Short Course on Parallel Computing – Exercises

Peter Bastian, Olaf Ippisch, Jorrit Fahlke, Panos Drouvelis

Exercise 6 GNBP with Cuda

`nbody_cuda.cu` contains a version of the GNBP-program that can take advantage of the processing power of Nvidia GPUs. Cuda is a C-like language, but Cuda programs contain both code that runs on the CPU and other code that run on the GPU. Please make yourself familiar with the code and play around with it. Maybe you can even make it more efficient?

Note: Since most Nvidia GPU only support single precision arithmetic `nbody_cuda.cu` has been written to use `float`s. That means that the softening parameter `epsilon2` had to be adjusted, since the one from `nbody_vanilla.c` would have no effect if added to a `float` value near `1.0`. As a result, `nbody_vanilla.c` cannot be used to check the results of `nbody_cuda.cu`.

Luckily `nbody_cuda.cu` contains two flavors of the `acceleration()` function: `acceleration_gpu()` which uses the GPU for computation and `acceleration_cpu()` which uses the CPU. Typing `make` will generate two binaries `nbody_cuda` and `nbody_cuda_cpu`, using the respective `acceleration()` functions. The resulting `.vtk` files from these binaries should contain the same numbers, except for rounding errors.