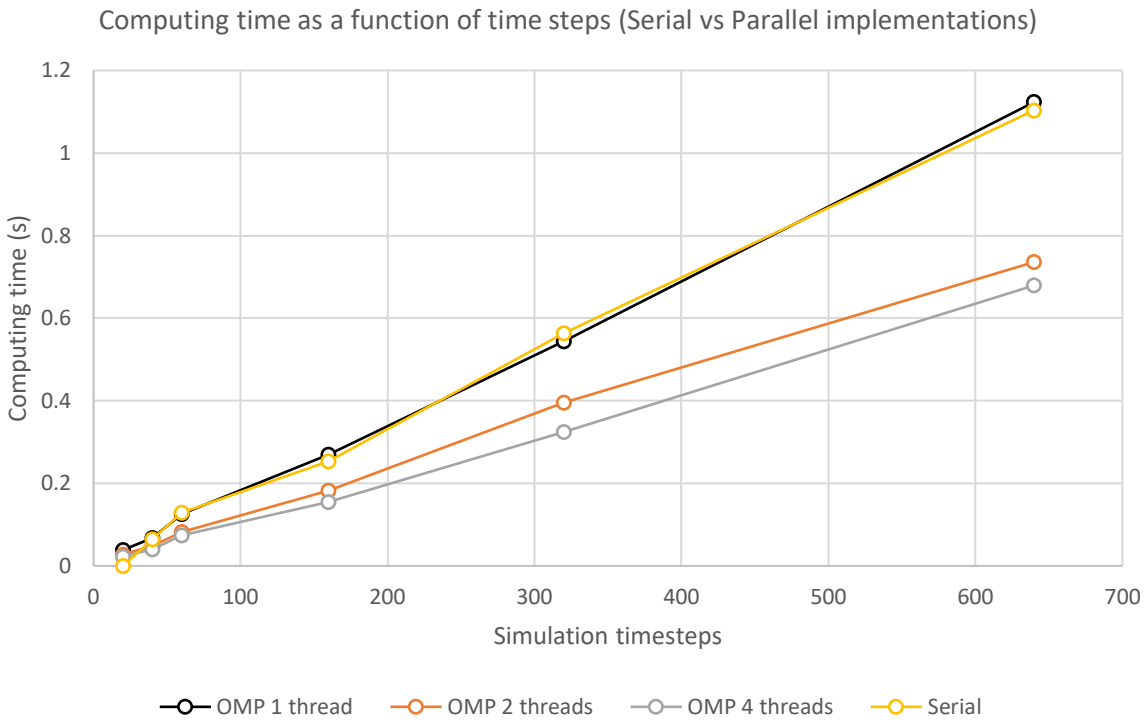
Parallelisation (using OMP on 1, 2 and 4 threads)

The performance of the code is studied by measuring its timings on a system with 2 cores and 4 threads that runs on Mac OS Catalina. The code has been executed on a single machine in serial and in parallel (using 1, 2 and 4 parallel threads). The computing time has been measured as a function of the number of time steps. Time intervals between 20 and 640 has shown the expected trend in parallel speed-up.

The loops to be parallelised have been carefully selected. Loops that might lead to race conditions or that create an overhead that results in a slowdown of the program have been ruled out. The loops that have been parallelised are those used for smoothing the density, calculating the derivatives, and updating the velocity, position, pressure and density of the particles.



For a low number of iterations, the serial and parallel versions result in similar computing times. The plot above, however, shows how parallelisation through OMP enhances the scalability of the program. For a high number of iterations, the parallelised code (running in 2 and 4 parallel threads) results in a reduction of the computing time of around 40%.

The correctness of the results after parallelisation has been ensured by visualising them for a long time interval on Paraview. Further work would involve implementing domain decomposition on MPI.