Гетманова Елена Евгеньевна

ИСПОЛЬЗОВАНИЕ VISUAL PYTHON ДЛЯ МОДЕЛИРОВАНИЯ ФИЗИЧЕСКИХ ПРОЦЕССОВ

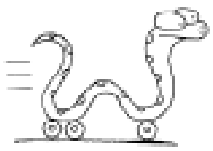
От редакции. В прошлом номере журнала был описан язык Visual Python, широко используемый международным сообществом для моделирования физических явлений. Данная статья продолжает тему. В ней описываются несколько лабораторных работ по физике. Часть работ соответствует школьной программе по физике, другая часть – вузовской программе.

Использование языка Visual Python для моделирования физических явлений целесообразно начинать с повторения простых программ приведенных в разделе «Описание языка Visual Python» статьи «Visual Python – язык для моделирования физических явлений» и создать простые геометрические объекты. Затем усложнить программу, задать объектам скорость, ускорение и промоделировать движение тела в определенном направлении. Далее подключить модуль, создающий графики изменения пути или скорости от времени. При этом можно изменять цвет и масштаб графиков. Затем включать обработчики событий, которые состоят, например, в совпадении координат двух объектов (столкновении объектов). В качестве обработчика может выступать, например, изменение прозрачности одного из объектов, Поскольку синтаксис языка достаточно очевиден, то при таком поэтапном усложнении кода, программирование на VPython осваивается достаточно быстро.

Векторы могут трактоваться как векторные физические величины (сила, скорость, ускорение и т. д.). Изменяя компоненты векторной величины, можно увидеть их влияние на характер протекания физического процесса. Это позволяет четко представить, например, как изменится движение тела при изменении компонент начальной скорости. Или, к примеру, как изменится движение тела при изменении компонент силы, действующей на тело. Такое ясное понимание, основанное на визуальном восприятии 3D-графики, практически невозможно осуществить при натурных экспериментах.

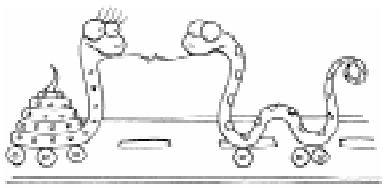
Работы включают в себя моделирование физических процессов в механике.

При изучении поступательного движения набирается код, показанный в листинге 1.



При запуске программы создается поверхность, по которой двигаются два объекта (рисунок 1).

Код программы подключает графический модуль, модуль для создания графиков, создает объекты и задает законы движения. Одновременно с движением объектов, строятся графики зависимости пройденного пути от времени (рисунок 1). Когда объекты встречаются, графики пересекаются и определяется время и место встречи. Данная



лабораторная работа помогает понять физический смысл отрицательных ускорений и скоростей. Работа включает элементы компьютерных игр. Например, можно поставить задачу о нахождении скорости, при кото-

Листинг 1.

```
from visual import *
from visual.graph import *
win=500
scene=display(title='TRANSLATION MOTION', width=win, height=win, x=0,y=0)
floor=box(length=50, height=0.1, width=10, color=color.red)
body=box(length=1,height=1, width=1, color=color.blue)
body.pos=vector(0,0.5,0)
body1=box(length=1, height=1, color=color.green)
body1.pos=vector(0,0.5,-1)
body.velocity=vector(-1,0,0)
body1.velocity=vector(1,0,0)
ppos=gdisplay(x=0,xmax=1,y=win,ymax=5,width=win, height=win/2,xtitle='time',
              ytitle='coordinate x')
path=gcurve(color=color.cyan)
path1=gcurve(color=color.red)
dt=0.01
for t in arange(0,1.01,0.01):
    path1.plot(pos=(t,-1*t+5*t*t/2))
    path.plot(pos=(t,1*t+3*t*t/2))
while 1:
    rate(50)
    body1.pos.x=body1.pos.x+body1.velocity.x*dt
    body.pos.x=body.pos.x+body.velocity.x*dt
    body1.velocity.x=body1.velocity.x+7*dt
    body.velocity.x=body.velocity.x+3*dt
```

рой один объект догонит второй. Во многих задачах кинематики требуется найти время и место встречи двух движущихся объектов. С помощью данной лабораторной работы может быть проверено умение решать задачи кинематики. Если задача решена правильно, то время и место встречи, показанное на графике, будет совпадать с полученным при решении.

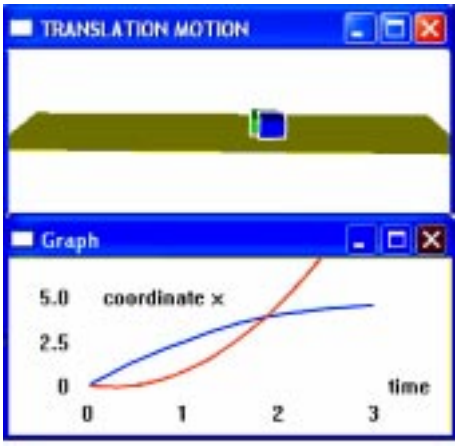
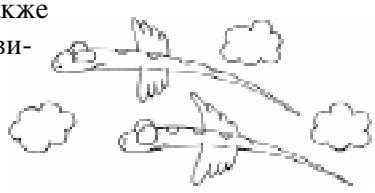


Рисунок 1.

В следующей работе изучается движение тела, брошенного под углом к горизонту. Код программы показан в листинге 2.

Одновременно с первым телом на некоторой высоте параллельно поверхности двигается второе тело. Когда координаты тел совпадают, второй объект исчезает (сбивается первым) (рисунок 2).



Строятся также траектории движущихся тел (рисунок 2).

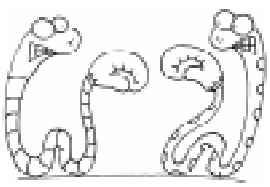
Данная лабораторная работа также может быть использована не только как демонстрация особенностей движения двух тел при различных начальных скоростях, высотах, на которой движется тело, и углах, под которыми бросается тело. Работа подойдет и для проверки умений решать задачи подобного типа. Вначале ученики решают задачу и находят, например,

Листинг 2.

```
from visual import *
from visual.graph import *
win=500
scene = display(title='Projectile Motion', width=win, height=win, x=0,y=0,
                background=(1,1,1))

floor = box(length=10, height=0.5, width=4, color=color.cyan)
ball1 = sphere(pos=(0,0,0), color=color.green)
ball1.velocity = vector(10,15,0)
ball = sphere(pos=(0,7,0), color=color.red)
ball.velocity = vector(10,0,0)

ppos=gdisplay(x=0,xmax=30, y=win, ymax=20, width=win, height=win/2.,
         xtitle='coordinate x', ytitle='coordinate y', background=(0.7,0.7,0.7))
path=gcurve(color=color.blue)
path1=gcurve(color=color.red)

dt = 0.01
for x in arange(0., 30.01, 0.01):
   path1.plot(pos=(x,1.5*x-9.8*x*x/(2*10*10)))
   path.plot(pos=(x,7))
while 1:
    rate(30)
    ball1.pos = ball1.pos + ball1.velocity*dt

    ball1.velocity.y=ball1.velocity.y - 9.8*dt
    ball.pos=ball.pos + ball.velocity*dt

if ball1.pos <= ball.pos:
        ball.visible = 1

 else:
        ball.visible = 0
```

координаты и время встречи двух тел. Далее программа запускается на выполнение и решение проверяется. А поскольку можно построить и график изменения расстояния между двумя телами в зависимости от времени, и изменение скорости от времени для каждого тела, то проверить можно достаточно большой спектр знаний и умений.

Одна из лабораторных работ посвящена изучению второго закона Ньютона и рассмотрению движения центра масс системы двух тел. В программе описаны два тела, движение которых определяется действующей на тела силой. Код



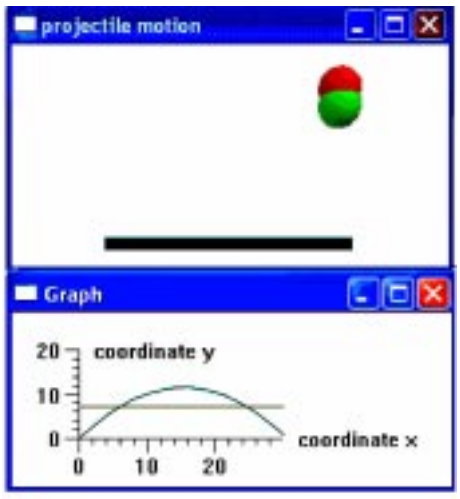
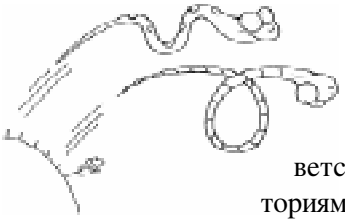


Рисунок 2.

программы показан в листинге 3 (этот и последующие листинги находятся на диске к журналу).



Под действием приложенной силы оба тела двигаются по соответствующим траекториям (рисунок 3). График показывает изменение расстояния между объектами в зависимости от времени (рисунок 3). В качестве самостоятельной работы предлагается изменить компоненты силы. На основе данной лабораторной работы можно проверить знания студентов по кинематике и динамике. Можно предложить студентам решить данную задачу аналитически и найти расстояние между объектами в определенный момент времени. Затем запустить программу на выполнение и проверить полученный результат.

При изучении моментов инерции моделируются вращения тел различной формы. Код данной программы показан в листинге 4). На экране показывается направление угловой и линейной скоростей (рисунок 4).

Известно, что в общем случае момент инерции тела зависит от оси вращения. Данная работа направлена на понимание свойств момента инерции. Рассматривается вращение четырех тел: диска, сферы, куба и тора. Строятся графики изменения моментов инерции относительно оси, проходящей через центр масс тел, в зависимости от массы тел.

В качестве самостоятельной работы предлагается изменить ось вращения. Изменение оси вращения тел в общем случае приводит к изменению момента инерции. Данную работу также можно использовать в качестве обучения и проверки умений решать задачи на нахождение момента инерции. Можно предложить учащимся найти моменты инерции, к примеру, тора относительно двух взаимно перпендикулярных осей, проходящих через центр масс. Далее построить график зависимости полученных моментов инерции от массы тела, объяснить ход изменения зависимостей. Написать соответствующий код программы, запустить программу на выполнение и сравнить графики, полученные аналитически и с помощью графического модуля.

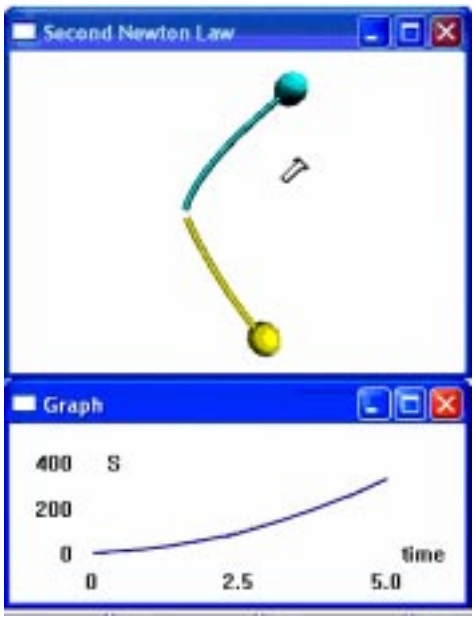


Рисунок 3.

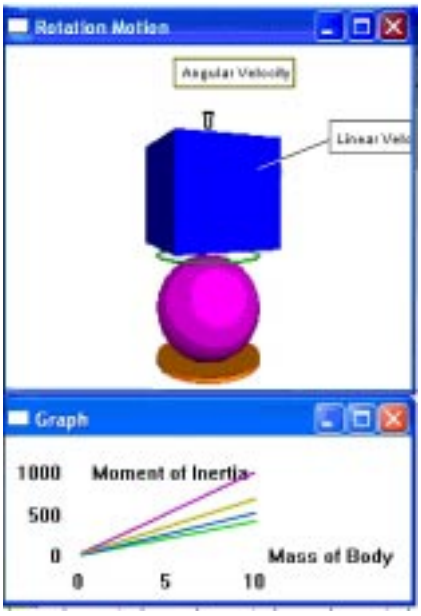
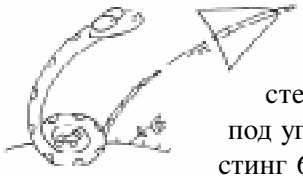


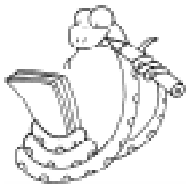
Рисунок 4.

В одной из работ изменяются скорости вращательного и поступательного движения тела при помощи ползунков и переключателей. В качестве самостоятельной работы предлагается рассмотреть различные тела – цилиндр, сферу, тор, которые совершают плоское движение. Строятся графики зависимости кинетической энергии тел от угловой скорости вращения при плоском движении. Данная лабораторная работа также позволяет проверить знания моментов инерции, кинетической энергии при плоском движении, кинетической энергии твердого тела при вращательном движении. Код работы показан в листинге 5.



В лабораторных работах исследуется движение стержня, брошенного под углом к горизонту (листинг 6).

Подобного рода задачи часто возникают при создании игр. В общем случае, как известно, такое движение представляет собой поступательное и вращательное движения. В зависимости от точки приложения силы, характер движения тела меняется (рисунок 5).



В конце каждой работы приведены контрольные вопросы и контрольные задания. Контрольные задания состоят в изменении условий моделирования данного физического процесса. Таким образом, лабораторные работы представляют собой сочетание традиционных лабораторных и практических занятий. Можно предположить, что подобный подход к изучению физики является динамичным и эффективным. Желание понять и активно использовать данную программу продиктовано еще и тем, что она используется при написании компьютерных игр.

Лабораторные работы позволяют изучать как физику, так и основы программирования.

Следует также отметить, что такой подход в образовании представляет собой обучение через действие, воспринимается учениками и студентами как имеющий непосредственное отношение к их личным целям и вызывает большой интерес.

Литература

1. *Сузи Р.А.* Python. СПб: БВХ-Петербург, 2002. 768 с.

2. *Гетманова Е.Е.* Моделирование физических процессов в VPython.-Финарт, Харьков, 2004. 69 с.

Гетманова Елена Евгеньевна
кандидат физико-математических наук, доцент кафедры физики Харьковского национального университета радиоэлектроники.