SPH MODELING OF LAVA FLOWS WITH GPU IMPLEMENTATION

Alexis Hérault
Sezione di Catania
Istituto Nazionale di Geofisica e Vulcanologia
Italy
herault@ct.ingv.it

Giuseppe Bilotta
Dipartimento di Matematica e Informatica
Università di Catania
Italy
bilotta@dmi.unict.it

Ciro Del Negro
Sezione di Catania
INGV
Italy
delnegro@ct.ingv.it

Giovanni Russo
DMI
Università di Catania
Italy
russo@dmi.unict.it

Annamaria Vicari
Sezione di Catania
INGV
Italy
vicari@ct.ingv.it

Abstract
We describe the implementation of the Smoothed Particle Hydrodynamics (SPH) method on graphical processing units (GPU) using the Compute Unified Device Architecture (CUDA) developed by NVIDIA. The entire algorithm is executed on the GPU, fully exploiting its computational power. The code faces all three main components of an SPH simulation: neighbor list constructions, force computation, integration of the equation of motion. The simulation speed achieved is one to two orders of magnitude higher than the equivalent CPU code. Applications are shown for simulating the paths of lava flows during volcano eruptions. Both static problems with purely thermal effects (such as lava lake solidification) and dynamic problems with a complete lava flow were simulated.

Key words
SPH, GPU, lava, fluid dynamics, CUDA

1 Introduction

A complete modeling of lava flow is challenging from the modelistic, numerical and computational points of view. The described phenomenon has at its core a complex fluid dynamic problem with free boundaries: the natural topography irregularities, the dynamic free boundaries and phenomena such as solidification, presence of floating solid bodies or other obstacles and their eventual fragmentation make the problem difficult to solve using traditional numerical methods (finite volumes, finite elements).

Recent developments by researchers at INGV have led to the creation of increasingly sophisticated models with ever more detailed representations of the mechanical and thermal aspects of lava flows. Simple models based on the concept of maximum slope and stochastic perturbation of topography (DOWNFLOW; (Favalli *et al.*, 2005)) were integrated by models that included a more complete physical description, based on Cellular Nonlinear Networks and Cellular Automata (MAGFLOW; (Vicari *et al.*, 2005; Del Negro *et al.*, 2008)), able to describe the spatial and temporal evolution of lava flow on the basis of given eruptive parameters. These models have been applied with success in collaboration with the Dipartimento di Protezione Civile for the creation of scenarios during the Mt Etna eruptions over the last years.

Although the latter models include a detailed and accurate physical description of the lava rheology, including thermal effects such as radiation, solidification and the dependency of the physical parameters on the temperature, they are inadequate for the description of more sophisticated thermal-based phenomena such as crust and lava tube formation and their rupture with consequent ephemeral vent opening.

Traditional methods such as finite volumes or finite elements also meet significant challenges in the simulation of lava flows, being tied to spatial discretization with fixed or adaptive meshes. The need to refine the discretization grid in correspondence of high gradients, when possible, is computationally expensive and with an often inadequate control of the error; for real-world applications, moreover, the information needed by the grid refinement may not be available (e.g. because the Digital Elevation Models are too coarse). Eulerian discretization has an additional problem with boundary tracking, which for complex fluids such as lava is further complicated by the presence of internal boundaries given by fluid inhomogeneity and presence of solidification fronts. Another problem is given by the need to solve the implicit system of equations needed to determine the pressure at every time-step for every grid-point. Lagrangian methods such as finite elements, instead, are challenged by the problems related to the continuous and deformable nature of the lava flow, which inevitably leads to significant deformations in the finite element structure, with consequent loss of accuracy and the need for remeshing.

An alternative approach is offered instead by meshfree particle methods (Hockney and Eastwood, 1988) that solve, in a natural way, most of the problems connected to the dynamics of complex fluids. Particle methods discretize the fluid using nodes which are not forced on a given topological structure: boundary treatment is therefore implicit and automatic; the movement freedom of the particles also permits the treatment of deformations without incurring in any significant penalty; finally, the accuracy is easily controlled by the insertion of new particles where needed.

2 The Smoothed Particle Hydrodynamics (SPH) Method

To this purpose, a new model has been developed, based on the Smoothed Particle Hydrodynamics (SPH) meshless method. Formulated by at the end of the '70s by (Gingold and Monaghan, 1977) and (Lucy, 1977) for astrophysics problem, the SPH method has recently seen a growing involvement for fluid dynamic applications (Monaghan, 1994; Dalrymple *et al.*, 2009). As a particle method, SPH doesn't suffer from the limitations traditional mesh-based numerical methods (finite differences, finite volumes, finite elements) encounter when describing a complex, free-surface fluid flow; in comparison to other particle methods, SPH also provides additional benefits such as the automatic preservation of mass and the direct computation of most physical quantities (e.g. pressure) without resorting to large, sparse implicit systems.

The underside of the SPH method is that it is necessary to employ a number of particles higher than the number of nodes in grid methods to achieve simulations of comparable resolution, thus increasing the computational requirements. However, since most calculations in SPH algorithms are direct, this method can be parallelized to a much higher degree than most traditional mesh methods, a characteristic that makes the SPH method particularly favorable to implementation on highly parallel computational hardware such as modern video cards.

3 Scientific computing on graphic cards

Since the introduction of 3D rendering on computers, video cards have evolved from simple devices dedicated to video output into powerful parallel computing devices. The graphical processing units (GPUs) on modern video cards often surpasses the computational power of the CPU that drives them. Until recently,

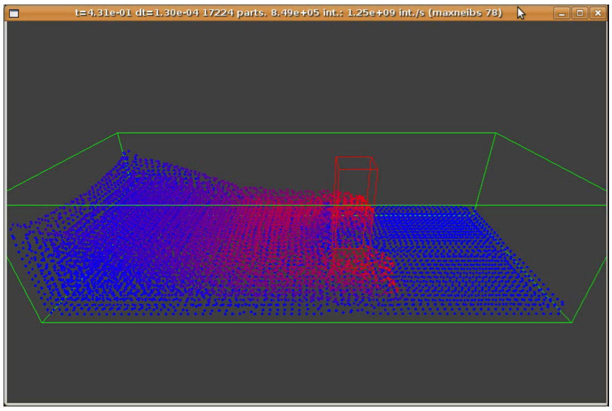

Figure 1. GPU SPH simulation running on an NVIDIA GTX 280, with an average 1.2×10^9 interactions per second

however, such computational power has been limited to the rendering of complex 3D scenes, satisfying the needs of computer gamers and professional designers.

The increasing computational power of GPUs has led to a growing interest in their usage for computation beyond video rendering; their computational power these days allows turning a desktop computer into a teraflop high-performance computer able to match the most expensive clusters in terms of performance, but at a fraction of cost, both in terms of initial price and total cost of ownership. However, full exploitation of their capabilities requires appropriate tools and problems that are computational rather than data-intensive.

Previously, General Programming for the GPU (GPGPU) has relied mostly on the OpenGL standard, an architecture designed to standardize programming of the most common operations needed to render detailed static and dynamic 3D scenes. Its usage for more generic numerical computations requires an intimate knowledge of computer graphics and a number of programming tricks to convert mathematical operations into equivalent graphical rendering operations and, conversely, to interpret the rendered scene as mathematical results of the operations. These transformations exact a significant coding cost and impose a number of constraints on the operations that can be performed.

4 SPH on CUDA

The CUDA architecture and programming language, introduced by NVIDIA in the spring of 2007, works around these limitations by allowing GPU programming using the C language extended to handle the specific needs of their GPU and its interfacing with the CPU host. While traditional GPGPU programming uses the GPU as a coprocessor for the CPU, performing only the most expensive computations on the graphics card while keeping much of the algorithm structure on the CPU host, CUDA encourages porting nearly all computations to the GPU.

Our lava simulation model uses the SPH method with a GPU implementation in CUDA to achieve a high computational performance. Both static problems with purely thermal effects (such as lava lake solidification) and dynamic problems with a complete lava flow can be simulated.

A direct comparison between SPH and finite elements for the lava lake solidification shows the superiority of the SPH method, that guarantees a significantly improved accuracy in proximity of the contact area of two or more solidification fronts. Although purely static problems are less of a direct interest for hazard assessment, they are still an important tool for risk management, when used in conjunction with dynamic problems, for example when evaluating the effects of barrier formation with respect to lava flow diversion or halting, and the consequent formation of lava lakes.

For the dynamic part of the model, the SPH algorithms are based on the ones of the SPHysics simulator, with the addition of thermal effects and the treatment of non-Newtonian fluids. Following recent developments in the physical modeling of lava flow rheology, both Bingham and power-law fluids can be simulated by our code.

References

Dalrymple, R.A., M. Gómez-Gesteira, B.D. Rogers, Panizzo A., Zou S., A.J.C. Crespo, G. Cuomo and M. Narayanaswamy (2009). *Smoothed Particle Hydrodynamics for Water Waves*. Q.Ma, ed., World Scientific Press.

Del Negro, C., L. Fortuna, A. Hérault and A. Vicari (2008). Simulations of the 2004 lava flow at etna volcano by the magflow cellular automata model. *Bull. Volcanol.* **70**(7), 805–812.

Favalli, M., M.T. Pareschi, A. Neri and I. Isola (2005). Forecasting lava flow paths by a stochastic approach. *Geophys. Res. Lett.* **32**, L03305.

Gingold, R.A. and J.J. Monaghan (1977). Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon. Not. R. Astr. Soc.* **181**, 375–389.

Hockney, R.W. and J.W. Eastwood (1988). *Computer simulation using particles*. Hilger, Bristol U.K.

Lucy, L. (1977). A numerical approach to the testing of fusion process. *Journal Astronomical* **82**, 1013–1024.

Monaghan, J.J. (1994). Simulating free surface flows with sph. *J. Comp. Phys.* **110**, 399–406.

Vicari, A., A. Hérault and C. Del Negro (2005). Modeling lava flows by smoothed particle hydrodynamics. In: *XXIV GNGTS*. oral presentation.