

Лабораторная работа по физическому моделированию №1, 16 января

На последнем практическом занятии мы с вами моделировали падение камня с некоторой высоты в поле тяжести Земли. Ответ на такую простую задачу вы могли получить и без компьютера с помощью простого аналитического вычисления. Сегодня мы рассмотрим чуть более сложную задачу: мы изучим падение с трением. Итак, в этой нехитрой работе вам предстоит промоделировать высадку десанта на вражескую территорию. :-)

Высадка происходит в два этапа: десантник выпрыгивает из самолета и некоторое время падает в поле тяжести с небольшим сопротивлением воздуха, затем он раскрывает парашют, сопротивление воздуха значительно усиливается, за счет чего посадка происходит с относительно небольшой скоростью.

1 Без парашюта и воздуха

Для начала сделаем то, что мы уже умеем. Пусть нет никакого сопротивления воздуха и парашюта. Нам нужно построить графики координата-время и скорость-время для свободного падения в поле тяжести $\vec{g} = 9.81 \frac{m}{s^2}$. Мы будем делать это итеративно: разобьем время на маленькие промежутки Δt и будем вычислять скорость \vec{v} и высоту h в каждый момент $t + \Delta t$.

- Создайте программу с названием *without_friction.py*.
- Проимпортируем необходимые библиотеки

```
import matplotlib.pyplot as plt
import numpy as np
```

- Теперь создадим функцию, которая будет возвращать массивы времен, скоростей и координат. Что она будет принимать в качестве аргументов? Высоту, с которой начинается падение, и интервал Δt . Итак, функция выглядит следующим образом:

```
def Fall(height, delta_t):
    list_of_velocities = []
    list_of_times = []
    list_of_heights = []
    # ...
    # something happens here
    # ...
    return list_of_velocities, list_of_times, list_of_heights
```

Внутри функции вам следует вычислять координаты и скорости до тех пор, пока высота больше нуля.

- Теперь, после того как функция написана, осталось только нарисовать графики. Например, для высоты от времени:

```
v, t, h = Fall(2000, 0.001)
plt.title('Fall')
plt.plot(t, h)
plt.xlabel("Time")
```

```
plt.ylabel("Height")
```

Первое упражнение закончено. Надеюсь, это было не сложно.

2 Высадка десанта

Ну а теперь приступим к задаче! Мы будем высаживать десантника с низколетящего бомбардировщика, летящего на высоте $h = 2000$ м. Масса десантника со снаряжением $m = 90$ кг, а высота, на которой он раскрывает парашют, — 1000 метров. Сила действующая на десантника складывается из силы притяжения и силы сопротивления воздуха. Силу сопротивления будем определять эмпирической формулой:

$$\vec{F} = -0.65S\vec{v}|v|,$$

где v — скорость падения и S — площадь поперечного сечения. (На самом деле, численный коэффициент 0.65 очень трудно получить из теоретических соображений, поэтому его предпочитают измерять экспериментально для различных сред. Более того, к вышеприведенной формуле нужно относиться с осторожностью, ее критерии применимости — отдельный, не очень простой, вопрос.)

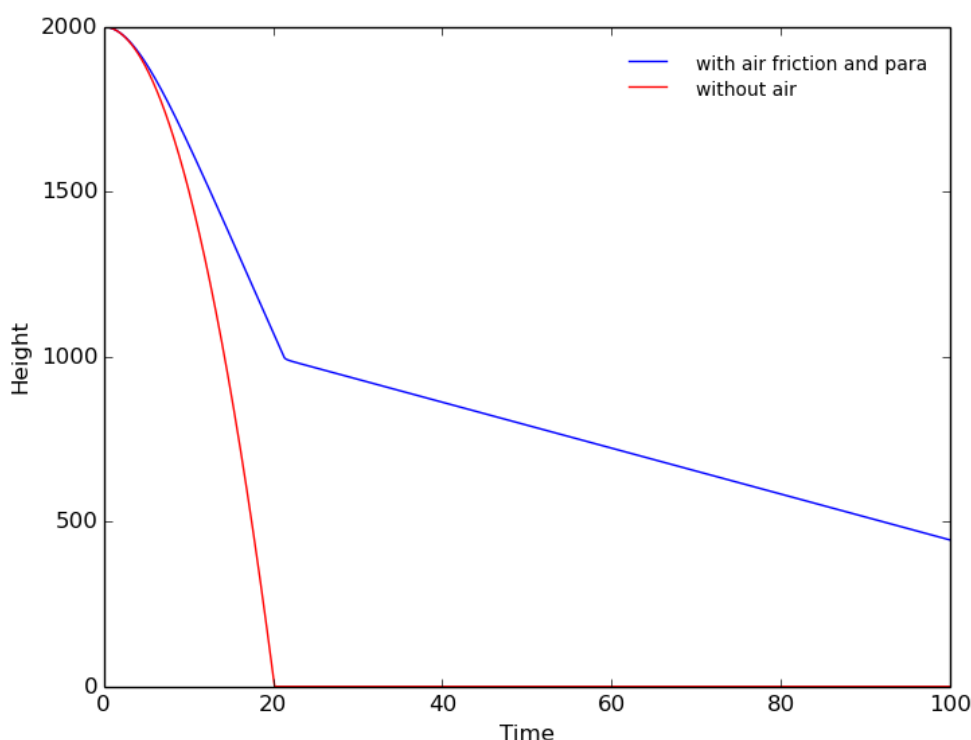
- Создайте отдельную программу *skydiving.py*, подключите необходимые библиотеки.
- Создайте функцию для вычисления координат и скоростей, передавая ей в качестве параметров массу парашютиста (заметьте, что масса играет роль только тогда, когда мы "включаем" трение), начальную высоту, высоту раскрытия парашюта и интервал Δt .

```
def Skydive(mass, position, position_open,
            delta_t):
    list_of_velocities = []
    list_of_times = []
    list_of_heights = []
    # ...
    # something happens here
    # ...
    return list_of_velocities, list_of_times, list_of_heights
```

- Что же происходит внутри функции? Опять, нужно организовать цикл, пока высота больше нуля. Внутри цикла нужно высчитывать ускорение парашютиста. Оно зависит от параметра S . До раскрытия парашюта $S = 0.4$ квадратных метров, после — $S = 28$ квадратных метров. В итоге, цикл выглядит примерно так:

```
while height > 0:
    if (height > 1000):
        S = 0.4
        air_friction = -0.65 * S * velocity * abs(velocity)
        ...
    else:
        S = 28
        air_friction = -0.65 * S * velocity * abs(velocity)
        ...
```

- Теперь осталось только выполнить функцию при для аргументов $m = 90$, $position = 2000$, $position_open = 1000$, $delta_t = 0.001$ и построить график.
- Постройте на одном графике падение без трения и приземление с парашютом. Результат должен получиться как на рисунке ниже (если ваш преподаватель нигде не налажал :-).



3 Немного рефлексии

Остается два вопроса важных вопроса.

- Можете ли вы получить последний результат аналитически? Вычислите явную формулу для времени приземления парашютиста. Обязательно сделайте это полезное упражнение!
- Очевидно, что высадку десанта на вражескую территорию нужно осуществить за как можно более короткое время. Вычислите моделированием на компьютере и аналитически, на какой высоте нужно раскрыть парашют, чтобы приземлиться как можно быстрее, считая, что критическая скорость, на которой десантник может приземлиться без повреждений, равна 10 м/с.

Как всегда, присылайте работы на почты dontwritehim@gmail.com и i_khisambeev@mail.ru.