High performance computing 3D SPH model: Sphere impacting the free-surface of water

Pierre Maruzewski
Ecole Polytechnique fédérale de Lausanne
EPFL – STI - LMH
Lausanne, Switzerland
pierre.maruzewski@epfl.ch

Guillaume Oger
HydrOcean
Nantes, France
guillaume.oger@hydrocean.fr

David Le Touzé
Ecole Centrale de Nantes
ECN
Nantes, France
david.letouze@ec-nantes.fr

John Biddisccombe
Swiss National Supercomputing Centre
CSCS
Manno, Switzerland
john.biddiscombe@cscs.ch

***Abstract*—In this work, an analysis based on a three-dimensional parallelized SPH model developed by ECN and applied to free surface impact simulations is presented. The aim of this work is to show that SPH simulations can be performed on huge computer as EPFL IBM Blue Gene/L with 8'192 cores. This paper presents improvements concerning namely the memory consumption, which remains quite subtle because of the variable-H scheme constraints. These improvements have made possible the simulation of test cases involving hundreds of millions particles computed by using more than thousand cores. This is illustrated in the paper on a water entry problem, namely on a test case involving a sphere impacting the free surface at high velocity. In the first part, a scalability study using from 124'517 particles to 124'105'571 particles is realized. In the second part, a complete simulation of a sphere impacting the free surface of water is done with 1'235'279 particles. A convergence study is achieved on pressure signals recorded by probes located on the sphere surface. Furthermore, ParaView-meshless developed by CSCS, is used to show the pressure field and the effect of impact.**

I. INTRODUCTION

Smoothed Particle Hydrodynamic method is well suited to model complex free surface problems of fast dynamics such as the water entry of 3D objects of complex geometry. With respect to classical Level-Set or VOF interface tracking, the free surface remains always precisely described by the particles in their Lagrangian motion with no need for adapting a mesh. Presently, most of the models present in the free surface SPH related literature are two-dimensional, and thus do not really suffer from high computational cost difficulties. But actual engineering applications are indeed three-dimensional, which dramatically increases the computational cost, and finally limits the use of the SPH method. Millions of particles and an efficient parallelization are then required. In this context, being an explicit method, SPH presents the other asset of a rather straightforward parallelization with respect to classical Finite Difference Method, Finite Volume Method or Finite Element Method used in Computational Fluid Dynamics. Ecole Centrale de Nantes decided to develop a parallelized model. This latter is tested on the Cray XD1 cluster with 32 cores [1]. Furthermore, Sbalzarini has developed a highly efficient parallel particle-mesh library which provides a computational tool for mesh-based method [2]. Finally, Ecole Polytechnique Fédérale de Lausanne has shown that SPH parallel simulation is available on High Performance Computing machine like an IBM Blue Gene/L with 8'192 cores [3].

II. SPH SOLVER

A. *Method*

The SoPHy-N model [1], developed by Ecole Centrale de Nantes, relies on renormalization kernel method and implementation of an exact Riemann solver with Godunov numerical scheme as described by Vila et al [4]. The latter presents various advantages such as avoiding artificial viscosity required in standard SPH, decreasing numerical dissipations and increasing stability, in a way inspired from compressible finite-difference and finite-volume schemes. Boundary conditions are imposed using ghost particles.

Specific developments have been realized to extend this technique from a flat boundary to an arbitrary 3D shape. Furthermore, SoPHy-N also integrates a variable smoothing length scheme. This technique is well adapted to the test case of water entry of body, providing accuracy in the impact area. The spatial distribution of particles is then slowly relaxed from the limit of this zone up to the body border.

B. *Local pressure capture*

A lot of general fluid problems require the knowledge of local pressure on solid boundaries. Note that the correct evaluation of these pressures through SPH formalism still remains a quite subtle task, and is unusually discussed in the SPH related literature. The procedure retained in this paper is based on a specific treatment related on the sampling of near boundary particles around a given point M where the pressure is determined.

This procedure consists in sampling the SPH particles in the near boundary area, within a distance d from the boundary that is proportional to the smoothing length h, and within the width S_{sensor} of the sensor, Figure 1. This area is an image of a pressure distribution seen by a pressure sensor during an experimental measurement. The local pressure at M is finally defined as the mean of the sampled particle pressures. As a consequence, the approximation tends towards the exact pressure at point M as the sensor surface narrows. Thus, some numerous particles contribute in the estimation of the boundary pressure at M, and give a correct approach of the mean pressure at this point.

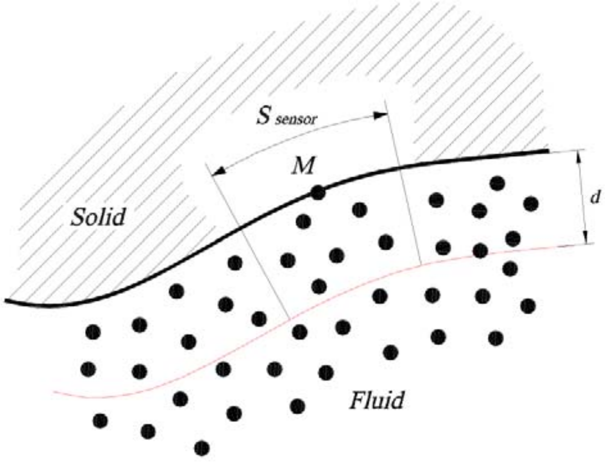

Figure 1. Particle sampling around *M*.

C. *Parallelization description*

For computational cost saving and memory requirement reasons, the practical implementation of three dimensional SoPHy-N code implies its parallelization. This parallelization is achieved using the standard Message Passing Interface, MPI, library for inter-process communications with non-blocking communications.

The domain decomposition as parallelization strategy consists in splitting the whole fluid domain into sub-domains. Each sub-domain is allocated to one core at the beginning of the calculation. Its size is adapted during the simulation. So by this way, each core has approximately the same number of particles to treat, in order to make the global calculation as efficient as possible. Let us consider a given core of interest. This core owns a large number of particles for which it is able to compute the interactions everywhere except near its limits. Indeed, its neighbor cores contain the missing particles, that we call now “foreign particles”. Its neighbor cores must therefore communicate to it the foreign particle data as position, velocity, pressure in order to allow the core of interest to complete its calculation, namely to account for all of the neighbors of its own particles. Thus, the core of interest has first to communicate to its neighbor cores the limits of the areas.

Furthermore, a dedicated algorithm ensures to get balanced loads afterwards. Rectangular sub-domains of optimized shape are adopted to avoid complicated distribution of particles.

III. IMPACT OF A BILLARD BALL

A. *Case definition*

The academic case of a billard ball impacting the water free surface described by Laverty [5] is used for smoothing length analysis, scalability performances and local flow analysis, Figure 2. The Figure 3. represents the numerical domain with 2 sub-domains: a rigid body, the billard ball, and a half spherical tank of water.

TABLE I. GEOMETRIC AND FLOW CONDITIONS

Tank of water	
Diameter d_w	6 m
Density ρ_w	1’000 kg.m^{-3}
Billard ball	
Diameter d_s	2 m
Density ρ_s	1 kg.m^{-3}
Impact velocity V_s	6 m.s^{-1}
Flow field properties	
Reynolds number $Re = \dfrac{\rho_w V_s d_s}{\mu_w}$	1.18 10^{7}
Froude number $Fr = \dfrac{V_s}{\sqrt{g d_s}}$	1.354

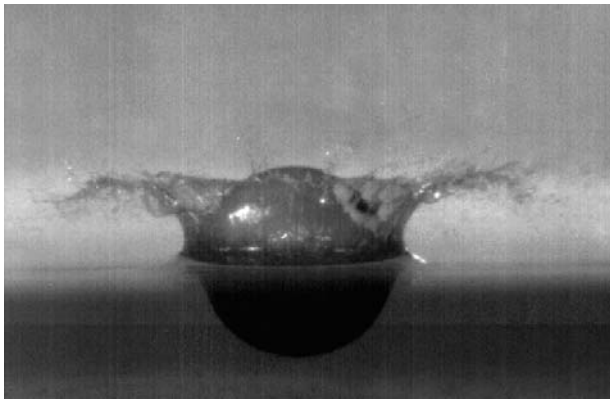

Figure 2. Billard ball impacting the water free surface.

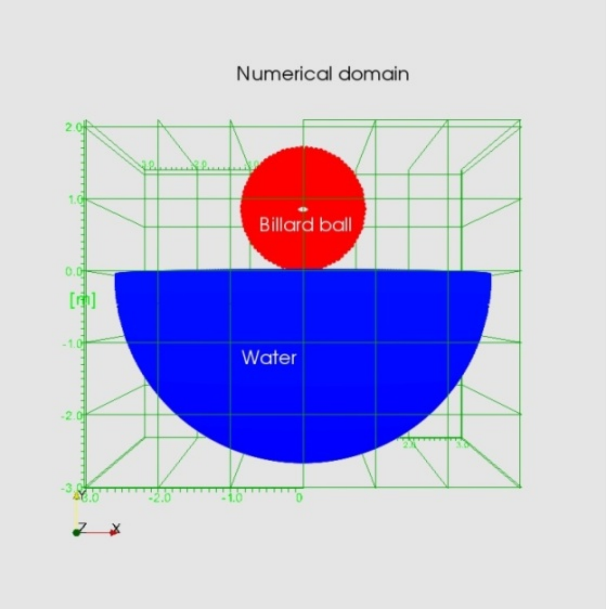

Figure 3. SPH sub-domains.

B. *Smoothing length impact*

The smoothing length *h* is the essential SPH parameter to achieve a good accuracy of results for both defining the influence area of the smoothing kernel function and for drawing a realistic analysis. Three different values of smoothing length *h* are used for evaluating the space convergence and the number of particles, TABLE II.

TABLE II. NUMBER OF PARTICLES

Smoothing length *h* [m]	Number of particles
0.09	82'161
0.0625	197'193
0.025	1'235'279

The water free surface during the impact of the billard ball is represented for these three smoothing lengths from Figure 4. to Figure 6. At high smoothing length, the wet surface does not cover the bottom half of the billard ball. Furthermore, the splashing is not absolutely developed. However, for fine smoothing length, the wet surface and the splash seem realistic.

Result obtains with a smoothing length *h* of 0.025 m on Figure 6 are in good agreement with Laverty's experimental results shown Figure 2 and with Richardson's measurement in literature [6].

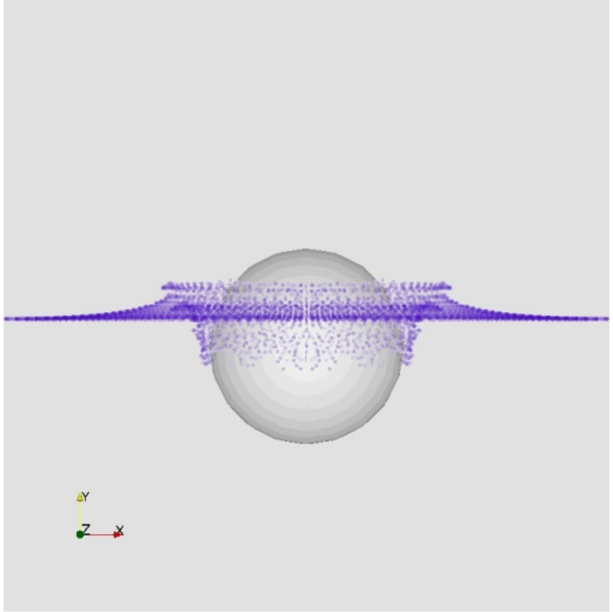

Figure 4. Smoothing length *h* = 0.09 m.

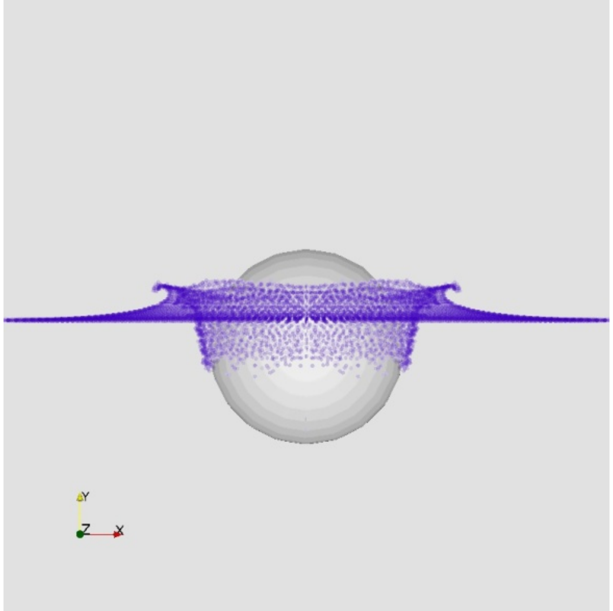

Figure 5. Smoothing length *h* = 0.0625 m

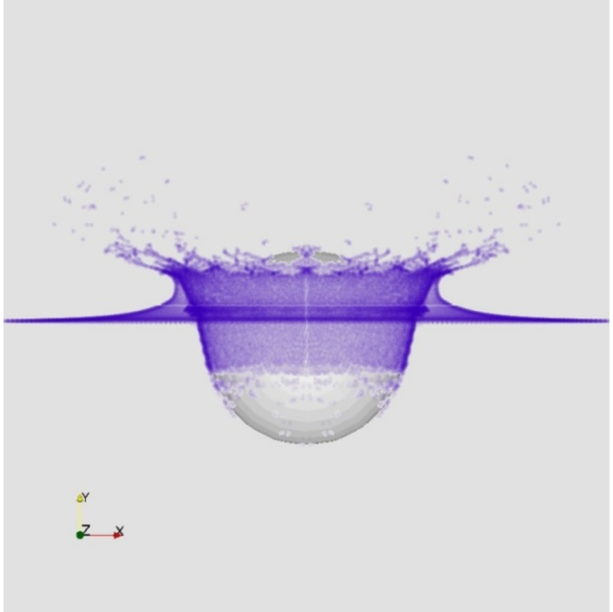

Figure 6. Smoothing length *h* = 0.025 m

C. *High performance computing speedup and efficiency*

Performance of SoPHy-N code is tested for both fixed-size and a scaled-size problem by decreasing the smoothing length *h*. In the fixed-size problems, the number of particles is kept constant. In fact, the work load per core decreases with number of cores. In the scaled-size problem, particle numbers grow proportionally to the number of cores, resulting in a constant load per core. Timing and parallel efficiency figures are collected on the IBM Blue Gene/L computer of EPFL. The 8'192 cores machine consists of 8 mid-planes packed in 4 racks. Each mid-plane has 512 dual-core computing nodes. Each core is a PowerPC 440 with 512 MB of memory.

For different smoothing length cases, we measure the elapsed wall-clock time *t* and compute speedup *S* and efficiency η by:

$$S(N_{proc}) = \frac{t(p)}{t(N_{proc})} \frac{N(N_{poc})}{N(p)}$$

$$\eta(N_{proc}) = \frac{S(N_{proc})}{N_{proc}}$$

Where *t(p)* is the time on the *p* minimal number of cores, linearly extrapolated if not measured, $t(N_{proc})$ is the time on N_{proc} cores, $N(p)$ is the problem size on *p* cores and $N(N_{proc})$ is the problem size on N_{proc} cores.

The speedup and parallel efficiency of SoPHy-N on IBM Blue Gene/L are plotted on Figure 7. and Figure 8. for the fixed and scaled-size problems, respectively. We are tested 11 fixed-size cases starting from *h* = 0.076 m for 124'517 particles to *h* = 0.00158 m for 124'105'571 particles. The largest case with about 124 million particles allows a parallel efficiency of 58% on 2'048 cores. The case with 7 and half million particles achieves a parallel efficiency of 72% on 128 cores. And finally, the case with 3 and half million particles reaches a parallel efficiency of 87% on 64 cores.

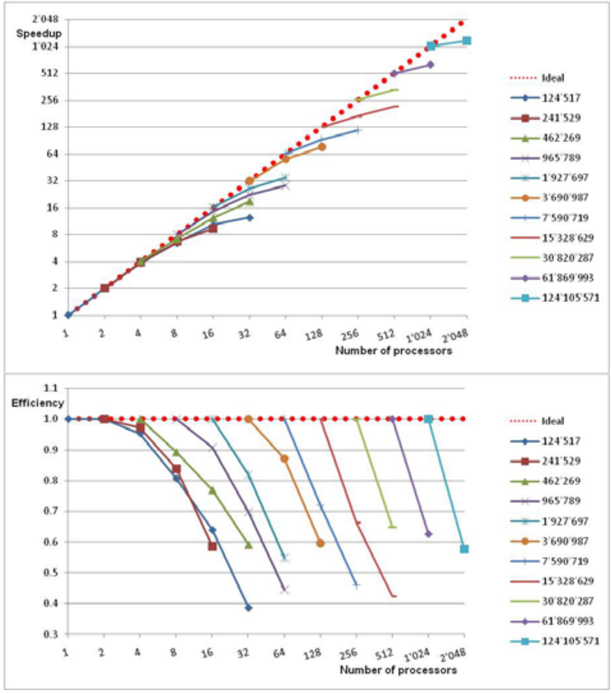

Figure 7. Parallel speedup and efficiency of SoPHy-N on IBM Blue Gene/L for different fixed-size problems starting with 124 thousands particles to 124 million particles.

For the scaled-size problem, we start from a high smoothing length *h* = 0.076 m and we decrease it by increasing by twice the number of particles between two consecutive smoothing lengths, Figure 8. We fixed the constant load to about 120'000 particles per core. It is the maximum load available for this HPC machine. The last case is obtained with 124 million particles on 1'024 cores. Unfortunately, we cannot test more than this last case due to the source code and the low memory per core. We noticed that SoPHy-N has a high parallel efficiency until 8 cores for the maximum constant load. Finally, performing a simulation with 120'000 particles per core from 16 to 1'024 cores seems to be equivalent as a simulation with only 16 cores. So, we are face to a bottle neck between load balancing, cores communications and low memory per core. The existing parallel implementation should be modified for the Blue Gene architecture if we want to perform a simulation for billion particles on 8'192 cores.

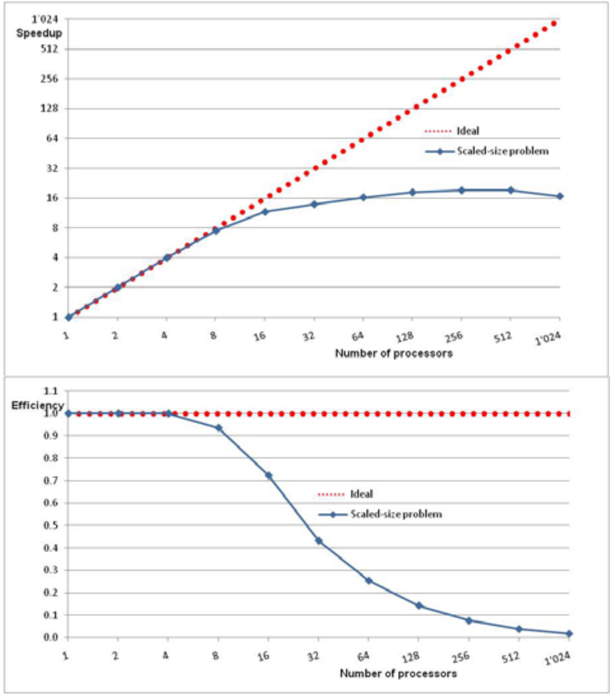

Figure 8. Parallel speedup and efficiency of SoPHy-N on BGL for scaled-size problem with 120 thousands particles per core.

D. *Local flow analysis*

The result of the billard ball impacting the water free surface with a smoothing length h = 0.025 m for 1'235'279 particles is shown from Figure 9. at t = 0 s to Figure 14. at t = 0.4 s. All visualization figures are performed with the open source ParaView-Meshless software developed at CSCS [7].

At t = 0 s, Figure 9. , the water reaches an hydrostatic state and the billard ball starts passing vertically downward with a constant velocity of 6 m.s^{-1}.

At t = 0.014 s, the billard ball touches the water free surface as shown by the red high pressure spot at the bottom of the ball on Figure 10. Furthermore, if we plot the time dependent local pressure profile along 4 probes located at the bottom surface of the ball, we notice a first high pressure peak at t = 0.014 s on Figure 15.

We observe the downward pressure wave propagation on red color from the water free surface to the bottom of the tank on Figure 11. Then the pressure wave reaches the bottom at t = 0.084 s on Figure 12.

At t = 0.132 s, the back pressure wave reaches the bottom of the billard ball on Figure 13. From the dependent local pressure profile on Figure 15. , we observe the second high pressure peak at this current time.

At t = 0.4 s, end of the simulation, the billard ball reaches the bottom of the tank. The water free surface develops a water splash.

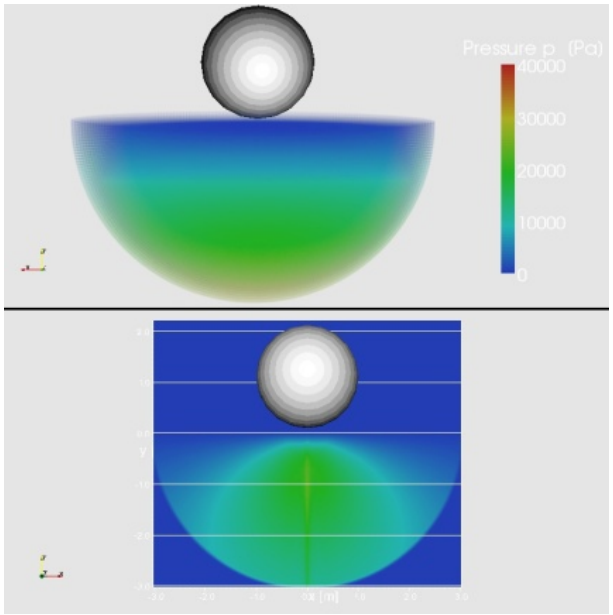

Figure 9. Billard ball impact at t = 0 s

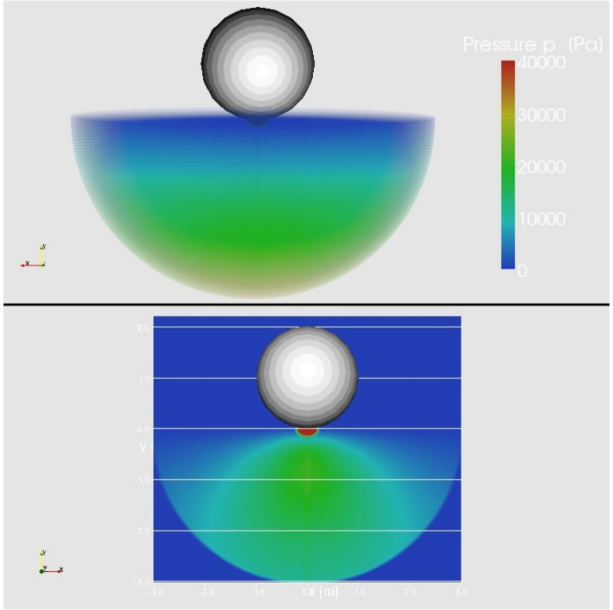

Figure 10. Billard ball impact at t = 0.014 s

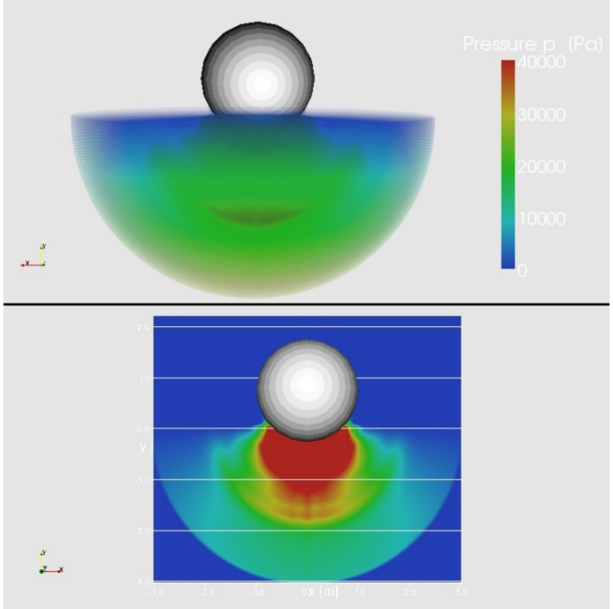

Figure 11. Billard ball impact at *t* = 0.06 s.

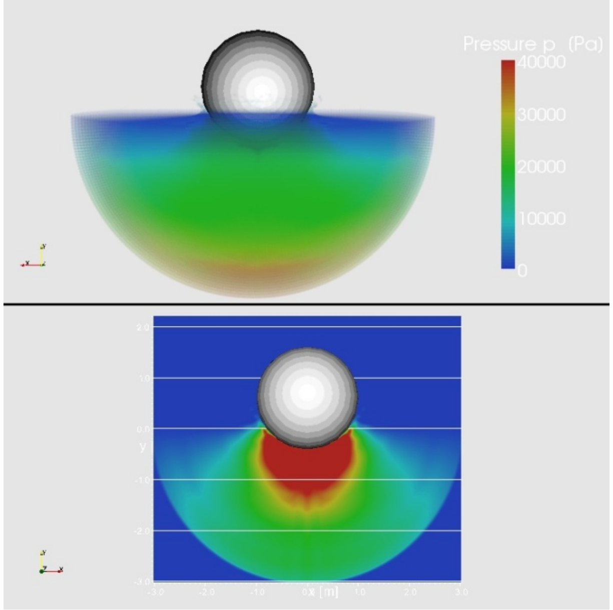

Figure 12. Billard ball impact at *t* = 0.084 s.

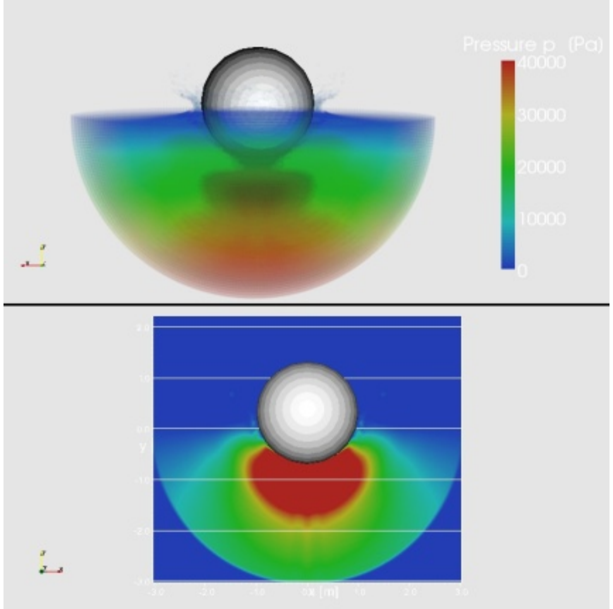

Figure 13. Billard ball impact at *t* = 0.132 s.

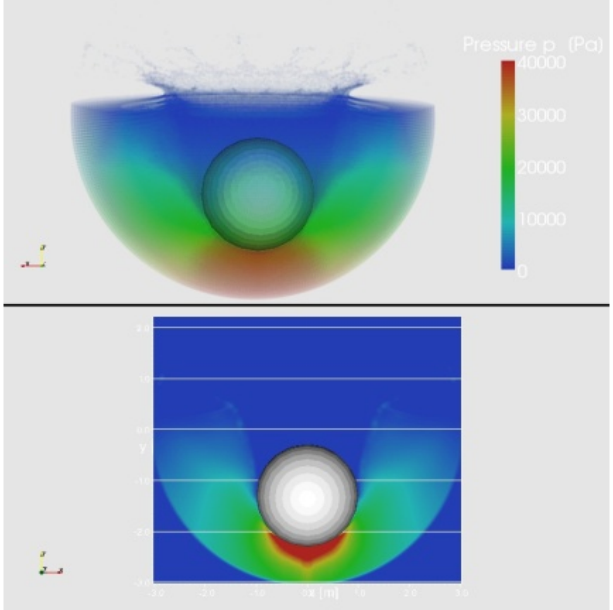

Figure 14. Billard impact at *t* = 0.4 s.

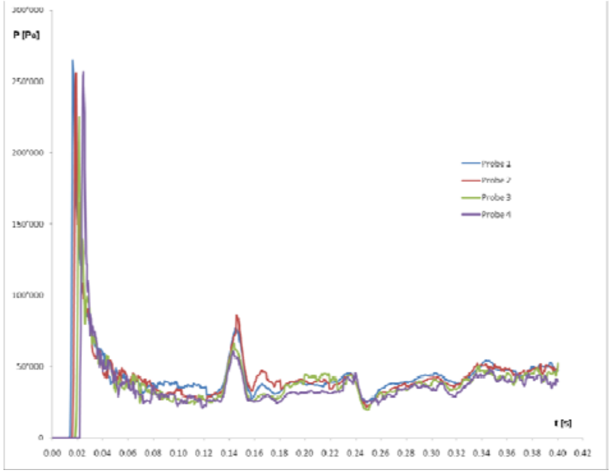

Figure 15. Time dependent local pressure profile along for probes located at the bottom of the billard ball.

IV. Conclusion

In this paper, we evaluate the implementation of parallel SPH code available on cluster to a high performance computer as an IBM Blue Gene/L. We notice that the existing SPH code encounters a bottle neck trouble for scaled-sized problem with a constant load per core. Nevertheless, a 124 million particles simulation is performed on 2'048 cores. In the second hand, we analyze the impact of rigid body to the water free surface. We visualize the pressure wave propagation from the body to the water free surface, to the bottom of the tank and the back pressure wave behavior.

Acknowledgement

The authors would like to thank Steering Committee of High Performance Computing of EPFL and its chairman, Professor François Avellan, the IT Department, and especially, Christian Clémencon, EPFL Blue Gene system manager for his precious advices.

References

[1] G. Oger, M. Doring, B. Alessandrini and P. Ferrant, “An improved SPH method: towards higer-order conergence,” Journal of Computational Physics, vol. 225, pp. 1472-1492, 2007.

[2] I. F. Sbalzarini, J. H. Walther, M. Bergdorf, S. E. Hieber, E. M. Kotsalis and P. Kounoutsakos, “PPM – A highly efficient parallel particle-mesh library for the simulation of continuum systems”, Journal of Computational Physics, vol, 215, pp. 566-588, 2006.

[3] P. Maruzewski, S. Shepel, “The level set interface-tracking model coupled to SPH method”, 1st SPHERIC Workshop, Rome, 2006.

[4] J.-P. Vila, “On particle weighted methods ans Smoothed Particles Hydrodynamics”, Mathematical Models and Methods in Applied Sciences, vol. 9, pp. 161-210, 1999.

[5] S. M. Laverty,” Experimental hydrodynamics of spherical projectiles impacting on a free surface using high speed imaging techniques”, Master in science in ocean engineering at the MIT, 2003.

[6] E. G. Richardson, “The impact of a solid on a liquid surface”, Proceedings Physics Society, vol. 61, pp. 352-367, 1948.

[7] J. Biddiscombe, “PV-meshless Wiki-website”, 2007.