Применение математических знаний при написании программ

Г. В. Гаркавенко, email: g.garkavenko@mail.ru[1]

Воронежский государственный педагогический университет

Аннотация. *В данной статье рассматривается то, как можно исследовать работу математических формул с использованием программирования. И демонстрируется, что получаемые в школе и вузе математические знания можно и нужно применять при решении различных задач, в частности в программировании.*

Ключевые слова: *исследовательская работа, программирование, математика, движение по окружности.*

Введение

Очень часто на вопрос «Для чего надо изучать математику?» школьники отвечают «Для поступления в вуз» или в лучшем случае «Для подсчета стоимости покупки». Ненамного лучше дело обстоит и со студентами, даже математических специальностей. Они знают, что «вот это» нужно для вычисления «вот того-то», и даже знают, как «это» решить. Но, как и где это можно применить на практике, как для решения задачи использовать комбинацию различных методов [1], вот здесь возникают затруднения.

Сейчас востребованы профессии, связанные с программированием. Студенты, изучая язык программирования пытаются сразу написать программу из команд, порой не задумываясь, что главным является алгоритм [2], а не просто синтаксически правильно написанные команды, которые выдают порой неправильный ответ, при том, что программа не содержит синтаксических ошибок. При создании алгоритма решения задачи требуется не только знание языка и приемов программирования, но и знание математических формул и законов. Тем, кто посвятил себя программированию необходимо знать математику, конечно уровень этих знаний может варьироваться в зависимости от специализации. Например, многие не понимают зачем изучать делимость чисел и сравнимость чисел по модулю. А ведь все эти вещи используются в шифровании, при защите информации. Одним из популярных направлений сейчас для освоения является робототехника. В связи с созданием роботов-манипуляторов рассматриваются прямая и обратная задачи кинематики [3], в зависимости от вида сочленений

роботов относят к одной из категорий, а соответственно они отличаются тем, что работают в различных системах координат. Для понимания того, по какой траектории движется манипулятор нужны знания, полученные при изучении линейной алгебры и аналитической геометрии, знание того как описываются различные траектории движения и преобразуются координаты. На занятиях по высшей математике изучение таких формул осуществляется с помощью бумаги и ручки, а хотелось бы, чтобы студенты провели исследование работы таких формул на каком-либо более практическом примере. Организация исследовательской деятельность обучающихся, одновременно и сложная и полезная задача [4]. Итак, одним из движений, которое осуществляет робот-манипулятор, является движение по дуге. В соответствии с этим, рассмотрим небольшой пример задания движения по окружности.

1. Пример применение школьных знаний по математике

Пусть необходимо написать программу, создающую простую графическую анимацию: вращение луны вокруг Земли.

Так как в тексте статьи невозможно продемонстрировать само движение для наглядной демонстрации изобразим на плоскости в декартовой системе координат окружность, которая будет изображать траекторию движения (орбиту луны) и небольшие круги (модель луны), расположенные вдоль этой траектории, вернее на круговой траектории будут находится центры малых кругов.

Для написания программы будем использовать язык программирования Python и библиотеку Tkinter.

Простое, прямолинейное движение какого-либо объекта создать несложно, чаще всего его создают, давая приращения координатам объекта. Также можно использовать уравнение прямой $y = ax + b$, задавая значения x и вычисляя соответствующие значения y.

А как изменять координаты, чтобы объект перемещался по окружности. Вспомним, что в школьном курсе математики изучается уравнение окружности $x^2 + y^2 = R^2$, и можно попробовать выразить y через x, $y = \pm\sqrt{R^2 - x^2}$, а затем, задавая значения переменной x, вычислить вторую координату. Но надо помнить, что под корнем не должно быть отрицательного числа, и корень взятый с плюсом или с минусом описывает только половину окружности, в верхней и нижней полуплоскостях. В нашей задаче R – это радиус окружности, задающей траекторию движения «луны», а координаты точек (x, y) задают положение центра маленького круга, изображающего «луну».

Итак, в Python, с использованием библиотеки Tkinter, зададим графическое окно с холстом для рисования и изобразим по центру координатные оси. При создании графических объектов отсчет их координат идет от левого верхнего угла окна. Будем организовывать отсчет координат от точки пересечения прямых, изображающих координатные оси, и обозначим координаты центра (x0, y0).

Листинг 1

Создание графического окна

```
from tkinter import *

size=400
my_win=Tk()
my_win.title("ПО окружности")
ris=Canvas(my_win, width=size, height=size, bg='white')
ris.pack()
m=10
color="#0bb"
## Начало координат
x0=size/2
y0=size/2
## Рисование оси OY
ris.create_line((x0,0),(x0,size))
## Рисование оси OX
ris.create_line(0,y0,size,y0)
```

Далее зададим радиус окружности, задающей траекторию движения R и радиус «луны» r. В цикле, задающем изменения *x* на отрезке [-R, R], будем вычислять соответствующий *y*, и рисовать маленькие окружности с центрами в точках (*x, y)* и (*x, -y).* Причем начинать будем в точке с координатами (R, 0) и движение осуществлять в верхней полуплоскости против часовой стрелки, как это принято в математике.

Листинг 2

Изображение движения по окружности

```
r=5
R=size/2-50
ris.create_oval(x0-R,y0-R,x0+R,y0+R)
x=R
y=0
x1=x0+x
y1=y0+y
ris.create_oval(x1-r,y1-r,x1+r,y1+r,fill=color)
while x > -R:
    x=x-10
    y=int((R*R-x*x)**(0.5))
    x1=x0+x
    y1=y0-y
```

```
    y2=y0+y
    ris.create_oval(x1-r,y1-r,x1+r,y1+r,fill=color)
    ris.create_oval(x1-r,y2-r,x1+r,y2+r,fill=color)
my_win.mainloop()
```

Результат работы программы представлен на рисунке 1 ниже.

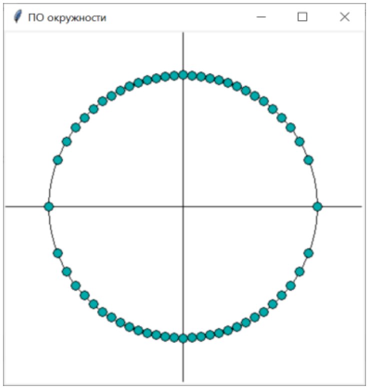

Рис. 1. Изображение движения по окружности на основе школьных знаний

Конечно, для анимации движения маленькой окружности по траектории, которую задает большая окружность, надо написать два цикл первый при изменении x от R до –R задавать движение по верхней половине окружности, а второй при изменении x от -R до R задавать движение по нижней половине окружности. Как бы мы не подбирали шаг для изменения x, скачков около оси OX, как это видно из рисунка 1 избежать не удастся.

А как задать движение по траектории, представляющей эллипс? А если оси эллипса не совпадают с осями координат?

В школьном курсе математики другого представления для окружности не изучается, хотя если хорошо подумать, можно вспомнить о тригонометрической окружности и заподозрить, что какая-то связь между тригонометрической окружностью и уравнением окружности есть.

2. Использование знания высшей математики для создания траектории движения по окружности

Студенты, в курсе математического анализа или высшей математики изучают параметрическое задание кривых, в частности окружности. И порой не знают зачем это нужно и как это можно применить. А применить можно, например, для создания движения по нужной траектории.

Параметрическое задание эллипса имеет вид:

$$x = R_1 \cos t$$

$$y = R_2 \sin t$$

Здесь t параметр, задающий угол на который переместится точка от оси OX вдоль окружности. R_1 - радиус эллипса по оси OX, R_2 - радиус эллипса по ось OY. Если $R_1 = R_2 = R$, то мы получим параметрическое задание окружности.

Тогда цикл рисования маленьких окружностей вдоль траектории, задаваемой большой окружностью примет вид:

Листинг 3

Программа рисующая движение по окружности с использованием параметрического задания

```
x1=x0+R
y1=y0
t=0
while t < 2*pi:
    t=t+0.05
    x=R*cos(t)
    y=R*sin(t)
    x1=x0+int(x)
    y1=y0-int(y)
    ris.create_oval(x1-r,y1-r,x1+r,y1+r,fill=color)
```

Результат работы такой программы представлен на рисунке 2. Параметр *t* пробегает полный круг от 0 радиан до 2π. Правда для работы с тригонометрическими функциями в Python пришлось подключить библиотеку math.

И последнее, необходимо задать движение по эллипсу, оси которого не совпадают с осями исходной системы координат, то есть не являются горизонтальной и вертикальной линиями. Пусть эллипс повернут относительно оси OX на угол α (его будем задавать в радианах).

В этом случае также поможет знание математики, в частности, формулы, позволяющие осуществлять переход от одной системы координат к другой системе координат. В нашем случае центр новой системы координат будет совпадать с центром исходной системы координат, точкой (x0, y0). В общем случае это может быть не так. А оси новой системы координат будут совпадать с осями нашего эллипса, задающего траекторию движения, то есть повернут относительно исходных осей на угол α. Вспомним из аналитической геометрии формулы позволяющие вычислить координаты точки относительно старой (исходной) системы координат, если известны координаты точки в новой системе координат:

$$x = x'\cos\alpha - y'\sin\alpha;$$

$$y = x'\sin\alpha + y'\cos\alpha.$$

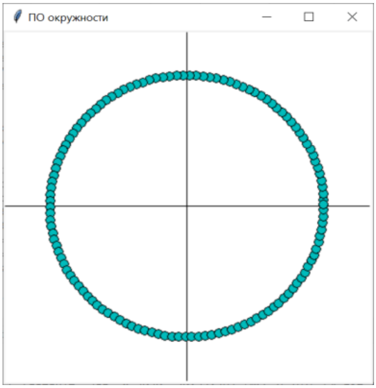

Рис. 2. Вид расположения маленьких кружков по круговой траектории при использовании параметрического задания окружности

Напишем программу, которая для сравнения рисует движение маленького круга по эллиптической траектории в исходной системе координат и в системе координат с поворотом осей на 60^0.

Листинг 4

Программа рисования движения по различным эллипсам

```
from tkinter import *
from math import *
size=400
my_win=Tk()
my_win.title("ПО окружности")
ris=Canvas(my_win, width=size, height=size, bg='white')
ris.pack()
color="#eed"
ris.create_line((size/2,0),(size/2,size))
ris.create_line(0,size/2,size,size/2)
r=5
x0=size/2
y0=size/2
R=size/2-50
ris.create_oval(x0-R,y0-R,x0+R,y0+R)
R1=R
R2=R-100
x1=x0+R
```

```
y1=y0
t=0
while t < 2*pi:
    t=t+0.05
    x=R1*cos(t)
    y=R2*sin(t)
    x1=x0+int(x)
    y1=y0-int(y)
    ris.create_oval(x1-r,y1-r,x1+r,y1+r,fill=color)
x1=x0+R
y1=y0
alpha=pi/3
t=0
while t < 2*pi:
    t=t+0.05
    x=R1*cos(t)
    y=R2*sin(t)
    x1=x*cos(alpha)-y*sin(alpha)
    y1=x*sin(alpha)+y*cos(alpha)
    x1=int(x1)+x0
    y1=-int(y1)+y0
    ris.create_oval(x1-r,y1-r,x1+r,y1+r,fill="#ccd")
my_win.mainloop()
```

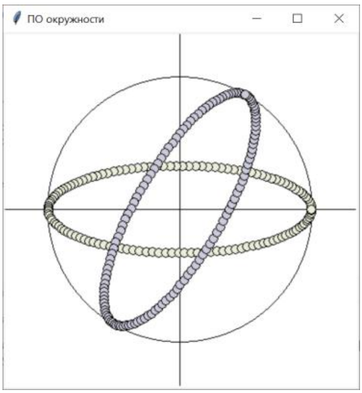

Рис. 3. Результат работы программы из листинга 4

Заключение

Наука математика, состоит из различных разделов, каждый из которых изучается отдельно. Но для решения какой-либо практической задачи зачастую требуются знания из различных областей математики, умение использовать комбинацию различных методов [1]. А вот демонстрация того, как использовать математические знания на практике, нетривиальная задача. Часто для решения математических задач используется программирование, и наоборот, при решении проблем, возникающих в программировании [5], используются математические методы. Сейчас в сети Интернет можно найти множество средств, позволяющих произвести математические вычисления [6], но их использование возможно только тогда, когда имеется математическая модель и известен метод решения поставленной задачи. А вот научиться строить математические модели и находить методы их исследования сложно.

Список литературы

1. Гаркавенко, Г. В. Изучение численных методов на одном модельном примере / Г. В. Гаркавенко, В. Н. Турбин // Информационные технологии в образовательном процессе вуза и школы. Материалы XIV Всероссийской научно-практической конференции. Редколлегия: В.В. Малев (науч. ред.), А.А. Малева (отв. ред.), М.В. Дюжакова, С.О. Башарина. 2020. – С. 73-77.
2. Гаркавенко, Г. В. О проблемах преподавания языков программирования в педагогическом вузе / Г. В. Гаркавенко, Е.А. Кубряков// Информатика: проблемы, методология, технологии. Информатика в образовании материалы XVIII Международной школы-конференции. – Воронеж, 2018. – С. 17-22.
3. Колтыгин, Д. С. Метод и программа решения прямой и обратной задачи кинематики для управления роботом-манипулятором / Д. С. Колтыгин, И. А. Седельников // Системы. Методы. Технологии. – 2020 – № 4 (48) – С. 65-74
4. Гаркавенко, Г. В. Пример исследовательской деятельности на уроках информатики / Г. В. Гаркавенко, В. В. Морозова // В сборнике: Актуальные проблемы методики обучения информатике и математике в современной школе. Материалы международной научно-практической интернет-конференции. Москва, 2021. С. 92-98.
5. Башарина, С. О. Исследование особенностей работы с вещественными числами / С. О. Башарина, Г. В. Гаркавенко, Е. Р. Найденкина // Информатика: проблемы, методы, технологии. Материалы XX Международной научно-методической конференции. Под редакцией А.А. Зацаринного, Д.Н. Борисова. (Воронеж, 13-14 февраля 2020 г.) – Воронеж, 2020. – С. 1791-1797.
6. Гаркавенко, Г. В. Использование информационных технологий при изучении численного интегрирования / Г. В. Гаркавенко, В. В. Морозова // Информационные технологии в образовательном процессе вуза и школы. Материалы XV Всероссийской научно-практической конференции. Редколлегия: Р.М. Чудинский (науч. ред.) [и др.]. Воронеж, 2021. С. 93-98.