Getting Started with NumPy

Guest Lectures at TTK University of Applied Sciences, Tallinn, Estonia

8 – 12 December 2025

Andy Louwyck & Dominique Stove

Vives University of Applied Sciences, Kortrijk, Belgium

TALLINNA TEHNIKAKÕRGKOOL
TTK UNIVERSITY OF APPLIED SCIENCES

vives University of Applied Sciences

Who’s Teaching Today?

Andy Louwyck

- Master and Doctor in Science: Geology
- Associate Degree in Programming
- Micro Degree in AI & Data Science
- Lecturer in IT at Vives University of Applied Sciences
- AI Coordinator at Flanders Environment Agency

Dominique Stove

- Master in Applied Economics
- Teaching Master’s Degree in Economics, Mathematics, and Physics
- Lecturer in IT at Vives University of Applied Sciences
- IT Consultant in Infrastructure
- Founder and Business Owner of IqPro

vives University of Applied Sciences

Vives Campus in the City of Kortrijk

Informatics Program for Exchange Students

https://www.vives.be/en/commercial-sciences-business-management-and-informatics/informatics-kortrijk

The Informatics-programme is a programme consisting of lectures, group work, visits and projects in the field of Business and Informatics. Evaluation follows the rules of the European Credit Transfer System (ECTS). Incoming students can select a programme of up to 30 ECTS credits per semester.

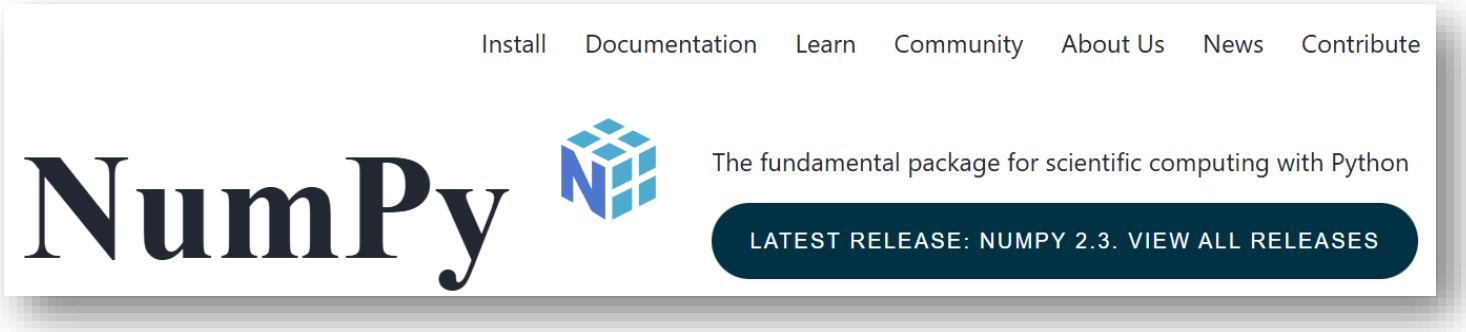
New full-year program!

Title	ECTS	hours/week S1	hours/week S2	Semester
Introduction to Artificial Intellingence	5	3	0	1
Programming in Python	3	2	0	1
Digital Workplace	3	2	0	1
Android App Development	5	3	0	1
E-business en E-marketing	3	2	0	1
Introduction to linux	3	2	0	1
Cybersecurity	5	3	0	1
Professional and International Communication 3 (English)	3	2	0	1
	30	**19**	0	
Machine Learning - Fundamentals	6	0	4	2
IT-Project	5	0	3	2
Power Tools	3	0	3	2
Full-Stack Development in .NET	6	0	4	2
Mobile App Development iOS	5	0	4	2
Data Engineering	5	0	3	2
Node.js Development	3	0	2	2
	33	0	23	

Let’s Dive In!

- What is NumPy?
- Why NumPy?
- MATLAB versus NumPy
- Demo

What is NumPy?



- NumPy = **Numerical Python**
- The fundamental **numerical computing library** for Python
- **Website**: numpy.org
- **Provides**:
  - Efficient **N-dimensional arrays**
  - **Vectorized** operations
  - Linear algebra, random numbers, FFT, and more
- The base for **scientific Python stack**: SciPy, Pandas, scikit-learn, Matplotlib, …

Why Engineers and Scientists Should Learn NumPy

- **Free** and **open-source**
- Part of the **Python ecosystem** with a huge community
- Widely used in **academia and industry**
- **Works with** AI, machine learning, simulation, optimization, data analysis, etc.
- **Easy integration with**:
    - Machine learning frameworks such as PyTorch and TensorFlow
    - Cloud computing
    - Web APIs
    - Sensors and microcontrollers
    - …

NumPy versus pure Python Performance

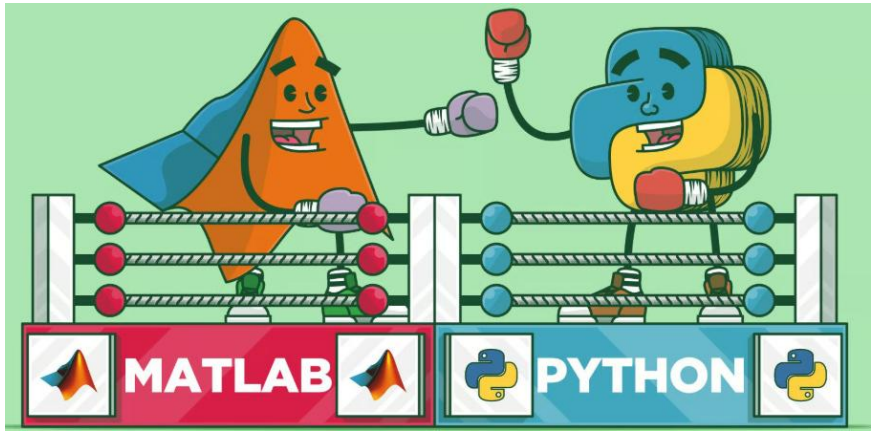
- NumPy uses **optimized C and Fortran** underneath
  (just like MATLAB)
- **Vectorization** replaces loops
  (just like in MATLAB)
- **10 to 100 times faster** for numerical tasks
  (just like MATLAB)

```python
import numpy as np
from time import time

# initialize 1000 x 1000 matrix with ones
A = np.ones((1000, 1000))

# double each element of A using nested loops
start = time()
for i in range(1000):
    for j in range(1000):
        A[i, j] = A[i, j] * 2
print(f"Time taken using loops: {time() - start:.4f} seconds")

# double each element of A using a vectorized expression
start = time()
A = A * 2
print(f"Time taken using vectorized expression: {time() - start:.4f} seconds")
```

```
Time taken using loops: 0.4709 seconds
Time taken using vectorized expression: 0.0038 seconds
```

MATLAB versus NumPy

Feature	Matlab	NumPy
Primary use	Numerical computing, engineering simulations	Within Python ecosystem
Language	Proprietary MATLAB language	Python
Cost	Paid license (expensive)	Free and open-source
Ease of use	Very user-friendly for matrix operations	Requires Python knowledge
Performance	Highly optimized for matrix and vector operations	Fast for arrays; integrates with C for speed
Toolboxes	Extensive specialized toolboxes	Base for scientific Python libraries: SciPy, Pandas, scikit-learn, …
Visualization	Built-in, high-quality plotting tools	Matplotlib and Seaborn (= separate libraries)
Community support	Strong in engineering and scientific fields	Huge global Python community
Integration	Limited outside MATLAB ecosystem	Works with ML frameworks, web apps, etc.
Portability	Runs mainly on MATLAB environment	Cross-platform, Jupyter, cloud-ready
Learning curve	Easier for engineers without coding skills	Easier for developers familiar with Python



MATLAB

- ✅ Excellent for matrix operations and linear algebra
- ✅ Rich built-in functions and engineering toolboxes
- ✅ Great visualization out-of-the-box
- ❌ Expensive license
- ❌ Proprietary language, less flexible
- ❌ Limited scalability for modern data science

NumPy

- ✅ Free and open-source
- ✅ Part of Python ecosystem (SciPy, Pandas, TensorFlow)
- ✅ Highly portable and integrates with modern tools
- ❌ Requires Python knowledge
- ❌ Visualization and specialized functions need extra libraries
- ❌ Some operations slower unless optimized

MATLAB Arrays versus NumPy Arrays

Concept	MATLAB	NumPy
Main data structure	array	ndarray
Indexing	1-based	0-based
Slicing	round brackets e.g.: `A(1:5)`	square brackets e.g.: `A[0:5]`
Matrix operations	default e.g.: `A * B`	explicit e.g.: `A @ B`
Broadcasting	implicit	more explicit rules

Creating Arrays

MATLAB

```
a = [1 2 3];
b = zeros(3,3);
c = ones(1,5);
d = 0:2:8;
e = linspace(0,1,5);
```

Python / NumPy

```
import numpy as np


a = np.array([1, 2, 3])
b = np.zeros((3, 3))
c = np.ones(5)
d = np.arange(0, 10, 2)
e = np.linspace(0, 1, 5)
```

Array Attributes

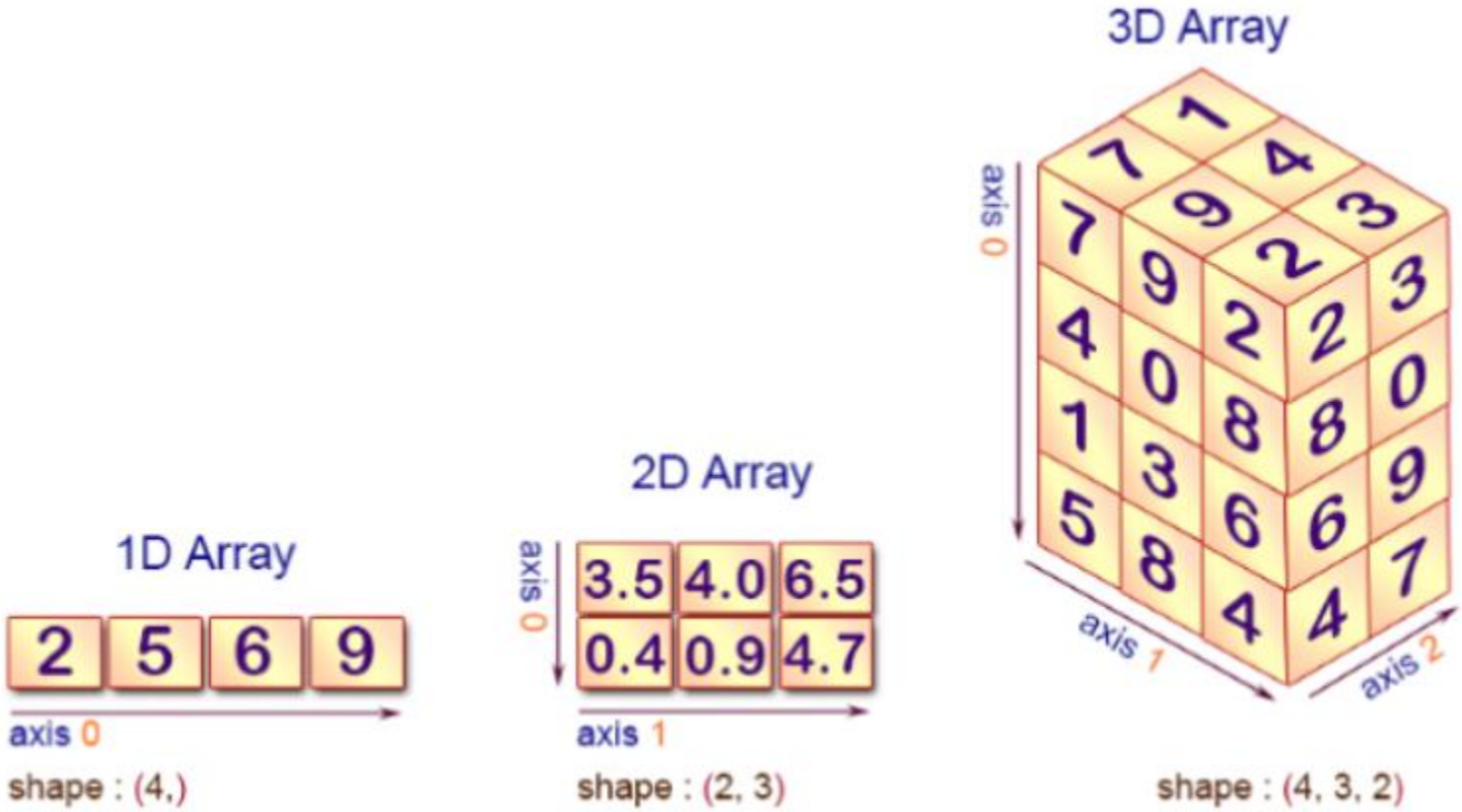
MATLAB

```
size(A)
ndims(A)
numel(A)
class(A)
```

Python / NumPy

```
A.shape
A.ndim
A.size
A.dtype
```

NumPy Axes


1D Array
2 5 6 9
axis 0
shape : (4,)
2D Array
axis 0
3.5 4.0 6.5
0.4 0.9 4.7
axis 1
shape : (2, 3)
3D Array
axis 0
axis 1
axis 2
shape : (4, 3, 2)

Indexing and Slicing

MATLAB

```
A(1,2)
A(:,2)
A(1:3,:)
A(end:-1:1)
```

Python / NumPy

```
A[0, 1]
A[:, 1]
A[0:3, :]
A[::-1]
```

Vectorized Operations

MATLAB

```
x = [0 1 2];
y = x.^2 + 3.*x + 1;
```

Python / NumPy

```
import numpy as np

x = np.array([0, 1, 2])
y = x**2 + 3*x + 1
```

Broadcasting

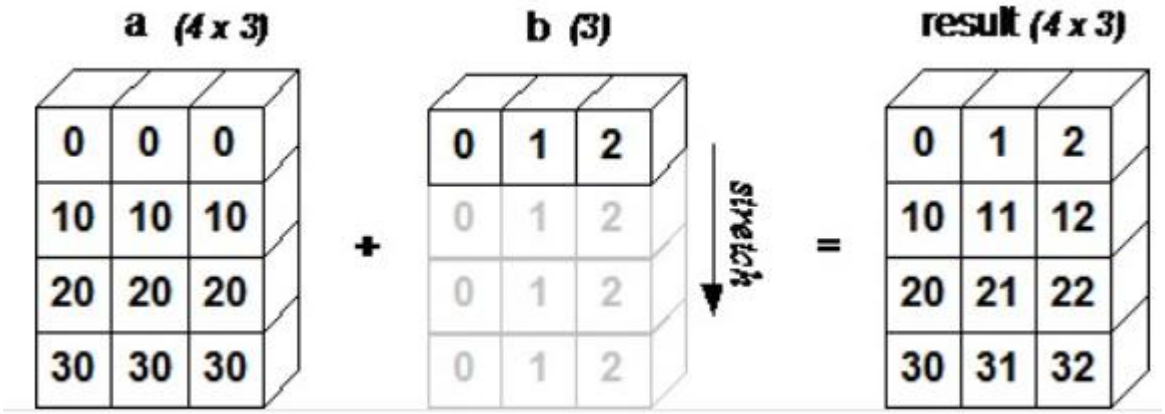
MATLAB

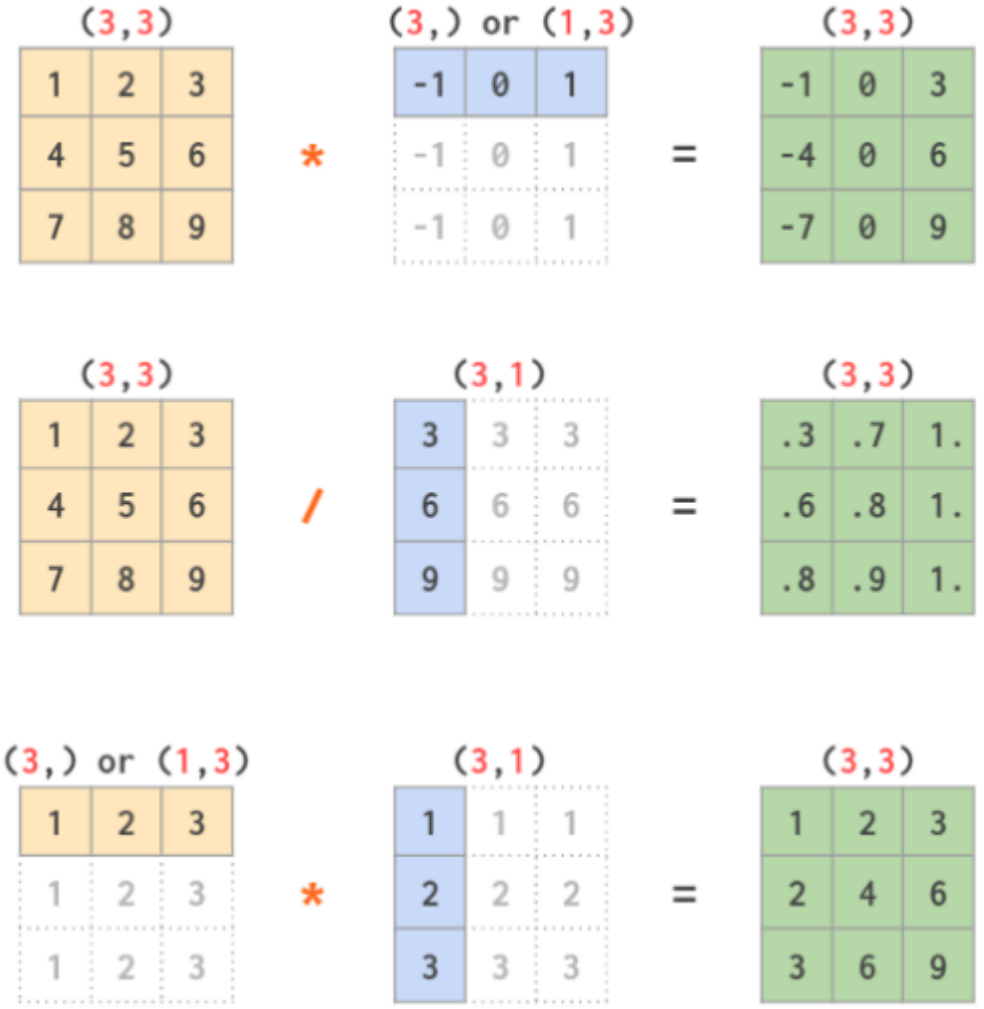
```
A = ones(3,3);
b = [1 2 3];
A + b
```

Python / NumPy

```
import numpy as np

A = np.ones((3, 3))
b = np.array([1, 2, 3])
A + b
```

Broadcasting in NumPy





TALLINNA TEHNIKAKÕRGKOOL
TTK UNIVERSITY OF APPLIED SCIENCES

vives University of Applied Sciences

Linear Algebra

MATLAB

```
A * B
inv(A)
det(A)
x = A \ b
```

Python / NumPy

```
import numpy as np


A @ B
np.linalg.inv(A)
np.linalg.det(A)
x = np.linalg.solve(A, b)
```

Random Numbers

MATLAB

```
rand(3,3)
randn(1,1000)
```

Python / NumPy

```
import numpy as np


np.random.rand(3, 3)
np.random.normal(0, 1, 1000)
```

Plotting

MATLAB

```
x = linspace(0,2*pi,200);
y = sin(x);

plot(x,y)
```

Python / NumPy

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

x = np.linspace(0, 2*np.pi, 200)
y = np.sin(x)

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

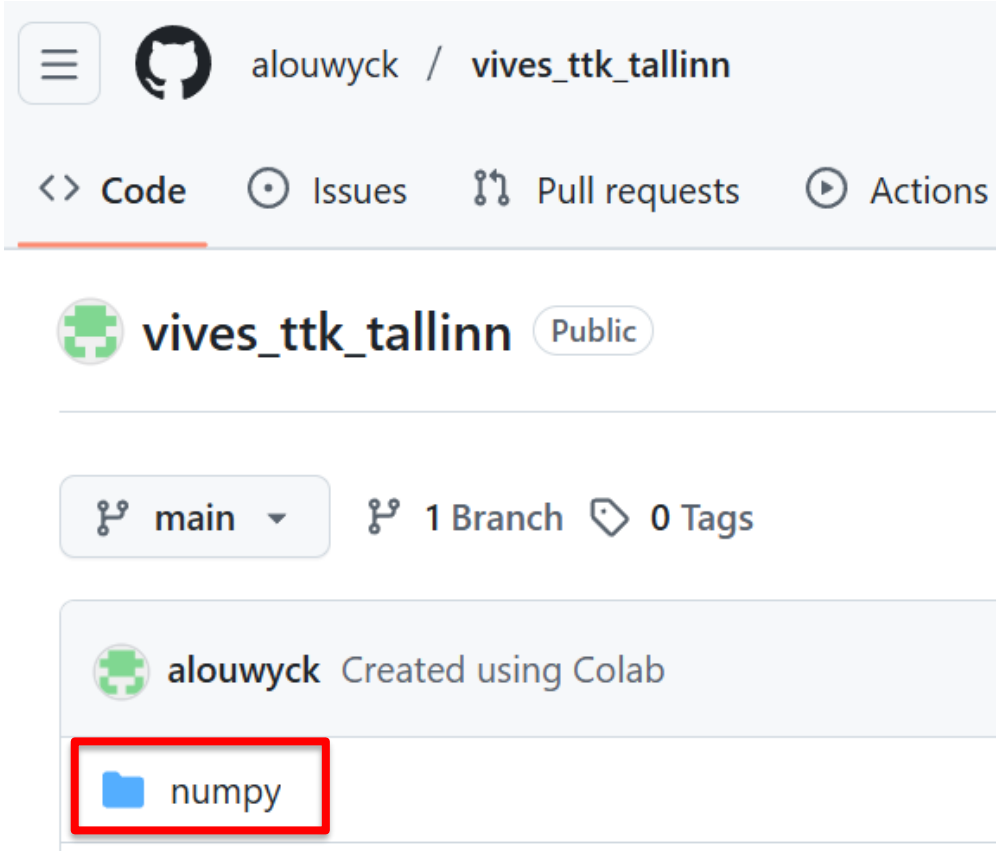
Summary

NumPy provides:

- Fast, memory-efficient numerical arrays
- MATLAB-like computations and linear algebra
- Works seamlessly with Matplotlib
- Foundation for engineering, data science, and AI workflows

GitHub Repo

https://github.com/alouwyck/vives_ttk_tallinn



TALLINNA TEHNIKAKÕRGKOOL
TTK UNIVERSITY OF APPLIED SCIENCES

vives University of Applied Sciences

Sources

- https://numpy.org/doc/stable/user/numpy-for-matlab-users.html
- https://realpython.com/matlab-vs-python/
- https://numpy.org/doc/stable/user/basics.broadcasting.html
- https://python.plainenglish.io/numpy-a-short-and-sweet-introduction-youll-actually-enjoy-1c5dc4f856a0
- https://www.legitpython.com/p/learn-numpy-from-scratch-step-by-step-part-1