ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»

НАУЧНО-ОБРАЗОВАТЕЛЬНОЕ СОРЕВНОВАНИЕ «ШАГ В БУДУЩЕЕ, МОСКВА»

Информатика, искусственный интеллект и системы управления

Программное обеспечение ЭВМ и информационные технологии

Моделирование тела брошенного под углом к горизонту

11 класс, ГБОУ Школа №1532,
Шахнович Дмитрий Сергеевич
Регистрационный номер: 311
Руководитель: учитель информатики, ГБОУ Школа №1532,
Сергиенко Антон Борисович

_____

Подпись научного руководителя

Москва, 2022

Аннотация

Предмет исследования: математическая модель тела, брошенного под углом к горизонту

Цель: Написание программного комплекса для решения и визуализации задачи внешней баллистики для шарообразного тела

Методы, использованные в работе: анализ литературы, обобщение, формализация, моделирование

Результаты: подготовлены теоретические материалы по задаче, рабочая программа для решения и моделирования процесса

Область применения: упрощение и автоматизация расчетов, обучение

Предложения: оптимизация вычислений дифференциальных уравнений, увеличение баз данных физических характеристик, решение для тел других форм.

Использованные технические средства: Python, QT

Содержание

Введение 4
Цели и задачи работы 4
Актуальность работы 4
Основная часть 5
Физическое решение задачи 5
Решение задачи без учёта силы лобового сопротивления 5
Решение задачи с учетом сопротивления воздуха 7
Программное реализация решения 9
Визуальная составляющая программы 11
Использование разработки 14
Результат 14
Список используемой литературы 15

Введение

Механика – область физики, изучающая движение материальных тел и их взаимодействие между собой. Механика включает в себя много областей, одной из которых является баллистика, описывающая движения снарядов с помощью методов математики.

Задача внешней баллистики или задача тела, брошенного под углом к горизонту, состоит в расчете траектории движения этого тела после силового взаимодействия.

Цели и задачи работы

Целью работы является написание программы с графическим интерфейсом для решения и визуализации задачи тела, брошенного под углом к горизонту, для шарообразных тел.

Для выполнения поставленной цели необходимо решение следующих задач:

1. Обобщение физического решения задачи.
2. Реализация решения задачи на языке Python по заданным данным.
3. Создание графического интерфейса программы.
4. Объединение интерфейса и решения задачи в единый программный комплекс.

Актуальность работы

Актуальность работы связана с отсутствием аналогов для вычисления и построения графиков тела, брошенного под углом к горизонту. Аналогов данной программе найдено не было. Найденные программы являются либо калькуляторами контрольных значений(макс. высота, время полёта и другие), либо строят график без учёта сопротивления воздуха

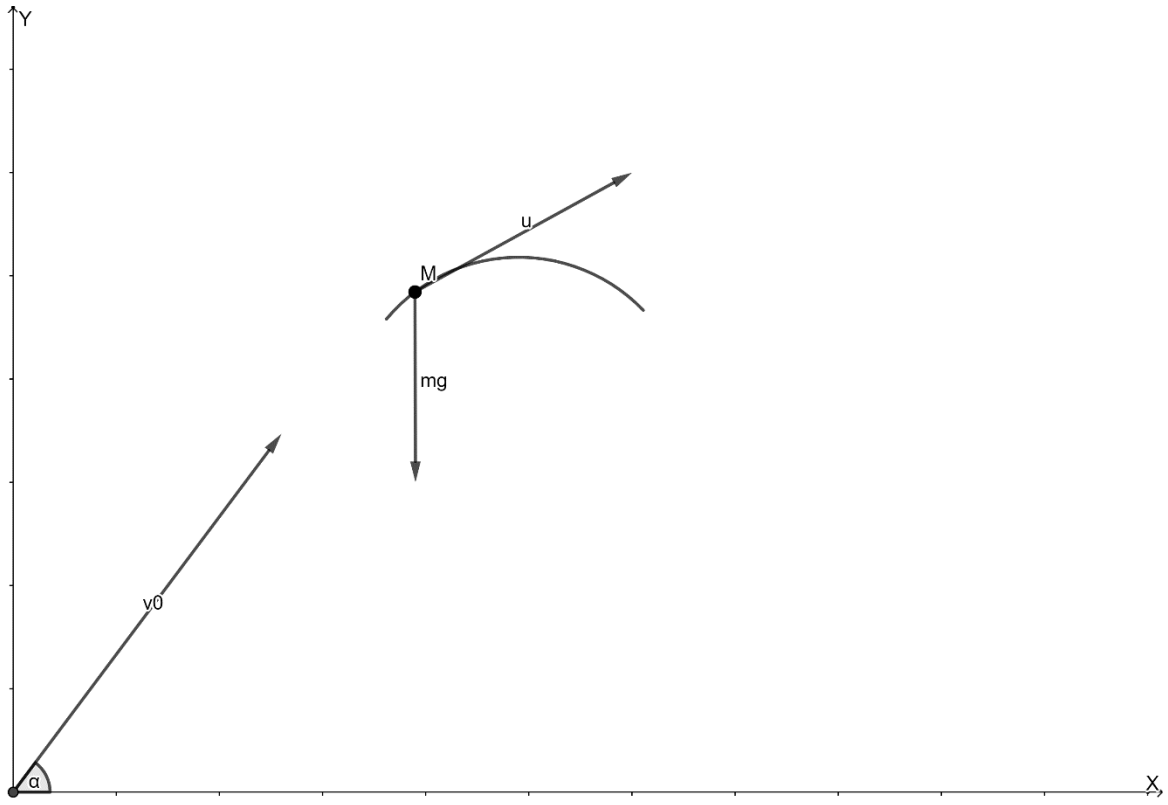
Основная часть

Физическое решение задачи

Так как задача внешней баллистики рассматривает движение тела после силового взаимодействия, то тело будет иметь какую-то начальную скорость. При этом вектор этой скорости будет образовывать с линией горизонта угол больше 0 градусов, но меньше 90. В общем случае на тело будут действовать две силы: сила лобового сопротивления и сила тяжести.

Решение задачи без учёта силы лобового сопротивления

Если пренебречь силой сопротивления, то остается только одна сила, действующая на тело: сила тяжести. В таком случае тело имеет постоянное ускорение равное ускорению свободного падения, а траектория тела представляет собой параболическую кривую. Также движение тела можно представить как наложение независимых движений вдоль разных осей, например оси X, параллельной плоскости поверхности земли, и оси Y, перпендикулярной плоскости поверхности земли.



Тогда, относительно оси X движение будет равномерным, а движение относительно оси Y — равноускоренным. Проекции ускорения тела для данных осей будут выглядеть так:

$$a_x = 0$$

$$a_y = -g$$

где g – ускорение свободного падения, м/с^2;

a_x – ускорение по оси X, м/с^2;

a_y – ускорение по оси Y, м/с^2.

Так как по оси X тело движется равномерно, по Y- равноускорено, то для него справедливы следующие равенства:

$$v_x(t) = v_{x0} = v_0 \cos\alpha$$

$$v_y(t) = v_{y0} - gt = v_0 \sin\alpha - gt$$

$$x(t) = x_0 + v_{x0}t = x_0 + v_0 \cos\alpha\, t$$

$$y(t) = y_0 + v_{y0}t - \frac{gt^2}{2} = y_0 + v_0 \sin\alpha\, t - \frac{gt^2}{2}$$

где $v_x(t)$, $v_y(t)$ – скорость по осям X и Y в момент времени t, м/с;

v_{x0}, v_{y0} – начальная скорость тела по осям X и Y, м/с;

v_0 – начальная скорость тела, м/с;

$\boldsymbol{x(t), y(t)}$ **–** координаты тела в момент времени t по осям X и Y, м;

x_0, y_0 – начальные координаты тела по осям X и Y, м.

Решение задачи с учетом сопротивления воздуха

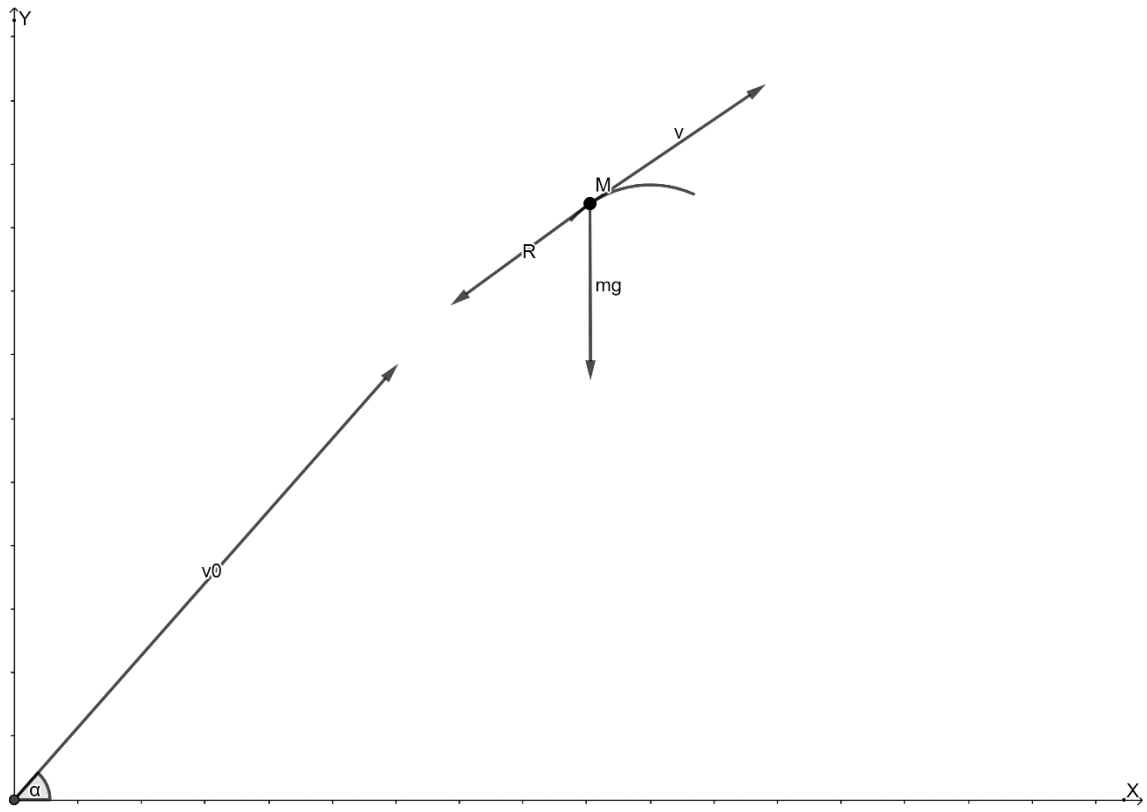
Сила сопротивления направлена против скорости движения, её величина пропорциональна характерной площади S, плотности среды ρ и квадрату скорости $\vec{v}^2$:

$$\vec{R} = -C_f \frac{\rho \vec{v}^2}{2} S$$

где C_f – безразмерный аэродинамический коэффициент сопротивления, получается из критериев подобия, например чисел Рейнольдса и Фруда в аэродинамике.

Определение характерной площади S зависит от формы тела:

- в простейшем случае (шар) — площадь поперечного сечения;
- для крыльев и оперения — площадь крыла/оперения в плане;
- для пропеллеров и несущих винтов вертолётов — либо площадь лопастей, либо ометаемая площадь винта;
- для подводных объектов обтекаемой формы — площадь смачиваемой поверхности;



В отличие от случая без сопротивления воздуха, тело будет иметь различные ускорения в разные промежутки времени.

Из второго закона Ньютона можно получить следующее уравнение:

$$m\vec{a} = m\vec{g} - C_f \frac{\rho \vec{v}^2}{2} S$$

$$\vec{a} = \vec{g} - C_f \frac{\rho v^2}{2m} S$$

Соответственно, проекции на оси будут выглядеть следующим образом:

$$a_x = -C_f \frac{\rho v_x^2}{2} S$$

$$a_y = -g - C_f \frac{\rho v_y^2}{2} S$$

По определению $a_x = \ddot{x}, a_y = \ddot{y}$:

$$\ddot{x} = -kv_x^2$$

$$\ddot{y} = -g - kv_y^2$$

Где k – коэффициент, определяющий вклад силы сопротивления в ускорение снаряда:

$$k = \frac{C_f S \rho}{2m}$$

Если взять, что бросаемое тело имеет шарообразную форму, то коэффициент можно упростить:

$$m = \gamma V = \gamma \frac{4}{3}\pi r^3, S = \pi r^2$$

$$k = \frac{C_f \pi \rho r^2}{\frac{8}{3}\gamma r^3 m} = \frac{3}{8}\frac{C_f \rho}{r\gamma}$$

Где γ - плотность материала тела, r - радиус шарообразного тела.

Таким образом получается система дифференциальных уравнений второго порядка в форме Коши:

$$\dot{x} = v_x$$

$$\dot{y} = v_y$$

$$\ddot{x} = -kv_x^2$$

$$\ddot{y} = -g - kv_y^2$$

Решив которую для начальных данных, получим траекторию полёта снаряда.

Программное реализация решения

Программу можно найти в репозитории https://github.com/Impervguin/Projectile-Simulation

В качестве основного языка программирования был выбран python. Прежде всего выбор обусловлен удобным синтаксисом языка, а также наличием большого числа библиотек, которые требовались для решения задачи.

Для решения дифференциальных уравнений была выбрана библиотека scipy.

Для отрисовки интерфейса рассматривались такие библиотеки как: tkinter, pyqt5 и wxpython. Выбрана была библиотека pyqt5, за счёт обширного набора

стандартных элементов, удобным интерфейсом разработки и обширным сообществом.

Также в работе использовались следующие библиотеки:

Numpy – для расчетов тригонометрических функций, квадратных и кубических корней.

PyqtGraph – для создания графика внутри графического интерфейса.

QtWebEngine – для отрисовки html-страницы внутри интерфейса.

Csv – для удобной работы с csv-файлами

Sqlite3 – для работы с базами данных

Функции для решения задачи на языке Python находится в файле calculate.py и содержит внутри себя две функции: для расчета траектории без учета сопротивления и с учетом сопротивления.

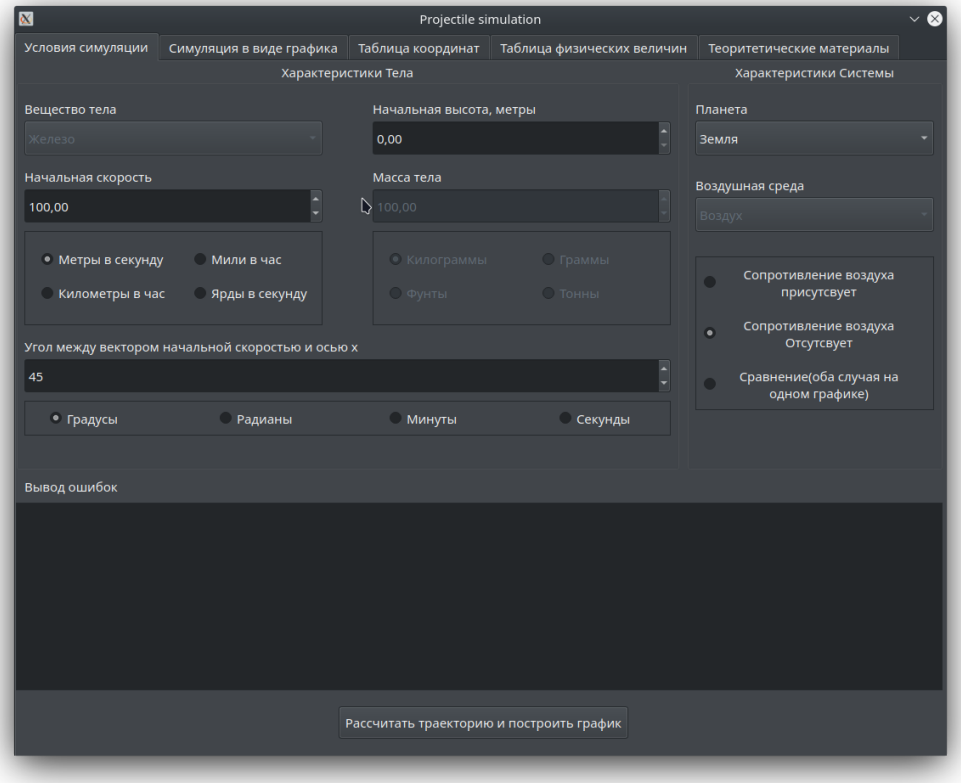
Функция по расчету без учета сопротивления воздуха имеет название calculate_without_air_resistance. В качестве аргументов функция принимает начальную скорость тела, угол между вектором начальной скорости и осью X, ускорение свободного падения, начальную координату по оси Y. Внутри себя функция просчитывает момент падения тела и с определенным шагом просчитывает траекторию полета тела. Функция возвращает три массива одинаковых размеров: координаты по оси X, координаты по оси Y, массив времени.

Функция по расчету с учетом сопротивления воздуха называется calculate_with_air_resistance. В качестве аргументов функция принимает начальную скорость тела, угол между вектором начальной скорости и осью X, ускорение свободного падения, начальную координату по оси Y, плотность воздушной среды, плотность вещества шарообразного тела, массу тела. Внутри себя функция аналогично первой функции рассчитывает время полета тела и с определенным шагом рассчитывает траекторию полета. Возвращаемые значения функции аналогичны возвращаемым значениям первой функции.

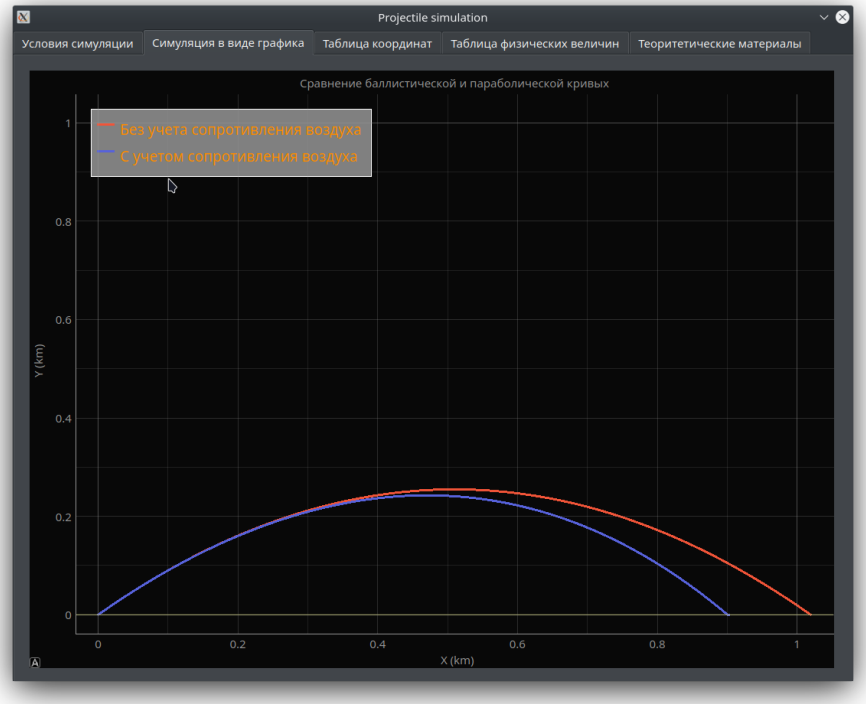
Визуальная составляющая программы

Интерфейс программы находится в файле ui_file.py. Он состоит из пяти страниц:

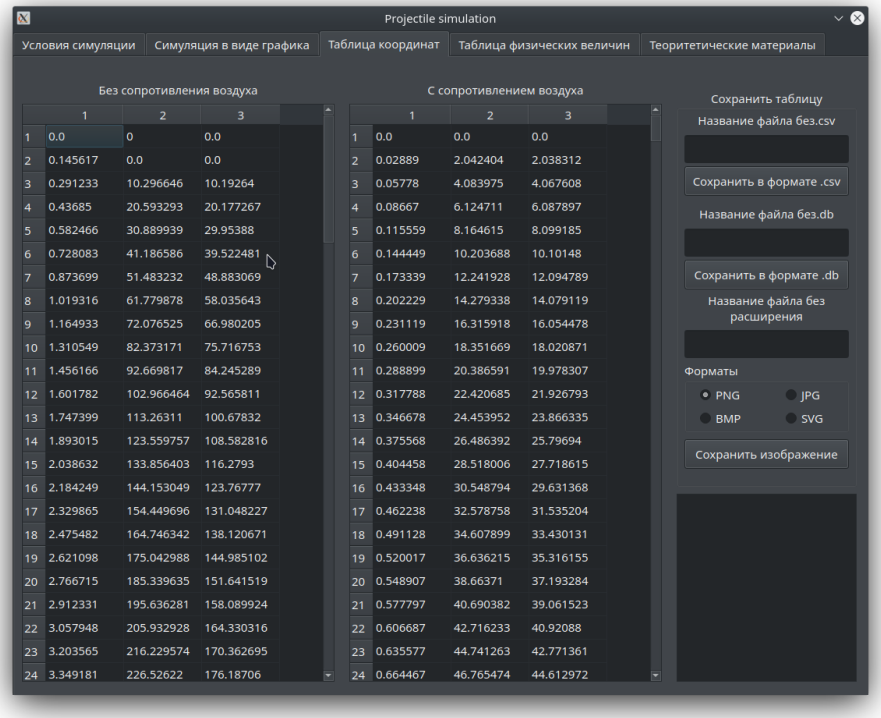
1. Страница ввода данных. На данной странице пользователь может ввести начальные данные в различных величинах, выбрать материал тела, воздушную среду и планету, на которой происходит бросок тела. Также он может выбрать тип графика: без учета или с учетом сопротивления воздуха, а также сравнение обоих графиков.



2. Страница графика. На данной странице пользователь может изучить полученный график с помощью соответствующего элемента.



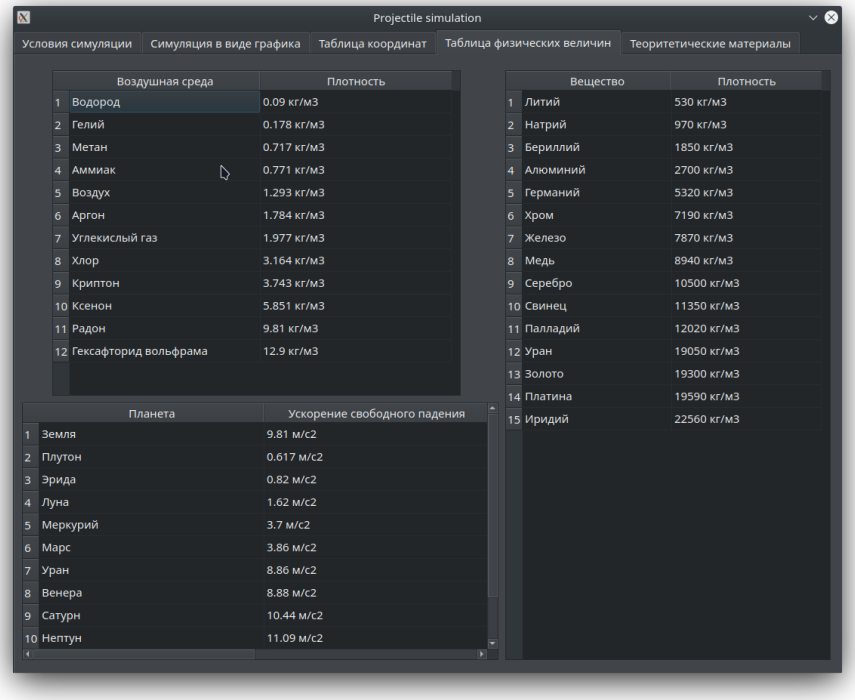
3. Страница таблиц координат. На данной странице выведены таблицы всех координат, полученных при расчете траектории. Пользователь может сохранить их в формате CSV или DB, а также сохранить график в виде растрового или векторного изображения.



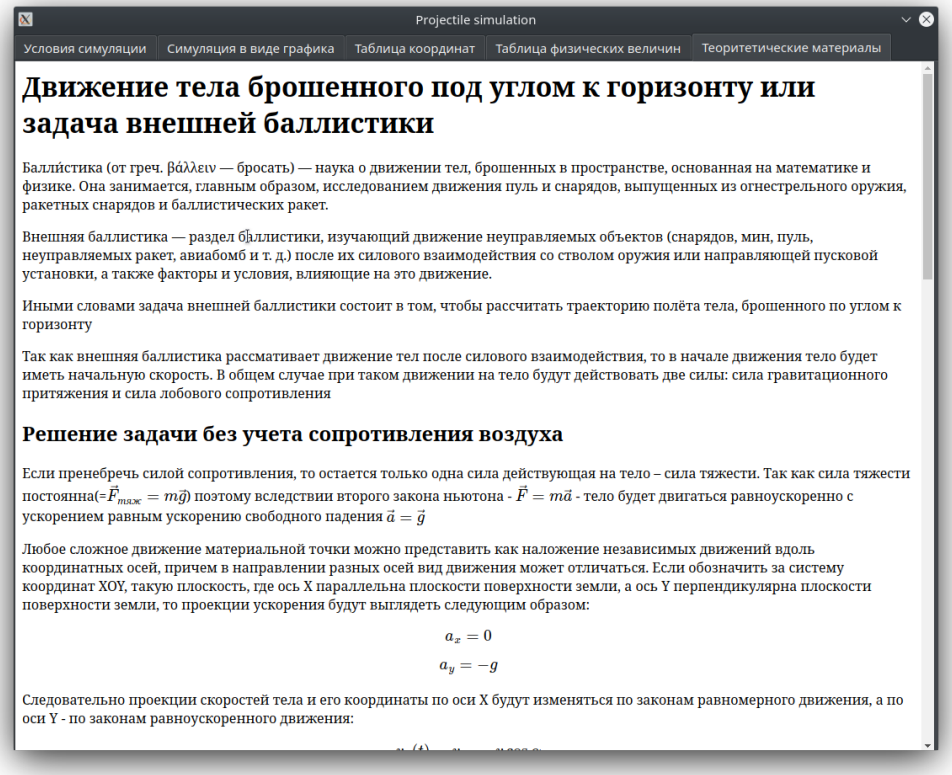
	1	2	3
1	0.0	0	0.0
2	0.145617	0.0	0.0
3	0.291233	10.296646	10.19264
4	0.43685	20.593293	20.177267
5	0.582466	30.889939	29.95388
6	0.728083	41.186586	39.522481
7	0.873699	51.483232	48.883069
8	1.019316	61.779878	58.035643
9	1.164933	72.076525	66.980205
10	1.310549	82.373171	75.716753
11	1.456166	92.669817	84.245289
12	1.601782	102.966464	92.565811
13	1.747399	113.26311	100.67832
14	1.893015	123.559757	108.582816
15	2.038632	133.856403	116.2793
16	2.184249	144.153049	123.76777
17	2.329865	154.449696	131.048227
18	2.475482	164.746342	138.120671
19	2.621098	175.042988	144.985102
20	2.766715	185.339635	151.641519
21	2.912331	195.636281	158.089924
22	3.057948	205.932928	164.330316
23	3.203565	216.229574	170.362695
24	3.349181	226.52622	176.18706

	1	2	3
1	0.0	0.0	0.0
2	0.02889	2.042404	2.038312
3	0.05778	4.083975	4.067608
4	0.08667	6.124711	6.087897
5	0.115559	8.164615	8.099185
6	0.144449	10.203688	10.10148
7	0.173339	12.241928	12.094789
8	0.202229	14.279338	14.079119
9	0.231119	16.315918	16.054478
10	0.260009	18.351669	18.020871
11	0.288899	20.386591	19.978307
12	0.317788	22.420685	21.926793
13	0.346678	24.453952	23.866335
14	0.375568	26.486392	25.79694
15	0.404458	28.518006	27.718615
16	0.433348	30.548794	29.631368
17	0.462238	32.578758	31.535204
18	0.491128	34.607899	33.430131
19	0.520017	36.636215	35.316155
20	0.548907	38.66371	37.193284
21	0.577797	40.690382	39.061523
22	0.606687	42.716233	40.92088
23	0.635577	44.741263	42.771361
24	0.664467	46.765474	44.612972

4. Таблица физических величин. На данной странице пользователь может получить информацию о характеристиках воздушных сред, материалов и планет, доступных для выбора при расчетах.



5. Теоретические материалы. На данной странице пользователь может получить теоретическую базу о решении задачи внешней баллистики.



Использование разработки

Полученную в ходе работы программу можно использовать для автоматизации расчетов в решении физических задач. Также работу можно использовать для обучения школьников решению баллистических задач

Результат

В результате работы было получено обобщённое решение задачи внешней баллистики, а также реализован программный комплекс для её решения. Дальнейшие направления развития программы: улучшение решения дифференциальных уравнений, обобщение и реализация решений для тел других форм.

Список используемой литературы

1. Данжу Джульен Путь Python. Черный пояс по разработке, масштабированию, тестированию и развертыванию.[Текст] -СПб.: Питер, 2020. -256 с.: ил. — (Серия «Библиотека программиста»).
2. Мэтиз Эрик Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2017. — 496 с.: ил. — (Серия «Библиотека программиста»).
3. Python 3 и PyQt 5. Разработка приложений. — 2-е изд., перераб. и доп. / Н. А. Прохоренок, В. А. Дронов. — СПб.: БХВ-Петербург, 2018. — 832 с.: ил. — (Профессиональное программирование)
4. Python и анализ данных / пер. с анг. А. А. Слинкина. – М.: ДМК Пресс, 2020. – 540 с.: ил.