Проект в рамках конкурса "Инженеры будущего"

по направлению "Программирование"

«Моделирование траектории движения брошенного тела»

Ученика 10 класса «Г»

Бауманской инженерной школы № 1580

Шомполова Максима Андреевича

Аннотация

Задача:

Создать приложение для моделирования траектории движения брошенного тела для различных случаев и расчёта различных характеристик этого броска.

Описание решения:

Разработанная программа позволяет рассчитывать характеристики различных бросков и моделировать их траекторию в зависимости как от начальной высоты, скорости, угла бросания, максимальной высоты, времени полёта и дальности броска, так и от среды, в которой находится тело, расстояния от центра планеты и характеристик этой планеты.

Актуальность решения

Разработанное решение, относящееся к области ИТ программирование, может помочь пользователю в следующих ситуациях:

1) Решение школьных задач по физике по темам: «Бросок под углом к горизонту», «Движение планет и искусственных спутников».
2) Моделирование реальных физических явлений для решения практических задач.
3) Моделирование реальных физических явлений для изучения связанных с этими явлениями задач.
4) Необходимость использования как русского, так и английского языков
5) Ознакомление с теорией по теме «Бросок под углом к горизонту»
6) Ознакомление с реальными физическими явлениями для общего развития

Цель

Создать приложение, которое смогло бы:

1) Моделировать траекторию и рассчитывать характеристики (начальная высота, скорость, угол бросания, дальность броска, максимальные и минимальные скорость и высота, время полёта) бросков следующих видов:
   а) Бросок вертикально вверх
   б) Бросок с начальной скоростью, направленной горизонтально
   в) Бросок под углом к горизонту
   г) Бросок с учётом сопротивления среды, её изменения с увеличением высоты над поверхностью планеты и изменения силы притяжения в зависимости от характеристик планеты (масса, радиус) и расстояния от тела до её центра (далее бросок в среде)
2) Использовать различные языки: английский, русский
3) Предоставлять теорию для ознакомления тремя способами: вывод на экран, открытие вебсайта или файла с теорией
4) Создавать модель используя данные, полученные с помощью взаимодействия с пользователем через достаточно простой интерфейс или прочитанные из файла с известным полным именем
5) Сохранять результаты, полученные пользователем при работе с приложением локально

Задачи

1) Ознакомиться с теорией, необходимой для создания интерфейса (Python 3 (Tkinter)), открытия файлов и вебсайтов
2) Анализ функций, которые должна выполнять программа, и структуры интерфейса, необходимых для выполнения поставленных целей
3) Ознакомиться с теорией по теме «Бросок с учётом сопротивления среды, её изменения с увеличением высоты над поверхностью планеты и изменения силы притяжения в зависимости от характеристик планеты (масса, радиус) и расстояния от тела до её центра»
4) Создать алгоритмы для выполнения целей 1 (а, б, в) (необходимые функции и классы)
5) Создать достаточно точный и оптимальный алгоритм для выполнения цели 1 (г) (необходимые функции и классы)
6) Найти необходимые теоретические материалы и создать интерфейс для цели 3
7) Создать возможность сохранения локально (в том числе и форму хранения данных)
8) Создать необходимые алгоритмы для выбора языка и перевода на этот язык и подготовить необходимые текстовые данные.

Существующие решения

1) Моделирование с помощью средств приложений Excel или Word. Решение даёт возможность достаточно точно построить графики различных зависимостей, но имеет достаточно ограниченные возможности визуализации и неприглядный и сложный интерфейс.
2) Моделирование с помощью средств приложений для построения графиков функций (например https://www.mathway.com/ru/graph). Данное решение имеет хорошие интерфейс и точность, но требует от пользователя знания темы «Бросок под углом к горизонту», так как требует знания законов движения.
3) Wolframalpha (https://www.wolframalpha.com). Решение, позволяющее рассчитать траекторию по различным характеристикам, с высокой точностью, и неплохим интерфейсом, но не предоставляет возможности для работы на русском языке и имеет платные функции.
4) Другие онлайн решения (https://planetcalc.ru/1508/ или https://www.fxyz.ru/формулы_по_физике/механика/кинематика/падение_тел/движение_тела_брошенного_горизонтально/уравнение_траектории_тела_брошенного_горизонтально/) не позволяют производить расчёты основываясь на различных данных и имеют интерфейс, не позволяющий получить достаточное количество информации о явлении.

Новизна решения

1) Решение имеет опции для работы с различными характеристиками явления и позволяет пользователю самому выбрать, какие данные он хочет ввести.
2) Решение предоставляет возможность работы с моделью среды, имеющей сопротивление.
3) Решение предоставляет возможность для сохранения результатов локально.
4) Решение позволяет ознакомиться с теорией по теме тремя различными способами: из файла, прочитав на экране, на вебсайте.
5) Решение предоставляет возможность работы с русским и английским языками.

Описание решения

Выбор средств для реализации целей и задач:

Был выбран язык Python 3, так как он позволяет производить расчёты с большими числами и позволяет реализовать многие функции (в том числе интерфейс), и решение потребности в длительных расчётах, а следовательно язык позволяет создать достаточно оптимизированное решение. Для создания интерфейса была выбрана библиотека Tkinter, так как она позволяет создать достаточно простой интерфейс и имеет все необходимые для решения функции. Также были использованы библиотеки math (функции, необходимые для расчётов), os (для открытия файлов) и webbrowser (для открытия вебсайтов). Не были использованы специализированные библиотеки для построения графиков, так как они не могут в точности позволить реализовать задуманный функционал.

Проделанная работа:

1) Разработал интерфейс для стартового окна
2) Разработал интерфейс для всех видов бросков, кроме броска в среде
3) Разработал классы для хранения и обработки характеристик для всех видов бросков, кроме броска в среде (в том числе для расчёта характеристик)
4) Разработал функции для создания моделей для всех видов бросков, кроме броска в среде
5) Разработал интерфейс для броска в среде
6) Разработал класс хранения и обработки характеристик для броска в среде
7) Разработал функции для создания моделей для броска в среде
8) Разработал функции и интерфейс для локального сохранения, открытия сохранённых результатов и чтения характеристик броска из файла
9) Разработал функции и интерфейс для ознакомления с теорией

10) Разработал интерфейс и функции и получил текстовые данные для возможности работы с русским или английским языками
11) Провёл необходимые тесты

Алгоритмы и методы:

1) Интерфейс был реализован при помощи библиотеки Tkinter:

   Функции:

   goto_theory – интерфейс для выбора способа ознакомления с теорией

   goto_language – интерфейс для выбора языка

   goto_start – интерфейс для стартового окна

   goto_save – интерфейс для сохранения

   goto_saved – интерфейс для открытия сохранённых данных и получения данных из файла

   goto_vertical_model – интерфейс для создания модели вертикального броска

   goto_model – интерфейс для создания модели броска под углом к горизонту

   goto_hormodel - интерфейс для создания модели горизонтального броска (задание дополнительных настроек для goto_model)

   goto_special_model – интерфейс для создания модели броска в среде (пред введены характеристики Земли)

   goto_choose_model – интерфейс для выбора вида броска

   del_all – удаление всех имеющихся элементов интерфейса в нашем окне
2) Алгоритм для моделирования вертикального броска (функция make_vertical_model). Сначала проверим правильность и достаточность данных. Очевидно, что траекторией вертикального броска является вертикальная прямая, которую мы и рисуем. Необходимые и дополнительные характеристики рассчитываем при помощи функции класса vertical_mods, о котором я расскажу позже. В случае, если моделирование невозможно, выводим сообщение об ошибке.

3) Алгоритм для моделирования броска под углом к горизонту и горизонтального броска, как частного случая, (функция chart). Сначала проверим правильность и достаточность данных. Необходимые и дополнительные характеристики рассчитываем при помощи функции класса mods, о котором я расскажу позже. Сначала определяем точки с максимальными координатами по осям x и y и с их помощью определяем масштаб. Далее при помощи уравнения траектории строим график. В случае, если моделирование невозможно, выводим сообщение об ошибке.
4) Алгоритм для моделирования броска в среде (функция make_special_model). Сначала проверим правильность и достаточность данных. Из-за невозможности определить данное движение каким-либо уравнением, выберем малый промежуток времени dt и будем на каждом из этих промежутков рассматривать наше движение, как равноускоренное. Неточностью можно пренебречь, так как dt мал. dt будем выбирать в зависимости от необходимой точности, а следовательно, времени движения (вращение на орбите, падение) и количества получающихся промежутков (для оптимизации). Движение будем рассматривать при помощи системы координат, в которой точка O лежит в центре Земли (Земля в нашей модели шар). Сила сопротивления уменьшается в зависимости от высоты над поверхностью планеты. Определив точки с максимальными координатами по осям x и y, находим масштаб. Далее строим траекторию и изображение Земли. При рассмотрении движения определяем дополнительные характеристики и выводим их. В случае, если моделирование невозможно, выводим сообщение об ошибке.
5) Класс mods.
    a) Хранит все необходимые данные для броска под углом к горизонту (начальная высота (y0), начальная скорость (v0), угол бросания (alpha), максимальная высота (ym), время полёта (t_fl) и дальность броска (xm)).
    b) Функции:

i) update – обновляет все денные, основываясь на данных, введённых пользователем
ii) calculate_extra – рассчитывает дополнительные характеристики явления
iii) setts – возвращает характеристики в удобном для сохранения виде
iv) enough – проверяет достаточно ли данных для проведения расчётов
v) right – проверяет непротиворечивость данных
vi) correct – проверяет, возможно ли начать моделирование, основываясь на имеющихся данных
vii) pull – возвращает имеющиеся данные пользователю
viii) clear – очищает все данные
ix) calculate – производит необходимые расчёты для получения полных данных из имеющихся

6) Классы vertical_mods (начальная высота (y0), начальная скорость (v0), максимальная высота (ym) и время полёта (t_fl)) и special_mods (начальная высота (y0), начальная скорость (v0), угол бросания (alpha), масса тела (m), масса планеты (M), радиус планеты (R), коэффициент сопротивления среды (k). Их функции выполняют те же роли, что и одноимённые функции класса mods.
7) Класс my_entry. Состоит из заголовка поля ввода (Label) и самого поля. Функции:
   a) get – возвращает полученные данные или в виде вещественного числа, или в виде пустой строки.
   b) destroy – удаляет все элементы класса из окна.
   c) pull – возвращает в поле полученные данные.
8) Важные переменные:
   a) items – список всех элементов интерфейса (для их удаления)
   b) settings – класс с текущими данными
   c) window – наше окно
   d) leng – текущий язык (об этом далее)

e) tr – словарь с текущим текстом для интерфейса
f) status – текущий вид броска

9) Функция сохранения (save). Сохраняет необходимые денные в файл с названием, введённым пользователем.
10) Функция для открытия сохранённых данных и получения данных из файла (goto_saved_model). По статусу и данным из файла создаёт интерфейс и модель (если это возможно).
11) Функция для возвращения к модели то сохранения (go_back).
12) Функции для выбора русского (choose_russian) или английского (choose_english). Переход при помощи изменения переменной tr.
13) Функции для ознакомления с теорией в окне (show_theory), в файле (open_theory_file) или в веб-браузере (open_webbrowser).

Полученные результаты:

Программа может:

1) Моделировать траекторию и рассчитывать характеристики (начальная высота, скорость, угол бросания, дальность броска, максимальные и минимальные скорость и высота, время полёта) бросков следующих видов:
    a. Бросок вертикально вверх
    b. Бросок с начальной скоростью, направленной горизонтально
    c. Бросок под углом к горизонту
    d. Бросок в среде
2) Использовать различные языки: английский, русский
3) Предоставлять теорию для ознакомления тремя способами: вывод на экран, открытие вебсайта или файла с теорией
4) Создавать модель используя данные, полученные с помощью взаимодействия с пользователем через достаточно простой интерфейс или прочитанные из файла с известным полным именем

5) Сохранять результаты, полученные пользователем при работе с приложением локально

Рекомендации для работы:

При вводе данных при помощи файла с расширением txt. Название файла писать без расширения. Сам файл должен иметь следующий вид.

1) В первой строке должно быть введено одно из следующих слов: usual (бросок под углом к горизонту), horizontal (горизонтальный бросок), vertical (вертикальный бросок), special (бросок в среде).
2) В последующих строках должны быть введены следующие данные в указанном порядке (в случае если данная характеристика явления не вводится вместо числа должна быть пустая строка).
    a. Бросок под углом к горизонту: начальная высота, скорость, угол бросания, время полёта, дальность броска и максимальная высота.
    b. Вертикальный бросок: начальная высота, скорость, время полёта и максимальная высота.
    c. Горизонтальный бросок: начальная высота, скорость, угол бросания (0), время полёта, дальность броска и максимальная высота.
    d. Бросок в среде: начальная высота, скорость, угол бросания, масса планеты, масса тела, радиус планеты, коэффициент сопротивления среды для материальной точки (сила прямо пропорциональна квадрату скорости.
3) Данные вводятся в системе СИ (углы в радианах)
4) Возможно представление в виде выражения (например 3*10**3 (** - возведение в степень) или 89/180*pi (pi -число пи). То же верно дся ввода через интерфейс.
5) Пример ввода:
   usual
   1

1

1

Расшифровка: Бросок под углом к горизонту с начальной высотой 1 метр, начальной скоростью 1 метр в секунду и углом броска 1 радиан.

Примеры работы с интерфейсом – смотри приложение.

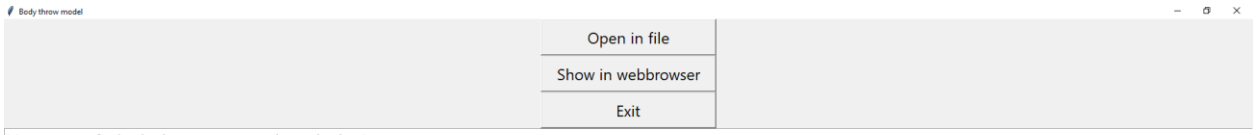
Перспективы развития проекта

1) Увеличение допустимой сложности бросков, не в ущерб оптимизации.
2) Добавление других характеристик для расчёта и дополнительных характеристик.
3) Добавление новых сред, в том числе сред, создающих значительную силу выталкивания.
4) Добавление возможности работы не только с материальной точкой.
5) Увеличение функциональности интерфейса.
6) Добавление возможности работы при помощи веб-сайта.
7) Добавление дополнительных языков.
8) Улучшение и добавление других способов сохранения результатов.
9) Улучшение теоретического материала.
10) Ввести дополнительные единицы измерения.

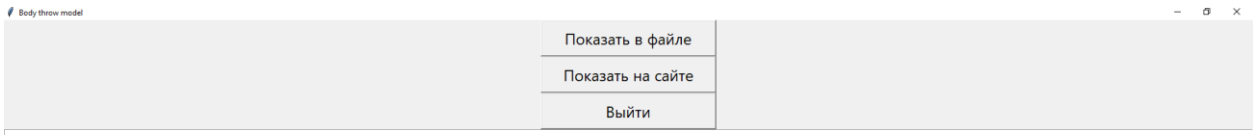
Библиографический список

1) https://www.wolframalpha.com
2) https://planetcalc.ru/1508/
3) https://www.fxyz.ru/формулы_по_физике/механика/кинематика/падение_тел/движение_тела_брошенного_горизонтально/уравнение_траектории_тела_брошенного_горизонтально/
4) https://ido.tsu.ru/schools/physmat/data/res/virtlab/text/m2_1.html
5) https://www.matematicus.ru/en/mechanics-and-physics/movement-of-a-body-thrown-at-an-angle-to-the-horizon
6) https://pythonru.com/uroki/obuchenie-python-gui-uroki-po-tkinter
7) https://riptutorial.com/Download/tkinter-ru.pdf
8) https://wiki.fenix.help/fizika/sila-soprotivleniya

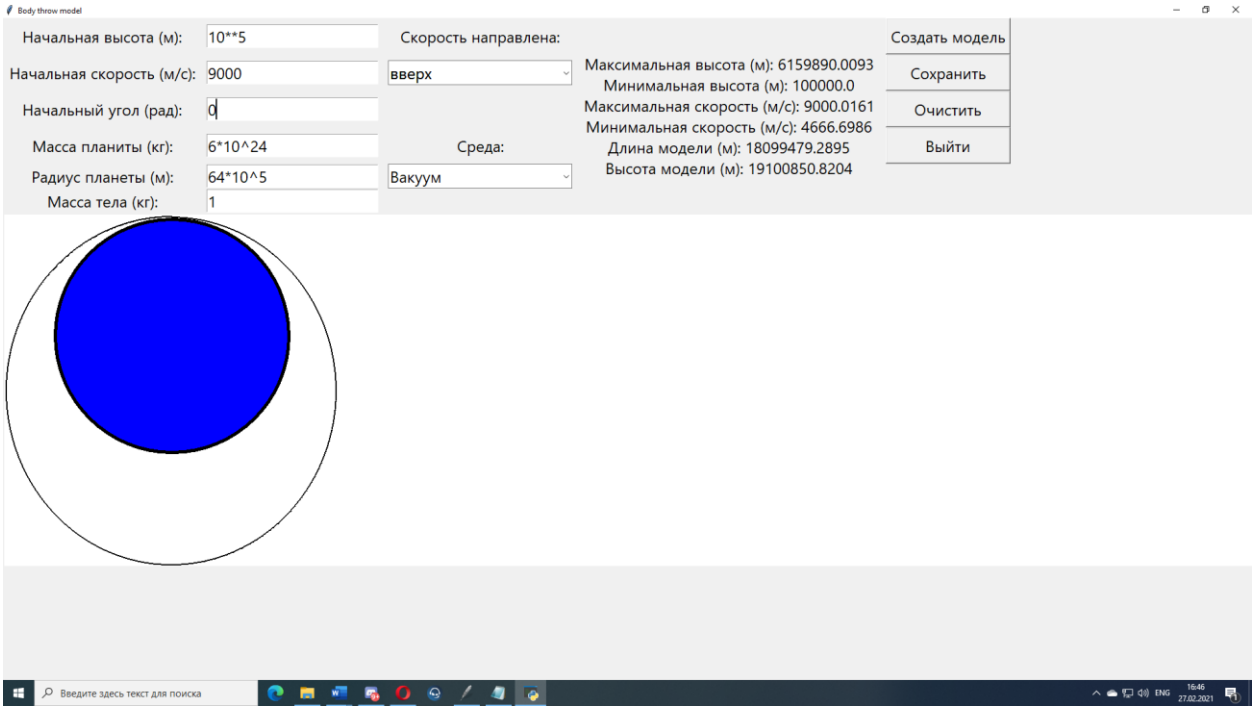
Приложение – примеру работы с интерфейсом

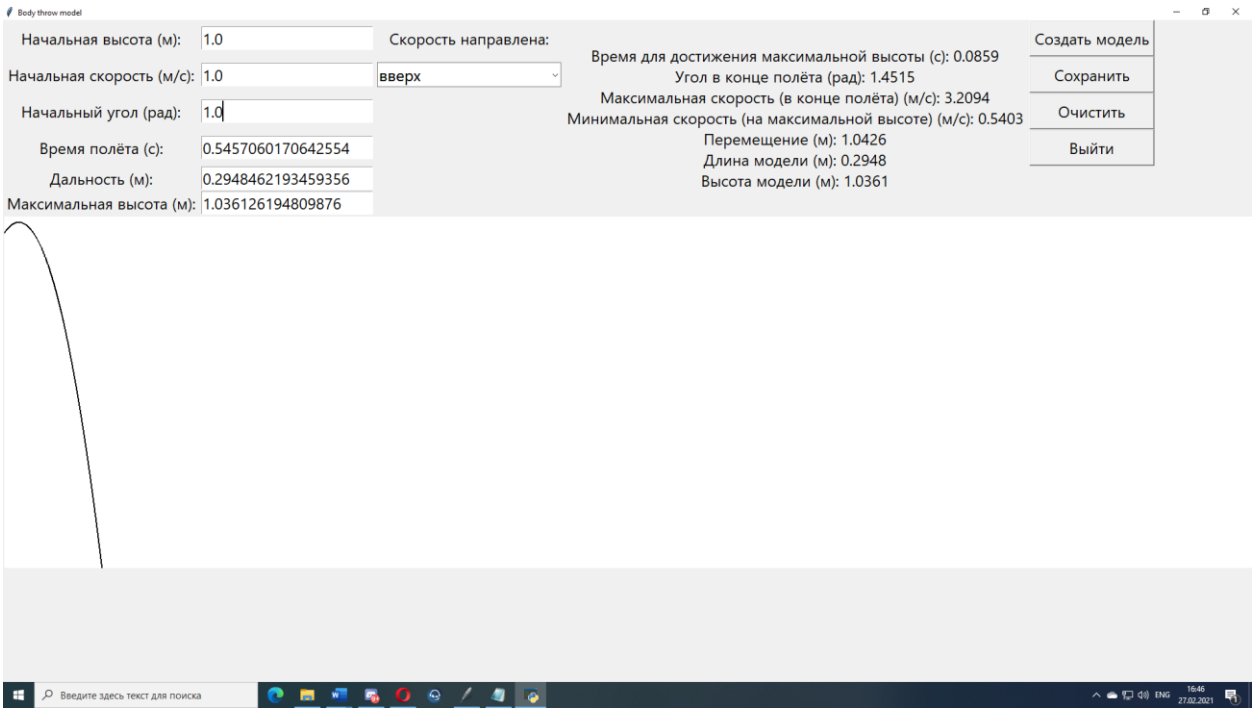


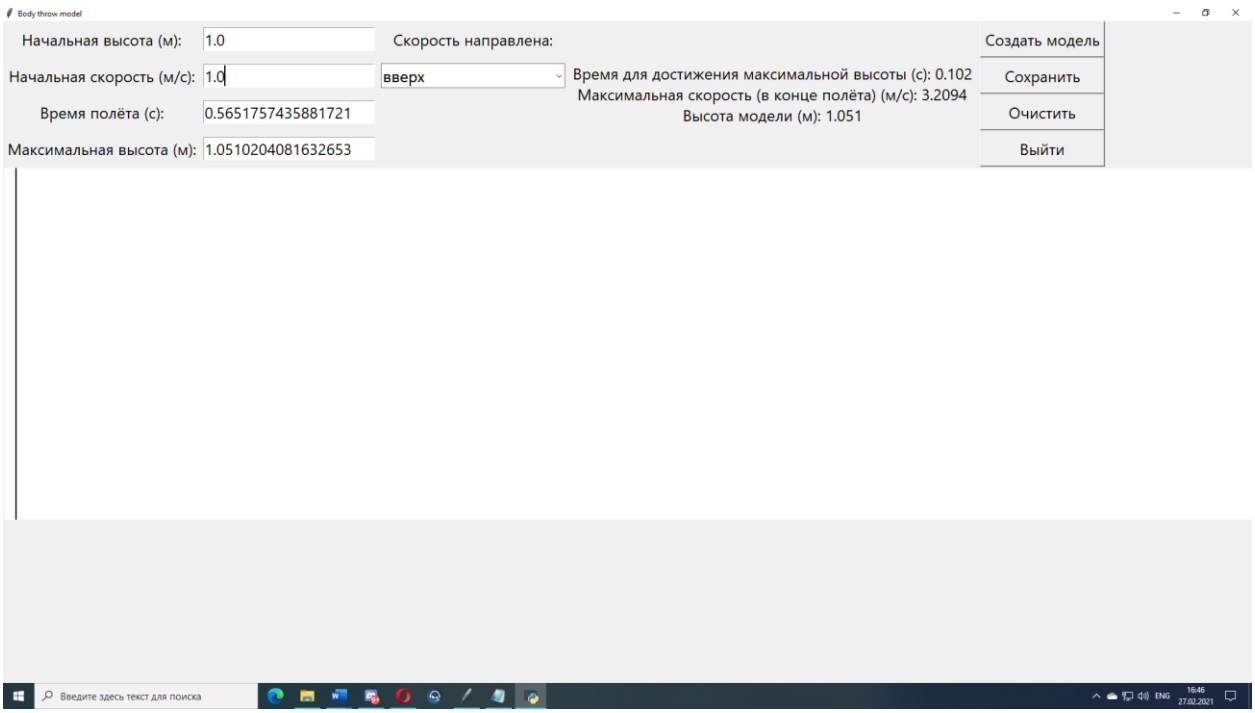
Movement of a body thrown at an angle to the horizon
Consider the motion of a body in the Earth's gravity field, but we will not take into account the air resistance. Let the initial velocity of the thrown body be directed at an angle to the horizon α. The body is thrown from a height of y=h0; x0=0.
Then, at the initial moment of time, the body has horizontal (vx) and vertical (vy) components of velocity. The projection of the velocity on the axis of coordinates at t=0 is equal to:
{v0_x=v0cosα, v0_y=v0sinα. (1).
The acceleration of a body is equal to the acceleration of gravity and is always directed down:
a=g(2).
Hence, the projection of the acceleration on the X-axis is equal to zero, and on the Y axis is equal to ay=g.
Since the acceleration component is zero on the X-axis, the velocity of the body in this direction is a constant value and is equal to the projection of the initial velocity on the X-axis (see(1)). The movement of the body along the X-axis is uniform.
In the situation shown in Figure 1, the Y-axis body will move first up and then down. In this case, the acceleration of the body in both cases is equal to the acceleration of g. The body spends the same amount of time going up from an arbitrary height y=h0 to the maximum lifting height (h) as it does going down from h to y=h0. Therefore, the points that are symmetrical with respect to the top of the body lift lie at the same height. It turns out that the trajectory of the body is symmetrical with respect to the point-the top of the rise - and this is a parabola.
The velocity of a body thrown at an angle to the horizon can be expressed by the formula:
v(t)=v0+gt (3), where v0 is the velocity of the body at the time of the throw. Formula (3) can be considered as the result of the addition of the velocities of two independent movements in straight lines, in which the body participates.
The expressions for the velocity projection on the axis take the form:
{vx=v0cosa, vy=v0sina-gt (4).
The equation for moving a body when moving in the field of gravity:
s(t)=s0+v0t+gt^2/2 (5), where s0 is the displacement of the body at the initial moment of time.
Projecting equation (5) on the X and Y coordinate axes, we get:
{x=v0cos(α)·t, y=h0+v0sin(α)·t-gt^2/2 (6).



Движение тела, брошенного под углом к горизонту
Рассмотрим движение тела в поле тяжести Земли, сопротивление воздуха учитывать не будем. Пусть начальная скорость брошенного тела направлена под углом к горизонту α. Тело брошено с высоты y=h0; x0=0.
Тогда в начальный момент времени тело имеет горизонтальную (vx) и вертикальную (vy) составляющие скорости. Проекции скорости на оси координат при t=0 равны:
{v0_x=v0cosα, v0_y=v0sinα. (1).
Ускорение тела равно ускорению свободного падения и все время направлено вниз:
a=g(2).
Значит, проекция ускорения на ось X равна нулю, а на ось Y равна ay=g.
Так как по оси X составляющая ускорения равна нулю, то скорость движения тела в этом направлении является постоянной величиной и равна проекции начальной скорости на ось X (см.(1)). Движение тела по оси X равномерное.
При ситуации, изображенной на рис.1 тело по оси Y будет двигаться сначала вверх, а затем виз. При этом ускорение движения тела в обоих случаях равно ускорению g.
На прохождение пути вверх от произвольной высоты y=h0 до максимальной высоты подъема (h) тело тратит столько же времени, сколько на падение вниз от h до y=h0.
Следовательно, точки симметричные относительно вершины подъема тела лежат на одинаковой высоте. Получается, что траектория движения тела симметрична относительно точки-вершины подъема - и это парабола.
Скорость движения тела, брошенного под углом к горизонту можно выразить формулой:
v(t)=v0+gt (3), где v0 - скорость тела в момент броска. Формулу (3) можно рассматривать как результат сложения скоростей двух независимых движений по прямым линиям, в которых участвует тело.
Выражения для проекции скорости на оси принимают вид:
{vx=v0cosα, vy=v0sinα−gt (4).
Уравнение для перемещения тела при движении в поле тяжести:
s(t)=s0+v0t+gt^2/2 (5), где s0 - смещение тела в начальный момент времени.
Проектируя уравнение (5) на оси координат X и Y, получим:
{x=v0cos(α)·t, y=h0+v0sin(α)·t−gt^2/2 (6).
Тело, двигаясь вверх, имеет по оси Y сначала равнозамедленное перемещение, после того, как тело достигает вершины, движение по оси Y становится


Body throw model
Начальная высота (м): 10**5
Начальная скорость (м/с): 9000
Начальный угол (рад): 0
Масса планиты (кг): 6*10^24
Радиус планеты (м): 64*10^5
Масса тела (кг): 1
Скорость направлена:
вверх
Среда:
Вакуум
Максимальная высота (м): 6159890.0093
Минимальная высота (м): 100000.0
Максимальная скорость (м/с): 9000.0161
Минимальная скорость (м/с): 4666.6986
Длина модели (м): 18099479.2895
Высота модели (м): 19100850.8204
Создать модель
Сохранить
Очистить
Выйти
Введите здесь текст для поиска
ENG
16:46
27.02.2021


Body throw model
Начальная высота (м): 1.0
Начальная скорость (м/с): 1.0
Начальный угол (рад): 1.0
Время полёта (с): 0.5457060170642554
Дальность (м): 0.2948462193459356
Максимальная высота (м): 1.036126194809876
Скорость направлена:
вверх
Время для достижения максимальной высоты (с): 0.0859
Угол в конце полёта (рад): 1.4515
Максимальная скорость (в конце полёта) (м/с): 3.2094
Минимальная скорость (на максимальной высоте) (м/с): 0.5403
Перемещение (м): 1.0426
Длина модели (м): 0.2948
Высота модели (м): 1.0361
Создать модель
Сохранить
Очистить
Выйти
Введите здесь текст для поиска
ENG
16:46
27.02.2021


Body throw model
Начальная высота (м):
1.0
Скорость направлена:
Создать модель
Начальная скорость (м/с):
1.0
вверх
Время для достижения максимальной высоты (с): 0.102
Максимальная скорость (в конце полёта) (м/с): 3.2094
Высота модели (м): 1.051
Сохранить
Время полёта (с):
0.5651757435881721
Очистить
Максимальная высота (м):
1.0510204081632653
Выйти
Введите здесь текст для поиска
ENG
16:46
27.02.2021