HPC for Smoothed Particle Hydrodynamics using MultiGPU implementation

J.M. Domínguez, A.J.C. Crespo*, A. Barreiro, M. Gómez-Gesteira

[*] EPHYSLAB, Universidade de Vigo, Campus As Lagoas s/n
32004, Ourense, Spain
e-mail: alexbexe@uvigo.es

ABSTRACT

The different SPH codes developed within the SPH community are reaching a sophisticated state of maturity in terms of improved formulations and validated calibrations for specific applications. However, one of the main drawbacks of the SPH method is still its high computational cost when real engineering problems must be studied using a huge number of particles. A parallel SPH scheme using heterogeneous clusters of Central Processing Units (CPUs) and Graphics Processing Units (GPUs) has been developed to make the SPH codes more attractive to compete with the other commercial CFD software and have been shown to be an accessible choice for research groups and companies.
Apart from the CUDA implementation to use one GPU card, a Message Passing Interface (MPI) is also used to communicate different CPU devices that can host several GPUs. Communication among devices uses a MPI implementation which addresses some of the well-known drawbacks of MPI such as including a dynamic load balancing and overlapping data communications with computation tasks. Results of efficiency were analysed to prove the capability of the multi-GPU implementation if real applications that require high resolutions must be performed at reasonable computational times [1]. Hence, the multi-GPU implementation has shown an efficiency close to 100% using 128 GPUs of the Barcelona Supercomputing Center, when 8 million particles per GPU have been simulated. Moreover, an application with more than 10^9 particles is presented to show the capability of the code to handle simulations that would require large CPU clusters or supercomputers otherwise.

The use of several GPUs has led to an important increase in the size of the simulations and new problems of precision mainly appear when the domain is huge in comparison to the distance of interaction between particles. Two solutions have been implemented with particle interaction and system update in double precision; "*PosDouble*" uses double precision only for position variables and "*PosCell*" maintains single precision for position, but also storing the relative position of the cell that particle belongs to. When using *PosDouble*, 20% of loss of efficiency is achieved with the Tesla K20 and using *PosCell* less than 8% of loss is achieved in Tesla K20

The issue of double precision is crucial for real engineering problems such as wave propagation in large domains or the study of runoff on realistic terrains where large systems are simulated using high resolution (see Figure 1).

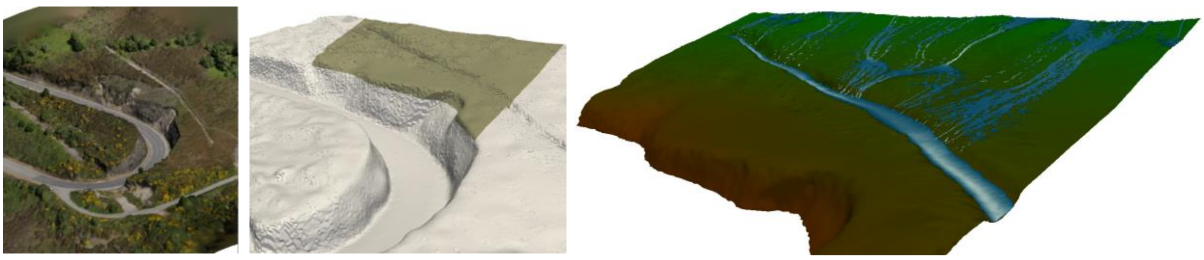

Figure 1. Study of runoff problems due to the intense rain on a terrain.

REFERENCES

[1] J.M. Domínguez, A.J.C. Crespo, D. Valdez-Balderas, B. Rogers and M. Gómez-Gesteira, "New multi-GPU implementation for SPH on heterogeneous clusters", *Computer Physics Communications*, **184**, 1848-1860 (2013).