INSTITUTO SUPERIOR TÉCNICO

ELECTRICAL AND COMPUTER ENGINEERING

PARALLEL AND DISTRIBUTED COMPUTING

2ND SEMESTER 2018/2019

PARTICLE SIMULATION

SERIAL + OMP

84037 – Eduardo Costa
84038 – Eduardo Melo
84087 – João Sebastião

Group: 25
Shift: Wednesday, 12:30
Professor: Nuno Roma

April 5th 2019

1 Program Structure

The program structure is presented in figure 1.

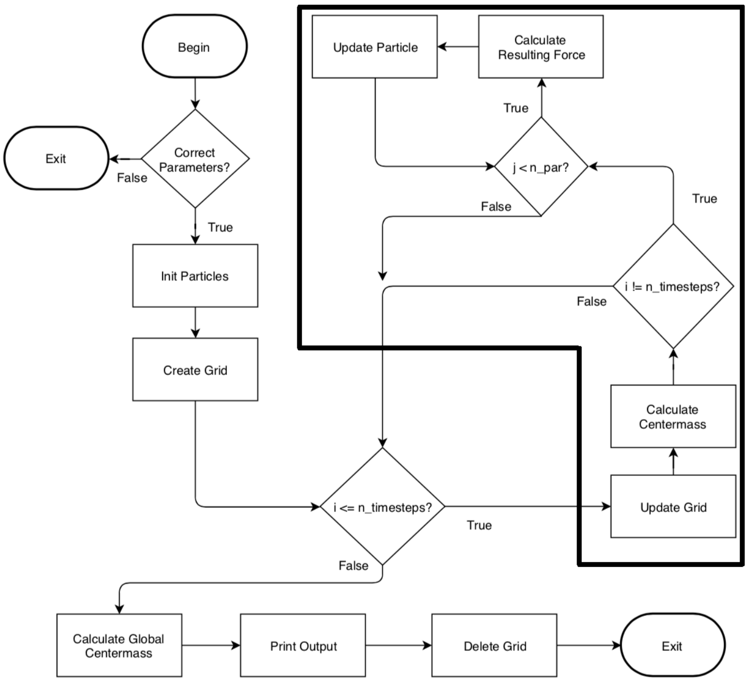

Figure 1: Fluxogram of the program structure.

Considering there is a certain cost associated with parallelization, only a few sections of the code were parallelized while the others remain in serial, since the cost-to-performance ratio with parallelization was higher than the serial execution.

The parallelized section of the program is presented inside the boxed area in the fluxogram above. This section only runs in parallel if the number of particles exceeds 1000. There is only one region in which synchronization becomes a problem, which is inside the *Update Grid* section. For this reason, the atomic clause was used in order to ensure the correct execution of the program. The workload is distributed evenly among the threads to achieve the best possible performance.

2 Performance Results

In what comes to performance, several tests were performed to first analyze the influence in the increase of the number of threads and, afterwards, the number of particles. Regarding the first, 4 tests were analyzed, with different grid size, time steps and number of particles while varying the number of threads. Regarding the second, 2 tests were analyzed, with different grid sizes and time steps and both running with 4 threads while varying the number of particles. The results obtained for the first and second analysis are presented in the figures 2 and 3 respectively.

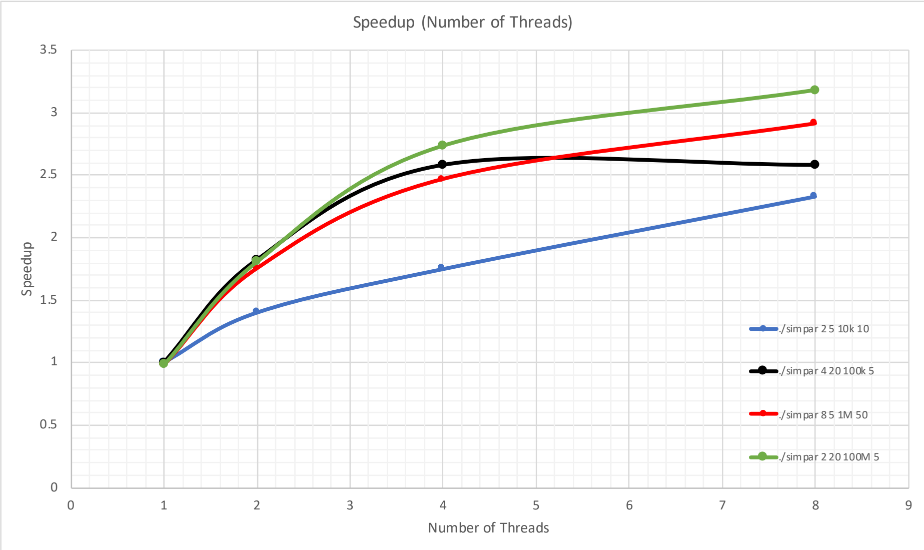

Figure 2: Speedup obtained on each test for different number of threads used.

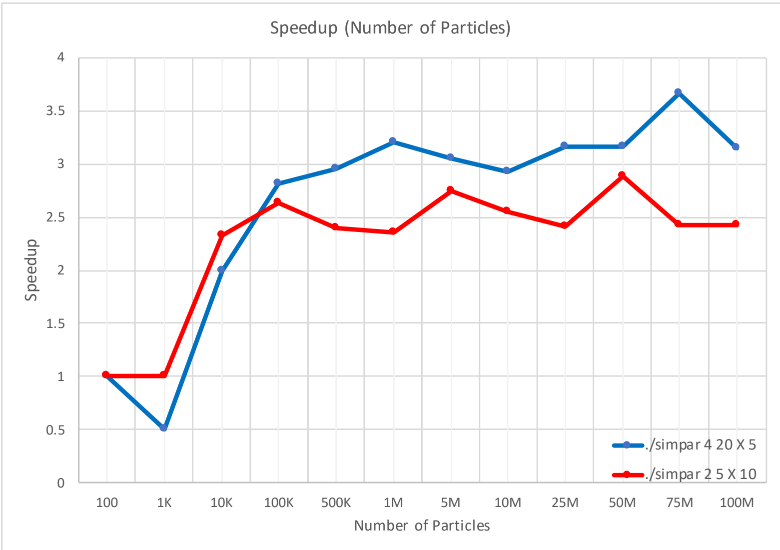

Figure 3: Speedup obtained in 2 tests using 4 threads for different number of particles.

Analyzing figures 2 and 3, it is possible to conclude that, while increasing the number of threads used, the speedup for every test gets increasingly more distant from the maximum speedup ($speedup = p$), and that with the increase in the number of particles, the speedup initially increases, although increasing at a slower rate, showing values of speedup around approximately 3.12 for the first test and 2.52 for the second test. Both of these results can be justified by the increased overhead in the management of the threads and synchronization, which is necessary to guarantee the correct execution at the cost of performance since threads can block each other, wasting time and not doing active work. Moreover, in the second case, as the number of particles increases to very high numbers, synchronization becomes the main overhead that, even with divided workload, the number of times that threads block each other is also very high, leading to approximately the same speedup.