

## Parallel computing with LMGC90

Part of the developments of LMGC90 are currently dealing with the optimization of tasks using parallel computing. However, given the complexity of the code and the structure of the Contact Dynamics Method (i.e., the algorithm behinds LMGC90), not all can be coded for multithreading processing.

Between the most time-consuming operations the code runs, we find on top: 1) the solver of interactions (the resolution of contact forces and particles' velocities), and 2) the contact detection. The first is parallelized but not the second.

Other parts of the code, such as the reading/writing of files, the preparation of the simulation (this is the computation of rigid body properties, external forces) are not parallelized either.

Now, how many processors is it worth using with LMGC90? Bellow, you find a figure from the Ph.D. dissertation of Mathieu Renouf (*Optimisation numérique et calcul parallèle pour l'étude de milieux divisés bi- et tridimensionnels*, 2004) presenting the speed-up (SP) of LMGC90 as a function of the number of processors (P). The different lines correspond to different data structures the code can use. For the moment, the most stable and that we usually use is the SDL ("Stored Delassus Loops"). As you can see, the speed-up quickly reaches a plateau around 4. That means that, at most, we can divide the computing time by 4 times when using around 16 processors. It is probably not worth using 16 processors for just a speed-up of 4 when using 4 processors already splits the computing time by almost 2. So, I recommend not using more than 6 processors for your simulations when you are running parallel jobs with LMGC90.

$$y = mx + b \tag{1}$$