L,H], color=
COLOR(RGB,0.3,0.3,0.3),thickness=2); # построение линии,
соединяющей начальные положения тел
anim1:=fall(x0,y0,V,alpha,pred,clr1,r1);
anim2:=fall(L,H,0,0,pred,clr2,r2);
```

```
plots[display](direct,anim1,anim2);
end proc:
```

Задаем начальные положения обоих тел, модуль начальной скорости, цвет снаряда, цвет падающего тела, радиусы снаряда и тела.

```
>monkey(0,0,50,130,30,black,blue,0.5,1);
```



2.3 Моделирование попадания снаряда, выпущенного под углом к горизонту, в прямолинейно и равномерно движущуюся горизонтально мишень

Мишень летит горизонтально на высоте H со скоростью v_2 в направлении пушки, причем в момент времени $t = 0$ мишень находилась на расстоянии L от пушки по горизонтали. В момент времени $t = 0$ пушка стреляет снарядом, вылетающим со скоростью v_1 под углом α. Найти угол α, если известно, что снаряд и мишень сталкиваются. Сопротивлением воздуха пренебречь.

Решение:

Выберем декартову систему координат так, чтобы пушка находилась в начале координат, ось OX совпадала с горизонталью и была направлена в сторону вертикали по которой летит мишень. Ось OY направим вертикально вверх. Момент начала движения снаряда и мишени выберем равным нулю. Движение снаряда, запущенного под углом α к горизонту со скоростью v_1 из точки O в пренебрежении сопротивления воздуха можно описать следующим образом:

$$\begin{cases} x_c = v_1 \cos(\alpha) t, \\ y_c = v_1 \sin(\alpha) t - \dfrac{at^2}{2}, \end{cases} \tag{21}$$

где

$v_1 \cos(\alpha)$ - горизонтальная составляющая начальной скорости снаряда,

$v_1 \sin(\alpha)$ - вертикальная составляющая начальной скорости снаряда,

a - ускорение свободного падения.

Запишем закон движения мишени:

$$\begin{cases} x_m = L - v_2 t, \\ y_m = H. \end{cases} \tag{22}$$

Если тела встретились, то очевидно, что их координаты в этот момент времени совпадают. Приравняем x_c к x_m и y_c к y_m и и найдем угол, под которым производится выстрел α^*:

$$\begin{cases} v_1 \cos(\alpha) t = L - v_2 t, \\ v_1 \sin(\alpha) t - \frac{at^2}{2} = H. \end{cases} \tag{23}$$

$$v_1 \cos(\alpha) t = L - v_2 t \;\Rightarrow\; t = \frac{L}{v_1 \cos(\alpha) + v_2} \tag{24}$$

$$v_1 \sin(\alpha) t - \frac{at^2}{2} = H \;\Rightarrow\; \frac{at^2}{2} - v_1 \sin(\alpha) t + H = 0 \tag{25}$$

Получим квадратное уравнение относительно t, решив его, мы получим:

$$t = \frac{v_1 \sin(\alpha) \pm \sqrt{v_1^2 \sin^2(\alpha) - 2gH}}{g} \tag{26}$$

$$\frac{v_1 \sin(\alpha) \pm \sqrt{v_1^2 \sin^2(\alpha) - 2gH}}{g} = \frac{L}{v_1 \cos(\alpha) + v_2} \tag{27}$$

$$(v_1 \sin(\alpha) \pm \sqrt{v_1^2 \sin^2(\alpha) - 2gH}) \cdot (v_1 \cos(\alpha) + v_2) = Lg \tag{28}$$

$$\pm(v_1 \cos(\alpha) + v_2) \cdot \sqrt{v_1^2 \sin^2(\alpha) - 2gH} = Lg - v_{c1}^2 \sin(\alpha)\cos(\alpha) - v_1 v_2 \sin(\alpha) \tag{29}$$

$$(v_1 \cos(\alpha) + v_2)^2 \cdot (v_1^2 \sin^2(\alpha) - 2gH) = (Lg - v_1^2 \sin(\alpha)\cos(\alpha) - v_1 v_2 \sin(\alpha))^2 \tag{30}$$

Выполнив необходимые алгебраические операции, мы имеем:

$$(Lg)^2 + 2gH(v_1 \cos(\alpha) + v_2)^2 - 2Lgv_1 \sin(\alpha)(v_1 \cos(\alpha) + v_2) = 0 \tag{31}$$

Это уравнение можно решить численно относительно α.

Решение этой задачи в СКМ Maple имеет вид:

```
> restart:
Нужные библиотеки.
>with(plots): with(plottools):
Процедура построения шарика
>kr:=proc(x,y,r,clr)
display(disk([x,y],r,color=clr))
end proc:
Процедура создания анимации движения в поле тяготения Земли
из начальной точки (x1,y1) с начальной скоростью V1,
направленной под углом alpha к горизонту
>projectile:=proc(x1,y1,V1,alpha,pred,clr,r)
(нач.x-координата; нач.y-координата; модуль начальной
скорости; угол; время полета; цвет линии; радиус круга)
local V1x, V1y, a, xn, yn, t, T; # локальные переменные
a:=10; # ускорение свободного падения
V1x:=V1*cos(alpha); # проекция нач. скорости на ось x
V1y:=V1*sin(alpha); # проекция нач. скорости на ось y
xc(t):=x1+V1x*t; # закон движения вдоль оси x
yc(t):=y1+V1y*t-a*t^2/2; # закон движения вдоль оси y
animate(plot,[[xc(t),yc(t),t=0..T],color=clr],T=0..pred,
scaling=constrained,thickness=2, frames=50);
animate(kr,[xc(t),yc(t),r,clr],t=0..pred,
```

```
scaling=constrained,frames=50);
display(%,%%);
end proc:
```

Процедура создания анимации движения из начальной точки (x2,y2) с начальной скоростью V2.

```
>samolet:=proc(x2,y2,V2,pred,clr,r)
```

(нач.x-координата; нач.y-координата; модуль начальной скорости; время полета; цвет линии; радиус круга)

```
local V2x, V2y, a, xm, ym, t, T; # локальные переменные
a:=10; # ускорение свободного падения
V2x:=x2-V2y*t; #проекция нач. скорости на ось x
V2y:=y2; # проекция нач. скорости на ось y
xm(t):=x2-V2y*t; # закон движения вдоль оси x
ym(t):=y2; #закон движения вдоль оси y
animate(plot,[[xm(t),ym(t),t=0..T],color=clr],T=0..pred,
scaling=constrained,thickness=2, frames=50);
animate(kr,[xm(t),ym(t),r,clr],t=0..pred,
scaling=constrained,frames=50);
display(%,%%);
end proc:
```

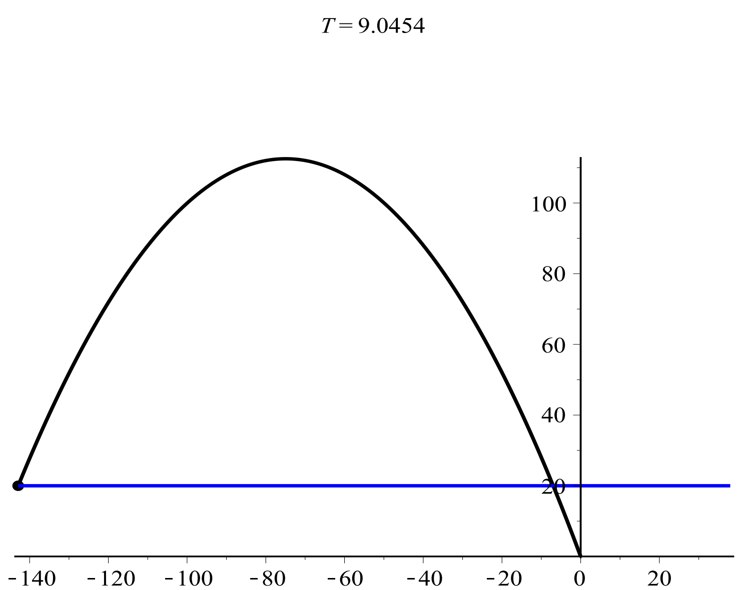
Процедура создания анимации полета снаряда из точки (x1,y1) с начальной скоростью V1 и движение самолета из точки (L,H) с начальной скоростью V2

```
>blast:=proc(x1,y1,L,H,V1,V2,clr1,clr2,r1,r2)
```

```
local alpha,pred,direct,anim1,anim2,a;
a:=10; # ускорение свободного падения
alpha:=fsolve((V1*cos(alpha)+V2)*sqrt(V1^2*sin(alpha)^2-2*a*H)=
L*a-V1^2*sin(alpha)*cos(alpha)-V1*V2*sin(alpha),alpha=0..3.14);
# вычисление угла наклона начальной скорости снаряда
pred:=solve(L-V2*t=x1+V1*cos(alpha)*t,t); # вычисление времени
столкновения тел
anim1:=projectile(x1,y1,V1,alpha,pred,clr1,r1);
anim2:=samolet(L,H,V2,pred,clr2,r2);
plots[display](anim1,anim2);
end proc:
```

Задаем начальные положения обоих тел, модули начальной скорости обоих тел, цвет снаряда, цвет самолета, радиусы снаряда и самолета.

```
>blast(0,0,38,20,50,20,black,blue,0.5,1);
```



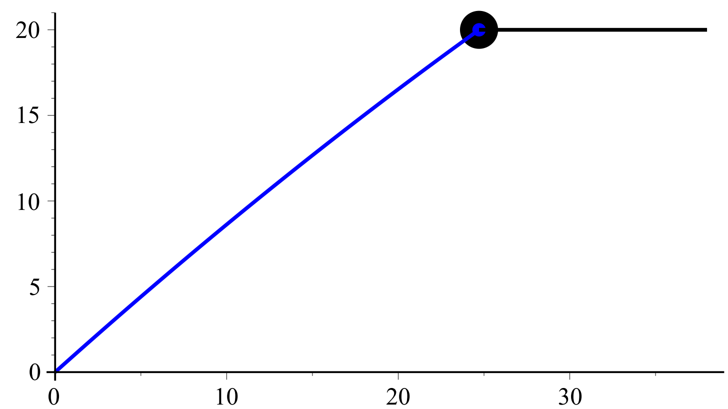
Процедура создания анимации полета снаряда из точки (x1,y1) с начальной скоростью V1 и движение самолета из точки (L,H) со скоростью V2.

```
>blast1:=proc(x1,y1,L,H,V1,V2,clr1,clr2,r1,r2)
local alpha,pred,direct,anim1,anim2,a;
a:=10;
alpha:=fsolve(-(V1*cos(alpha)+V2)*sqrt(V1^2*sin(alpha)^2-2*a*H)-L*a+
+V1^2*sin(alpha)*cos(alpha)+V1*V2*sin(alpha),alpha=0..3.14/2); # вычи-
сление угла наклона начальной скорости снаряда
pred:=solve(L-V2*t=x1+V1*cos(alpha)*t,t); # вычисление времени
столкновения тел
anim1:=projectile(x1,y1,V1,alpha,pred,clr1,r1);
anim2:=samolet(L,H,V2,pred,clr2,r2);
plots[display](anim1,anim2);
end proc:
```

Задаем начальные положения обоих тел, модули начальной скорости обоих тел, цвет снаряда, цвет самолета, радиусы снаряда и самолета.

```
>blast1(0,0,38,20,50,20,blue,black,0.5,1);
```

$T = 0.66447$



Очевидно, что задача имеет два решения. Первое, когда снаряд попадает в мишень до того,как он достигнет максимальной высоты (решение 2). Второе решение, соответствует попаданию снаряда в мишень после того, как снаряд достигнет максимальной высоты (решение 1).

Отметим, что если $v_c <$ критической, то максимальная высота снаряда будет меньше высоты полета самолета и задача решений иметь не будет.

2.4 Моделирование попадания снаряда в свободно падающую (не вертикально) мишень

Мишень, в момент времени $t = 0$, сброшена с самолета, летящего под углом β к горизонту со скоростью v_2, в этот момент самолет находится на высоте H и на расстоянии L по горизонтали от пушки. В момент времени $t = 0$ пушка стреляет снарядом, вылетающим со скоростью v_1 под некоторым углом к горизонту. Найти этот угол α, если известно, что снаряд и мишень сталкиваются. Сопротивлением воздуха пренебречь.

Решение:

Выберем декартову систему координат так, чтобы пушка находилась в начале координат, ось OX совпадала с горизонталью и была направлена в сторону мишени. Ось OY направим вертикально вверх. Момент начала движения снаряда и мишени выберем равным нулю.

Движение снаряда, запущенного под углом α к горизонту со скоростью v_1 из точки O в пренебрежении сопротивления воздуха можно описать следующим образом:

$$\begin{cases} x_c = v_1^x t; \\ y_c = v_1^y t - \dfrac{at^2}{2}, \end{cases}$$

где

$v_1^x = v_1 \cos(\alpha)$–горизонтальная составляющая начальной скорости снаряда,

$v_1^y = v_1 \sin(\alpha)$–вертикальная составляющая начальной скорости снаряда,

a - ускорение свободного падения.

Запишем закон движения мишени:

$$\begin{cases} x_m = L + v_2^x t; \\ y_m = H + v_2^y t - \dfrac{at^2}{2}. \end{cases}$$

где

$v_2^x = v_2 \cos(\beta)$–горизонтальная составляющая начальной скорости мишени,

$v_2^y = v_2 \sin(\beta)$–вертикальная составляющая начальной скорости мишени,

a - ускорение свободного падения.

Если тела встретились, то очевидно, что их координаты в этот момент времени совпадают. Приравняем x_c к x_m и y_c к y_m и найдем t:

$$\begin{cases} v_1^x t = L + v_2^x t; \\ v_1^y t - \dfrac{at^2}{2} = H + v_2^y t - \dfrac{at^2}{2}. \end{cases}$$

$$v_1 \cos(\alpha) t = L + v_2 \cos(\beta) t \Rightarrow t = \frac{L}{v_1 \cos(\alpha) - v_2 \cos(\beta)} \tag{32}$$

$$v_1 \sin(\alpha) t - \frac{at^2}{2} = H + v_2 \sin(\beta) - \frac{at^2}{2} \Rightarrow t = \frac{H}{v_1 \sin(\alpha) - v_2 \cos(\beta)} \tag{33}$$

$$\frac{L}{v_1 \cos(\alpha) - v_2^x} = \frac{H}{v_1 \sin(\alpha) - v_2^y} \Rightarrow (v_1 \sin(\alpha) - v_2^y) L = (v_1 \cos(\alpha) - v_2^x H) \tag{34}$$

$$v_1 L \sin(\alpha) - v_1 H \cos(\alpha) = L v_2^y - H v_2^x \tag{35}$$

$$\sqrt{v_1^2 (L^2 + H^2)} \sin(\alpha) - \operatorname{arctg}\left(\frac{H}{L}\right) = L v_2^y - H v_2^x \tag{36}$$

$$\alpha = \operatorname{arctg}\left(\frac{H}{L}\right) + \arcsin\left(\frac{L v_2^y - H v_2^x}{v_1 \sqrt{(L^2 + H^2)}}\right) \tag{37}$$

Решение этой задачи в СКМ Maple имеет вид:

```
> restart:
```

Нужные библиотеки.

```
>with(plots): with(plottools):
```

Процедура построения шарика

```
>kr:=proc(x,y,r,clr)
display(disk([x,y],r,color=clr))
end proc:
```

Процедура создания анимации движения в поле тяготения Земли из начальной точки (x0,y0) с начальной скоростью V1, направленной под углом alpha к горизонту

```
>projectile:=proc(V1,alpha,pred,clr,r) #(модуль начальной
скорости; угол; время полета; цвет линии; радиус круга)
local V1x, V1y, a, xc, yc, t, T; # локальные переменные
a:=10; # ускорение свободного падения
V1x:=V1*cos(alpha); # проекция нач. скорости на ось x
V1y:=V1*sin(alpha); # проекция нач. скорости на ось y
xc(t):=V1x*t; # закон движения вдоль оси x
yc(t):=V1y*t-a*t^2/2; # закон движения вдоль оси y
animate(plot,[[xc(t),yc(t),t=0..T],color=clr],T=0..pred,
scaling=constrained,thickness=2, frames=50);
animate(kr,[xc(t),yc(t),r,clr],t=0..pred,
scaling=constrained,frames=50);
display(%,%%);
end proc:
```

Процедура создания анимации движения в поле тяготения Земли из начальной точки (L,H) с начальной скоростью V2, направленной под углом beta к горизонту

```
>bomba:=proc(L,H,V2,beta,pred,clr,r)
```

(нач.x-координата; нач.y-координата; модуль начальной скорости; угол; время полета; цвет линии; радиус круга)

```
local V2x, V2y, a, xm, ym, t, T; # локальные переменные
a:=10; # ускорение свободного падения
V2x:=V2*cos(beta); #проекция нач. скорости на ось x
V2y:=V2*sin(beta); # проекция нач. скорости на ось y
xm(t):=L+V2x*t; # закон движения вдоль оси x
ym(t):=H+V2y*t-a*t^2/2; #закон движения вдоль оси y
animate(plot,[[xm(t),ym(t),t=0..T],color=clr],T=0..pred,
scaling=constrained,thickness=2, frames=50);
animate(kr,[xm(t),ym(t),r,clr],t=0..pred,
scaling=constrained,frames=50);
display(%,%%);
end proc:
```

Процедура, эллюстрирующая движение самолета, проходящего через точку (L,H) с начальной скоростью V2, направленной под углом beta к горизонту

```
>samolet:=proc(L,H,V2,beta,pred,clr,r)
```

(нач.x-координата; нач.y-координата; модуль начальной скорости; угол; время полета; цвет линии; радиус круга)

```
local V2x, V2y, a, xm, ym, t, T,x0,y0; # локальные переменные
x0:=L/2:
y0:=H-tan(beta)*(L-x0)/2:
V2x:=V2*cos(beta);
V2y:=V2*sin(beta);
plot(y0+tan(beta)*(x-x0)^2/(2*(L-x0)),x=0..100,color=clr,
scaling=constrained,thickness=2, frames=50,view=[0..100,0..40]);
end proc:
```

Процедура создания анимации полета самолета из точки (x0,y0) с начальной скоростью V1, направленной под углом alpha к горизонту и движение мишени из точки (L,H) со скоростью V2, направленной под углом beta к горизонту.

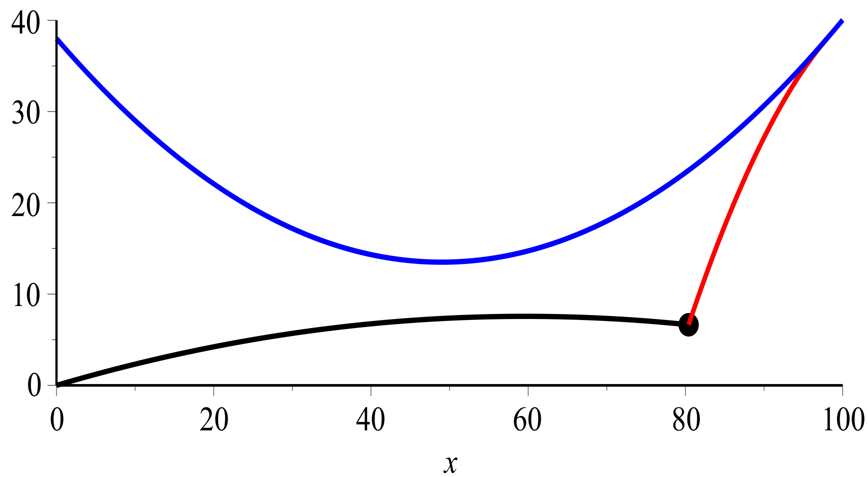
```
>blast:=proc(L,H,V1,V2,beta,clr1,clr2,clr3,r1,r2,r3)
local alpha,pred,direct,anim1,anim2,anim3,a,V2x,V2y;
a:=10;# ускорение свободного падения
V2x:=V2*cos(beta);
V2y:=V2*sin(beta);
alpha:=(arctan(H/L)+arcsin((-H*V2*cos(beta)+L*V2*sin(beta))/(V1*
sqrt(L^2+H^2))));
# вычисление угла наклона начальной скорости снаряда
pred:=L/(V1*cos(alpha)-V2*cos(beta)): # вычисление времени
столкновения тел
anim1:=projectile(V1,alpha,pred,clr1,r1);
anim2:=samolet(L,H,V2,beta,pred,clr2,r2);
```

```
anim3:=bomba(L,H,V2,beta,pred,clr3,r3);
plots[display](anim1,anim2,anim3);
end proc:
```

Задаем начальное положения мишени, модули начальной скорости мишени и снаряда, цвет снаряда, цвет самолета, цвет мишени, радиусы снаряда, мишени и самолета.

```
>blast(98,38,50,15,Pi+Pi/4,black,blue,red,0.5,1,1);
```

$$T = 1.6590$$

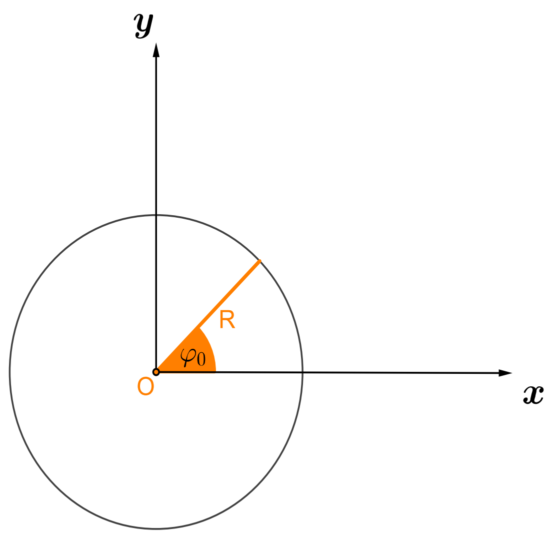


2.5 Моделирование движения гвоздя, вбитого в обод прямолинейно и равномерно движущегося колеса

Описать закон движения гвоздя, вбитого в обод колеса, если известно, что колесо катится по горизонтальной дороге со скоростью v.

Решение:

Закон движения гвоздя в системе координат начало которой совмещено с центром колеса, ось $\widetilde{OX}$ которой направлена горизонтально в направлении движения колеса, а ось $\widetilde{OY}$ вертикально вверх



Положение гвоздя в момент времени $t = 0$

есть закон равномерного вращения

$$\begin{cases} \tilde{x}(t) = R\cos\left(\frac{2\pi}{T}t + \varphi_0\right), \\ \tilde{y}(t) = R\sin\left(\frac{2\pi}{T}t + \varphi_0\right), \end{cases} \tag{38}$$

где R - радиус колеса, T - время за которое гвоздь совершает полный оборот, φ_0 - угол к оси $\widetilde{OX}$, под которым находится гвоздь в момент времени $t = 0$.

Движение центра колеса легко описать в системе координат в которой начало находится в нижней точке колеса в момент времени $t = 0$, ось OX которой направлена горизонтально в направлении движения колеса, а ось OY вертикально вверх. Это движение является прямолинейным и равномерным

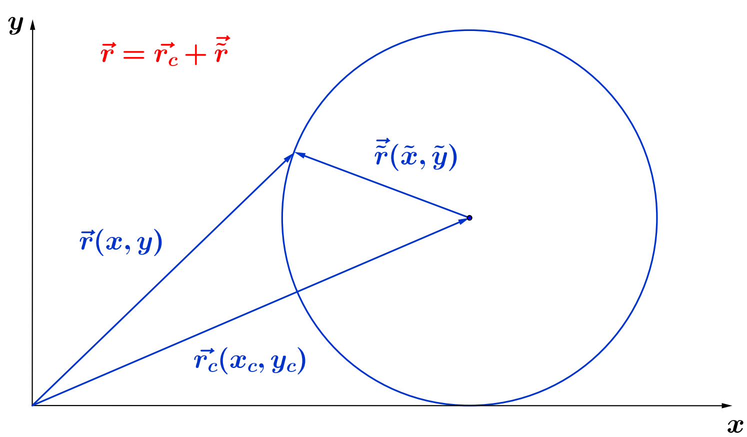
$$\begin{cases} x_c(t) = Vt; \\ y_c(t) = R, \end{cases} \tag{39}$$

где R - радиус колеса, V - линейная скорость центра колеса ($V = 2\pi/T$). Тогда закон движения центра колеса можно переписать в виде:

$$\begin{cases} x_c(t) = \frac{2\pi}{T}t; \\ y_c(t) = R. \end{cases} \tag{40}$$

Положение гвоздя в этой системе координат можно описать учитывая, что

$$\vec{r}(x,y) = \vec{r_c}(x_c, y_c) + \vec{\tilde{r}}(\tilde{x}, \tilde{y}) \tag{41}$$



В координатах равенство (41), с учетом (39) и (40), будет иметь вид

$$\begin{cases} x(t) = x_c(t) + \tilde{x}(t) = \frac{2\pi}{T}t + R\cos\left(\frac{2\pi}{T}t + \varphi_0\right), \\ y(t) = y_c(t) + \tilde{y}(t) = R + R\sin\left(\frac{2\pi}{T}t + \varphi_0\right). \end{cases} \tag{42}$$

Это дает возможность моделировать движение гвоздя.

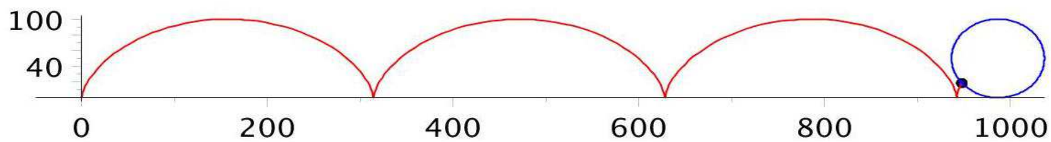
Если

$$\varphi_0 = -\pi/2, \tag{43}$$

что соответствует нахождению гвоздя в нижней точке в момент времени $t = 0$, то закон движения гвоздя можно переписать в виде

$$\begin{cases} x(t) = x_c(t) + \tilde{x}(t) = \frac{2\pi}{T}t + R\sin\left(\frac{2\pi}{T}t\right), \\ y(t) = y_c(t) + \tilde{y}(t) = R - R\cos\left(\frac{2\pi}{T}t\right). \end{cases} \tag{44}$$

Эту линию при $R = 50, T = 5$ можно изобразить так



Решение этой задачи в СКМ Maple имеет вид:

```
> restart:
Нужные библиотеки.
>with(plots): with(plottools):
Процедура построения гвоздя
>kr:=proc(x,y,r,clr1)
display(disk([x,y],r,color=clr1))
end proc:
>F := proc(t,R,T)
plots[display](circle([2*Pi*R*t/T, R], R, color = blue));
end proc:
Процедура создания анимации движения гвоздя вбитого в обот
колеса
>F1:=proc(R,T,phi,clr1,clr2,r)
local x,y,t; # локальные переменные
```

```
x(t):=R*cos(-2*Pi*t/T+phi)+2*Pi*R*t/T; # закон движения вдоль
оси x
y(t):=R*sin(-2*Pi*t/T+phi)+R; # закон движения вдоль оси y
animate(plot,[[x(t),y(t),t=0..A],color=clr2],A=0..5*Pi,
scaling=constrained,thickness=2, frames=400):
animate(kr,[x(t),y(t),r,clr1],t=0..5*Pi,scaling=constrained,
frames=400):
animate(F,[t,R,T],t=0..5*Pi,scaling=constrained,frames=400):
display(%,%%,%%%);
end proc:
>F1(50,5,-Pi/2,blue,red,5);
```

3 Заключение

В работе

- создан элективный курс введения в компьютерное моделирование,
- написан ряд программ в системе компьютерной математики Maple, моделирующих столкновение двух тел, двигающихся либо равномерно и прямолинейно, либо равноускоренно на плоскости.

Список литературы

[1] Р.Р. Шарафутдинова. Практикум по введению в компьютерное моделирование. //Международная научно-практическая конференция ИТОН-2014. IV-й международный семинар и международная школа "Математическое и компьютерное моделирование фундаментальных объектов и явлений в системах компьютерной математики". Материалы конференции и труды семинара. Под общей редакцией Ю.Г. Игнатьева, Казань: Изд-во "Фолиант 2014. - с. 164-166.

[2] В.П. Дьяконов. Maple 9.5/10 в математике, физике и образовании. М:Солон-Пресс , 2006. - 720 с.

[3] Ю.Г. Игнатьев. Проблемы информационных технологий в математическом образовании. Казань: Изд-во ТГППУ, 2005. - 118 с.

[4] Тихоненко А.В. Компьютерный практикум по общей физике. Часть 1. Классическая механика. Учебное пособие по курсу «Общая физика». Обнинск: ИАТЭ, 2003. – 84 с.