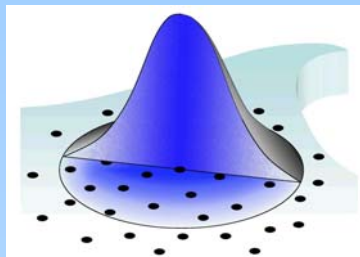


SPH European Research Interest Community

<http://wiki.manchester.ac.uk/spheric/>

Contact: damien.violeau@edf.fr



D. Violeau, Chairman
N. Quinlan, Secretary
B. Rogers, Webmaster

Steering Committee

EDF R&D / LNHE (France)
National University of Ireland, Galway
University of Manchester (UK)
University of Vigo (Spain)
L'Aquila University (Italy)
Ecole Centrale de Nantes (France)
VA TECH Hydro (Switzerland)
University of Plymouth (UK)
Swiss National Supercomputing Centre
INSEAN (Italy)
ESI-Group Netherlands
University of Heidelberg (Germany)

Members

Novosibirsk State University (Russia)
University of Sydney (Australia)
Ecole Centrale de Lyon (France)
Universidad Nacional de Educacion a Distancia (Spain)
Johns Hopkins University (USA)
University of Nottingham (UK)
North China Electric Power University
University of Lancaster (UK)
Université de Montpellier (France)
Technical University of Madrid (Spain)
Groupe-Conseil LaSalle (Canada)
CSIRO Mathematical and Information Sciences (Australia)
Shanghai Jiao Tong University (China)
Ecole Polytechnique Fédérale de Lausanne (Switzerland)
Université du Havre (France)
Université de Rennes (France)
Université de Savoie (France)
Swiss Federal Institute of Technology (Switzerland)
CESI RICERCA (Italy)
CIMNE Barcelona (Spain)
University of Palermo (Italy)
University of Genova (Italy)
CEDEX (Spain)
University of Pavia (Italy)
Dublin Institute of Technology (Ireland)
Imperial College London (UK)
Institute for Plasma Research (India)
BAE Systems (UK)
University of Umeå (Sweden)
Institut Français du Pétrole (France)
University of West Bohemia (Czech Republic)
Tarbiat Modares University (Iran)
University of West Bohemia (Czech Republic)
Universidade de Santiago de Compostela (Spain)
CUGRI (Italy)
University of Hamburg (Germany)
Iran University of Science & Technology
University of Cambridge (UK)
ASR Limited (New Zealand)
Cranfield University (UK)
City University London (UK)
HydroOcean (France)
Laboratório Nacional de Engenharia Civil (Portugal)
Aristotle University of Thessaloniki (Greece)
Catholic University Leuven (Belgium)
University of Calabria (Italy)
University of Ljubljana (Slovenia)
Virginia Tech (USA)
SINTEF Materials and Chemistry (Norway)

SPHERIC III, EPFL Lausanne, June 3rd – 6th, 2008

Pierre Maruzewski, EPFL – LMH, Avenue de Cour 33 bis, CH-1007 Lausanne, Switzerland



The third international SPHERIC workshop on the Smoothed Particle Hydrodynamics (SPH) method took place in Lausanne in June 2008 from 3rd to 6th. The goals of this workshop were to share knowledge around the following main themes:

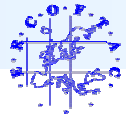
- to develop the basic scientific concepts including parallelism and post-processing,
- to communicate experience in the application of these technologies,
- to foster communication between industry and academia,
- to discuss currently available as well as new concepts,
- to give an overview of existing software and methods,
- to define and run benchmark test cases.

The workshop began with a training day on June the 3rd. The organizing committee was pleased that 30 researchers and students registered. The first half training day was about the free open-source SPHysics solver designed specifically for simulating free-surface flow phenomena. This short course, given by Prof. Robert A. Dalrymple and Dr. Benedict Rogers, was designed to introduce students and practising engineers to the basic SPHysics code and use it for problems in coastal engineering and hydrodynamics (see page 9). The second part of the training day, organized by Dr. John Biddiscombe and Dr. Yun Jang, was about post-processing of SPH simulations by using the pv-meshless software, developed on the ParaView platform at the Swiss National Supercomputing Center (see page 10).



From 4th to 6th, the 13 plenary sessions were attended by about 81 researchers, industry representatives and students. They took benefit from three keynote lectures given by Prof. Jean-Paul Vila (INSA Toulouse), Prof. Javier Bonet (Swansea University) and Dr. Peter Berczik (Astronomisches Rechen-Institut). The session topics covered a large scale of applications of SPH:

- Advances in SPH models,
- Free-surface flows,
- Wave impact,
- Incompressible methods,
- Turbulence,
- High Performance Computing (see pages 5-8),
- Non Newtonian Fluids,
- Fluid-Structure interactions,
- Multiphase flows,
- Astrophysics.



A poster session (6 posters) was held during the three days. Below is the detailed workshop programme. During the workshop, 15 students were nominated for the Libersky's student award. The Steering Committee of SPHERIC decided to present it to Ruairi Nestor, from the National University of Ireland, Galway, for his paper "Moving boundary problems in the finite volume particle method". Ruairi's interview was published in 24Heures, a Swiss newspaper.

Following the above few comments, which cannot do justice to the vivid and stimulating presentation and discussion held during the Workshop, I would like to thank all the contributors for keeping the tight schedule for the submission of papers and Prof. François Avellan, Director of EPFL Laboratory for Hydraulic Machines for agreeing to organize this workshop. I would like also to acknowledge the tremendous work of Mrs. Valérie Jacquot-Descombes in touch with all EPFL services. I further thank as well as the local team, Dr. Cécile Munch-Alligné, Dr. Mohamed Fahrati, Olivier Braun, Alireza Zobeiri and Vlad Hasmatuchi of the EPFL Laboratory for Hydraulic Machines for helping in the publication of the proceedings. Special thank goes to Dr. Philippe Cerrutti who made possible SPHERIC IIIrd website and to Mrs. Isabelle Stoudmann for her administrative support. Besides, this event would not have been possible without the commitment of Dr. Etienne Parkinson and Dr. Jean-Christophe Marongiu, Dr. Jean Favre and John Biddiscombe, Co-Chairmen of the Workshop.

We are very grateful to our sponsors, who financially support the edition of the proceedings, namely EPFL, CSCS, VATECH Hydro Andritz and ERCOFTAC. Further thanks to Lausanne Tourisme for their help in organizing the tourism guide.

Contact: pierre.maruzewski@epfl.ch

I. DAY 1: WEDNESDAY, JUNE 4th, 2008

A. Session 1: Advances in SPH models – 1

Smoothed Particle Hydrodynamics stochastic model for flow and transport in porous media, A. M. Tartakovsky, Pacific Northwest National Laboratory, Richland, D. M. Tartakovsky, University of California, San Diego, P. Meakin, Idaho National Laboratory.

Oblique impact of a jet on a plane surface solved by SPH: suggestions to improve the results of the pressure profiles, D. Molteni, Dipartimento di Fisica e Tecnologie Relative, Università di Palermo, A. Colagrossi, INSEAM, Italian Ship Model Basin, Roma, Italy.

A hybrid Boussinesq-SPH model for coastal wave propagation, A. J. C. Crespo, M. Gómez-Gesteira, Grupo de Física de la Atmósfera y del Océano, Universidad de Vigo, Ourense, Spain, R. A. Dalrymple, Department of Civil Engineering, Johns Hopkins University, Baltimore, USA.

B. Session 2: Free-surface flows

Simulation of interfacial and free-surface flows using a new SPH formulation, A. Colagrossi, M. Antuono, INSEAM, Italian Ship Model Basin, Roma, Italy, N. Grenier, D. Le Touzé, Laboratoire de Mécanique des Fluides, Ecole Centrale de Nantes, France, D. Molteni, Dipartimento di Fisica e Tecnologie Relative, Università di Palermo, Italy.

Swimming with and without skin, J. Kajtar, Joe Monaghan, School of Mathematical Sciences, Monash University, Melbourne, Australia.

A new 3D parallel SPH scheme for free-surface flows, A. Ferrari, M. Dumbser, E. F. Toro, A. Armanini, Department of Civil and Environmental Engineering, University of Trento, Italy.

C. Session 3: Wave impact – 1

SPH simulation of a floating body forced by regular waves, S. Manenti, A. Panizzo, University of Rome "La Sapienza", P. Ruol, University of Padova, L. Martinelli, University of Bologna, Italy.

Wave impact simulations using incompressible and weakly-compressible SPH models, J. P. Huges, D. I. Graham, P. W. James, School of Mathematics and Statistics, D. E. Reeve, A. J. Chadwick, J. Lawrence, School of Engineering, University of Plymouth, UK.

Coastal flow simulation using a SPH formulation modelling the non-linear shallow water equations, M. De Leffe, D. Le Touzé, B. Alessandrini, Laboratoire de Mécanique des Fluides, Ecole Centrale de Nantes, France.

D. Session 4: Incompressible method

Simulation of vortex spindown and Taylor-Green vortices with incompressible SPH method, R. Xu, P. Stansby, B. D. Rogers, University of Manchester, C. Moulinec, Daresbury Laboratory, Science and Tech. Facilities Council, UK.

A constant-density approach for incompressible multiphase SPH, X. Y. Hu, N. A. Adams, Lehrstuhl für Aerodynamik, Technische Universität München, Germany.

Permeable and Non-reflecting Boundary Conditions in SPH, M. M. Lastiwka, N. J. Quinlan, M. Basa, Department of Mechanical and Biomedical Engineering, National University of Ireland, Galway, Ireland.

E. Session 5: Turbulence

Experiences of SPH with the lid driven cavity problem, A. Panizzo, T. Capone, S. Marrone, Department of Hydraulic Engineering, University of Rome, “La Sapienza”, Roma, Italy.

Forced 2D wall-bounded turbulence using SPH, M. Robinson, J. Monaghan, School of Mathematical Sciences, Monash University, Melbourne, Australia.

Modelling a fish passage with SPH and Eulerina codes: the influence of turbulent closure, D. Violeau, R. Issa, Saint-Venant Laboratory for Hydraulic, Paris-Est University, J. Chorda, M.-M. Maubourguet, Institut de Mécanique des Fluides de Toulouse, France.

II. DAY 2: THURSDAY, JUNE 5th, 2008

A. Session 6: Advances in SPH models -2

Conventional SPH revisited, R. Vignjevic, J. Campbell, Cranfield Universtiy, UK.

Riemann solves and efficient boundary treatments: an hybrid SPH-finite volume numerical method, J.-C. Marongiu, F. Leboeuf, Laboratoire de Mécanique des Fluides et d’Acoustique, Ecole Centrale de Lyon, France, E. Parkinson, VATECH Hydro Andritz, Vevey, Suisse.

Moving boundary problems in the finite volume particle method, R. Nestor, M. Basa, N. Qiuinlan, Department of Mechanical and Biomedical Engineering, National University of Ireland, Galway, Ireland.

B. Session 7: High Performance Computing

High-performance computing SPH: Towards a hundred million of particle simulation, C. Moulinec, D. R. Emerson, X. J. Gu, Daresbury Laboratory, Science and Technology Facilities Council, UK, R. Issa, EDF R&D, National Hydraulics and Environment Laboratory, Chatou, France.

HPC for Spartacus-3D SPH code and applications to real environmental flows, R. Issa, D. Violeau, Saint-Venant Laboratory for Hydraulic, Paris-Est University, France, C. Moulinec, Daresbury Laboratory, Science and Technology Facilities Council, UK, D. Latino, IBM systems & Technology Group, Dubai, United Arab Emirates, J. Biddiscombe, CSCS, Manno, Switzerland, G. Thibaud, EDF R&D, SINETICS, Clamart, France.

High-performance computing 3D SPH model: Sphere impacting the free-surface of water, P. Maruzewski, EPFL-LMH, Lausanne, Switzerland, G. Oger, HydrOcéan, Nantes, France, D. Le Touzé, Laboratoire de Mécanique des Fluides, Ecole Centrale de Nantes, France, J. Biddiscombe, CSCS, Manno, Switzerland.

C. Session 8: Non Newtonian fluids

SPH simulation of non-Newtonian mud flows, T. Capone, A. Panizzo, Department of Hydraulic Engineering, University of Rome, “La Sapienza”, Roma, Italy.

SPH molecules – A model of granular materials, T. Capone, Department of Hydraulic Engineering, University of Rome, “La Sapienza”, Roma, Italy, J. Kajar, J. Monaghan, School of Mathematical Sciences, Monash University, Melbourne, Australia.

A SPH thermal model for the cooling of a lava lake, A. Herault, Université Paris-Est, France, A. Vicari, C. Del Negro, Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Catania, Italy.

D. Session 9: Fluid - Structure

Modelling 3D fracture and fragmentation in a thin plate under high velocity projectile impact using SPH R. Das, P. W. Cleary, CSIRO Mathematical and Information Sciences, Melbourne, Australia.

SPH interaction of fluids and solids, L. Lobovsky, Department of Mechanics, University of West Bohemia, Plzeň, Czech Republic.

SPH framework to model fluid shell interactions, S. Potapov, EDF R&D, Clamart, B. Maurel, A. Combescure, LaMCoS INSA-Lyon, Villeurbanne, France.

E. Session 10: Multiphase flows

Simulating dynamic surface tension of lung surfactant using SPH, S. Adami, X. Y. Hu, N. A. Adams, Lehrstuhl für Aerodynamik, Technische Universität München, I. Mahle, MTU Aero Engines GmbH, Munich, Germany.

Two-phase flow simulations using a volume fraction SPH scheme with a Riemann solver, N. Grenier, D. Le Touzé, P. Ferrant, Laboratoire de Mécanique des Fluides, Ecole Centrale de Nantes, J.-P. Vila, LMIP INSA Toulouse, France.

Lifeboat water entry simulation by the hybrid SPH-FE method, P. H. L. Groenenboom, ESI Group, Delft, The Netherlands.

III. DAY 3: FRIDAY, JUNE 6th, 2008

A. Session 11: Advances in SPH models – 3

Splitting for highly dissipative smoothed particle dynamics, S. Litvinov, X. Y. Hu, N. A. Adams, Lehrstuhl für Aerodynamik, Technische Universität München, Germany.

A comparative study of ANSYS AUTODYN and RSPH simulations of blast waves, S. Børve, A. Bjerke, Norwegian Defence Research Establishment, M. Omang, J. Trulsen, Institute of Theoretical Astr., University of Oslo, Norway.

Analysis of SPH and mesh based simulations using point based post processing tool, Y. Jang, CSCS, Manno, Switzerland, J.-C. Marongiu, Laboratoire de Mécanique des Fluides et d'Acoustique, Ecole Centrale de Lyon, France, E. Parkinson, N. Gervais, H. Garcin, VA-TECH Hydro Andritz, Vevey, Switzerland.

B. Session 12: Astrophysics

Gas accretion from the elliptic gas disk to the binary system, Y. Imaeda, Kobe University, T. Tsuribe, Osaka University, S.-I. Inutsuka, Kyoto University, Japan.

Smoothed particle hydrodynamics in thermal phases of an one dimensional molecular cloud, M. Nejad-Asghar, Department of Physics, Damghan University of Basic Sciences, Iran, D. Molteni, Dipartimento di Fisica e Tecnologie Relative, Università di Palermo, Italy.

Accelerating smoothed particle hydrodynamics for astrophysical simulations: a comparison of FPGAs and GPUs, G. Marcus, A. Kugel, R. Männer, Dept. of computer science, University of Heidelberg, Mannheim, P. Berczik, I. Berentzen, R. Spurzem, Astronomisches Rechen-Institut, University of Heidelberg, T. Naab, M. Hilz, A. Bukert, University Observatory Munich, Germany.

C. Session 13: Wave impact – 2

Reynolds number and shallow depth sloshing, A. Colagrossi, INSEAN, Rome, Italy, L. Delorme, Eurocopter, Marignanne, France, J.-L. Cercós-Pita, A. Souto-Iglesias, Naval Architecture Department, Technical University of Madrid, Spain.

SPH conservation of circulation in breaking wave processes, M. Antuono, A. Colagrossi, INSEAN, Rome, Italy, J. Monaghan, School of Mathematical Sciences, Monash University, Melbourne, Australia, D. Le Touzé, Laboratoire de Mécanique des Fluides, Ecole Centrale de Nantes, France.

Simulation of wave impact pressure on vertical structures with the SPH method, F. Dentale, G. Viccione, E. Pugliese Carratelli, University of Salerno, Civil Engineering Department, Fisciano, Italy.

IV. POSTER SESSION

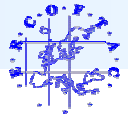
SPH study of high speed ship slamming, D. Veen, T. Gourlay, Center for Marine Science and Technology, Curtin University, Australia.

Investigation of wave loading on a half-submerged cylinder using SPH, P. Omidvar, B. D. Rogers, P. K. Stansby, School of Mechanical, Aerospace and Civil Engineering, The University of Manchester, UK.

New features and applications of the hybrid SPH-FE approach in PAM-CRASH, P. H. L. Groenenboom, ESI Group, Delft, The Netherlands.

A regularized Lagrangian finite point method for incompressible viscous flows, J. Fang, A. Parriaux, EPFL-GEOLEP, Lausanne Switzerland.

SPH simulation of the flow in a spring safety valve, S. Sibilla, Dipartimento di Ingegneria Idraulica e Ambientale, Università di Pavia, Italy.



Accelerating Astrophysical Smoothed Particle Hydrodynamics with GPU

P. Berczik, I. Berentzen, R. Spurzem, *Astronomisches Rechen-Institute, University of Heidelberg, Heidelberg, Germany*
G. Marcus, A. Kugel, R. Manner, *Dept. of Computer Science V, ZITI, University of Heidelberg, Mannheim, Germany*

Particle-based astrophysical simulation determines the motion of individual particles according to a model for the interaction forces. Long-range gravity and short-range hydrodynamical forces are amongst the most important ones for many systems. The most time consuming part of these algorithms has always been the gravity interactions, and its acceleration has been widely explored, first with ASICs like the GRAPE boards (Fukushige *et al.*, 2005), Field Programmable Gate Arrays (FPGAs) (Hamada *et al.*, 1998) and more recently with Graphics Processing Units (GPUs) (Hamada and Itaka, 2007; Nguyen, 2008; Bellemana, 2008). Current implementations achieve over ~500 GFlops from a single board. This is quite logical, as it represents >90% of time for pure-CPU algorithms, it is ~N² in its more basic form, and highly parallelizable.

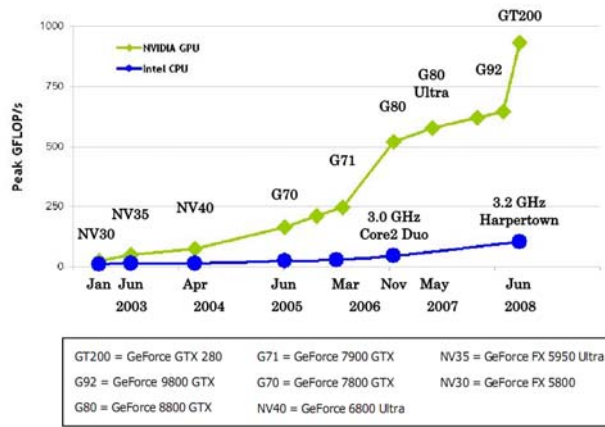


Figure 1 – Theoretical speed performance of different NVIDIA GPU's in the last few years.

With particle numbers >10k the latter SPH part is computationally very demanding, once the gravity has been accelerated. It is also more complex, as it is based on interaction lists between particles. We support such calculations, which can be applied to a wide variety of practical problems besides astrophysics, by programmable hardware. Related work include the developments of Nakasato and Hamada (2005) and similar work has been done for Molecular Dynamics (MD) (Scrofano, 2007), as the algorithms are comparable.

Nowadays Graphic Processing Units (GPU) is highly parallel, programmable processors used primarily for image processing and visualization in workstations and entertainment systems. As they grow in complexity, they have added many features, including programmability, floating point capability and wider and faster interfaces. As a final addition, the latest versions include APIs to program them as custom accelerators, enabling us to test with relative ease the performance of the platform.

We use the standard SPH formulation as the base and modify it in order to reduce the actual number of computations done in hardware. As kernel functions, we use:

$$W'(x) = \begin{cases} 1 - \frac{3}{2}x^2 + \frac{3}{4}x^3 & \text{if } 0 \leq x < 1 \\ \frac{1}{4}(2-x)^3 & \text{if } 1 \leq x < 2 \\ 0 & \text{otherwise} \end{cases}$$

And the scalar part of the gradient of W as:

$$\Omega(x) = \begin{cases} -1 + \frac{3}{4}x & \text{if } 0 \leq x < 1 \\ -\frac{1}{4}x + 1 - \frac{1}{x} & \text{if } 1 \leq x < 2 \\ 0 & \text{otherwise} \end{cases}$$

In Step 1 we compute the density, curl and divergence of the velocity as:

$$\begin{aligned} \pi \rho_i &= \sum_j \frac{m_j}{h_{ij}^3} W'(|\vec{r}_{ij}|/h_{ij}) \\ -\frac{\pi}{3} \rho_i (\nabla \cdot \vec{v}) &= \sum_j \frac{m_j}{h_{ij}^5} (\vec{v}_{ij} \cdot \vec{r}_{ij}) \Omega(|\vec{r}_{ij}|/h_{ij}) \\ -\frac{\pi}{3} \rho_i (\nabla \times \vec{v}) &= \sum_j \frac{m_j}{h_{ij}^5} (\vec{v}_{ij} \times \vec{r}_{ij}) \Omega(|\vec{r}_{ij}|/h_{ij}) \end{aligned}$$

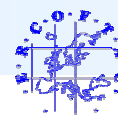
And in Step 2 the acceleration, including artificial viscosity, as:

$$\Pi_{ij} = \begin{cases} \frac{-\alpha c_{ij} \mu_{ij} + \beta \mu_{ij}^2}{\rho_{ij}} & \text{if } \vec{v}_{ij} \cdot \vec{r}_{ij} \leq 0 \\ 0 & \text{if } \vec{v}_{ij} \cdot \vec{r}_{ij} > 0 \end{cases}$$

$$\mu_{ij} = \frac{h_{ij}(\vec{v}_{ij} \cdot \vec{r}_{ij})}{\vec{r}_{ij}^2 + \eta^2 h_{ij}^2} f_{ij}$$

$$\frac{\pi}{3} \frac{d\vec{v}_i}{dt} = - \sum_j \frac{m_j}{h_{ij}^5} \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \Omega(|\vec{r}_{ij}|/h_{ij})$$

For each timestep, the coprocessor loads the particle data (position, velocity, mass, etc) into external memory. After selecting for step1 or step2 computations, neighbour lists are sent in sequence from the host in the format [ip, NB, jp1 ... jpN], being ip the index of the i-particle, NB the number of neighbours, and jpX the corresponding j-particles indices. Neighbour lists are received from the host and processed immediately at the rate of one neighbor interaction per cycle. Given the described scheme, it is clear the algorithm is order ~N*M, where N is the number of particles and M the average number of



neighbours. Therefore, the overall performance is driven by the communication time (how fast can the neighbour lists be sent to the board) and the clock frequency of the coprocessor (how fast an interaction can be dispatched). In order to use efficiently the capabilities of the coprocessor, we developed a software library for C/C++/FORTRAN languages. This library provides the user with a clean interface to the SPH functionality while hiding coprocessor-specific details. In addition, an emulation core is provided, that allows the library to perform the same operations with the host CPU only.

Particular attention was given to the interface of the library with existing applications. Since the coprocessor performance is directly proportional to the communication between the host and the board, a generic buffer management scheme (Marcus, 2006) was implemented, allowing the library to access data structures of the application directly for direct conversion between the formats of the application and the coprocessor. As mentioned, GPUs are in essence a highly parallel, programmable architecture. We currently use a board containing 128 parallel processing elements and 768 MB of high speed RAM over a 384-bit bus, but many versions of such a cards nowadays are available:

http://www.nvidia.com/object/geforce_family.html

Such a level of parallelism sacrifices memory hierarchy and coherency in order to commit most resources on-chip to computational resources instead of cache memory.

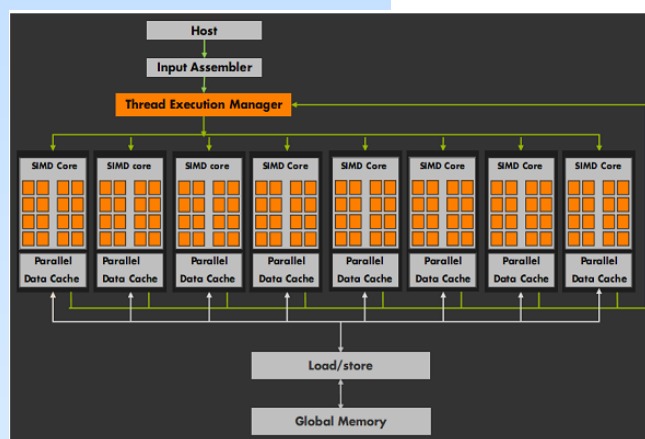


Figure 2 – Basic organization of the computational model inside GPU.

A sketch of the organization of an NVIDIA GPU is depicted in Fig. 2. The board (and processor), in particular a GeForce 8800 GTX, consists of 16 multiprocessors, each one capable of managing 8 threads in parallel in SIMD fashion for a total of 128 threads; and communicates to the host using a 16-lane PCI-Express bus. Each multiprocessor has several shared resources available to all his local threads, specifically a small, fast shared memory of 16KB in size, a big register file (8K registers), a constant memory area to store constant values, and interfaces to the main (global) memory.

It is important to note that none of these memories are cached. In contrast, special read-only regions referred to as constant memory (for constant values) and texture memory (for large data references/interpolated data) are cached. Having regions without cache, without write coherency and with several penalties to the memory access patterns adds complications to the implementation of the algorithms. Fortunately, NVIDIA provides an API and computational model to make efficient use of the processors:

http://www.nvidia.com/object/cuda_home.html

The CUDA (Compute Unified Device Architecture) library and tools provide a C-like programming language and compiler, with specific extensions for the platform. The library and API makes interfacing with the board a very easy task. In contrast to other architectures, NVIDIA GPUs based their computing model around several levels of units of work, where the most simple is very similar to lightweight threads (simply referred to as threads). Threads can be grouped in blocks, which are assigned to a single multiprocessor. Blocks are in turn organized in a grid, which represents the current workload. Blocks are assigned dynamically to available multiprocessors. This thread model allows the hardware to scale more easily to the number of computational units available, allowing to effectively hide the latency of memory accesses for each thread when using a very high number of threads. At the same time, it allows precise control on the work assigned to a single multiprocessor and the proper discovery of sibling threads for communication.

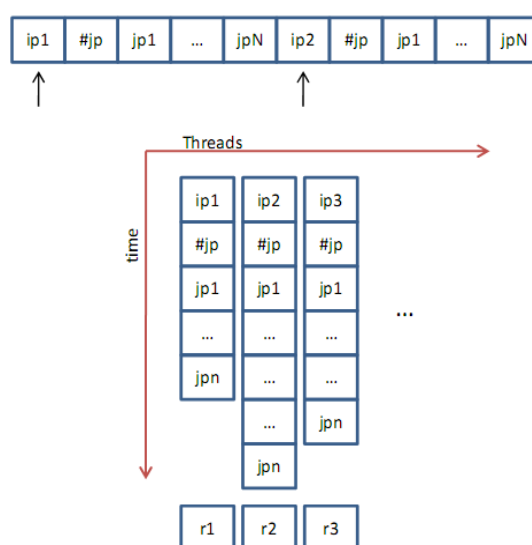
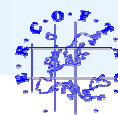


Figure 3 – Organization of the neighbour's threads in the GPU memory.

The current implementation of the SPH algorithm on the GPU is based in the communication interface developed originally for our FPGA accelerator.



However, data structures containing particle data and the neighbour lists are transferred once to the GPU memory instead of using small chunks as with the FPGA. Results are read at the end of each step. This favours better performance as a result of large transfers.

Once the data is in the GPU memory, Fig. 3 shows our scheme on how to parallelize the work among the threads. We assign each neighbour list to a separate thread, and process its corresponding segment of the big NL array independently. Each thread copies the *i*-particle and the current *j*-particle into shared memory, then computes the interaction and adds it to the result, which is also kept in shared memory. Each thread loads and process the *j*-particles of its own list one after another. Therefore, each thread is independent and their execution order is not important. Another advantage of this algorithm is that the accumulation is done per thread, which removes the necessity for a parallel reduction addition.

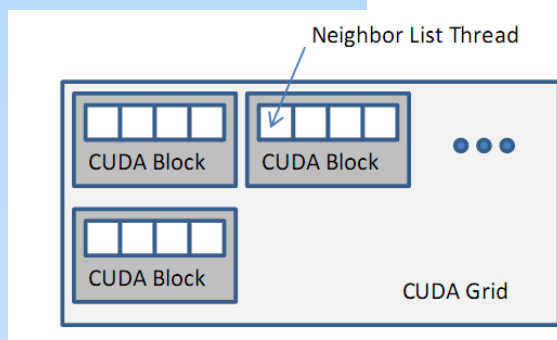


Figure 4 – Neighbour list threads on the CUDA grid.

This allows threads to be distributed as in Fig. 4, where the big NL array is simply divided into appropriate chunks, where the CUDA Grid spans the full NL array and is divided in the necessary blocks. The size of each CUDA block, *i.e.* the number of threads in a given block, is determined by several factors. In this case, each thread requires 2 particles and 1 result to be stored in shared memory, which uses 116 bytes at most. Because the size of shared memory is 16 KB, this means we can hold data for up to 141 threads per block. Increasing this value will be useful, if the algorithm is computationally bounded.

Our software library is designed from scratch to support multiple implementations of the supported SPH algorithms. Therefore, the effort of extending it to support GPUs was low. In addition, using the same interface allows us to use the same applications as before for FPGA board without any significant change in the program interface: just to change a switch in the initialization function, to define the type of processing core to use. CPU runs and GPU runs are from a workstation equipped with an Intel Core 2 Quad at 2.4 GHz, 4 GB of RAM and a GeForce 8800 GTX GPU with 768 MB. FPGA based MPRACE runs are from one node of the Titan cluster at the ARI-Heidelberg, with 2 Intel Xeon CPUs at 3.2 GHz, 4 GB of RAM, a GRAPE6a board and a MPRACE-1 board in a PCI-X slot. All presented runs

are serial runs, *i.e.* they use only one core. For the self gravity calculation between particles we use the self coded TREE-GRAPE gravity routine (Berczik *et al.*, 2007).

For performance and accuracy measures we use the timing results from simulation with gravity and SPH forces, running for one single step with shared timestep integration. Accuracy is compared relative to the original double precision implementation on the CPU. In the case of the GPU, it has IEEE-754 compliant single precision operators for most operators, but has limited range for several others, particularly division and square root. The accuracy of the results is high, comparable to the use of SSE instructions in the CPU. The results for the time evolution of energies during adiabatic collapse test runs with different particle numbers (ranging from 10k to 100k) show that the absolute error in total energy conservation during the whole period of integration was less than 0.1%. For more detailed information, consult (Berczik *et al.*, 2007).

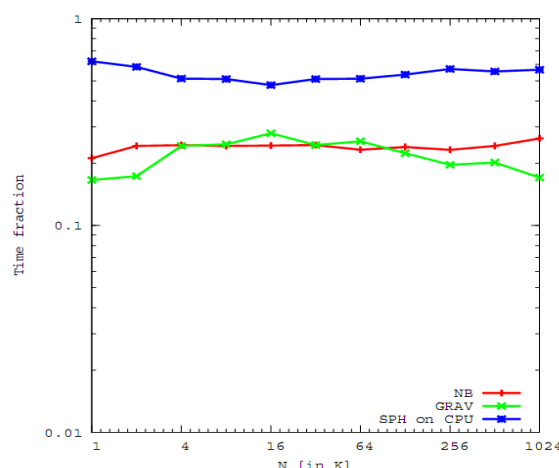
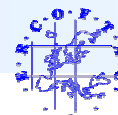


Figure 5 – Time fraction in different functions in the CPU for a given number of particles.

In Fig. 5 the fraction of the total used by each significant part of the algorithm can be observed. Gravity (GRAV) represents the time spent in computing the gravitational forces using a GRAPE accelerator board. A version without the accelerator is not shown, as that will represent over 95% of the total computational time. The neighbour list generation is represented by the NB order $\sim N \cdot M$, where *N* is the number of particles and *M* the average number of neighbours. Therefore, the overall performance is driven by the communication time (how fast can the neighbour lists be sent to the board) and the clock frequency of the coprocessor (how fast an interaction can be dispatched). Both NB and SPH are computed by the CPU. In this condition, GRAV and NB are balanced, but SPH consumes more time than the others. At around 60%, this is the critical part of the



algorithm. Fig. 6 shows the time fractions once SPH is computed in the GPU. As it can be seen, it is reduced from ~60% to less than 10% (less than 8% for bigger sets). Fig. 7 shows the time spent in the SPH computations in all different cases, as well as the speedup (ratio) against the CPU time. From this plot it can be seen how the speedup is sustained for the particle sets, and is ~10 times for the FPGA and about ~20 times for the GPU. Assuming an average speedup of ~20 times for the SPH computations, one can use Amdahl's Law to compute the overall speedup for an overall application. With the SPH consuming 60% of the time, this gives a maximum speedup of ~2.3. This ratio will be much higher for simulations of another nature without self gravity, where computational time will be spent almost exclusively in SPH related computations.

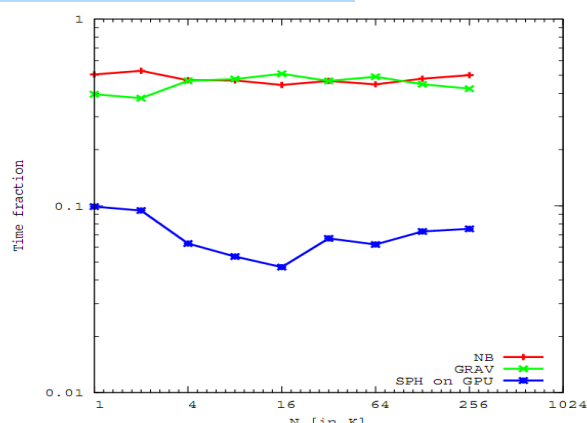


Figure 6 – Time fraction in different functions computing SPH in the GPU for a given number of particles.

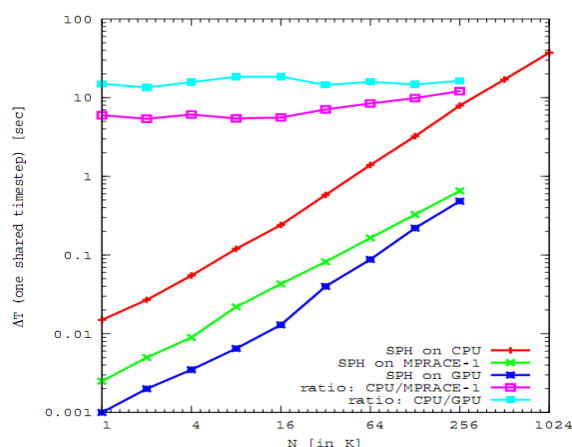


Figure 7 – Time and Speedup of the SPH computations for the FPGA and the GPU.

When considering the computational efficiency, the estimated peak performance for GPU is ~500 GFlops, but our current SPH implementation has only ~50 GFlops, at most. This means the GPU is quite far from the maximum, leading us to believe there is plenty of possibilities to improve the algorithm and obtain better performance. The

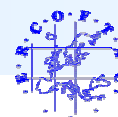
access patterns observed on global memory raise the need for a better neighbour list structure, one that allow for efficient computation of neighbour lists in parallel.

Acknowledgments: This research work has been performed in the framework of the **GRACE** project and was funded by the Volkswagen Foundation (Ref: I/80 040) and the Ministry of Science, Research and the Arts of the State of Baden-Württemberg, Germany. P.B. acknowledges support from the German Science Foundation under **SFB 439** (sub-project B11) "Galaxies in the Young Universe" at the University of Heidelberg.

Contact: berczik@ari.uni-heidelberg.de
guillermo.marcus@ziti.uni-heidelberg.de

References

- Bellemana, R., Bdorfa, J., and Zwart, S.P. (2008). *High performance direct gravitational N-body simulations on graphics processing unit's (II): An implementation in CUDA*. New Astronomy, **13**(2):103–112.
- Berczik, P., Nakasato, N., Berentzen, I., Spurzem, R., Marcus, G., Lienhart, G., Kugel, A., Männer, R., Burkert, A., Wetzstein, M., Naab, T., Vazquez, H., and Vinogradov, S. (2007). *Special hardware accelerated, parallel SPH code for galaxy evolution*. Proc. 2nd SPHERIC workshop, Madrid, pp. 5–8.
- Fukushige, T., Makino, J., and Kawai, A. (2005). *GRAPE6a: A single-card GRAPE6 for parallel PC-GRAPE cluster system*. PASJ, **57**:1009–1021.
- Hamada, T., Fukushige, T., Kawai, A., and Makino, J. (1998). *Progrape-1: A programmable special-purpose computer for many-body simulations*. FPGAs for Custom Computing Machines, Proceedings IEEE Symposium, pp. 256–257.
- Hamada, T. and Iitaka, T. (2007). *The chamomile scheme: An optimized algorithm for N-body simulations on programmable graphics processing units*. <http://arxiv.org/abs/astro-ph/0703100v1>.
- Marcus, G., Lienhart, G., Kugel, A., and Männer, R. (2006). *Buffer management strategies for high performance computing with reconfigurable hardware*. Proc. FPL (IEEE), pp. 343–348.
- Nakasato, N. and Hamada, T. (2005). *Astrophysical hydro-dynamics simulations on a reconfigurable system*. Proc. FCCM (IEEE), pp. 279–280.
- Nguyen H. *GPU Gems 3*. Addison-Wesley, New York, 2008.
- Scrofano, R., Gokhale, M.B., Trouw, F. and Prasanna, V.K. (2007). *Accelerating molecular dynamics simulations with reconfigurable computers*. IEEE Transactions on Parallel and Distributed Systems.



SPHERIC Training Day

Benedict D. Rogers, University of Manchester, School of Mech. Aero & Civil Engineering, U.K.

John Biddiscombe, Swiss National Supercomputing Centre (CSCS), Manno, Switzerland

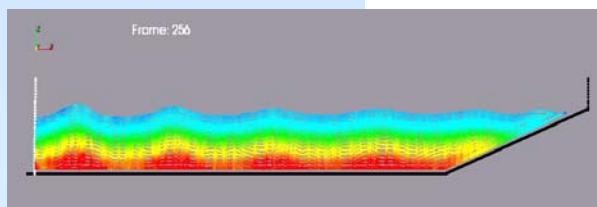
The Training Day at the 3rd International SPHERIC Workshop was the first time that such an activity had been part of a SPHERIC event. The SPHERIC Training Day took place the day before the main workshop started in a computer teaching laboratory on the EPFL campus. The course was well attended with over 30 people registered for the two courses.

The day consisted of two parts – essentially two separate but closely related ½-day courses. The morning session was an introduction to the SPHysics code – a recently released open-source SPH code for free-surface hydro-dynamics. The afternoon session was dedicated to the meshfree visualisation package pv-meshless – an open-source variant of the ParaView software with new developments and features specifically designed for the visualisation of meshfree particle simulations.



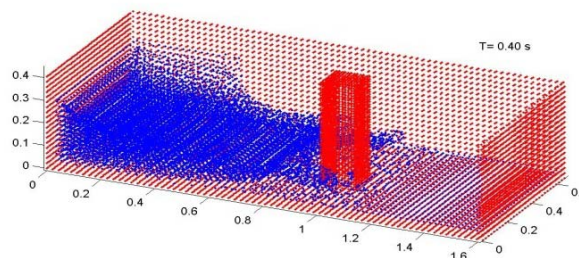
Morning Session: SPHysics – Simulation

SPHysics is a free open-source SPH solver designed specifically for simulating free-surface flow phenomena. SPHysics is the result of collaboration between four universities and since its release, the code has been downloaded over 600 times. This short course was designed to introduce students and practising engineers to the basic SPHysics code, how to run, and use it for problems in coastal engineering and hydrodynamics.



SPHysics – Example wave propagation.

The morning session on SPHysics consisted of an introductory talk given by Professor Tony Dalrymple on the theory underlying SPH used by the SPHysics code. This was followed by a guide to the actual SPHysics code by Dr Ben Rogers so that attendees would have an idea of how the code was structured and how they could modify the code to suit their own purposes.



SPHysics: 3-D Dam break case.

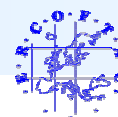
The last part of the SPHysics course consisted of a hands-on practical session run by Dr. Alex Crespo and Prof. Moncho Gomez Gesteira where the attendees were able to run the code and adjust input parameters to see how the code behaved. This final session was as useful for the course organisers as it was for the attendees since the feedback allowed the SPHysics developers to see where users of the code had problems and were able to share experiences and suggestions.



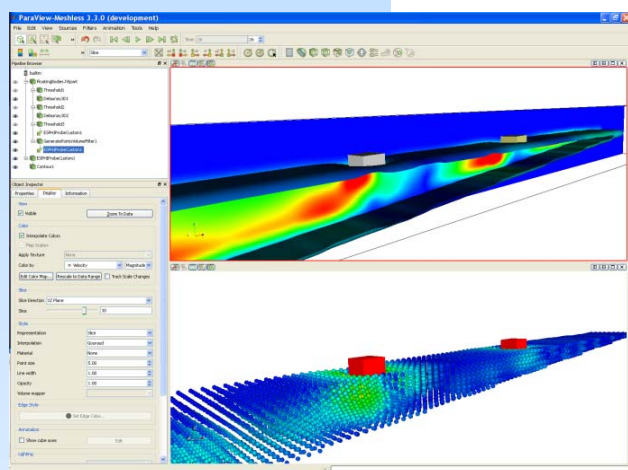
Afternoon Session: pv-meshless – Visualisation

Post-processing of SPH simulation results asks for specific approaches because of its particular data structure. Atop developments from CSCS, a share of the EU Marie Curie project ESPHI is dedicated to generic open-source SPH post-processing tools built within the ParaView software. This short course was designed to introduce students and practising engineers to the basic post-processing, how to import data, run it, and use it for problems in hydraulic engineering and hydrodynamics.

The afternoon session on pv-meshless consisted of an introductory talk given by John Biddiscombe on the basic structure of ParaView, how one creates visualizations of data by applying filters and how to create different views or representations of results. Importing data into pv-meshless using some data converters provided with the software was also covered.



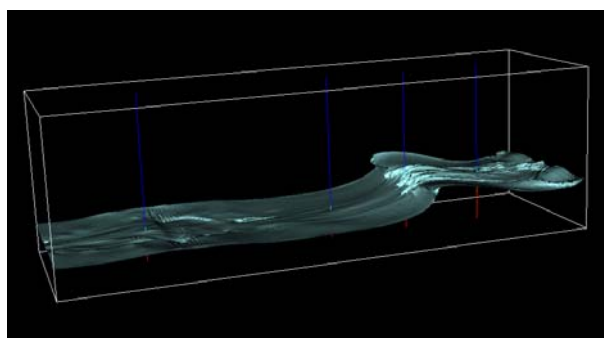
Then Dr. Jun Yang introduced some of the latest developments generated by his work as part of the ESPHI project. The work consists of a series of filters which can be used to process both meshless SPH and conventional CFD data using a common library. This enables direct comparison between the results of one and another, an important part of the validation process. The tools created consist of interpolation filters for point based data, projection (of probe regions) and integration filters which compute flow parameters of use to a practicing engineer. The final part of the tutorial had been intended as a ‘hands on’ work-through on some example datasets (in particular some of those generated using SPHysics code) to teach the participants how to generate images such as those shown in the example snapshot above, as well as how to select individual particles and display their parameters. An important part of the developments towards pv-meshless has been the creation of SPH probe filters which allow the user to slice particle data and generate plots of values on lines, planes and also to create contours of the free surface. The slicing filters were briefly explained, along with the parameters necessary to control them together with some of the more recent features of ParaView/pv-meshless, such as the ability to plot x/y scatter plots, histograms, and 2D data plots as well as a spreadsheet view of raw (or generated) data values – which can be combined with *selections* of data in the 3D views.



pv-meshless: example floating-bodies dataset visualization.

Due to network problems, the tutorial was unable to complete as intended, however the afternoon finished with a very useful Question and Answer session about pv-meshless regarding how to perform specific tasks, how to create animations of data, and the direction of future developments.

Feedback from the pv-meshless training session indicated that future events should be targeted more towards hands-on examples of how to create interesting visualizations and less on lecture-style explanations of features.



pv-meshless: example free-surface visualization.

SPHERIC Training Day 2009

The aim of the Training Day 2008 was for attendees who might be new students or practising engineers to gain an introduction to SPH, be introduced to a validated code and finally be able to visualise the results, all of which can be downloaded from the internet.

Following the success of the Training Day in Lausanne, this will be repeated for the 4th SPHERIC workshop to be held in 2009 in Nantes, France. The SPHERIC Training Day will include more of the developments that are being generated as part of the E.U.-funded ESPHI Project.

References

<http://wiki.manchester.ac.uk/sphysics>

<https://twiki.cscs.ch/twiki/bin/view/ParaViewMeshless>