Numerical Methods for Physicists

Introduction

Titus Beu
University "Babeș-Bolyai"
Department of Biomolecular Physics
Cluj-Napoca, Romania

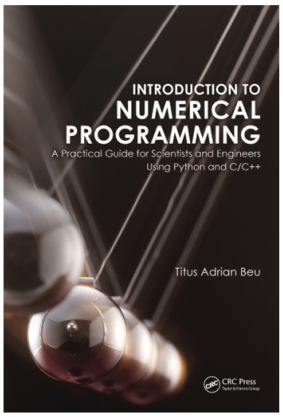
Course structure

1. Introduction
2. Basic programming techniques in Python
3. Elements of scientific graphics
4. Evaluation of functions
5. Systems of Linear Algebraic Equations
6. Eigenvalue problems
7. Approximation of tabulated functions – interpolation and regression
8. Integration of functions
9. Ordinary Differential Equations
10. Partial Differential Equations
11. Random Walks

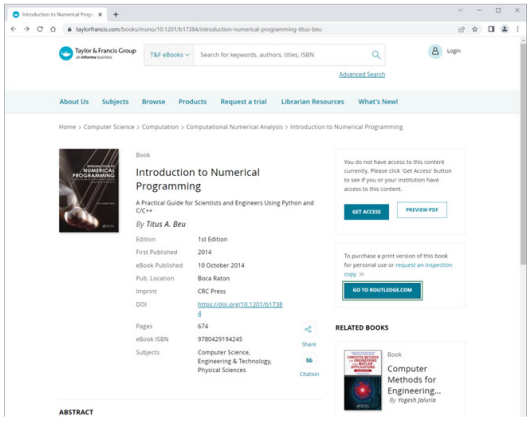
Bibliography

- Beu, T. A., *Introduction to Numerical Programming: A Practical Guide for Scientists and Engineers Using Python and C/C++*, (CRC Press / Taylor & Francis, 2014).
- Beu, T. A., *Calcul numeric în C*, Ediția a III-a, (Editura Albastră, MicroInformatica, Cluj-Napoca, 2004).
- http://phys.ubbcluj.ro/~tbeu/INP/index.html
- https://www.python.org/
- Kernighan, B. and P. J. Plauger (1978). *The Elements of Programming Style, 2nd Edition*. New York: McGraw-Hill.
- Knuth, D. E. (1998). *The Art of Computer Programming (3rd Ed.), vol. 1: Fundamental Algorithms*. Reading, MA: Addison-Wesley Professional.
- E. Yourdon, *Techniques of Program Structure and Design* (Prentice-Hall, Englewood Cliffs, NJ, 1975).

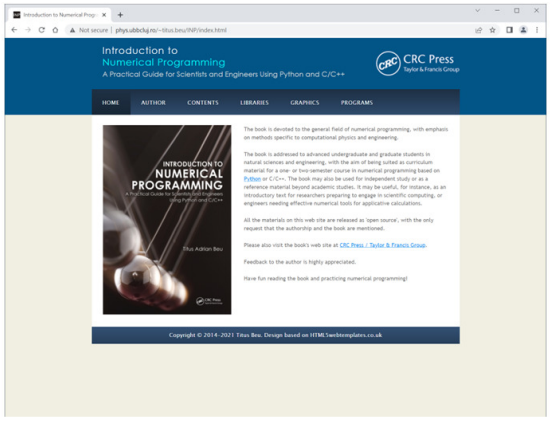
Introduction to Numerical Programming



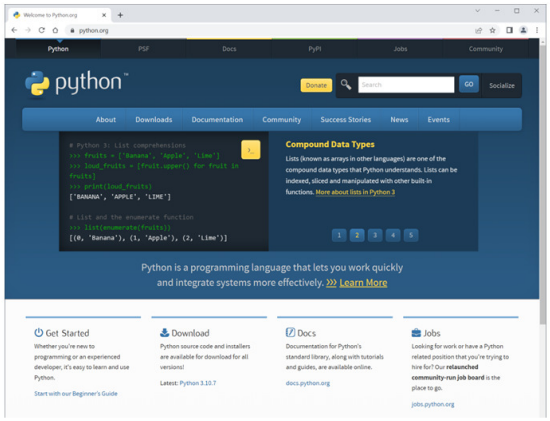
Introduction to Numerical Programming



Introduction to Numerical Programming

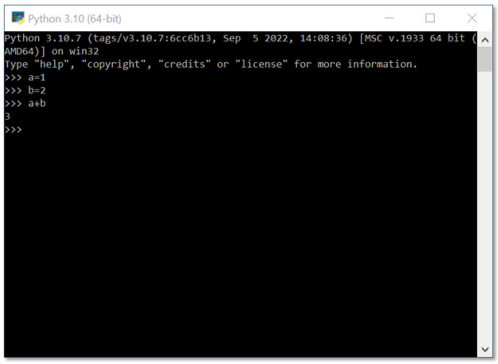


Python



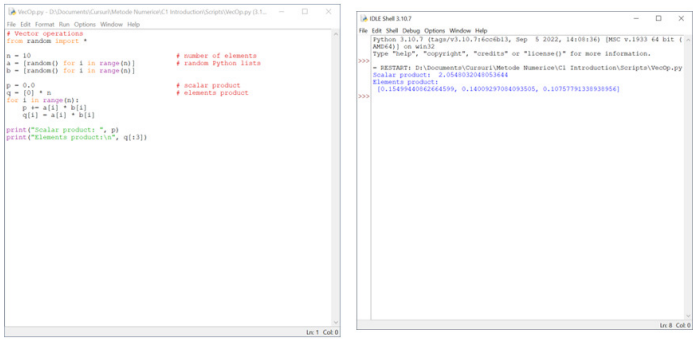
Python

- Python console



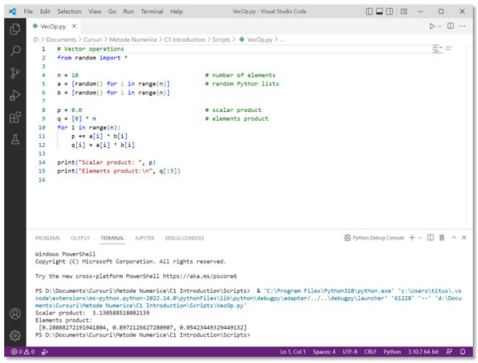
Python

IDLE - Integrated Development and Learning Environment



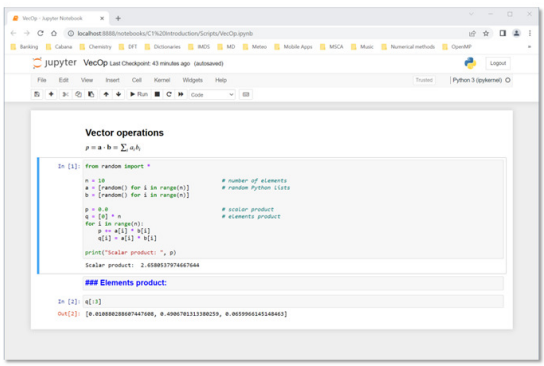
Visual Studio Code

- **Visual Studio Code** – free code editor optimized for building modern web / cloud applications.



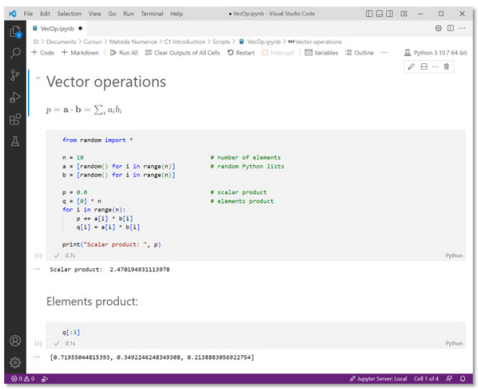
Jupyter Notebook

- **Jupyter Notebook** - web application for creating and sharing computational documents.

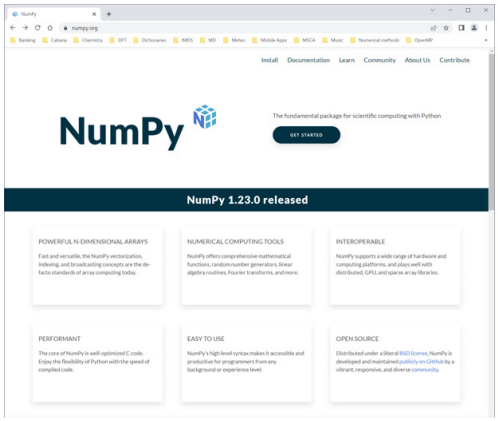


Visual Studio Code

- **Jupyter notebook**



NumPy



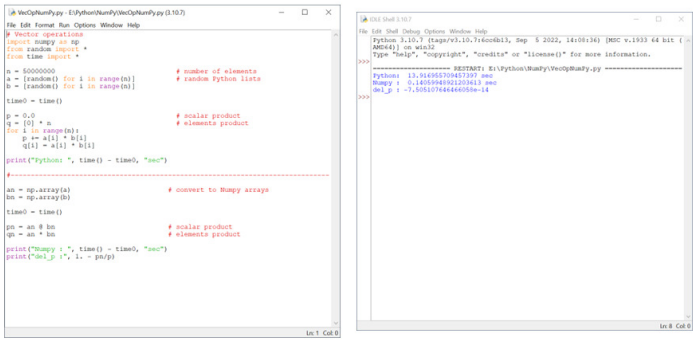
NumPy

What is NumPy?

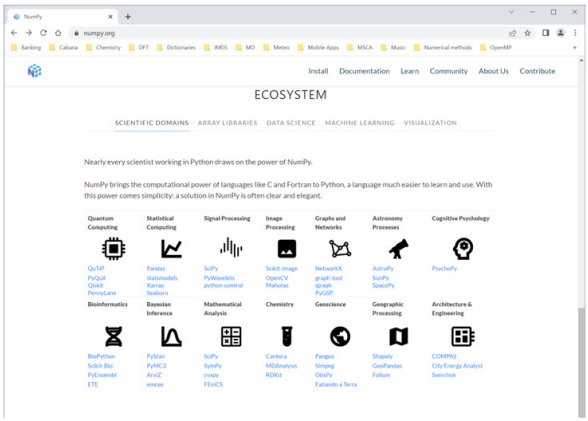
https://numpy.org/doc/stable/user/whatisnumpy.html

- NumPy is the **fundamental package/library for scientific computing** in Python.
- At the center of NumPy is the ndarray object - encapsulates n-dimensional contiguous arrays of homogeneous data types.
- For performance, many operations with ndarrays are performed in compiled code, using efficient vectorization (low-level parallelization).
- Differences between NumPy arrays and standard Python sequences:
  - NumPy arrays have a fixed size (set at creation), unlike Python lists (which can grow dynamically). Changing the size of an ndarray will create a new array and delete the original.
  - The elements in a NumPy array are all of the same data type and occupy the same size in memory. Exception: arrays of Python or NumPy objects.
  - NumPy arrays facilitate advanced operations on large data sets, which are executed more efficiently than is possible using Python's built-in sequences.

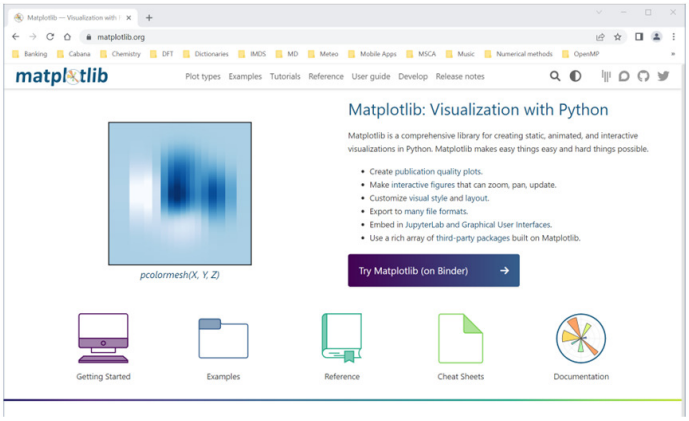
NumPy



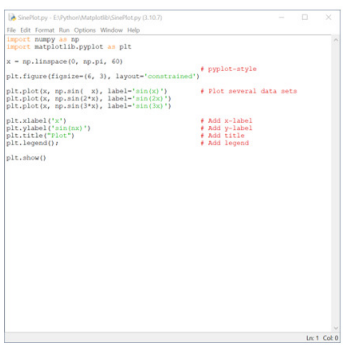
NumPy

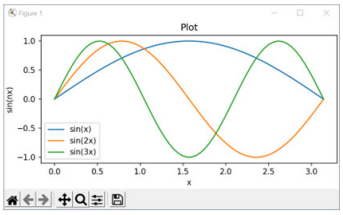


Matplotlib



Matplotlib





Matplotlib

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

x = np.linspace(-10, 10, 101)

xm, ym = np.meshgrid(x, x)
zm = np.cos(np.sqrt(xm**2 + ym**2))

h = plt.contourf(xm, ym, zm)

plt.axis('scaled')
plt.colorbar()

plt.show()
```