

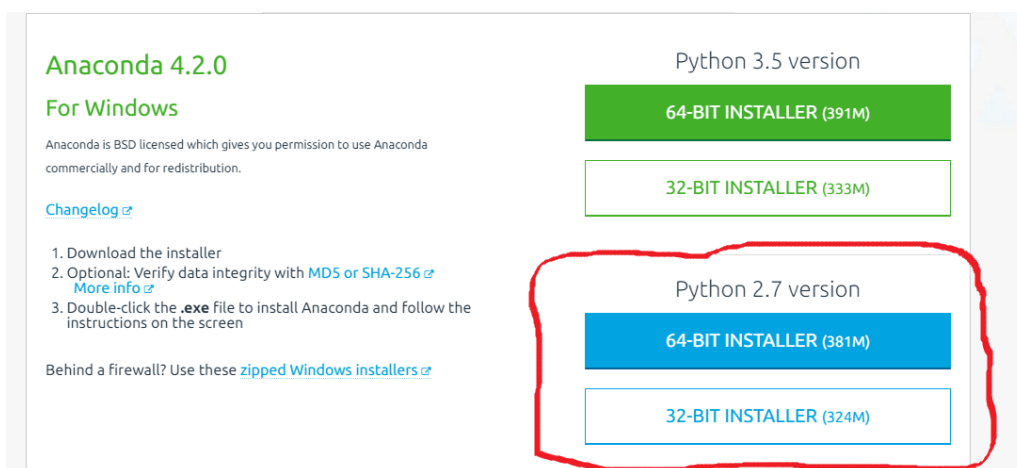
Руководство

Это руководство по установке и использованию всех необходимых программ для рисования графиков функций, в том числе в среде Minecraft с помощью языка программирования Python.

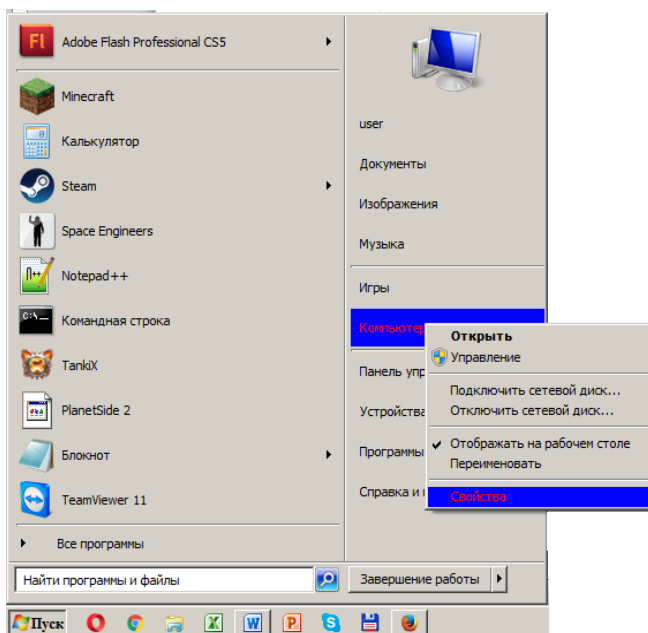
Установка Python и необходимых математических библиотек.

Скачаем и установим пакет программ под названием Анаконда (Anaconda). Страница для скачивания находится вот по адресу: <https://www.continuum.io/downloads>

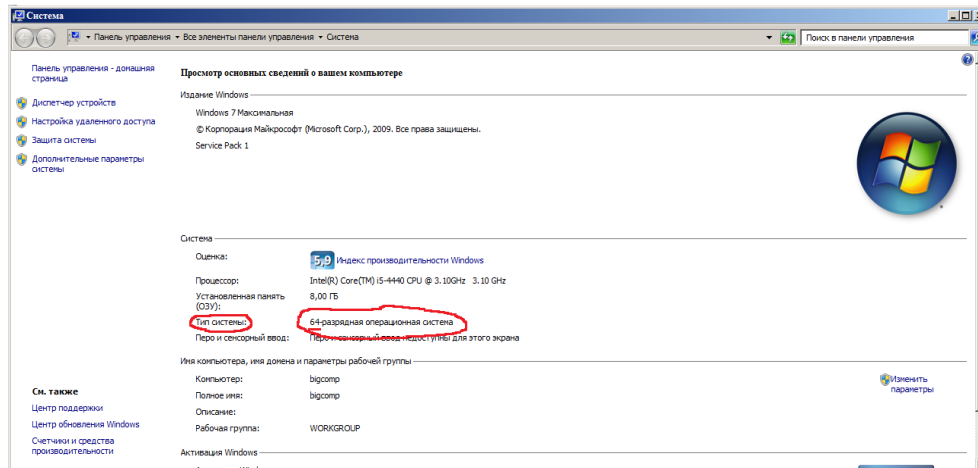
По ссылке предлагается скачать программу для разных операционных систем, но, скорее всего, у вас на компьютере установлена операционная система Windows. Поэтому нам надо выбрать один из вариантов *синего цвета*.



Нам предлагают скачать вариант программы для 32-разрядной операционной системы, или для 64-разрядной (про разрядность можно почитать, например, [здесь](#)). Чтобы понять, какой вариант нам нужен, нажмем левой кнопкой мыши на кнопку <Пуск>, выберем пункт меню <Компьютер>, нажмем на него правой кнопкой мыши и выберем пункт меню <Свойства>.



В открывшемся окне кроме других характеристик вашего компьютера и операционной системы указана также ее разрядность. На моем компьютере это выглядит так:



Скачиваем программу и по окончании загрузки устанавливаем ее. Теперь у вас на компьютере есть все необходимое для работы – анаконда установит ruPython и все нужные библиотеки.

Математика

Давайте теперь напишем простейшую программу, рисующую график уравнения параболы: $y = ax^2 + bx + c$.

Создадим новый файл, назовем его parabola.py и запишем в него следующие строки:

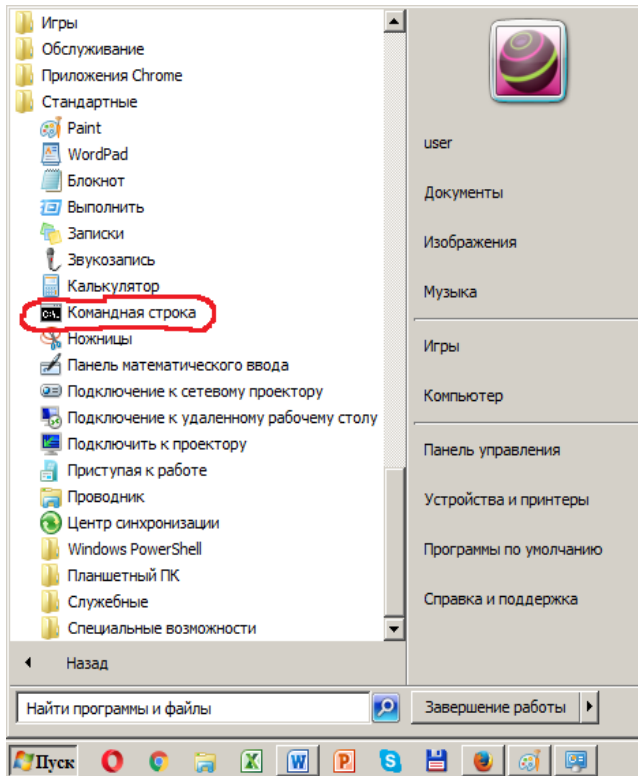
```
import numpy as np
import matplotlib.pyplot as plt
```

```
x = np.linspace(-10, 10, 1000)
y = x**2 + 2*x + 2
```

```
plt.plot(x, y)
```

```
plt.grid()
plt.show()
```

Теперь сохраним файл и запустим его. Для этого надо открыть консоль windows:



и выполнить в ней команду `python parabola.py`:

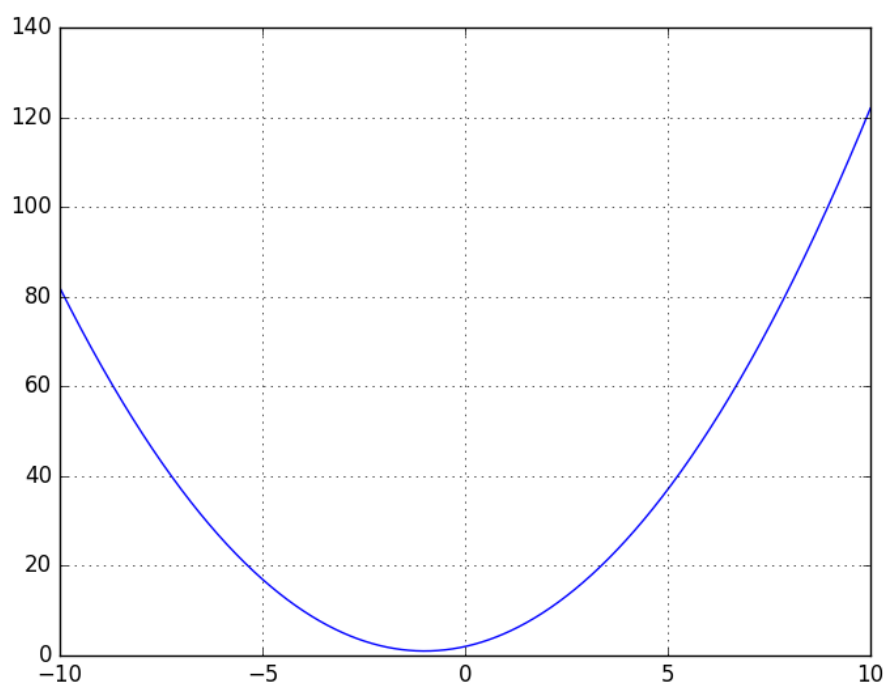
```
Command Prompt - python parabola.py
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Andrew.Tkachenko>cd
C:\Users\Andrew.Tkachenko

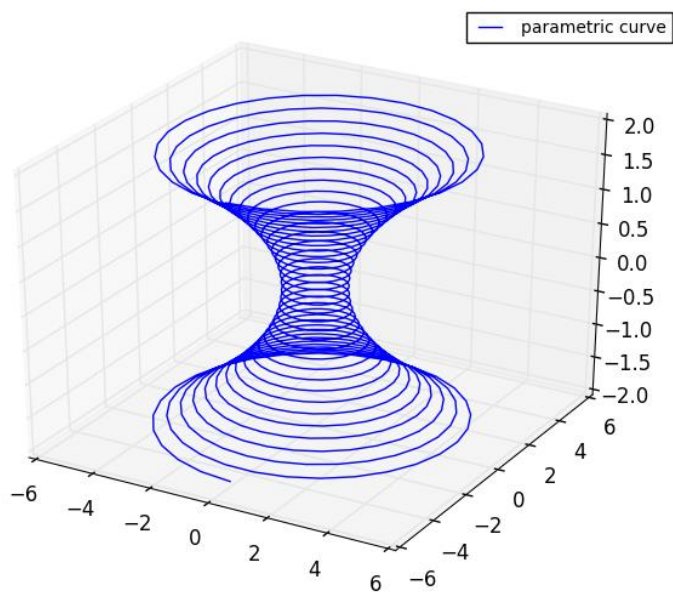
C:\Users\Andrew.Tkachenko>clear
'clear' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Andrew.Tkachenko>cd mproject
C:\Users\Andrew.Tkachenko\mproject>python parabola.py
```

В результате выполнения программы откроется новое окно с нарисованным графиком. График можно сохранить как отдельное изображение.



Библиотека также позволяет строить графики в трехмерной координатной системе:



Устанавливаем с Minecraft

Для работы с Майнкрафт необходимо установить на компьютер пакет программ, которые запускают локальный сервер Minecraft и графического клиента.

Устанавливаем сервер

Скачаем из интернета специальный пакет [Adventure in Minecraft](#). Это архив (про архивы подробнее можно посмотреть [здесь](#)), поэтому перед использованием его надо распаковать. После распаковки архива на жестком диске создастся каталог <Adventures In Minecraft>, в котором лежит сервер minecraft и библиотека, помогающая работать с Minecraft из Python.

В каталоге <Adventure in Minecraft> находим подкаталог <My Adventures>. В этом каталоге мы будем создавать и запускать наши программы.

В каталоге <Adventures In Minecraft> находится файл **StartBukkit.bat**. Чтобы запустить сервер достаточно выполнить этот файл.

Устанавливаем клиента

Первым делом необходимо скачать [minecraft версии 1.6.4](#). Открываем и запускаем Майнкрафт. После загрузки программы выбираем режим «Multiplayer», затем «Direct Connect» и далее выбираем сервер «localhost».

Построение графиков функций в Minecraft

Напишем программу, рисующую график затухающих колебаний по формуле:

$$\sin(2\pi x)e^{-x}$$

Вот её код:

```
import pylab

import numpy as np
import matplotlib.pyplot as plt

import mcpi.minecraft as minecraft
import mcpi.block as block
import time

x = np.arange(0, 30 * np.pi, .01)
y = np.multiply(np.sin(2*np.pi*x), np.exp(-x))

mc = minecraft.Minecraft.create()
pos = mc.player.getTilePos()

for i in range(0, x.size):
    xo=x[i]*20
    yo=y[i]*20
    mc.setBlocks(pos.x+xo, pos.y+yo, pos.z, pos.x+xo+1, pos.y+yo+1, pos.z, block.STONE.id)
```

Соответствующее коду изображение:



Пример более сложной фигуры

