
HPC for numerical methods and data analysis

Fall Semester 2023

Prof. Laura Grigori

Assistant: Mariana Martínez Aguilar

Session 1 – September 10, 2024

Matrices and vectors in Python and MPI

Exercise I Matrices in Python

Find two efficient ways in Python to assign the following matrix:

$$M = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix},$$

without entering manually each element (*hint*: create two row vectors and then combine them to form a matrix).

Suitably use the Python commands to:

- a) extract the element in the first row, third column of A ;
- b) extract the entire second row of A ;
- c) extract the first two columns of A ;
- d) extract the vector containing all the elements of the second row of A except for the third element.

Modify your code accordingly to assign the following matrices as well, calculate the time it takes for Python to assign them in both your methods. Which method is better? Compare your results with one of your classmates.

$$M_1 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \\ 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \\ 19 & 20 & 21 \\ 22 & 23 & 24 \\ 25 & 26 & 27 \\ 28 & 29 & 30 \\ 31 & 32 & 33 \end{bmatrix}$$

$$M_4 = \begin{bmatrix} 1 & 2 & \dots & 250 \\ \vdots & \vdots & \ddots & \vdots \\ 124751 & 124752 & \dots & 250 * 500 \end{bmatrix}.$$

Exercise II Function in Python

We want to compute the function $f(x) = (\sqrt{1+x} - 1)/x$ for different values of x in a neighborhood of 0. We first notice that $f(x)$ can be equivalently written as $f(x) = 1/(\sqrt{1+x} + 1)$ and also as $f(x) = 1/2 - x/8 + x^2/16 - 5x^3/128 + o(x^4)$.

Create three function handles representing the above definitions of $f(x)$ (*hint*: the term $o(x^4)$ can be neglected in the computation) and, for each function handle,

- evaluate $f(x)$ at $X = [10^{-10} \ 10^{-12} \ 10^{-14} \ 10^{-16}]$ using a `for` loop;
- evaluate $f(x)$ at the same points given in a) using Python vector algebra;
- display the results and comment on the importance of *round-off errors* for this example;
- make sure you document your functions correctly;
- if $x = 0$ then your first function raises an error explaining why.

Exercise III Matrix-vector multiplication in Python

- Consider the multiplication of a matrix $A \in \mathbb{R}^{m \times n}$ with a vector $v \in \mathbb{R}^n$. Write a Python file containing a script that:
 - creates a matrix of dimension $m \times n$

- creates a vector of dimension n
- define a function that computes Av by using two nested loops

Exercise IV Matrix-vector multiplication with NumPy

a) Consider the same operation as in the previous exercise, the multiplication of a matrix $A \in \mathbb{R}^{m \times n}$ with a vector $v \in \mathbb{R}^n$. Compute matrix-vector multiplication by using numpy library:

- create a matrix of dimension $m \times n$, a vector of dimension n
- define a function that computes Av by using two nested loops
- compare the performance obtained for different values of m and n between the two nested loops code and the code using numpy library and draw a plot displaying the obtained performance

Exercise V Hello world with Python and MPI

Execute the following simple code on 4 processors several times.

```
from mpi4py import MPI
import numpy as np

comm = MPI.COMM_WORLD
rank = comm.Get_rank()
print("I am rank = ", rank )
```

To execute this code, do

```
$ mpiexec -n 4 python script.py
```

Observe the order in which the prints take place.