

Simulating complex fluids with Smoothed Particle Hydrodynamics

Vito Zago^{1,*}, Giuseppe Bilotta¹, Annalisa Cappello¹, Robert A. Dalrymple², Luigi Fortuna³, Gaetana Ganci¹, Alexis Hérault^{1,4}, Ciro Del Negro¹

¹ Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Catania, Osservatorio Etneo, Italy

² Department of Civil Engineering, Johns Hopkins University, Baltimore (MD), USA

³ Dipartimento di Ingegneria Elettrica, Elettronica e Informatica, Università di Catania, Italy

⁴ Conservatoire National des Arts et Métiers, Département Ingénierie Mathématique, Paris, France

Article history

Received January 20, 2017; accepted November 12, 2017.

Subject classification:

Algorithms and implementation; Computational geophysics: General or miscellaneous.

ABSTRACT

Complex fluid dynamics encompasses a large variety of flows, such as fluids with non-Newtonian rheology, multi-phase and multi-fluid flows (suspensions, lather, solid/fluid interaction with floating objects, etc.), violent flows breaking waves, dam-breaks, etc.), fluids with thermal dependencies and phase transition or free-surface flows. Correctly modeling the behavior of such flows can be quite challenging, and has led to significant advances in the field of Computational Fluid Dynamics (CFD). Recently, the Smoothed Particle Hydrodynamics (SPH) method has emerged as a powerful alternative to more classic CFD methods (such as finite volumes or finite elements) in many fields, including oceanography, volcanology, structural engineering, nuclear physics and medicine. With SPH, the fluid is discretized by means of particles and thanks to the meshless, Lagrangian nature of the model, it easily allows the modeling and simulation of both simple and complex fluids, simplifying the treatment of aspects that can be challenging with more traditional methods: dynamic free surfaces, large deformations, phase transition, fluid/solid interaction and complex geometries. In addition, the most common SPH formulations are fully parallelizable, which favors implementation on high-performance parallel computing hardware, such as modern Graphics Processing Units (GPUs). We present here how GPUSPH, an implementation of the SPH method that runs on GPUs, can model a variety of complex fluids, highlighting the computational challenges that arise in its applications to problem of great interest in volcanology.

1. Introduction

The ability to accurately model the behavior of fluids under the most diverse conditions is extremely important in both scientific research and applied industry,

and has led to the development of Computational Fluid Dynamics (CFD).

Traditional approaches to CFD have been based on an Eulerian description of the fluid, with fixed reference grids (finite differences, finite volumes). These approaches are often computationally very efficient, but may have issues with complex geometries, moving boundaries or the tracking of the free surface and of internal interfaces (such as the boundaries between fluids in multi-fluid simulations, or the front in phase transitions). Lagrangian, mesh-based approaches such as finite elements can overcome some of these issues, but their application to highly dynamic flows such as dam-breaks is limited due to the possible fragmentation of the flow and the computationally expensive need for remeshing when the fluid undergoes large deformations.

These methods have been used to model and study geophysical flows under specific conditions, such as lava tubes [Filippucci and Dragoni 2013], channel flows [Costa and Macedonio 2005], or other simple geometries [Fernandez Nieto and Narbona-Reina 2016]. In some cases, it is also possible to find analytical solutions to the equations governing the flow [Tallarico and Dragoni 1999], which can then be used in models still relying on mesh-based approaches [Cappello et al. 2016]. However, many geophysical flows present characteristics, such as multiple fluids or phases, non-Newtonian or temperature-dependent rheology, violent or otherwise highly dynamic behavior, which are challenging to model with the classical approaches.

An alternative to mesh-based method has recently emerged in the form of Lagrangian, mesh-free approaches. Among these, Smoothed Particle Hydrodynamics (SPH) has found a large number of applications

in many fields [Monaghan 2005] since its inception in 1977. In SPH the problem is discretized by means of particles which act both as interpolation nodes and to carry physical information (mass, density, temperature, velocity, etc.) about a small volume of the fluid. The evolution of the properties of each particle is therefore determined by a discretized form of the physical equations (Navier-Stokes equations for the motion, thermal equation for the temperature, etc.), so that the properties of the fluid at any point in space can be computed by a smoothed average of the properties of the neighboring particles.

For the simulation of complex fluids, SPH presents a number of advantages over traditional mesh based methods, such as the implicit tracking of the free surface and of any internal interface (such as solidification fronts) and the lack of restrictions in the geometry of the fluid and of the containing geometries. The main downsides are lower accuracy and, for the classical formulation, the use of a weakly-compressible rather than a fully incompressible fluid model, such that the speed of sound can significantly limit the time-step size then used with explicit time-stepping schemes. On the other hand, the standard weakly-compressible SPH formulation has the benefit of being completely parallelizable, since the evolution of each particle at each time-step can be computed independently from that of the others, which makes SPH quite suitable for implementation on massively parallel hardware, such as modern Graphics Processing Units [H erault et al. 2010].

In this paper we present some applications of GPUSPH [H erault et al. 2011, Bilotta et al. 2016], a GPU implementation of the SPH method specialized for the modeling of complex fluids in three dimensions. We introduce first the equations governing the fluid motion and heat exchange, the approach used to discretize them using SPH, and finally some applications including validation against standard analytical problems and qualitative results in more sophisticated examples taken from applications of GPUSPH to both industrial topics and geophysics, with a particular attention to volcanology. In the application section we will highlight the computational challenges faced by the model, and discuss the possible approaches to resolve them.

2. Notation

Throughout this manuscript we use the following conventions:

- Vectors will be written in bold;
- The reference coordinate system is Cartesian orthogonal with axes x_1 , x_2 and x_3 ;
- Latin indices denote coordinates;
- Greek indices denote particles;

- Physical quantities and mathematical expressions follow the common SPH notation, summarized in Table 1.

| Symbol | Meaning | Symbol | Meaning | |
|----------------------------|--|--|--|--------------|
| δ | Dirac distribution | $W_{\beta\alpha}$ | $W(\mathbf{x}_{\beta\alpha}, h)$ | |
| \mathbf{n} | outward normal | $f_{\beta\alpha}$ | $f(\mathbf{x}_{\beta}) - f(\mathbf{x}_{\alpha})$ | |
| \mathbf{x} | position | P | pressure | |
| \mathbf{u} | velocity | c_p | heat capacity | |
| $\frac{D}{Dt}$ | total derivative operator | $\frac{\partial}{\partial t} + u_i \frac{\partial}{\partial x_i}$ | κ | conductivity |
| τ_{ij} | Reynolds viscous stress tensor | ϵ | emissivity | |
| ϵ_{ij} | strain tensor | $\frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$ | \mathbf{g} | gravity |
| T | temperature | ρ | density | |
| h | smoothing length | μ | dynamic viscosity | |
| Δp | average inter-particle spacing | ν | kinematic viscosity | |
| $\mathbf{x}_{\beta\alpha}$ | $\mathbf{x}_{\beta} - \mathbf{x}_{\alpha}$ | m | mass | |
| W | smoothing kernel | V | volume | |

Table 1. Mathematical notations.

3. Governing equations of fluid flow and heat transfer

We model the dynamics of fluids according to the Navier-Stokes equations for continuity of mass and forces balance:

$$\frac{D\rho}{Dt} = -\rho \nabla \cdot \mathbf{u} \quad (1)$$

(where ρ is the density, \mathbf{u} the velocity and D/Dt the total derivative) and

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla P + \nabla \cdot (\mu \nabla \mathbf{u}) + \mathbf{F} \quad (2)$$

(where P is the pressure, μ the dynamic viscosity, and \mathbf{F} represents external forces, such as gravity). The fluid is assumed to be weakly-compressible, so that the pressure can be obtained directly from the density, using Cole's equation of state [Cole 1948, Batchelor 1974]:

$$P(\rho) = c_0^2 \frac{\rho_0}{\gamma} \left(\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right) \quad (3)$$

where ρ_0 is the at-rest density, γ is the polytropic constant and c_0 the speed of sound. A weakly-compressible regime is achieved if c_0 is at least an order of magnitude higher than the maximum velocity experienced during the flow.

The thermal evolution is described by the heat equation:

$$\rho c_p \frac{DT}{Dt} = \nabla \cdot (\kappa \nabla T) \quad (4)$$

where T is the temperature, c_p the specific heat at constant pressure, and κ the thermal conductivity.

3.1. Viscosity models

Equation 2 only applies to Newtonian fluids, for which the shear stress $\boldsymbol{\tau}$ is proportional to the strain rate $\dot{\boldsymbol{\epsilon}}$. The dynamic viscosity is the coefficient of proportionality, and it may depend on local physical and chemical properties of the fluid (such as the temperature, density, composition, etc.), but not on the shear stress or strain rate themselves.

A fluid where the relationship between stress and strain rate is not linear is known as non-Newtonian fluid, and many fluids of great interest in industrial applications, biology and geophysics are indeed non-Newtonian. Among non-Newtonian fluids, there is a large subset of fluids (called generalized Newtonian fluids), for which it is still possible to use the same form (Equation 2) of the Navier-Stokes equations, provided that the dynamic viscosity μ is replaced by an effective viscosity function $\mu_{\text{eff}} = \mu_{\text{eff}}(\dot{\boldsymbol{\epsilon}})$ such that $\boldsymbol{\tau} = \mu_{\text{eff}}(\dot{\boldsymbol{\epsilon}})\dot{\boldsymbol{\epsilon}}$.

The effective viscosity may still depend on other physical properties of the fluid. For our model we consider the Herschel-Bulkley rheology, a generalized Newtonian rheology characterized by a yield strength τ_0 , a power law exponent n and a consistency index k such that $\boldsymbol{\epsilon} = \mathbf{0} \Leftrightarrow |\boldsymbol{\tau}| \leq \tau_0$, and

$$|\boldsymbol{\tau}| = \tau_0 + k|\dot{\boldsymbol{\epsilon}}|^n \quad (5)$$

otherwise. As special cases of the Herschel–Bulkley rheology, we obtain:

Newtonian fluids for $\tau_0 = 0, n=1$ (in which case k is the viscosity);

power-law fluids for $\tau_0 = 0, n \neq 1$;

Bingham fluids for $\tau_0 \neq 0, n=1$.

For an Herschel-Bulkley fluid, the effective viscosity can be written [Zhu et al. 2005] as.

$$\mu_{\text{eff}} = \mu_{\text{eff}}(\dot{\boldsymbol{\epsilon}}) = \frac{\tau_0}{|\dot{\boldsymbol{\epsilon}}|} + k|\dot{\boldsymbol{\epsilon}}|^{n-1} \quad (6)$$

for $|\boldsymbol{\tau}| > \tau_0$, with $|\dot{\boldsymbol{\epsilon}}| = \sqrt{(\boldsymbol{\epsilon} : \boldsymbol{\epsilon})/2}$ and

$$\boldsymbol{\epsilon} : \boldsymbol{\epsilon} = \epsilon_{xx}^2 + \epsilon_{yy}^2 + \epsilon_{zz}^2 + 2(\epsilon_{xy}^2 + \epsilon_{xz}^2 + \epsilon_{yz}^2) \quad (7)$$

4. Mathematical basis of Smoothed Particles Hydrodynamics

To solve numerically the Navier-Stokes and thermal equations seen in section 3, we discretize their right-hand sides using the Smoothed Particle Hydrodynamics method. In our presentation of the method and its application to fluid dynamics we follow Liu and Liu [2003] and Monaghan [2005].

4.1. SPH discretization of fields

Let us consider a field f defined on a domain Ω . By

definition of Dirac's delta distribution δ , the value assumed by f at any location $\bar{\mathbf{x}} \in \Omega$ can be expressed, with typical abuse of notation, as

$$f(\bar{\mathbf{x}}) = \int_{\Omega} f(\mathbf{x})\delta(\mathbf{x} - \bar{\mathbf{x}}) d\mathbf{x}. \quad (8)$$

Let us now approximate Dirac's delta with a family of functions $W(\cdot, h)$, parametrized by the length h , and satisfying

$$\int_{\Omega} W(\mathbf{x} - \bar{\mathbf{x}}, h) d\mathbf{x} = 1 \quad (9)$$

and

$$\lim_{h \rightarrow 0} W(\mathbf{x} - \bar{\mathbf{x}}, h) = \delta(\mathbf{x} - \bar{\mathbf{x}}) \quad (10)$$

where the limit is to be taken in the sense of distributions.

These W functions are the *smoothing kernels* and the parameter h is the *smoothing length*. By substituting the smoothing kernel into (8) we get an initial approximation of f as

$$f(\bar{\mathbf{x}}) \approx \int_{\Omega} f(\mathbf{x})W(\mathbf{x} - \bar{\mathbf{x}}, h) d\mathbf{x} \quad (11)$$

We can further approximate the integral with a summation of a finite set of particles $\mathbf{x}_1, \dots, \mathbf{x}_{\alpha}, \dots, \mathbf{x}_N$ with volume V_{α} , obtaining:

$$f(\bar{\mathbf{x}}) \approx \sum_{\alpha=1}^N f(\mathbf{x}_{\alpha})W(\mathbf{x}_{\alpha} - \bar{\mathbf{x}}, h)V_{\alpha} \quad (12)$$

where the summation is extended to all particles. However, if the W are chosen with compact support, the summation can be extended only to the particles in a small neighborhood of $\bar{\mathbf{x}}$.

From equations (11) and (12), we observe that the SPH approximation has two main sources of error:

1. approximation of the Dirac's delta with the smoothing kernels;
2. discretization of the domain by means of a finite set of particles.

The first approximation vanishes as $h \rightarrow 0$, the second error is controlled by the average inter-particle spacing Δp and vanishes for $\Delta p \rightarrow 0$. Additionally, Δp must tend to zero faster than h to ensure consistency for the method. Most applications adopt a fixed ratio $h/\Delta p$ usually in the range [1.3, 1.5].

4.2. SPH discretization of gradients

The field discretization with SPH can be used to reconstruct a field at any point given its values on a set of particles, but the same principle can also be used to

discretize spatial gradients. To this end, we will assume further that the smoothing kernels W have compact support and radial symmetry, so that they only depend on $|\mathbf{x} - \bar{\mathbf{x}}|$.

Let us consider the vector field constituted by $\nabla f(\mathbf{x})$; recalling equation (11) we can write

$$\nabla f(\bar{\mathbf{x}}) \approx \int_{\Omega} \nabla_{\mathbf{x}} f(\mathbf{x}) W(\mathbf{x} - \bar{\mathbf{x}}, h) d\mathbf{x} \quad (13)$$

By applying Green's theorem, we rewrite the right-hand side obtaining

$$\nabla f(\bar{\mathbf{x}}) \approx \int_{\Sigma} f(\mathbf{x}) W(\mathbf{x} - \bar{\mathbf{x}}, h) \mathbf{n} d\Sigma - \int_{\Omega} f(\mathbf{x}) \nabla_{\mathbf{x}} W(\mathbf{x} - \bar{\mathbf{x}}, h) d\mathbf{x} \quad (14)$$

where $\Sigma = \partial\Omega$ is the boundary of the domain and \mathbf{n} its normal. Given the compact support for W , the first integral in (14) is zero if $\bar{\mathbf{x}}$ is far from the boundary; additionally, by symmetry of W , we have $\nabla_{\mathbf{x}} W(\mathbf{x} - \bar{\mathbf{x}}, h) = -\nabla_{\bar{\mathbf{x}}} W(\mathbf{x} - \bar{\mathbf{x}}, h)$, so that the discretized version can be written as

$$\nabla f(\bar{\mathbf{x}}) \approx \sum_{\alpha=1}^N f(\mathbf{x}_{\alpha}) \nabla_{\bar{\mathbf{x}}} W(\mathbf{x}_{\alpha} - \bar{\mathbf{x}}, h) \mathbf{V}_{\alpha}. \quad (15)$$

This expression allows us to compute (an approximation of) the gradient of the field f without having its analytical expression, relying instead on the gradient of the smoothing kernel. When applied to a particle β , the equation takes the form:

$$\nabla f(\mathbf{x}_{\beta}) \approx \sum_{\alpha}^N f(\mathbf{x}_{\alpha}) \nabla_{\beta} W_{\alpha\beta} \mathbf{V}_{\alpha} \quad (16)$$

where $W_{\alpha\beta} = W(\mathbf{x}_{\alpha} - \mathbf{x}_{\beta}, h)$. This can be symmetrized (thus helping preserve conservation properties of the analytical equations) by subtracting the gradient of the function identically equal to $f(\mathbf{x}_{\beta})$, obtaining:

$$\nabla f(\mathbf{x}_{\beta}) \approx \sum_{\alpha}^N f_{\alpha\beta} \nabla_{\beta} W_{\alpha\beta} \mathbf{V}_{\alpha} \quad (17)$$

where $\mathbf{x}_{\alpha\beta} = f(\mathbf{x}_{\alpha}) - f(\mathbf{x}_{\beta})$.

Finally, we observe that due to the symmetry of W , its gradient can be written as:

$$\nabla_{\beta} W_{\alpha\beta} = \frac{\mathbf{x}_{\alpha\beta}}{|\mathbf{x}_{\alpha\beta}|} \left. \frac{\partial W(\mathbf{r}, h)}{\partial \mathbf{r}} \right|_{\mathbf{r}=\mathbf{x}_{\alpha\beta}}. \quad (18)$$

where $\mathbf{x}_{\alpha\beta} = \mathbf{x}_{\alpha} - \mathbf{x}_{\beta}$.

It is therefore convenient to choose a kernel such that

$$F(\mathbf{r}) = \frac{1}{r} \frac{\partial W}{\partial \mathbf{r}} \quad (19)$$

has an analytical expression, and given $F_{\alpha\beta} = F(\mathbf{x}_{\alpha\beta})$, we can write $\nabla_{\alpha} W_{\alpha\beta} = \mathbf{x}_{\alpha\beta} F_{\alpha\beta}$.

5. SPH discretization of the fluid equations

We can apply the SPH discretization to the continuity Equation (1), obtaining

$$\frac{D\rho_{\beta}}{Dt} = \sum_{\alpha} m_{\alpha} \mathbf{u}_{\alpha\beta} \nabla_{\beta} W_{\alpha\beta}. \quad (20)$$

For the momentum (Equation 2) and thermal (Equation 4) equations, we additionally need a discretization for the Laplacian, for which we follow Brookshaw [1985], Morris et al. [1997] and Cleary and Monaghan [1999]. The momentum equation then takes the form:

$$\frac{D\mathbf{u}_{\beta}}{Dt} = - \sum_{\alpha} \left(\frac{P_{\alpha}}{\rho_{\alpha}^2} + \frac{P_{\beta}}{\rho_{\beta}^2} \right) F_{\alpha\beta} m_{\alpha} \mathbf{x}_{\alpha\beta} + \sum_{\alpha} \frac{2\bar{\mu}_{\alpha\beta}}{\rho_{\alpha}\rho_{\beta}} F_{\alpha\beta} m_{\alpha} \mathbf{u}_{\alpha\beta} + \mathbf{g} \quad (21)$$

where $\bar{\mu}_{\alpha\beta}$ is the harmonic mean of μ_{α} and μ_{β} , and $\mathbf{g} = \mathbf{F}/\rho$, while the thermal equation becomes:

$$\frac{DT_{\beta}}{Dt} = \frac{1}{\rho_{\beta} c_p} \sum_{\alpha} \frac{k_{\alpha\beta} T_{\alpha\beta}}{\rho_{\alpha}} F_{\alpha\beta} \quad (22)$$

For non-Newtonian fluids, the viscosity μ_{β} of each particle is computed before the actual forces, using the specific rheological law, from the strain rate:

$$\dot{\boldsymbol{\epsilon}}_{\beta} = (\nabla \mathbf{u})_{\beta} + (\nabla \mathbf{u})_{\beta}^T \quad (23)$$

with the $(\nabla \mathbf{u})_{\beta}$ tensor computed in SPH form as:

$$(\nabla \mathbf{u})_{\beta} = \sum_{\alpha} \mathbf{u}_{\alpha\beta} \mathbf{r}_{\alpha\beta} F_{\alpha\beta} \frac{m_{\alpha}}{\rho_{\alpha}}. \quad (24)$$

The evolution of temperature, density and velocity for each particle is finally obtained by numerical integration of (20), (21) and (22) over time. GPUSPH uses a predictor-corrector time integration scheme, which can be described as follows, for a given time-step ∇t :

1. (in the non-Newtonian case) compute viscosities
 $\mu^{(n)} = \mu(\mathbf{x}^{(n)}, \mathbf{u}^{(n)}, \rho^{(n)}, T^{(n)})$
2. compute accelerations
 $\mathbf{a}^{(n)} = \mathbf{a}(\mathbf{x}^{(n)}, \mathbf{u}^{(n)}, \rho^{(n)}, T^{(n)}, \mu^{(n)})$,
 density derivatives
 $\dot{\rho}^{(n)} = \dot{\rho}(\mathbf{x}^{(n)}, \mathbf{u}^{(n)}, \rho^{(n)}, T^{(n)})$,
 and temperature derivatives
 $T^{(n)} = T(\mathbf{x}^{(n)}, \mathbf{u}^{(n)}, \rho^{(n)}, T^{(n)})$,
3. compute half-step intermediate positions, velocities, densities and temperatures:
 - a) $\mathbf{x}^{(n*)} = \mathbf{x}^{(n)} + \mathbf{u}^{(n)} \frac{dt}{2}$,
 - b) $\mathbf{u}^{(n*)} = \mathbf{u}^{(n)} + \mathbf{a}^{(n)} \frac{dt}{2}$,
 - c) $\rho^{(n*)} = \rho^{(n)} + \dot{\rho}^{(n)} \frac{dt}{2}$,
 - d) $T^{(n*)} = T^{(n)} + \dot{T}^{(n)} \frac{dt}{2}$,
4. (in the non-Newtonian case) compute intermediate viscosities

$$\mathbf{u}^{(n^*)} = \mathbf{u}(\mathbf{x}^{(n^*)}, \mathbf{u}^{(n^*)}, \rho^{(n^*)}, T^{(n^*)})$$

5. compute corrected accelerations

$$\mathbf{a}^{(n^*)} = \mathbf{a}(\mathbf{x}^{(n^*)}, \mathbf{u}^{(n^*)}, \rho^{(n^*)}, T^{(n^*)}, \mathbf{u}^{(n^*)}),$$

density derivatives

$$\dot{\rho}^{(n^*)} = \dot{\rho}(\mathbf{x}^{(n^*)}, \mathbf{u}^{(n^*)}, \rho^{(n^*)}),$$

and temperature derivatives

$$T^{(n^*)} = T(\mathbf{x}^{(n^*)}, \mathbf{u}^{(n^*)}, \rho^{(n^*)}, T^{(n^*)}),$$

6. compute new positions, velocities, densities and temperatures:

$$a) \quad \mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \left(\mathbf{u}^{(n)} + \mathbf{a}^{(n^*)} \frac{dt}{2} \right) dt,$$

$$b) \quad \mathbf{u}^{(n+1)} = \mathbf{u}^{(n)} + \mathbf{a}^{(n^*)} dt,$$

$$c) \quad \rho^{(n+1)} = \rho^{(n)} + \dot{\rho}^{(n^*)} dt,$$

$$d) \quad T^{(n+1)} = T^{(n)} + \dot{T}^{(n^*)} dt.$$

Adaptive time-stepping can be used, in which case the maximum allowed time-step is computed based on constraints, similar to the Courant-Friedrichs-Lewy conditions, derived from the sound speed, the forces magnitude and the viscous and thermal diffusivity:

$$\Delta t_{\beta} \leq \min \left\{ C_1 \sqrt{\frac{h}{\|\mathbf{a}_{\beta}\|}}, C_2 \frac{h}{c_{\beta}}, C_3 \frac{\rho_{\beta} h^2}{\mu_{\beta}}, C_4 \frac{\rho_{\beta} c_p h^2}{k} \right\} \quad (25)$$

where c_{β} is the sound speed at density ρ_{β} and the C_1, C_2, C_3, C_4 stability constants in GPUSPH take the values $C_1 = C_2 = 0.3, C_3 = 0.125$ and $C_4 = 0.1$. The effective time-step is then computed as $\Delta t = \min_{\beta} \Delta t_{\beta}$.

Some consideration can be done about applications involving highly viscous fluids, where the viscous stability condition may become predominant. The dependence on the square of the smoothing length causes in fact a stronger reduction of the time-step as the resolution increases. This constitutes a relevant problem in simulating geophysical flows, like lava, where large values of viscosity are commonly involved, causing simulation times to be very long.

6. GPUSPH applications and tests

We now show some applications of the GPUSPH platform, illustrating the wide range of possible applications and the benefits of the method. We will also discuss the computational performance of the model in the different contexts. All simulations were run with a smoothing length $h=1.3\Delta p$, which gives about 80 neighbors per particle. The kernel employed is the 5th order smoothing kernel proposed by Wendland [1995], defined by $W(r, h) = \tilde{W}(r, h)$, which in a general d-dimensional space takes the form

$$\tilde{W}(q) = C_{w,d}(2q+1)(1-q/2)^4 \quad 0 \leq q \leq 2; \quad (26)$$

the normalization constant in three dimensions is

$$C_{w,3} = \frac{21}{16\pi h^3}. \quad (27)$$

The Wendland kernel has an analytical expression for $F = 1/r (\partial W / \partial r)$, given by $F(r, h) = \tilde{F}(r, h)$, which in d dimensions is

$$\tilde{F}(q) = C_{F,d}(q-2)^3; \quad (28)$$

the normalization constant is given by

$$C_{F,d} = \frac{5}{8h^2} C_{w,d}. \quad (29)$$

In what follows, particular emphasis will be given on discussing simulation performances in reference to the number of particles, the allowed time-step length, and the resulting ratio between simulation run-time and simulated time. The simulations were all run on one or more NVIDIA TITAN X Maxwell GPUs, as discussed in the details of each problem.

6.1 High dynamic flows: Dam Break

The dam-break problem reproduces the breaking of a dam, a large body of water that suddenly gets in motion under gravitational effect and impacts the obstacles it encounters (Figure 1). This example shows the ability of GPUSPH to model highly dynamic flows, and makes it possible to appreciate the detail in the reproduction of the free surface, including splashes and droplets. The fluid density is $\rho=1000 \text{ kg/m}^3$ and the dynamic viscosity is $\mu = 10^{-4} \text{ Pa s}$. The simulation was run using two GPUs, showing the capability to distribute the computation across multiple domains, even with domain partitions not parallel to the computational axes [Rustico et al. 2012, 2014]. With a resolution of 170 particles per meter, about 500,000 particles are involved in the simulation, resulting in a time-step of 10^{-4} s , and a time ratio of 1:100 (one second of simulated time takes about 100 seconds of simulation).

6.2 Non-Newtonian Fluids

As discussed in subsection 3.1, GPUSPH is able to reproduce any rheology based on the Herschel–Bulkley model. We present here a validation of the implementation reproducing the Plane Poiseuille flow for a Bingham fluid. The distance between the top and bottom plane is 1m, and the domain is periodic in the flow and transverse directions (Figure 2). The employed fluid has consistency index $k=0.1 \text{ Pa s}$, density $\rho = 1 \text{ kg/m}^3$ and yield strength $\tau_0=0.01 \text{ Pa}$, and is driven by an external force conferring an acceleration $a = 0.05 \text{ m/s}^2$ result-

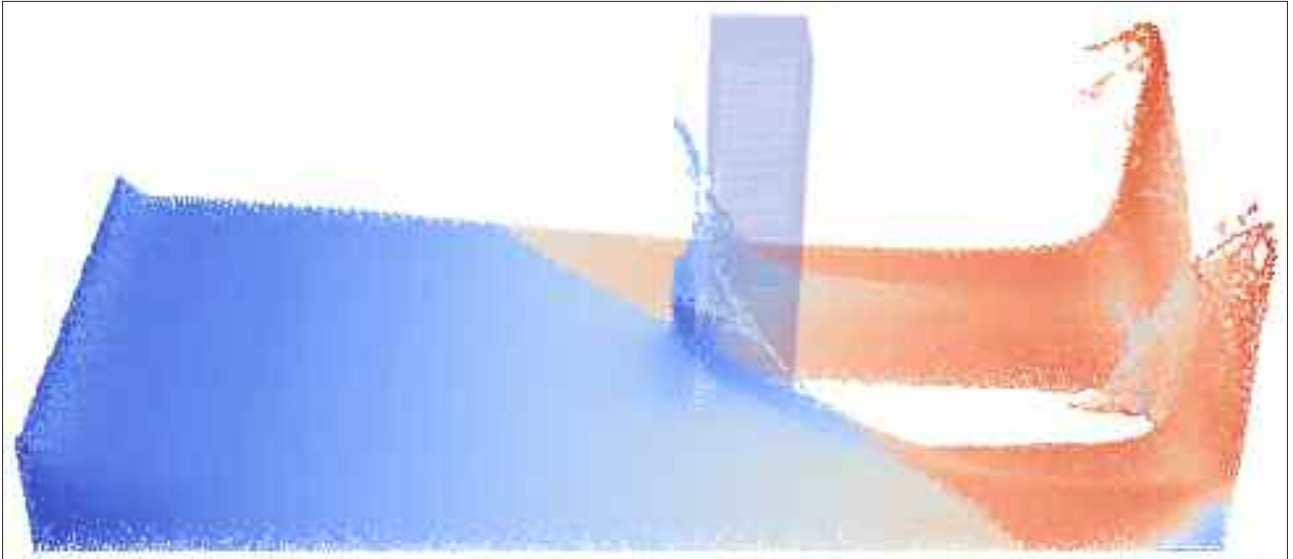


Figure 1. Dam break with obstacle. Particles are colored by device.

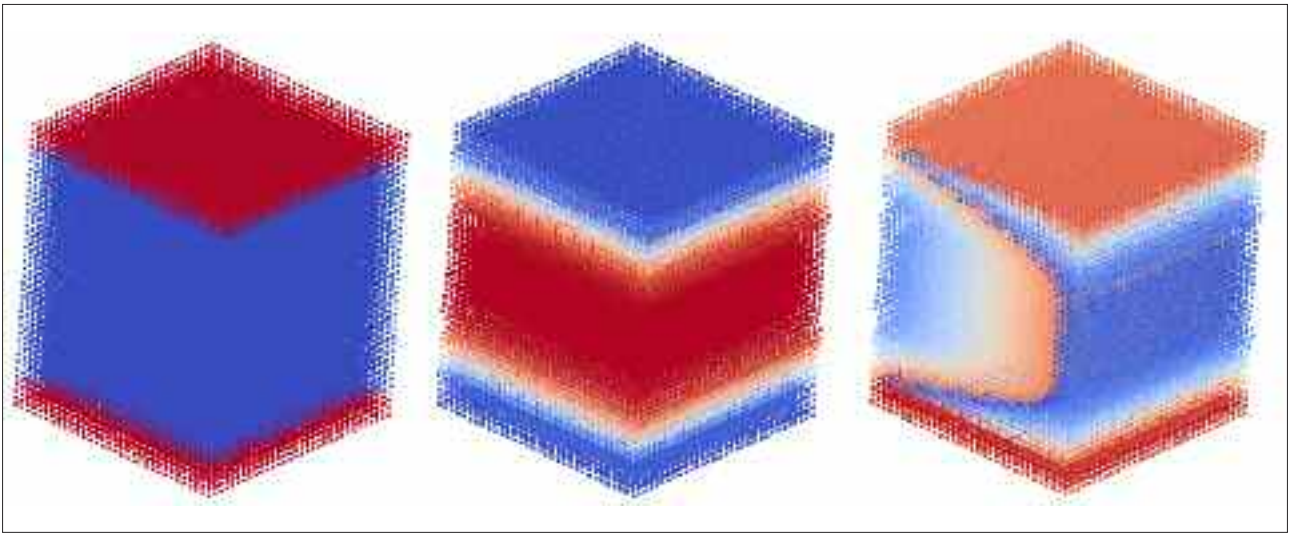


Figure 2. Plane Poiseuille flow. Particles colored by material (left), velocity (middle) and ID (right).

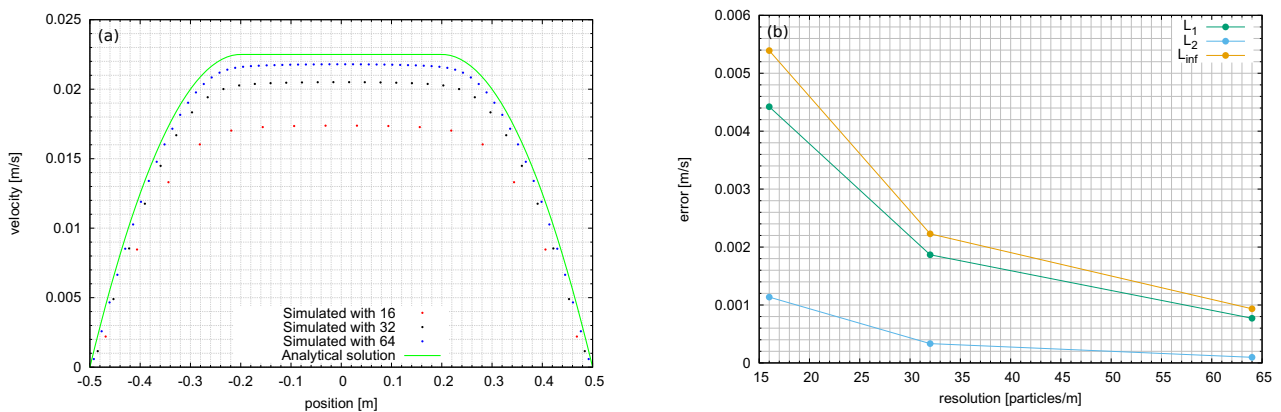


Figure 3. Validation of the GPUSPH model for the plane Poiseuille flow of a Bingham fluid. a) Velocity profiles of the Bingham Plane Poiseuille flow. Dotted lines are the simulated solutions, the green solid one is the analytical solution. b) L1, L2 and L_∞ error between numerical and analytical velocity profiles, for increasing resolution.

ing in a Reynolds number of 0.225.

The convergence of the numerical method (as discussed in section 4) is illustrated in Figure 3, showing

the convergence of the velocity profile (Figure 3a) and the decreasing error (Figure 3b) with increasing resolution.

The performance at different resolutions is shown in Table 2. We remark that the time-step in this case is limited by the viscosity, and thus decreases with the square of the linear resolution. This leads to a significant worsening of the ratio of simulated time to run-time, as discussed at the end of section 5, which in the worst case reaches here as low as 50 minutes to simulate 1 second.

| Resolution | Particles | Δt (seconds) | Time ratio |
|----------------|------------------|----------------------|------------|
| 16 particles/m | $6.1 \cdot 10^3$ | $5.5 \cdot 10^{-5}$ | 1 : 20 |
| 32 particles/m | $4.1 \cdot 10^4$ | $1.4 \cdot 10^{-5}$ | 1 : 170 |
| 64 particles/m | $3.0 \cdot 10^5$ | $3.5 \cdot 10^{-6}$ | 1 : 3000 |

Table 2. Performance of plane Poiseuille flow for a Bingham fluid.

6.3 Thermodynamics: natural convection

We tested the interaction of the thermal and mechanical models by simulating a thermal convection. We consider a box containing the fluid, with adiabatic walls, a heated bottom plate and a cooled top plate. As effect of the heating, the fluid gets in motion causing the inception of a Rayleigh-Bénard convective cell.

Additionally, the thermal model can be paired with the momentum equation by introducing temperature-dependent rheological parameters, as in Kaddiri et al. [2012], which alters the behavior of the convective cell as shown in Figure 4. The bottom plate is at temperature $T_b = 100$ K and the top plate at $T_t = 0$ K. Consider-

ing a Pearson constant $m = 10$, the kinematic viscosity at temperature $T = T_b$ is $\nu_b = 20\text{m}^2/\text{s}$ while at temperature $T = T_t$ is $\nu_t \approx 10^{-3}\text{m}^2/\text{s}$. The simulation is stable despite the large ratio (five orders of magnitude) between the minimum and maximum viscosity.

Again, the time-step is limited by the *maximum* viscosity, so that with a resolution of 33 particles per meter, resulting in about 55,000 particles and time-step in the order of 10^{-5} s, the simulated to run-time is nearly 1:400.

6.4 Multi-Fluid and multi-phase flows

GPUSPH allows the simultaneous simulation of several fluids, each one with its own rheological and thermal properties. SPH is quite appropriate for these kind of problems, since interfaces between two fluids are implicitly tracked.

Multiple liquids

Figure 5a illustrates a multi-fluid problem based on a lava-lamp: two fluids with opposite conditions in terms of density, rheology and thermal expansion coefficient are contained within an adiabatic box with heated bottom and cooled top. The red fluid has density $\rho_1 = 2\text{ kg/m}^3$, Bingham rheology with $\tau_0 = 0.001$ Pa, consistency index depending on the temperature as $k(T) = \exp(-2T)$ and thermal expansion coefficient $\alpha_1 = 0.6\text{ K}^{-1}$, while the blue one has $\rho_2 = 1\text{ kg/m}^3$, Newtonian rheology with kinematic viscosity $\mu_2 = 1\text{ m}^2/\text{s}$ and $\alpha_2 = 0.1\text{ K}^{-1}$. At a resolution of 33 particles per meter, about 30,800 particles are involved with a time-step in the order of 10^{-4} s, and using a single GPU the obtained time-ratio is 1:10.

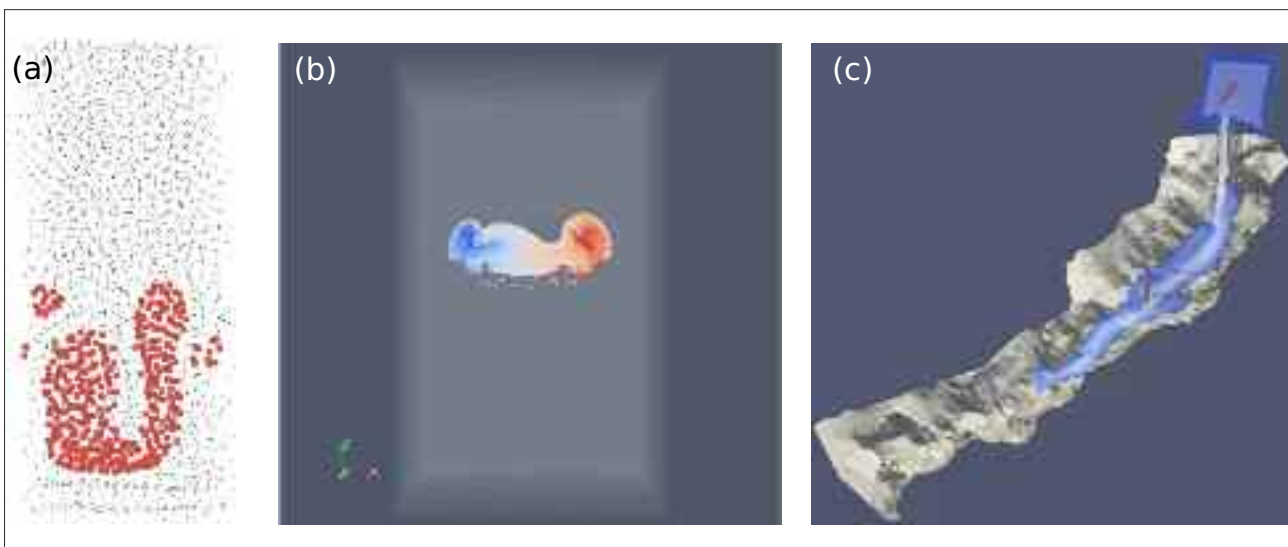


Figure 5. Multi-phase flows. a) Lava lamp with particles colored by fluid type. b) Rising bubble as example of gas-fluid interaction. c) Spill-way of the Goulours dam as example of solid-fluid interaction.

Liquid/Gas interaction

For multi-phase flows with liquid and gas phases, the entire system can be modeled using SPH. An example with an air bubble rising in water (1:1000 density ratio) is illustrated in Figure 5b, showing the fracture of the bubble surface towards the end of the ascension. As explained in detail in Bilotta [2014], for liquid/gas simulations where the compressibility does not play a major role, it is necessary to give the less dense fluid a higher sound-speed, so that both fluids are at the same quasi-compressible regimen. Due to this phenomenon, such multi-fluid simulations suffer from a time-step limitation dictated by the sound-speed of the less dense fluid. In the raising bubble example, a simulation with 16 particles in the bubble diameter requires over 500,000 particles, and the resulting time-step is in the order of 10^{-6} s. The time-ratio over 1:7400. It should be noted that in this case the time scale of the entire simulation is short, so that the impact of the high time-ratio is less significant than in larger-scale simulations.

Liquid/Solid interaction

For the fluid/solid interaction, GPUSPH computes the forces and torque acting on each moving object from the interaction of the boundary particles modeling the object with the neighboring fluid particles. The actual object dynamics, including object/object and objects/terrain interaction, is then handled by the ODE library (Smith, 2007). An example application is shown in Figure 5c, with floating tree trunks transported on the spillway of the Goulours Dam, on data

provided by EDF [Bilotta et al. 2014]. Water density is $\rho_w=1000 \text{ kg/m}^3$, whereas $\rho_t=0.7\rho_w$. Like in the standard dam-break case, this is a highly dynamic problem, but we now introduce a strong interaction with moving objects (the trunks) and complex geometries (the irregular floor of the spillway). The entire simulation consists of over 2,500,000 particles (inter-particle spacing of 0.2m), and the time-step (controlled by the sound speed of water) is in the order of 10^{-4} s. With a single GPU, we achieve a time ratio of about 1:500. This is due partly to the large number of particles, and partly to the use of the semi-analytical boundary model [Mayrhofer et al. 2015] for the fluid/terrain interaction, which requires a larger influence radius and thus a higher number of neighbors per particle.

6.5 Lava flows

Lava flows are highly complex flows exhibiting non-Newtonian rheology with temperature dependent viscosity and phase transition [Cappello et al. 2011]. The details of the rheology and its dependency on the physical and chemical properties of the lava are still under active research, and both analogical and numerical experiments are needed to improve our knowledge of lava rheology [Del Negro et al. 2005]. We illustrate here two proofs of concept cases of the application of GPUSPH for the simulation of lava flow emplacement.

Figure 6 shows the pouring of molten lava on a sloping ground, a typical set-up used to test the rheological properties of lava in analogical experiments: the ability to reproduce the same conditions is crucial to advance our knowledge of lava flows, since compar-

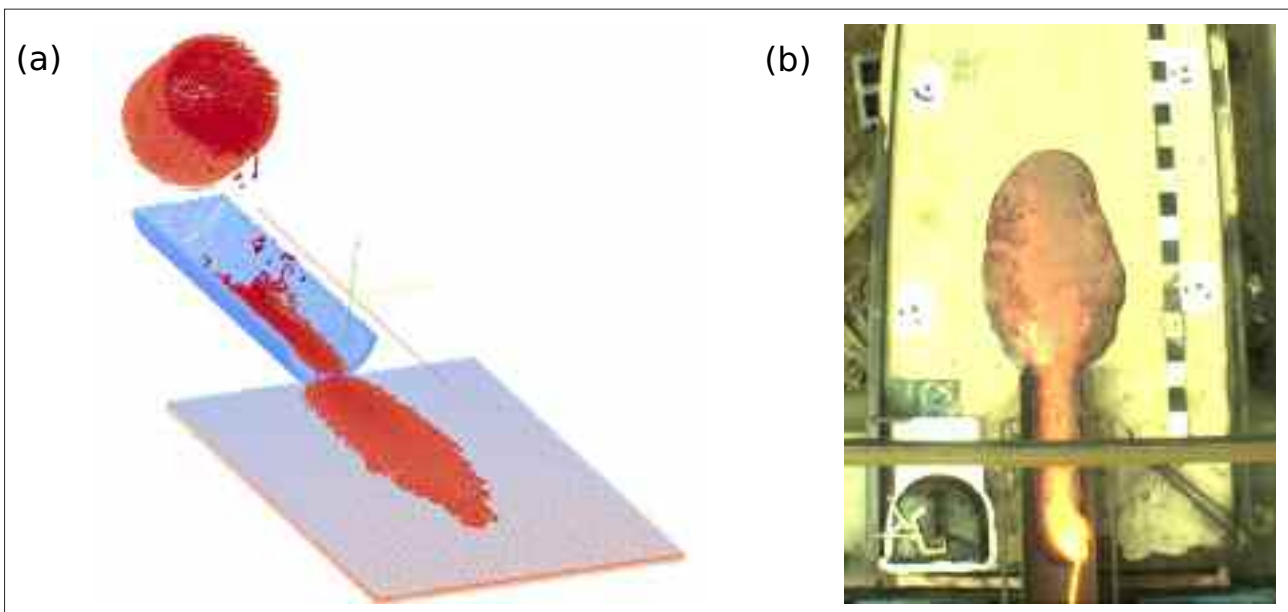


Figure 6. Simple lava flow emplacement. a) Simulated molten lava flow emplacement. b) Analogical experiment for molten lava flow emplacement, courtesy of USGS.

isons between numerical solutions and experimental data [Dietterich et al. 2015] allow both improvements to the numerical model and in an inversion process to better determine which kind of rheology better reproduces the experiment. The dynamical viscosity of the lava depends on the temperature following the Vogel-Fulcher-Tammann equation [Cordonnier et al. 2015], with coefficients determined experimentally [H. Dietterich, private communication]:

$$\log_{10}(\mu_{\text{exp}}) = -5.94 + \frac{5500}{T - 610} \quad (30)$$

where T is the temperature expressed in Kelvin and the density is $\rho = 2350 \text{ kg/m}^3$.

Thermal radiation is modeled as described in Bilotta et al. [2016], using heat capacity $c_p = 1600 \text{ J/(kg}\cdot\text{K)}$ and thermal emissivity $\epsilon = 0.96$.

A further step in our ongoing research is shown in Figure 7, illustrating a simple case of lava-water interaction. In this case, the sharp decrease in temperature caused by the contact with water results in a different emplacement compared to the lava flow behavior in absence of water. In our preliminary tests, simulations have been conducted with a viscosity that is 10 times lower than the experimental viscosity, due to computational constraints. Indeed, the main limitation in performing lava simulations is constituted by very long simulation times due to the high viscosity

1:150,000 (about two days for one simulated second) running on a cluster with four GPUs. Using the experimental law would have led to nearly one month of computational time to simulate a single second.

7. Conclusions and future work

We have shown how GPUSPH constitutes a very versatile fully three-dimensional platform for CFD, enabling the simulation of natural flows of Newtonian and non-Newtonian fluid with constant as well as temperature-dependent rheology.

The fully explicit nature of the SPH numerical method allows efficient parallelization on current high-performance parallel computing hardware such as GPUs, and can even scale to multiple GPUs. On the other hand, the explicit integration scheme can become a limiting factor in the simulation of highly viscous fluids, for which the time-step decreases with the *square* of the spatial resolution, quickly leading to very long simulation times.

In the application of GPUSPH to lava flows this can be of crucial importance, particularly in the cooling phase, where the viscosity of lava can grow by more than two orders of magnitude.

A solution to this issue, which we are currently working on, is the use of a semi-implicit integration scheme, in which the inviscid part of the Navier-Stokes equation is integrated with the explicit scheme

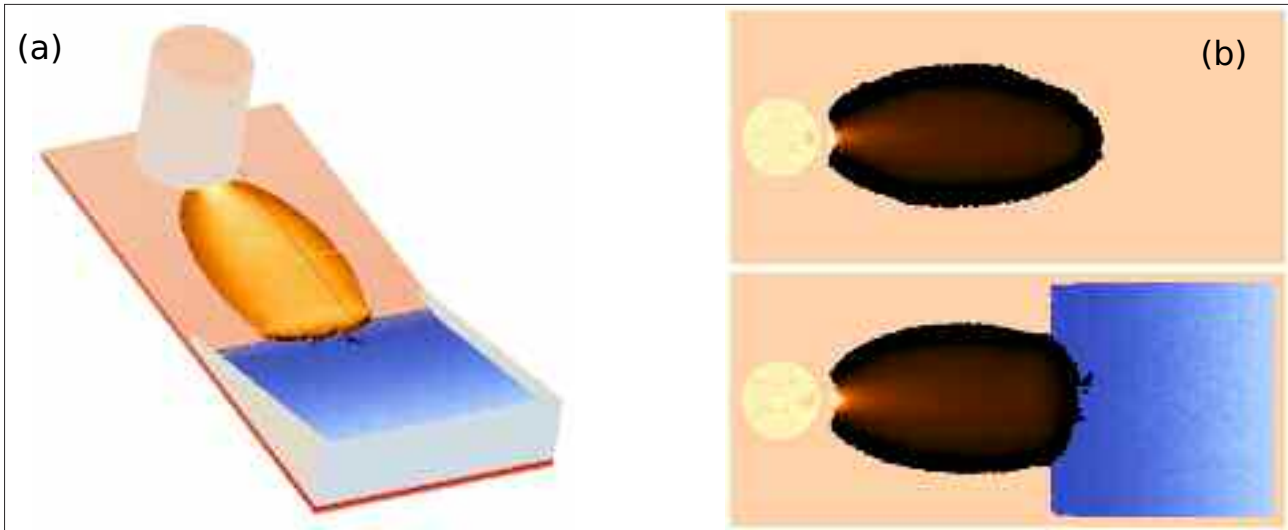


Figure 7. Lava flows simulations with GPUSPH. a) Lava-water interaction. b) Comparison between lava-water interaction and dry emplacement.

of lava, especially in the cooling phase. The high values of viscosity can determine time-steps in the order of 10^{-7} s or lower.

Considering a resolution of 16 particles per meter, each of the simulations above involved about 300,000 particles, with an effective time ratio of over

currently in use, whereas the viscous contribution is computed with an implicit scheme. While this incurs the need to solve a large, sparse implicit system, it provides significant benefits in computational times, while also improving the stability of the numerical method.

Acknowledgements. This work was developed within the framework of ATHOS Research Programme organized by INGV (Catania, Italy) and Johns Hopkins University (Baltimore, USA) and was partially supported from the DPC-INGV 2016 B2 Contract.

References

- Batchelor, G.K. (1974). An introduction to fluid mechanics. Cambridge University Press.
- Bilotta, G. (2014). GPU implementation and validation of fully three-dimensional multi-fluid SPH models. *Rapporti Tecnici INGV* (292).
- Bilotta, G., A. H erault, A. Cappello, G. Ganci, C. Del Negro (2016). GPUSPH: a smoothed particle hydrodynamics model for the thermal and rheological evolution of lava flows. *Geological Society, London, Special Publications*, 426, doi: 10.1144/SP426.24.
- Bilotta, G., A. H erault, A. Mayrhofer, D. Violeau (2014). Modelling real-life flows in hydraulic waterworks with GPUSPH. *Proceedings of the 9th SPHERIC international workshop*, 335-341.
- Brookshaw, L. (1985). A method of calculating heat diffusion in particle simulations. *Astronomical Society of Australia.*, 6 (2), 207.
- Cappello, A., A. H erault, G. Bilotta, G. Ganci, C. Del Negro (2016). MAGFLOW: a physics-based model for the dynamics of lava-ow emplacement. *Geological Society, London, Special Publications*, 426 (1), 357-373, doi: 10.1144/SP426.16.
- Cappello, A., A. Vicari, C. Del Negro (2011). Assessment and modeling of lava flow hazard on Mt. Etna volcano. *Bollettino di Geofisica Teorica e Applicata*, 52, 2, doi: 10.4430/bgta0003.
- Cleary, P.W., J.J. Monaghan (1999). Conduction modelling using Smoothed Particle Hydrodynamics. *Journal of Computational Physics*, 148, 227-264.
- Cole, R.H. (1948). *Underwater explosions*. Princeton University Press.
- Cordonnier, B., E. Lev, F. Garel (2015). Benchmarking lava-flow models. *Geological Society, London*, 426, Special Publications.
- Costa, A., G. Macedonio (2005). Numerical simulation of lava flows based on depth-averaged equations. *Geophysical Research Letters*, 32.
- Del Negro, C., L. Fortuna, A. Vicari (2005). Modelling lava flows by Cellular Nonlinear Networks (CNN): preliminary results. *Nonlinear Processes in Geophysics* 12 (4), 505-513, SRef-ID: 1607-7946/npg/2005-12-505.
- Dietterich, H.R., K.V. Cashman, A.C. Rust, E. Lev (2015). Diverting lava flows in the lab. *Nature Geoscience*, 8, 494-496.
- Fern andez-Nieto, E. D., J. Garres-D az, A.M., G. Narbona-Reina (2016). A multilayer shallow model for dry granular flows with the $\mu(i)$ -rheology: application to granular collapse on erodible beds. *Journal of Fluid Mechanics*, 798, 643-681.
- Filippucci, M., A.T., M. Dragoni (2013). Simulation of lava flows with power-law rheology. *Discrete & Continuous Dynamical Systems - Series S*, 6, 677-685.
- H erault, A., G. Bilotta, R.A. Dalrymple (2010). SPH on GPU with CUDA. *Journal of Hydraulic Research*, 48, 74-79.
- H erault, A., G. Bilotta, A. Vicari, E. Rustico, C. Del Negro (2011). Numerical simulation of lava flow using a GPU SPH model. *Annals of Geophysics*, 54 (5), doi: 10.4401/ag-5343.
- Kaddiri, M., M. Na imi, M. Raji, M. Hasnaoui (2012). Rayleigh-b enard convection of non-Newtonian power-law fluids with temperature-dependent viscosity. *ISRN Thermodynamics*.
- Liu, G., M.B. Liu (2003). *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*. World Scientific.
- Mayrhofer, A., M. Ferrand, C. Kassiotis, D. Violeau, F. Morel (2015). Unified semi-analytical wall boundary conditions in SPH: analytical extension to 3-D. *Numerical Algorithms*, 68: 15.
- Monaghan, J.J. (2005). *Smoothed Particle Hydrodynamics*. *Reports in Progress in Physics*, 68.
- Morris, J.P., P.J. Fox, Y. Zhu (1997). Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics*, 136, 214-226.
- Rustico, E., G. Bilotta, A. H erault, C. Del Negro, G. Gallo (2012). Advances in multi-GPU Smoothed Particle hydrodynamics Simulations. *IEEE transactions on Parallel and Distributed Systems*, 25, 43-52, 10.1109/TPDS.2012.340.
- Rustico, E., J. Jankowski, A. H erault, G. Bilotta, C. Del Negro (2014). Multi- GPU, multi-node SPH implementation with arbitrary domain decomposition. *Proceedings of the 9th International SPHERIC Workshop*.
- Smith, R.L. (2007). *Ode: Open dynamics engine*. URL <http://www.ode.org>
- Tallarico, A., M. Dragoni (1999). Viscous Newtonian laminar flow in a rectangular channel: application to Etna lava flows. *Bulletin of Volcanology*, 61, 40-47.
- Wendland, H. (1995). Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational*

Mathematics, 389-396.

Zhu, H., Y. Kim, D. Kee (2005). Non-Newtonian fluids with a yield stress. *Journal of Non-Newtonian Fluid Mechanics*, 129.

*Corresponding author: Vito Zago,
Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Catania,
Osservatorio Etneo, Italy; email: vito.zago@ingv.it.