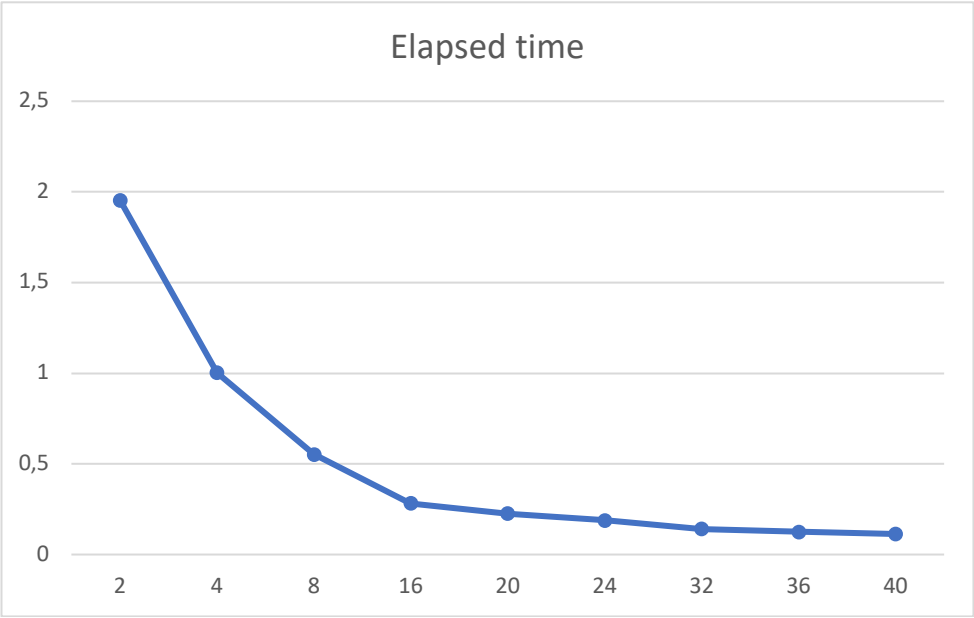
Exercise 3: MPI π approximation

Student: **Laura Balasso**

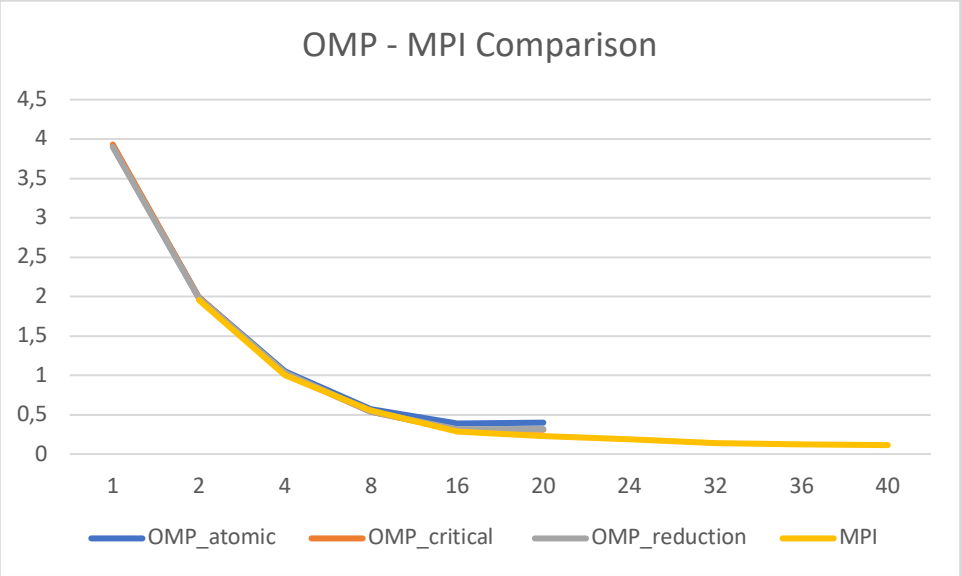
Academic year: 2018/19

The aim of this exercise is to compute the π approximation with the midpoint method (see exercise 1) using MPI.
I followed the same procedure used in exercise 1, this time using the process id to split the domain equally among the processes. Then, as requested, I reduced the global result in the last process with `MPI_Reduce`, transferred the result in process 0 and printed it.
I measured the execution time for different number of processes, to compare it with the OpenMP version.
The following table shows the elapsed time in seconds obtained as an average of five runs with $N = 10^8$.

Processes	**Time**
2	1.955263
4	1.003177
8	0.553079
16	0.282796
20	0.226137
24	0.188622
32	0.141480
36	0.125704
40	0.113300



The program scales up well. Below we can see a comparison with the OpenMP approach.



The two versions of the program are similar from the point of view of the runtime but the MPI code seems to perform better with a large number of processors (16 and 20). Moreover, since with MPI we can work on distributed memory, the MPI version is able to exploit a larger number of cores while OMP is limited to the maximum number of cores per node.