© Золотарев П.А., Колегов К.С. 2022

Основы работы с библиотеками SciPy и NumPy
САМОСТОЯТЕЛЬНАЯ РАБОТА

http://www.kimrt.ru
Онлайн-курс
Суперкомпьютерные технологии в задачах моделирования

NumPy — это расширение языка Python, добавляющее поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых математических функций для операций с этими массивами [1].

SciPy — это библиотека Python с открытым исходным кодом, предназначенная для решения научных и математических проблем [2].

Установка. Для установки *Numpy* [3] и *Scipy* пропишем в командной строке следующее [4].

```
pip3 install numpy scipy
```

Работа с NumPy. Импортируем *NumPy*.

```
import numpy as np
```

Теперь создадим массив с помощью *np.array()*.

```
Arr = np.array([2, 3, 1])
```

И выведем его на экран.

```
print(Arr)
```

```
E:\Python\Python39\python.exe "E:/Zolotarev/самостоятельные работы/numpy scipy/num.py"
[2 3 1]

Process finished with exit code 0
```

Также массивы можно создавать и другими способами, например, *np.zeros()* создает массив определенной длины и заполняет его нулями.

```
Arr = np.zeros(2)

print(Arr)
```

```
E:\Python\Python39\python.exe "E:/Zolotarev/самостоятельные работы/numpy scipy/num.py"
[0. 0.]

Process finished with exit code 0
```

Также можно создать двумерный массив, добавив еще аргумент.

```
Arr = np.zeros(2, 3)

print(Arr)
```

```
E:\Python\Python39\python.exe "E:/Zolotarev/самостоятельные работы/numpy scipy/num.py"
[[0. 0. 0.]
 [0. 0. 0.]]

Process finished with exit code 0
```

Аналогично можно создать массив заполненный единицами с помощью *np.ones()*, а помощью *np.eye()* можно создать единичную матрицу.

```
Arr = np.eye(3)

print(Arr)
```

```
E:\Python\Python39\python.exe "E:/Zolotarev/самостоятельные работы/numpy scipy/num.py"
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]

Process finished with exit code 0
```

Для создания последовательных массивов используется *np.arange(start, stop, step).*

```
Arr = np.arange(3)

print(Arr)
```

```
E:\Python\Python39\python.exe "E:/Zolotarev/самостоятельные работы/numpy scipy/num.py"
[0 1 2]

Process finished with exit code 0
```

```
Arr = np.arange(0,10,2)

print(Arr)
```

```
E:\Python\Python39\python.exe "E:/Zolotarev/самостоятельные работы/numpy scipy/num.py"
[0 2 4 6 8]

Process finished with exit code 0
```

Также можно использовать функцию *np.linspace()*, которая принимает количество элементов вместо шага.

```
Arr = np.linspace(0, 8, 5)

print(Arr)
```

```
E:\Python\Python39\python.exe "E:/Zolotarev/самостоятельные работы/numpy scipy/num.py"
[0. 2. 4. 6. 8.]

Process finished with exit code 0
```

© Золотарев П.А., Колегов К.С. 2022

```
Arr = np.linspace(0, 1, 5)
print(Arr)
```

```
E:\Python\Python39\python.exe "E:/Zolotarev/самостоятельные работы/numpy scipy/num.py"
[0.   0.25 0.5  0.75 1.  ]

Process finished with exit code 0
```

Создадим два массива a и b.

```
a = np.array([2,3,8,1])
b = np.array([4,5,9,7])
```

Сложим их и выведем на экран.

```
c = a + b
print(c)
```

```
E:\Python\Python39\python.exe "E:/Zolotarev/самостоятельные работы/numpy scipy/num.py"
[ 6  8 17  8]

Process finished with exit code 0
```

А теперь объединим эти массивы.

```
c = np.concatenate((a,b))
print(c)
```

```
E:\Python\Python39\python.exe "E:/Zolotarev/самостоятельные работы/numpy scipy/num.py"
[2 3 8 1 4 5 9 7]

Process finished with exit code 0
```

Отсортируем массив c.

```
c = np.concatenate((a,b))
c = np.sort(c)
print(c)
```

```
E:\Python\Python39\python.exe "E:/Zolotarev/самостоятельные работы/numpy scipy/num.py"
[1 2 3 4 5 7 8 9]

Process finished with exit code 0
```

Работа с SciPy.

Вычисление определенного интеграла функции одной переменной [5].

Импортируем инструменты для работы с интегралами из *scipy*.

```
from scipy import integrate
```

Объявим функцию с помощью *lambda*.

© Золотарев П.А., Колегов К.С. 2022

```
a = lambda x:x**2
```

Вычислим интеграл на интервале от 0 до 1.

```
b = integrate.quad(a,0,1)
```

Выведем результат.

```
print(b)
```

```
E:\Python\Python39\python.exe "E:/Zolotarev/самостоятельные работы/доработать/numpy scipy/sci.py"
(0.33333333333333337, 3.700743415417189e-15)

Process finished with exit code 0
```

Первое значение – определенный интеграл, второе – погрешность.

Определитель матрицы.

Импортируем *numpy* как *np* и линейную алгебру из *scipy*.

```
import numpy as np
from scipy import linalg
```

Создаем матрицу (массив).

```
a = np.array([[1,3,2],[1,5,8],[2,2,3]])
```

Вычисляем определитель.

```
b = linalg.det(a)
```

Выводим матрицу (массив) и определитель.

```
print(a)
print(b)
```

```
E:\Python\Python39\python.exe "E:/Zolotarev/самостоятельные работы/доработать/numpy scipy/sci.py"
[[1 3 2]
 [1 5 8]
 [2 2 3]]
22.0

Process finished with exit code 0
```

Одномерная интерполяция.

Импортируем *numpy*, *scipy* и *matlotlib*.

```
from scipy import interpolate
import matplotlib.pyplot as plt
import numpy as np
```

Сгенерируем массивы x и y с координатами функции $y=x^2$ на отрезке [-10;10].

```
x = np.arange(-10, 11)
```

© Золотарев П.А., Колегов К.С. 2022

```
y = x**2
```

Создаем одномерную интерполяцию для нашей функции.

```
f = interpolate.interp1d(x, y)
```

Сгенерируем массивы для интерполяции на отрезке [0;10].

```
x1 = np.arange(0,11)
```

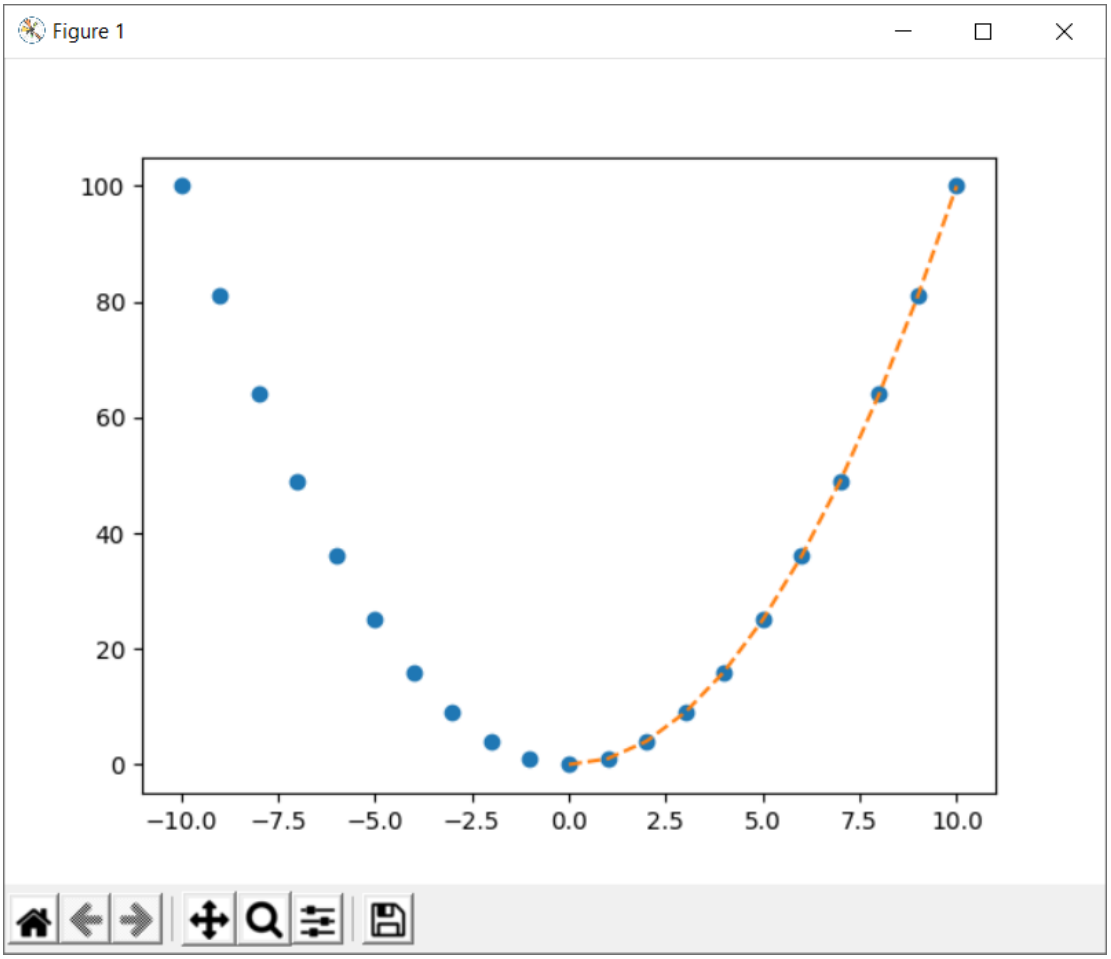
```
y1 = f(x1)
```

И нарисуем график с помощью *matlpotlib*.

```
plt.plot(x,y,'o',x1,y1,'--')
```

```
plt.show()
```

Получим следующий график.



Весь код программы представлен ниже.

```
from scipy import interpolate
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
x = np.arange(-10, 11)
```

© Золотарев П.А., Колегов К.С. 2022

```
y = x**2
f = interpolate.interp1d(x, y)
x1 = np.arange(0,11)
y1 = f(x1)
plt.plot(x,y,'o',x1,y1,'--')
plt.show()
```

http://www.kimrt.ru/index/course_stm/0-24

Проект реализуется победителем Конкурса на предоставление грантов преподавателям магистратуры 2020/2021 благотворительной программы «Стипендиальная программа Владимира Потанина» Благотворительного фонда Владимира Потанина.

Источники

1. NumPy. URL: https://pythonworld.ru/numpy
2. Руководство по SciPy. URL: https://pythonru.com/biblioteki/scipy-python
3. NumPy, часть 1: начало работы. URL: https://pythonworld.ru/numpy/1.html
4. Getting started SciPy. URL: https://docs.scipy.org/doc/scipy/getting_started.html#getting-started-ref
5. Scipy.integrate.quad. URL: https://docs.scipy.org/doc/scipy/reference/reference/generated/scipy.integrate.quad.html