Quantum Information and Computing Assignment 1

Marco Chiloiro, 30/11/2023

1. Setup

The usual "hello world" program was considered as a test work in Fortran, a program that simply prints "Hello world!".

- Locally (using VS Code)

```
(base) marco@marco-Vostro-5370:~/Assignment 1$ ./hello
 Hello world!
(base) marco@marco-Vostro-5370:~/Assignment 1$
```

- Remotely on a CloudVeneto VM (using Vim)

```
ubuntu@qic-chiloiro:~/ex1$ ./hello
 Hello world!
ubuntu@qic-chiloiro:~/ex1$
```

2. Number precision

Exploration of the limits of INTEGER and REAL in Fortran.

a) 2,000,000 + 1 by using:
   i) INTEGER*2: `Error: Arithmetic overflow converting INTEGER(4) to INTEGER(2)`
   ii) INTEGER*4: 2,000,001

This is because the maximum integer that can be memorized as INTEGER*2 is 2^15 - 1 which is smaller than 2,000,000, while the maximum integer that can be memorized as INTEGER*4 is 2^31 - 1.

b) $\pi \cdot 10^{32} + \sqrt{2} \cdot 10^{21}$ by using:
   i) REAL*4: 1.41421360E+21
   ii) REAL*8: 1.4142135995874421E+021

which is what we expected from single and double precision.

3. Testing performance

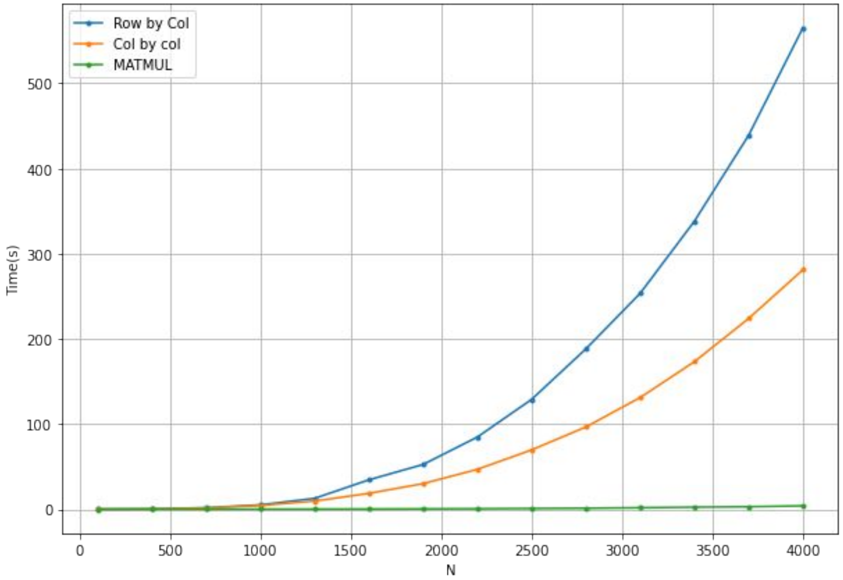
Multiplication of two random matrices of size $N \times N$ performed by:

1. row by column (the standard arithmetic method)
2. column by column (the optimal method w.r.t. the way Fortran stores matrices)
3. using the built-in function MATMUL

By increasing *N*, the execution times of the three methods are compared. This first two of the following graphs are done using the standard *gfortran* compiler optimization flag.

For the last one, different optimization flags available with the compiler (-O1, -O2 and -O3) are used.

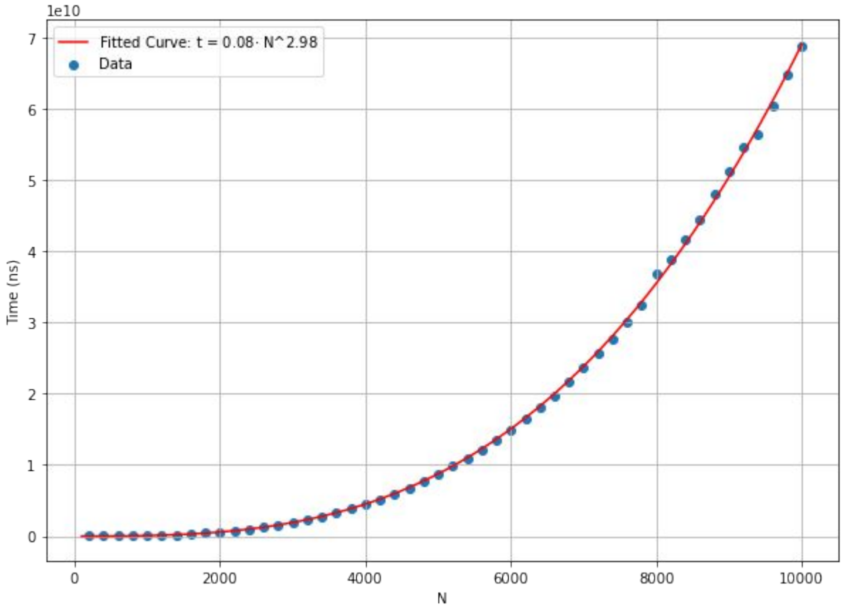
3. Results - Algorithm comparison



It is possible to see from this graph that the row by column multiplication method (blue) is slower than the column by column one (orange), as expected.

However, the built-in function MATMUL (green) is by far the fastest.

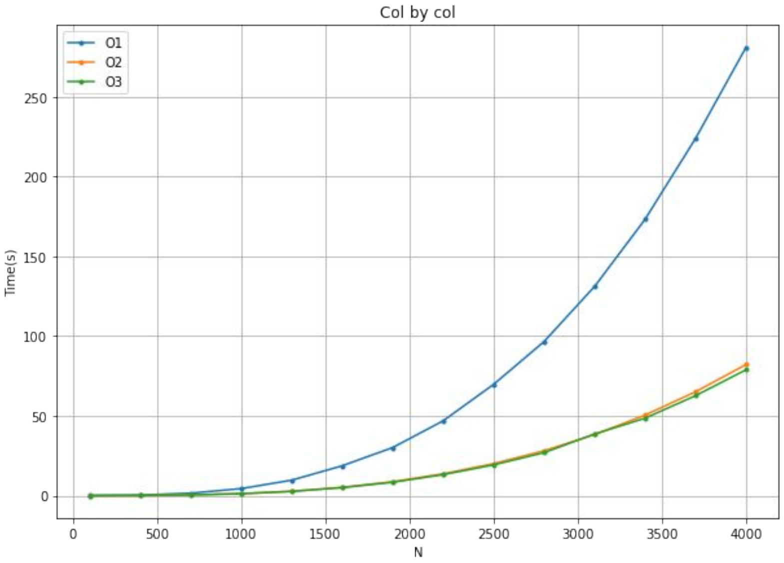
3. Results - Fit on MATMUL data

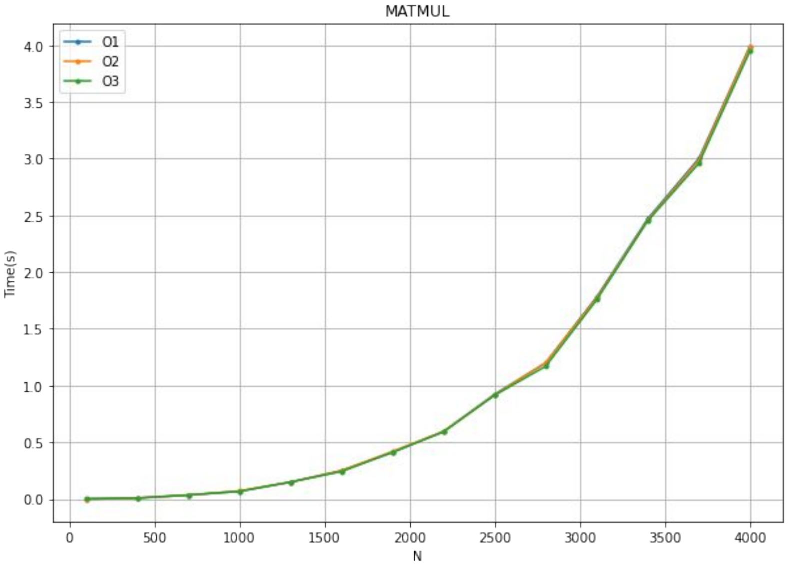


Data deriving from the use of the MATMUL algorithm for multiple values of *N* (up to 10000). By performing a parametric fit of the type *t = a*N^b,* we found that *b* is almost equal to 3, as we expected since the number of elementary operations necessary to perform a matrix multiplication is *N*^3, given *N* the dimension of the matrices.

Note that the unit of time is nanoseconds. This is because a common laptop has a clock frequency of the order of 1GHz, meaning it can perform ~1 operation per nanosecond.

3. Results - Optimization flags comparison





It is possible to see that for the column by column case (on the left), higher optimization levels lead to better performances, although there is not much difference between the -O2 and -O3 cases. On the other hand, by using MATMUL algorithm (on the right), compilation optimization does not lead to improvements in execution time.