

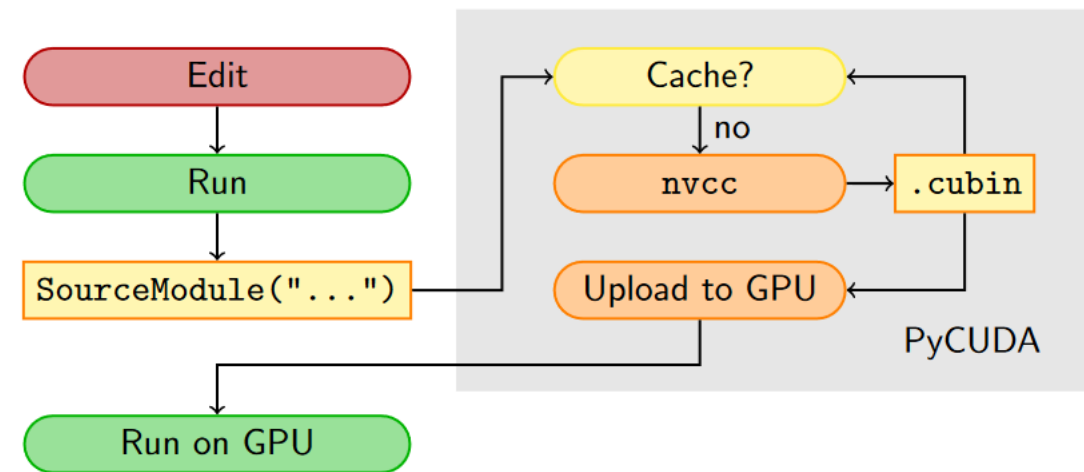
Introduction to GPU computing (3)

Computing Methods for Experimental Physics and Data Analysis
GPU & Python Hands-on : Lecture 4

gianluca.lamanna@unipi.it

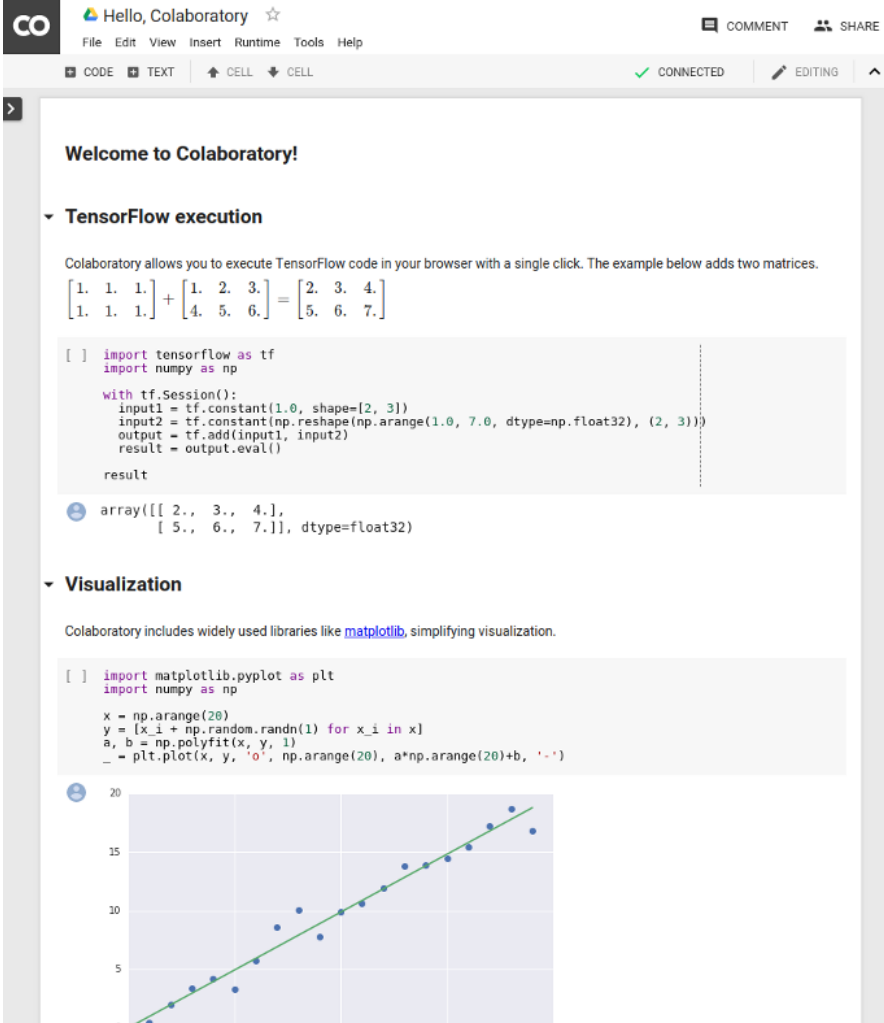
PyCuda Module & Numba

- PyCUDA lets you access Nvidia's CUDA parallel computation API from Python
 - All the CUDA features can be accessed through pyCUDA
- Supports Just-in-time compilation of the CUDA kernels in C
- Small overhead with respect to the C implementation to the GPU part
- Several additional features
 - Example: cuda exceptions translated to python exception
- One of the virtues of PyCUDA is that it allows us to use the class **GPUArray**
- We will use Numba to compile ufuncs on GPU
- <https://pypi.org/project/pycuda/>
- <https://documen.tician.de/pycuda/>
- <http://numba.pydata.org/numba-doc/latest/cuda/index.html>



Colab

- Also known as *Colaboratory*, is free *Jupyter* notebook running on Google Cloud
 - The notebooks are stored in Google Drive
 - <http://colab.research.google.com>
- The notebooks are environment to write text and run code based on Python3
 - It's possible to run on cloud computers housing GPUs
- Thanks to the IPython library it's possible to run shell commands (including compilers) on the cloud filesystem
- Possibility to add modules in the development environment



The screenshot displays the Google Colaboratory web interface. At the top, there's a header with the Colab logo, a greeting "Hello, Colaboratory", and navigation links like "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". Below this, a status bar shows "CODE", "TEXT", "CELL", and "CELL" tabs, along with "CONNECTED" and "EDITING" indicators. The main content area is titled "Welcome to Colaboratory!" and features two sections: "TensorFlow execution" and "Visualization".

TensorFlow execution

Colaboratory allows you to execute TensorFlow code in your browser with a single click. The example below adds two matrices.

$$\begin{bmatrix} 1. & 1. & 1. \\ 1. & 1. & 1. \end{bmatrix} + \begin{bmatrix} 1. & 2. & 3. \\ 4. & 5. & 6. \end{bmatrix} = \begin{bmatrix} 2. & 3. & 4. \\ 5. & 6. & 7. \end{bmatrix}$$

```
[ ] import tensorflow as tf
import numpy as np

with tf.Session():
    input1 = tf.constant(1.0, shape=[2, 3])
    input2 = tf.constant(np.reshape(np.arange(1.0, 7.0, dtype=np.float32), (2, 3)))
    output = tf.add(input1, input2)
    result = output.eval()
```

array([[2., 3., 4.],
 [5., 6., 7.]], dtype=float32)

Visualization

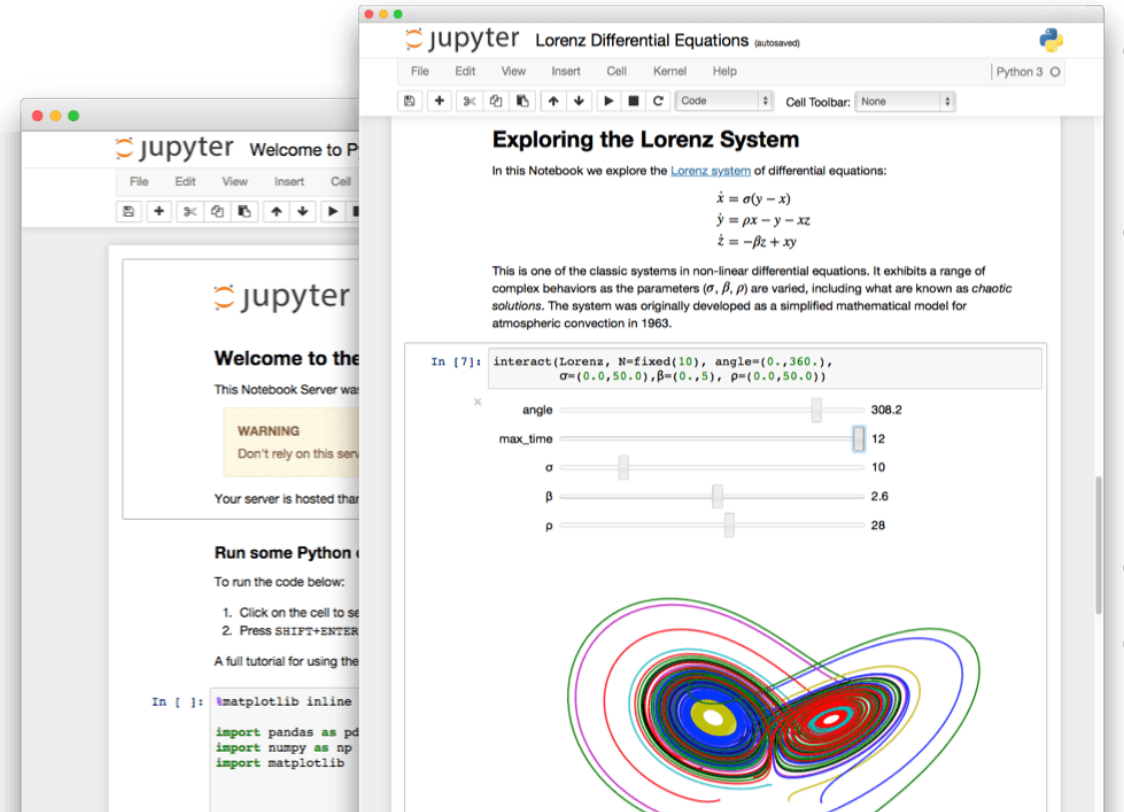
Colaboratory includes widely used libraries like `matplotlib`, simplifying visualization.

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

x = np.arange(20)
y = [x_i + np.random.randn(1) for x_i in x]
a, b = np.polyfit(x, y, 1)
_ = plt.plot(x, y, 'o', np.arange(20), a*np.arange(20)+b, '-')
```

The visualization shows a scatter plot of 20 data points (blue circles) with a linear regression line (green line) fitted to them. The x-axis ranges from 0 to 20, and the y-axis ranges from 0 to 20.

Jupyter



<https://www.geeksforgeeks.org/how-to-use-jupyter-notebook-an-ultimate-guide/>

- Colab implement a cloud version of the Jupyter notebook
→ <https://jupyter.org/>
- A Jupyter Notebook document is a JSON document
→ ordered list of input/output cells
→ can contain code, text, latex, mathematics, plots and media
→ ".ipynb" extension.
- It's free and open-source
- It implements a language shell (aka interactive toplevel) environment built on IPython library
→ IPython is command shell for interactive python
→ Jupyter is a web-based, graphics implementation of IPython
- Other programming languages (49) are supported including R, Matlab, Julia, etc.

