

“ PRELIMINARY VALIDATION OF LAVA BENCHMARK TESTS ON THE GPUSPH PARTICLE ENGINE ”

Vito Zago^{*,1,3}, Giuseppe Bilotta¹, Annalisa Cappello¹, Robert Anthony Dalrymple², Luigi Fortuna^{3,5}, Gaetana Ganci¹, Alexis Hérault^{1,4}, Ciro Del Negro¹

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

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

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

⁽⁴⁾ Laboratoire Modélisation Mathématique et Numérique, Conservatoire National des Arts et Métiers, France

⁽⁵⁾ CNR-IASI, Italian National Research Council - Institute for Systems Analysis and Computer Science "A. Ruberti", Rome, Italy

Article history

Received July 30, 2018; accepted September 7, 2018.

Subject classification:

SPH; Lava flows; Benchmark.

ABSTRACT

Lava flow modeling is important in many practical applications, such as the simulation of potential hazard scenarios and the planning of risk mitigation measures, as well as in scientific research to improve our understanding of the physical processes governing the dynamics of lava flow emplacement. Existing predictive models of lava flow behavior include various methods and solvers, each with its advantages and disadvantages. Codes differ in their physical implementations, numerical accuracy, and computational efficiency. In order to validate their efficiency and accuracy, several benchmark test cases for computational lava flow modeling have been established. Despite the popularity gained by the Smoothed Particle Hydrodynamics (SPH) method in Computational Fluid Dynamics (CFD), very few validations against lava flows have been successfully conducted. At the TecnoLab of INGV-Catania we designed GPUSPH, an implementation of the weakly-compressible SPH method running fully on Graphics Processing Units (GPUs). GPUSPH is a particle engine capable of modeling both Newtonian and non-Newtonian fluids, solving the three-dimensional Navier-Stokes equations, using either a fully explicit integration scheme, or a semi-implicit scheme in the case of highly viscous fluids. Thanks to the full coupling with the thermal equation, and its support for radiation, convection and phase transition, GPUSPH can be used to faithfully simulate lava flows. Here we present the preliminary results obtained with GPUSPH for a benchmark series for computational lava-flow modeling, including analytical, semi-analytical and experimental problems. The results are reported in terms of correctness and performance, highlighting the benefits and the drawbacks deriving from the use of SPH to simulate lava flows.

1. INTRODUCTION

The mechanisms controlling lava flow emplacement are not yet fully understood, due to the complexity of the fluid, its non-Newtonian rheology, and the strong effect on the rheology by the physical and chemical properties of the fluid (temperature, chemical composition, degree of crystallization, etc.) and their evolution over time. Mathematical modeling and computer simulations can play an essential role in improving our un-

derstanding of the lava flow patterns, its morphology, and thermal evolution [Del Negro et al., 2008; Ganci et al., 2018; Vicari et al., 2009]. The complex nature of the fluid, aspects such as free surface and irregular topographies, and phenomena like phase transition and the consequent formation of levees and tunnels make simulation of lava flows an extremely challenging task for Computational Fluid Dynamics (CFD). In the modeling of lava flow hazards [Cappello et al., 2011, 2016a; Del Negro et al., 2013, 2016; Ganci et al., 2013; Hérault

et al., 2009], common approaches to the simulation of lava flows start by reducing the complexity with a number of different strategies, such as reduced dimensionality, simplified thermal or dynamic models, or the use of stochastic approaches with little or no physical modeling [Costa and Macedonio, 2005]. These simplifications allow easier implementations and higher performance, and while the results may still be useful for real-time forecasting [Cappello et al., 2016b], risk mitigation [Scifoni, 2010] and the production of long-term scenarios [Del Negro et al., 2013], they are inadequate for a more thorough study of the behavior of the fluid and the laws underlying its rheology, which require the detailed modeling of the full three-dimensional flow and its rheological aspects.

The Smoothed Particle Hydrodynamics (SPH) method, recently introduced in the field of Computational Fluid Dynamics (CFD), is a Lagrangian mesh-free particle-based method that allows a three-dimensional modeling of the fluid, taking into account in an efficient way many physical aspects that are typical of lava flows, such as the free surface, solidification fronts, the high dynamicity of the fluid and the interaction with irregular boundaries, such as solid lava emplacements and natural topographies. One main drawback of the SPH method is low accuracy, since in its common form it is a first order method [Monaghan, 2005]. A second drawback comes from the frequent adoption of a weakly-compressible model [H erault et al., 2010] instead of a fully incompressible one. As stated by Cordonnier et al. [2016], this choice affects the quality of the simulation, requiring an adjustment of parameters in order to improve the results. Moreover, the adoption of a weakly-compressible model affects the simulation performance, since the time stepping is driven by the speed of sound [Zago et al., 2018]. On the other hand, a weakly-compressible model allows a complete parallelization of the computations that can be run on massively parallel hardware like Graphics Processing Units (GPU), giving an advantage in terms of performance [H erault et al., 2010].

The TecnoLab at the Istituto Nazionale di Geofisica e Vulcanologia (INGV) in Catania has created the GPUSPH particle engine [H erault et al., 2011; Bilotta, 2014; Zago et al., 2017], the first implementation of the Weakly-Compressible Smoothed Particle Hydrodynamics (WCSPH) method to run entirely on Graphic Processing Units (GPUs). To better handle the computational needs of lava flow simulations, GPUSPH has been extended to distribute computations across multiple GPUs [Rustico et al.,

2012], even across separate nodes in a cluster [Rustico et al., 2014], and it includes a semi-implicit integration scheme for highly viscous flows [Zago et al., 2018].

GPUSPH has already been validated in a number of contexts, both against classical theoretical problems, and real-world applications [Wei et al., 2015; 2016]. However, we need to evaluate the accuracy and robustness of the particle engine results in the context of lava flow simulation, starting from the benchmark tests introduced by Cordonnier et al. [2016]. These benchmark tests of growing complexity (from a simple dam break without thermal effects to a physical experiment including both dynamic and thermal aspects) can be used to validate lava flow models in terms of completeness, accuracy and computing performance. In their work, Cordonnier et al. [2016] also compare the models that represented the state of the art at that time, and illustrated the respective main features, advantages and disadvantages. Here we present preliminary results obtained with GPUSPH, discussing the influence of the model and formulation on the accuracy and computational performance of the results, and the possible strategies to improve them.

2. SPH IN GPUSPH FOR LAVA

The SPH method discretizes the fluid by means of particles that act as interpolation nodes. Each particle carries information about a small volume of fluid, such as velocity, position, density, mass, temperature and so on, and moves according to equations of motion. The basis of the method relies on the SPH smoothing, that is the way in which the fields are interpolated at the position of the particles [Monaghan, 2005]. For each particle the effect of its neighbors is weighted by the value of a function called *smoothing kernel*, that is centered in the particle position and is usually indicated with $W(r, h)$. Here, r indicates the distance from the neighbouring particle and h is a parameter called *smoothing length*. The smoothing kernel is usually chosen with a compact support, which radius is called influence radius, usually determined as a multiple of h .

The dynamics of fluid bodies is modeled according to the continuity equations for mass:

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

where ρ is the density, \mathbf{u} the velocity and D/Dt the total derivative with respect to time, and momentum (Navier–Stokes equations):

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

where P is the pressure, μ the dynamic viscosity, and \mathbf{G} represents external forces, such as gravity, and the thermal evolution is described by the heat equation

$$\rho c_p \frac{DT}{Dt} = \nabla(\kappa \nabla T), \quad (3)$$

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

Even though we are considering the incompressible form of the Navier-Stokes equations, solving a fully incompressible fluid model would be rather complex and computationally expensive, since it would imply to solve the pressure form Poisson equation. Moreover, incompressible SPH models are known for exhibiting unstable behaviors in the density field and issues dealing with free surfaces [Ihmsen et al., 2014]. As an alternative, in SPH one uses frequently a weakly compressible formulation where the pressure is derived from the density using an equation of state, such as [Cole, 1948]:

$$P(\rho) = c_0^2 \frac{\rho_0}{\gamma} \left(\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right), \quad (4)$$

with ρ_0 the at rest density, c_0 the speed of sound and γ the polytropic constant. By construction, the compressible model allows to integrate independently each particle, making possible the adoption of explicit integration schemes without the need to solve any linear system. While this constitutes an advantage in terms of parallelizability of computations [H erault et al., 2010], it also introduces a drawback in terms of time-stepping.

Explicit integration schemes exhibit stability requirements concerning the maximum allowed time step. The latter is in fact linked to several factors like the speed of sound, the viscosity, the maximum acceleration and the thermal diffusivity, by some CFL-like conditions [Monaghan, 1989; Monaghan and Kos, 1999; Morris et al., 1997, and references within]:

$$\Delta t_\beta \leq \min \left\{ C_1 \sqrt{\frac{h}{\|\mathbf{a}_\beta\|}}, C_2 \frac{h}{c_\beta}, C_3 \frac{\rho_\beta h^2}{\mu_\beta}, C_4 \frac{\rho_\beta c_p h^2}{\kappa} \right\}, \quad (5)$$

where \mathbf{a}_β is the acceleration of the particle β , c_β is the speed of sound at density ρ_β , and C_1, C_2, C_3 and C_4 are stability constants. In GPUSPH we use $C_1=C_2=0.3$, $C_3=0.125$ and $C_4=0.1$. The maximum allowed time step for the whole system is then the minimum time step over all particles, $\Delta t = \min_\beta \Delta t_\beta$.

Because of these constraints, the speed of sound

used for SPH simulations is usually taken smaller than the physical one, but still enough to prevent a compressible behavior. Weak compressibility is achieved as long as the speed of sound is large enough to bound density variations within certain ranges. A common choice is to take the sound speed one order of magnitude above the maximum velocity experienced in the flow, so that the maximum variation in density is contained within 1% of the fluid density. When the simulation involves high fluid columns, the hydrostatic velocity (i.e. the maximum velocity that a particle at the highest position would achieve in free-fall) can be predominant with respect of the maximum flow speed, and must be used instead. This is important to eliminate the vertical collapse of the fluid column due to excessive compressibility. Therefore, choosing a high speed of sound can help to get closer to the incompressible behavior, which would be more appropriate for lava, but such improvement is paid with a small time step, and then in terms of computational time and numerical precision.

2.1 SURFACE THERMAL DISSIPATION

Thermal dissipation is modeled both at the contact with the ground, using Equation (3) and on the free surface. The latter occurs according to two phenomena:

1. **Thermal radiation:** according to Stefan-Boltzmann law, we express the radiated heat per unit surface as:

$$\frac{DT}{Dt} = \frac{K_B \kappa \epsilon}{m c_p} (T^4 - T_a^4), \quad (6)$$

with K_B the Stefan-Boltzmann constant, ϵ the emissivity, m the mass and T_a the air temperature.

2. **Air convection:** We do not model air particles, but we account the heat lost due to air convection by means of a convection coefficient η , according to the following law, per unit surface:

$$\frac{DT}{Dt} = \frac{\eta}{m c_p} (T - T_a), \quad (7)$$

2.2 SPH DISCRETIZATION OF THE EQUATIONS

The dynamical equations seen above are discretized according to the SPH method as described in Bilotta et al. [2016], H erault et al. [2011] and Zago et al. [2018] obtaining:

$$\frac{D\rho_\beta}{Dt} = - \sum_\alpha m_\alpha \mathbf{u}_{\alpha\beta} \nabla_\beta W_{\alpha\beta}, \quad (8)$$

for the continuity Equation (1),

$$\frac{D\mathbf{u}_\beta}{Dt} = \sum_\alpha \left(\frac{P_\alpha}{\rho_\alpha^2} + \frac{P_\beta}{\rho_\beta^2} \right) F_{\alpha\beta} m_\alpha \mathbf{x}_{\alpha\beta} - \sum_\alpha \frac{2\bar{\mu}_{\alpha\beta}}{\rho_\alpha \rho_\beta} F_{\alpha\beta} m_\alpha \mathbf{u}_{\alpha\beta} + \mathbf{g}, \quad (9)$$

for the momentum Equation (2), and

$$\frac{DT_\beta}{Dt} = -\frac{1}{\rho_\beta c_p} \sum_\alpha \frac{2m_\alpha k_{\alpha\beta} T_{\alpha\beta}}{\rho_\alpha} F_{\alpha\beta}, \quad (10)$$

for the thermal Equation (3).

The two formulas for the surface thermal dissipation, introduced in section 2.1, are applied to surface particles that are identified as described in Bilotta et al. [2016] and H erault et al. [2011]. To apply these two per unit surface formulas, a simple way of computing the particle surface would be to consider it a square with the average inter-particle spacing as particle side. But we are dealing with flows that usually evolve towards a condition of under-resolved flow, then we would underestimate the surface in the regions where the simulation becomes more rarefied (i.e. where we have a thin flow). To face this problem we compute the particle surface using the numerical volume [Hu and Adams, 2006]

$$V_\beta = \frac{1}{\sum_\alpha W_{\alpha\beta}}, \quad (11)$$

and considering a spherical particle volume and a circular particle surface.

2.3 SPH SMOOTHING KERNEL

Given a symmetric SPH smoothing kernel W , its gradient can be written as:

$$\nabla_\beta W_{\alpha\beta} = -\frac{\mathbf{x}_{\alpha\beta}}{|\mathbf{x}_{\alpha\beta}|} \frac{\partial W(r, h)}{\partial r} \Big|_{r=|\mathbf{x}_{\alpha\beta}|}, \quad (12)$$

where $\mathbf{x}_{\alpha\beta} = \mathbf{x}_\alpha - \mathbf{x}_\beta$, and h is the kernel smoothing length. By choosing a kernel for which

$$\mathbf{F}(r) = \frac{1}{r} \frac{\partial W}{\partial r}, \quad (13)$$

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

For the simulations presented in this work we adopt a Wendland kernel [Wendland, 1995], defined as $W(r, h) = \tilde{W}(r/h)$ and $F(r, h) = \tilde{F}(r/h)$ with

$$\begin{aligned} \tilde{W}(q) &= C_w(2q+1) \left(1 - \frac{q}{2}\right)^4 & 0 \leq q \leq 2; \\ \tilde{F}(q) &= C_f(q-2)^3 \end{aligned} \quad (14)$$

where, working in three dimensions, $C_w = 21/(16 \pi h^3)$ and $C_f = 5 C_w / (8h^2)$.

2.4 BOUNDARY CONDITIONS AND BOUNDARY MODEL

From a mechanical point of view, at the interfaces between fluid and either walls and ground we impose a no-slip condition; this is obtained prescribing normal and tangential velocity along the analytical boundary:

$$\mathbf{u}_n(t) = 0 \quad (15)$$

$$\mathbf{u}_t(t) = \mathbf{v}_w \quad (16)$$

where \mathbf{v}_w is any physical sliding velocity of the wall (in our examples, we will always have $\mathbf{v}_w = 0$). In our SPH discretization, physical boundaries are implemented according to the Dummy boundary model [Adami et al., 2012], where solid walls are discretized with multiple layers of boundary particles, enough to cover a full influence radius and thus complete the kernel support for fluid particles adjacent to the boundary. No-slip boundary conditions with dummy boundaries is obtained assigning to the boundary particles a velocity obtained as \mathbf{v}_w plus the opposite of the Shepard-averaged velocity of the neighboring fluid:

$$\mathbf{u}_\beta = \mathbf{v}_w - \frac{\sum_{\alpha \in F} \mathbf{u}_\alpha W_{\alpha\beta}}{\sum_{\alpha \in F} W_{\alpha\beta}}, \quad (17)$$

(where F represents the set of fluid particles) while the density is computed to achieve a pressure that matches the Shepard-averaged pressure of the neighboring fluid:

$$P_\beta = \frac{\sum_{\alpha \in F} P_\alpha W_{\alpha\beta} + \mathbf{g} \cdot \sum_{\alpha \in F} \rho_{\alpha\beta} \mathbf{x}_{\alpha\beta} W_{\alpha\beta}}{\sum_{\alpha \in F} W_{\alpha\beta}}, \quad (18)$$

In order to reduce the accumulation of numerical error in the computations for the boundary model, all summations are performed using the Kahan method [Kahan, 1965]. For the thermal model we use absorbing boundary conditions, implemented using the sponge layer approach: the boundary is assumed to have a sufficiently large thickness H_s , through which heat propagates using the standard heat equation. Given a one-dimensional reference system with the origin on the boundary interface and oriented along the inwards normal \mathbf{n} , the conditions for the temperature $T(n, t)$ (with n the wall depth coordinate, and t the time) can be described analytically as

$$T(0, 0) = T_w, \quad (19)$$

$$T(-H_s, t) = T_w, \quad (20)$$

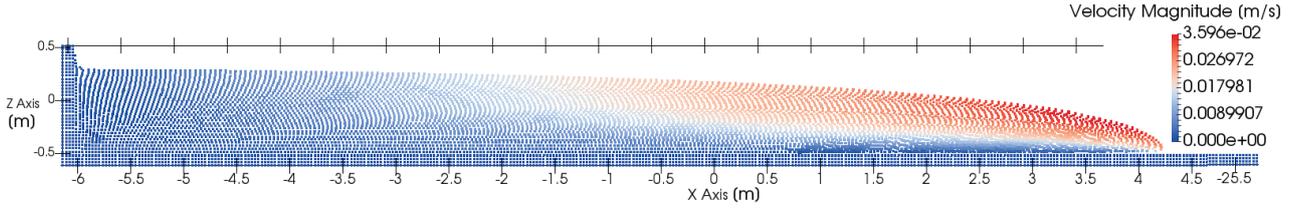


FIGURE 1. Lateral slice of $BM1_{HR}$ at $t = 10s$. Particles are colored by velocity magnitude.

where T_w is the initial physical temperature of the wall [H erault, 2008]. H_s must be chosen large enough to guarantee the last condition over all the simulation time.

3. BENCHMARKING THE GPUSPH MODEL

The SPH discretization, as any numerical method, introduces some errors that lead to a discrepancy between the simulation results and the exact solutions, which can be reduced adopting finer resolutions. In the following we will analyze the results of the simulation in terms of convergence with respect to the discretization fineness. In particular, the outputs of GPUSPH have been compared to the first three benchmark tests introduced by Cordonnier et al. [2016].

3.1 BM1: VISCOUS DAM-BREAK

Dam breaks are one of the simplest test cases in the field of CFD, and is described as a defined amount of confined fluid that is suddenly freed from one side and allowed to spread onto an horizontal plane, driven by gravity. In the simplest configurations, validation is made against the progress of the front of the flow over time.

In Cordonnier et al. [2016], the initial configuration of the fluid is a box with length $L = 6.6m$, height $H = 1m$ and width $W = 6.6m$. The fluid has density $\rho = 2700 kg/m^3$ and a dynamic viscosity $\mu = 10^4 Pa\cdot s$. According to Balmforth et al. [2007] and Saramito et al. [2013], the evolution of the front over time is analytically described by

$$\frac{x_f(t)}{L} = \begin{cases} 0.284 \left(\frac{t}{T}\right)^{\frac{1}{2}} & \text{if } t < 2.5T; \\ 1.133 \left(\frac{t}{T} + 1.221\right)^{\frac{1}{5}} - 1 & \text{if } t \geq 2.5T, \end{cases} \quad (21)$$

where T is the characteristic time of the problem, defined as $T = (L/H)^2 (\mu/\rho gh)$. In our case, we have $T = 16.45 s$.

1) *Implementation in GPUSPH*: The simulation do-

main consists of a base plane and three walls containing the fluid. The fourth side of the main fluid box is left free to allow the fluid to slump, simulating the sudden opening of a gate (Figure 1). The solid walls are modeled using dummy boundaries, as introduced in section 2.3. The density diffusion approach introduced by Molteni and Colagrossi [2009] is used to smooth out the noise that naturally develops in the density field. For the speed of sound, the usual choice in WSPH is to pick a value c_0 around 10 or 20 the maximum velocity value. Our experiments however show that much lower errors at a given spatial resolution can be obtained by using a higher speed of sound. There are diminishing returns in raising the value of c_0 , though, due to the smaller time-step, and even a reversal when the time-step becomes too small for the available precision. The main results that we illustrate are thus obtained with a speed of sound 100 times higher than the hydrostatic velocity, resulting in $c_0 = 443 m/s$.

2) *Results for BM1*: We show results for three different resolution, a Low Resolution with inter-particle distance $\Delta p_{LR} = 1/8 m = 0.125m$, an Intermediate Resolution with inter-particle distance $\Delta p_{IR} = \Delta p_{LR}/2 = 1/16 m = 0.0625m$, and a High Resolution with $\Delta p_{LHR} = \Delta p_{LR}/4 = 1/32 m = 0.03125m$. In the following we will refer to these three simulations as $BM1_{LR}$, $BM1_{IR}$ and $BM1_{HR}$. We measure the front of the fluid as the position of the furthest particle in the flow, plus $\Delta p/2$ to take into account the particle volume. Results for the front position at the three resolutions, compared to the front position predicted by Equations (21) are presented in Figure 2.

We observe that GPUSPH slightly overestimates the theoretical solution for the front progress, and that the error becomes smaller at higher resolutions. Table 1 shows the errors obtained as the difference between the simulated and theoretical front position in respectively

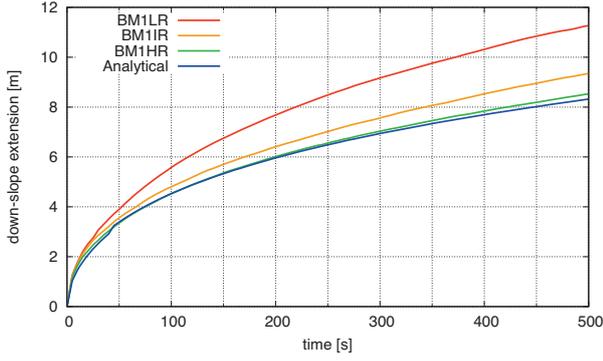


FIGURE 2. Front position over time for $BM1$. The comparison among the analytical solution and the solutions simulated with different resolutions reveals convergence of the method.

simulation becoming under-resolved as the flow progresses, due to the decrease in thickness, leading to larger errors and to the formation of artifacts, as shown in Figure 4, illustrating the situation for $BM1_{IR}$ at $t = 500s$: we can observe that the front profile is no more well reconstructed and some artifacts are arising, like the formation of a second head and the detachment of the fluid from the ground. For $BM1_{HR}$ we can further observe that at $t = 45s$ there is a temporary inversion, with the simulation being slightly behind the theoretical result. This may be explained by the change in the expression of the analytical law, that presents a small discontinuity at $t = 2.5T = 41.125s$, giving a bigger value from the right hand side.

Time (s)		$BM1_{LR}$ (8 parts/m)	$BM1_{IR}$ (16 parts/m)	$BM1_{HR}$ (32 parts/m)
10	error [m]	0.3085	0.2628	0.1822
10	err ratio	1.1739	1.4423	-
500	error [m]	2.9546	1.0267	0.2113
500	err ratio	2.8779	4.8599	-

TABLE 1. Errors for $BM1$ at short and long time. The error ratios are computed as the ratio of the error at lower resolution over the error at higher resolution.

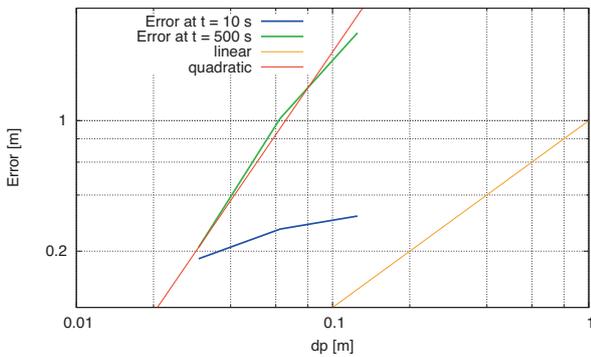


FIGURE 3. Logarithmic plot of the error for $BM1$ over the spatial discretization interval for different times. The convergence rate is higher at longer times due to the under-resolved condition of lower resolution simulations.

the low, intermediate and high resolution cases at time t , and the error ratios, obtained as the error at lower resolution over the error at higher resolution.

From Table 1 and Figure 3 we can assess the convergence of the model and that the order of convergence grows over time, being in the best case around a second order trend. The latter result is due to the low-resolution

Finally, concerning the fluid height, we have from Saramito et al. [2013] that for short times the fluid height at the dam position and the end of the reservoir should remain constant (i.e. $h(t,0)/H = 0.684$ and $h(t,-L)/H = 1$) while the surface shape rearranges, whereas for long times ($t \gg 2.5T$) the height at $x = -L$ and $x = 0m$ is the same, and evolves according to the law t^{-2} . For short times (less than around 100s) all of the three different resolution simulations match the analytical result, with an error of less than Δp .

3) *Simulation performance*: All the simulations were performed on a NVIDIA Titan X GPU (Maxwell architecture); with the adopted spatial discretization, $BM1_{HR}$ consists of $30'890'152$ particles and the time step (controlled by the speed of sound) is $\Delta t_{BM1_{HR}} = 2.54 \cdot 10^{-5} s$. 1s of $BM1_{HR}$ evolution is simulated in around 1950s of real time. $BM1_{LR}$ involves $8'080'744$ particles and a time step $\Delta t_{BM1_{LR}} = 5.63 \cdot 10^{-5} s$ and 1s of $BM1_{LR}$ evolution is simulated in around 168s. Finally, $BM1_{LR}$ involves $187'244$ particles and a time step

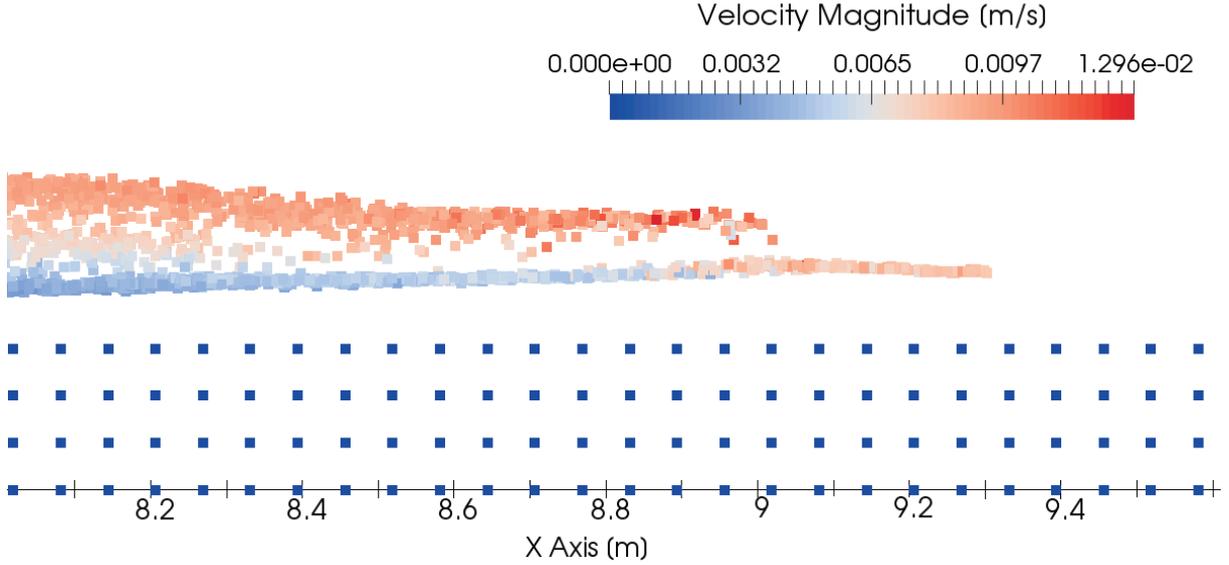


FIGURE 4. Under resolved flow front for $BM1_{IR}$ at $t = 500s$: this is one of the resolution issues arising in $BM1$ as the flow evolves.

$\Delta t BM1_{LR} = 5.6 \cdot 10^{-4} s$ and 1s of $BM1_{LR}$ evolution is simulated in around 18 s.

3.2 BM2: INCLINED VISCOUS ISOTHERMAL SPREADING

This benchmark test regards the simulation of a fluid spreading onto an inclined plane, and follows the analytical solution derived by Lister [1992]. The fluid has a Newtonian rheology and is injected at a constant rate Q from a point source through the plane. We are interested in the evolution of the down-slope and cross-slope extent (L_d and y_p , respectively as shown in Figure 5) of the flow. The plane is inclined by an angle $\alpha = 2.5^\circ$ and the fluid, with kinematic viscosity $\nu = \mu/\rho = 11.3 \cdot 10^{-4} m^2/s$, flows at a rate $Q = 1.48 \cdot 10^{-6} m^3/s$.

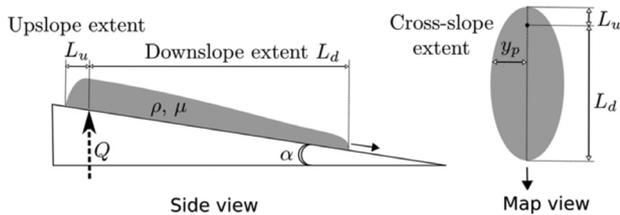


FIGURE 5. Setup for $BM2$. [from Cordonnier et al., 2016].

The density is left free by Cordonnier et al. [2016], and we have opted to use the same value adopted in $BM1$, $\rho = 2700 kg/m^3$. According to Cordonnier et al. [2016], the analytical solution at long time for the downslope extent over time is given by:

$$L_d \approx \left[\frac{(\rho g)^3 Q^4 \sin^5 \alpha}{(3\mu)^3 \cos^2 \alpha} \right]^{\frac{1}{9}} t^{\frac{7}{9}} = 0.0064 t^{\frac{7}{9}}, \quad (22)$$

and for the cross slope extent, at long times, is given by:

$$y_p \approx \left(\frac{Q \cos \alpha}{\sin \alpha} \right)^{\frac{1}{3}} t^{\frac{1}{3}} = 0.0324 t^{\frac{1}{3}}, \quad (23)$$

- 1) *Implementation in GPUSPH*: The simulation domain is constituted by a base plane and a piston, the latter used to obtain the constant flow rate Q , as shown in Figure 6. Also in this case we perform a convergence test using three levels of discretization: a low resolution with inter-particle distance $\Delta p_{LR} = 1/384 m = 2.6 \cdot 10^{-3} m$, an Intermediate resolution with $\Delta p_{IR} = \Delta p_{LR}/1.5 = 1/512 m = 1.95 \cdot 10^{-3} m$, and a High Resolution with $\Delta p_{HR} = \Delta p_{IR}/1.5 = 1/768 m = 1.3 \cdot 10^{-3} m$.

The fluid source, ideally a point source, is obtained by means of a hole in the plane, through which the fluid is extruded. Because of consistency requirements of the SPH method, the size of the source cannot be chosen arbitrarily small, but there is a lower bound dictated by the resolution. To avoid under-resolving the inlet, we impose a lower bound on the inlet diameter in order to have at least $8 \Delta p$ with the coarser resolution, then we use $2 cm$. For simplicity, the source has a squared shape. The width of the piston is $0.1m$. It is taken bigger than the hole section to avoid having a high fluid column that would require a very high speed of sound and conse-

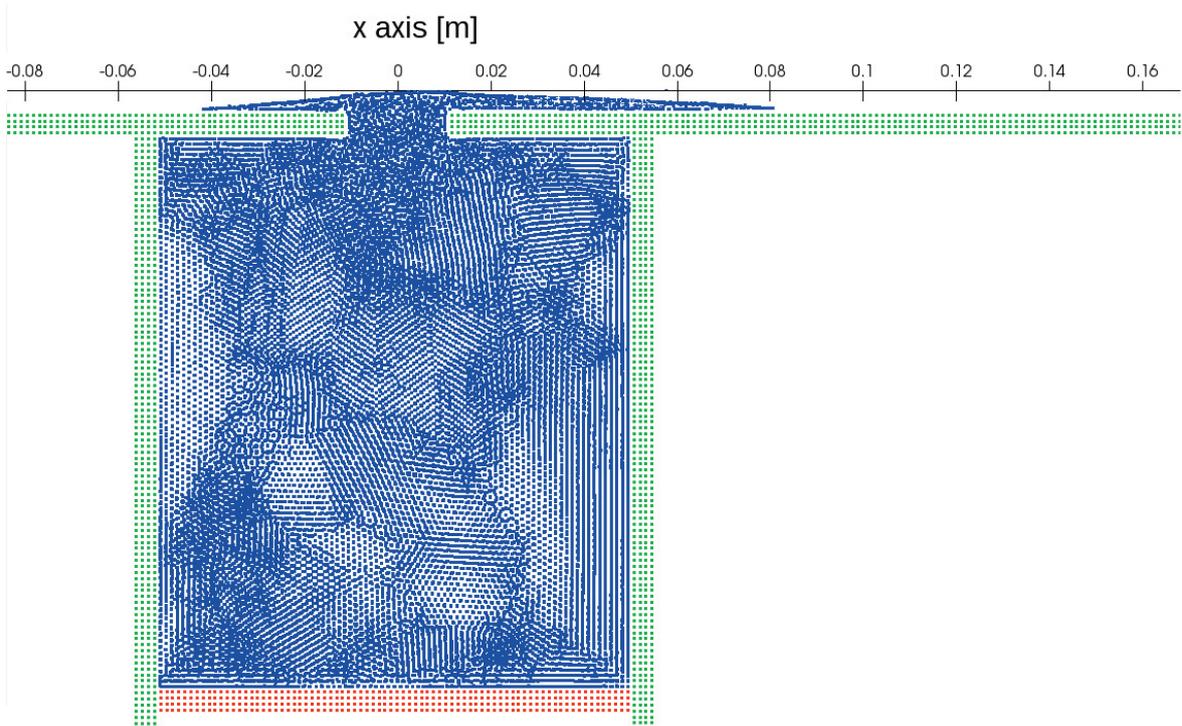


FIGURE 6. Implementation of *BM2* in GPUSPH, lateral view of a slice. Particles are colored by type: fluid in blue, walls in green and the moving piston in red.

quently very small time steps. On the other side, having a too large piston implies more stresses on the fluid that would lead to higher disorder in the flow, with consequent increase in the discretization error [Monaghan, 2005]. Moreover a large piston surface, coupled with the small value of Q , can determine small particles velocities that can be affected by numerical precision during the integration process.

In order to make a simpler definition of the geometry, with reduced numerical rounding, the floor is parallel to the xy coordinate plane, and gravity is oriented by an angle of $-\alpha$. A side view of the simulation at $t = 15s$ is presented in Figure 6.

As in *BM1*, the speed of sound is chosen so as to minimize compressibility while avoiding numerical in-

stabilities due to loss of precision. For *BM2*, we use $c_0 = 125,5$ m/s.

2) *Results for BM2*: The evolution of L_d is shown in Figure 7. The comparison between the simulated and analytical solutions is to be performed at long time, anyway an irregularity can be observed at the beginning of the simulation for both L_d and y_p , where, while a theoretical solution would have a growing trend, starting from zero, the simulated solutions maintain a non-zero constant value. This is due to the fact that the vent is not a point source and its size corresponds to the initial extension of the flow.

We observe that for L_d (Figure 7), the convergence is apparent, with a resolution increase leading to results closer to the theoretical solutions. As in *BM1*, as the flow spreads out, it becomes under-resolved at lower resolution, leading to considerably worse results that diverge from the analytical solution over time. Moreover, the under resolved flow front generates some artifacts that result in a shaky position advance.

The errors are reported in Table 2 and their trend is shown in Figure 8; as for *BM1* we have a convergence that gains orders over time. Also in here, this can be explained considering the thinning evolution of the flow, which makes the low-resolution simulation becoming under-resolved as the flow progresses.

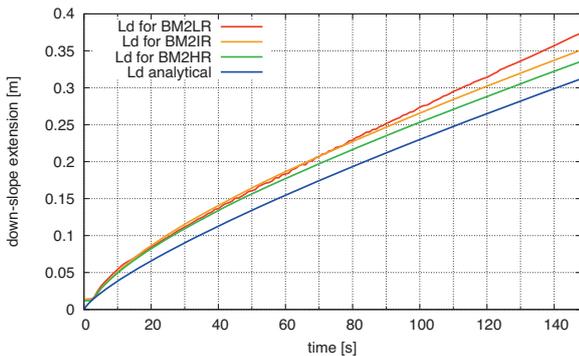


FIGURE 7. Time evolution of the down-slope extension for *BM2*.

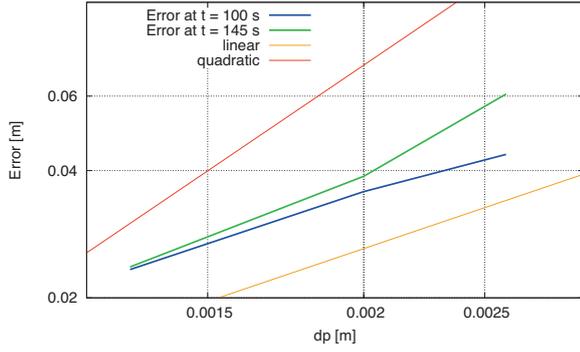


FIGURE 8. Logarithmic plot of the error for $BM2$ over the spatial discretization interval at different times. As for $BM1$, the convergence rate is higher at longer times due to the under-resolved condition of lower resolution simulations.

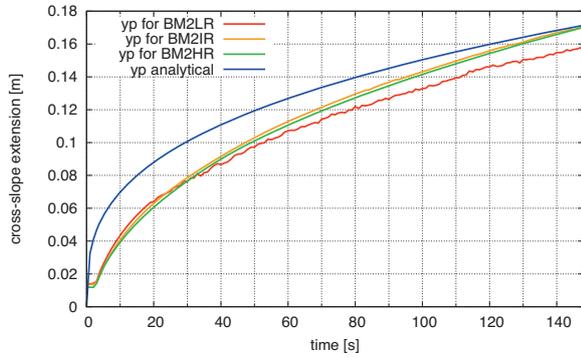


FIGURE 9. Time evolution of the cross-slope extension for $BM2$.

When it comes to the cross-slope extent, we have a discrepancy between the simulated and analytical extension, as shown in Figure 9, which leads at long time to a wider flow. The causes are not yet fully known, and could likely be due to the approximations done at numerical level such as finite size of the inlet, as well as compressibility. Although the value of the density does not affect the theoretical behavior of the problem, it can

be relevant in numerical terms. The impact that the density has in the discrepancies mentioned above is currently being investigated.

- 3) *Simulation performance*: $BM2_{HR}$ involves 2'761'836 particles and a time step $\Delta t_{BM2_{HR}} = 4.15 \cdot 10^{-6}$ s. Running on the same hardware as $BM1$, one second of $BM2_{HR}$ evolution is simulated in around 7'165 s. $BM2_{IR}$ involves 1'171'807 particles and a time step $\Delta t_{BM2_{IR}} = 6.22 \cdot 10^{-6}$ s, with one second of $BM2_{IR}$ evolution being simulated in around 1'673s. Finally, $BM2_{LR}$ involves 645'444 particles and a time step $\Delta t_{BM2_{LR}} = 8.27 \cdot 10^{-6}$ s, with one second of $BM2_{IR}$ evolution being simulated in around 646s.

3.3 BM3: AXISYMMETRIC COOLING AND SPREADING

This benchmark test deals with a non-isothermal flow. The setup exhibits many similarities with $BM2$, including a point source with fluid spreading on a plane. The floor is however horizontal in $BM3$, leading to axial symmetry in the flow emplacement. Thermal effects are also taken into account in $BM3$, with only one-way coupling, as the fluid rheology is assumed to be independent from the temperature, which therefore acts as a passive tracer. The parameters for $BM3$ are reported in Table 3.

The radius of the expanding fluid evolves according to:

$$R(t) \approx 0.715 \left(\frac{\rho g Q^3}{3\mu} \right)^{\frac{1}{8}} t^{\frac{1}{2}} = 2.23 \cdot 10^{-3} t^{\frac{1}{2}}, \quad (24)$$

- 1) *Implementation in GPUSPH*: The implementation in GPUSPH of $BM3$ is similar to that of $BM2$, though the difference in the magnitude of some physical parameters introduces more stringent constraints. The very small flow-rate sets an upper bound in the dimension of the source, since a larger inlet would result in a smaller velocity of the piston, that could lead to numerical issues for its

Time (s)		$BM2_{LR}$ (384 parts/m)	$BM2_{IR}$ (512 parts/m)	$BM2_{HR}$ (768 parts/m)
100	error [m]	0.0437	0.0357	0.0233
100	err ratio	1.2241	1.5322	-
145	error [m]	0.0606	0.0388	0.237
145	err ratio	1.5619	1.6373	

TABLE 2. Errors for $BM2$. The error ratios are computed as the ratio of the error at lower resolution over the error at higher resolution.

Parameter	Value	Parameter	Value
Density	886 kg/ m ³	Convective heat transfer	2 W m ⁻² K ⁻¹
Viscosity	3.4 Pa s	Emissivity	0.96
Specific heat	1500 J kg ⁻¹ K ⁻¹	Eruption temperature T_0	42 °C
Bed slope	0°	Ambient temperature T_a	20 °C
Effusion rate	2.2×10 ⁻⁸ m ³ s ⁻¹	Thermal conductivity	0.15 W m ⁻¹ K ⁻¹

TABLE 3. Parameters for *BM3*.

integration in single precision. The upper bound also imposes limitations on the coarsest resolution that can be used to discretize the problem and finally on the time-step. On the other hand, to be the piston capable of containing enough fluid to run the simulation, a smaller piston section implies a large piston height, requiring a higher speed of sound that would affect the time step. A solution to the fluid column could be to make a horizontal piston, parallel to the plane, but the junction with the hole would introduce disorder in the flow. The size of the piston has been chosen in order to approximate a trade-off with the issues just mentioned. The piston has a squared section, 0.005m wide and is 0.13m high. The plane has a side 0.12m long and the vent is a square hole placed in the middle. In order to avoid introducing disorder in the flux, the hole shape and size matches with the piston section. We use for the wall the same thermal parameters that we use for the fluid, and the thickness required by the dynamic boundary model is also sufficient to implement the absorbing conditions for the thermal model. The speed of sound is 40 m/s. We perform a convergence test using three levels of discretization: a low resolution with inter-particle distance $\Delta p_{LR} = 1/512 \text{ m} = 1.95 \cdot 10^{-3} \text{ m}$, an Intermediate resolution with $\Delta p_{IR} = \Delta p_{LR}/1.5 = 1/768 \text{ m} = 1.3 \cdot 10^{-3} \text{ m}$, and a High Resolution with $\Delta p_{HR} = \Delta p_{IR}/1.5 = 1/1024 \text{ m} = 9.7 \cdot 10^{-4} \text{ m}$.

2) *Results for BM3*: For the flow dynamics, the model convergence can be assessed by looking at the radius of the emplacement. We take as objective the simulated time $t = 144\text{s}$ the fluid is already spread enough to show a clear temperature profile, and artifacts due to the low resolution are yet to ap-

pear. We observe an over estimation of the emplacement radius, being $R_{BM3HR}(144) = 0.038 \text{ m}$ and $R(144) = 0.027 \text{ m}$, probably due to discrepancies between the numerical and analytical setup.

Concerning the temperature profile, we perform a graphical comparison between the simulated profile, and the analytical and measured ones, as shown in Figure 10, where we plot the normalized temperature $(T^* = (T - T_a)/(T_0 - T_a))$ with respect to the radius normalized by the current flow extension. In the two reference curves, the temperature at the vent is lower than the temperature specified in the problem data, which has been faced by setting a lower initial temperature of the simulated fluid. From a graphical estimation, the temperature is chosen as the 93% of the normalized temperature, that is 313.6 K.

We can see that although the simulated solutions do not apparently match the reference, they qualitatively tend to those as the resolution is increased. The high mismatch in the shape of the profile can be due to an incomplete implementation of the thermal model for the boundary that is not described in Codonnier et al. [2016]. The width of the simulated curves comes from the particles marked as surface, not being distributed on an imaginary smooth surface, but rather being recognized at different heights. This is also cause of the double curve effect (more apparent in the green curve) that is due to the lower surface particles cooling down for both the effect of surface dissipation and ground transmission.

3) *Simulation performance*: $BM3_{LR}$ involves 30'550 particles and a time step $\Delta t_{BM3_{LR}} = 1.97 \cdot 10^{-5} \text{ s}$. 1s of $BM3_{LR}$ evolution is simulated in around 15s on the same hardware used for the other benchmarks. $BM3_{IR}$ involves 60'432 particles and a time step $\Delta t_{BM3_{IR}} = 1.31 \cdot 10^{-5} \text{ s}$. 1s of $BM3_{IR}$ evolu-

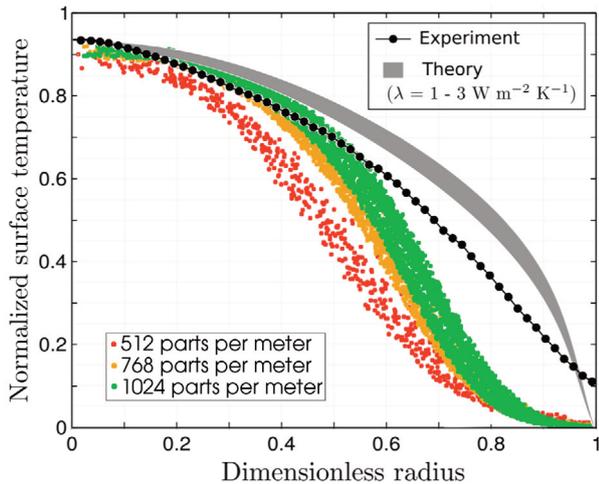


FIGURE 10. Time evolution of the cross-slope extension for $BM2$.

tion is simulated in around 134s. Finally, $BM3_{HR}$ involves 106'724 particles and a time step $\Delta t_{BM3_{LR}} = 9.83 \cdot 10^{-6}$ s. 1s of $BM3_{HR}$ evolution is simulated in around 267s.

4. CONCLUSIONS AND FUTURE WORK

We have presented the first three benchmark tests introduced by Cordonnier et al. [2016] simulated using GPUSPH. They state that the main drawback of SPH is the Weakly Compressible formulation and therefore a tuning of the parameters is needed to improve the quality of the simulations. Following this reasoning we mainly acted on the speed of sound and the boundary model in order to find a good compromise among accuracy, stability and simulation time. We have shown that for $BM1$ we have a strong convergence, while for the two benchmarks cases involving a fluid injection, larger discrepancies with the analytical results are present, but, we have proven the existence of a convergent behavior in most of the cases, and that the errors can be mitigated choosing a proper resolution for the spatial discretization. We have also encountered some issues related to under-resolved conditions of the problem, and again we have seen that they can be eliminated using finer discretization.

One main general improvement will be given by the introduction of an open boundary model. Although pistons are quite easy to implement, we have seen that it is worth to develop a more sophisticated way to implement a constant flux, in order to obtain a cleaner flow and a more stable simulation. For our future work we will focus on the introduction of the inlet feature, which

should lead to significant improvements on the results for $BM2$ and $BM3$.

Anyway, the biggest obstacle to reproduce the $BM3$ test case has been the incomplete description of the problem setup from Cordonnier et al. [2016], with respect to the thermal boundary conditions; while this can be partially solved referring to the tests from which the benchmark setup is taken [Garel et al., 2012]. We believe that more complete benchmarks should be developed.

Cordonnier et al. [2016] also set up a fourth benchmark test ($BM4$), regarding experimental lava flow and the simulation involves realistic lava parameters. Before passing to real lava we need to be sure that the first three tests are reproduced in a good way that has been preliminary done here. We measured the performance of GPUSPH reporting the simulation times; which are around two orders of magnitude lower than what required by naïve implementations of the WSPH method, as demonstrated by Hérault et al. [2010]. Further improvement will be sought in the next work introducing open boundaries, which will contribute to reduce the noise content within the simulation and have simpler setups. The inlet will also give advantage in terms of performance and robustness. In fact, the high viscosity values that will be involved in $BM4$ require the adoption of the semi implicit integration scheme, recently introduced in GPUSPH [Zago et al., 2018]; in that case, we shown that the performances of the semi implicit scheme have a strict dependence on the number of particles involved in the simulation, then simpler geometries obtained substituting the piston structure with an open boundary will be more suited for the adoption of the semi implicit scheme.

Acknowledgements. This work was developed within the framework of the Laboratory for Technological Advance in Volcano Geophysics (TecnoLab) of the INGV in Catania (Italy) and was supported from the ATHOS research programme. We thank two anonymous reviewers for the constructive and supportive comments that helped to improve the manuscript.

REFERENCES

- Adami S., X. Y. Hu and N. A. Adams, (2012). A generalized wall boundary condition for smoothed particle hydrodynamics, *Journal of Computational Physics*, vol. 231 (21), pp. 7057–7075. doi:10.1016/j.jcp.2012.05.005.

- Balmforth N. J., R. V. Craster, P. Perona, A. C. Rust, R. Sassi (2007). Viscoplastic dam breaks and the bostwick consistometer, *Journal of Non-Newtonian Fluid Mechanics*, 142, pp. 63–78. doi:10.1016/j.jnfm.2006.06.005.
- Bilotta G. (2014). GPU implementation and validation of fully three-dimensional multi-fluid SPH models. *Rapporti Tecnici INGV*, v. 292, pp. 1–46, ISSN 2039-7941.
- Bilotta G., A. H erault, A. Cappello, G. Ganci and C. Del Negro (2016). GPU SPH: a Smoothed Particle Hydrodynamics model for the thermal and rheological evolution on lava flows, in *Detecting, Modelling and Responding to Effusive Eruptions*, edited by A. Harris et al., Geological Society, London, Special Publications, 426, 387–408, doi:10.1144/SP426.24.
- Cappello A., G. Ganci, S. Calvari, N. M. P erez, P. A. Hern andez, S. V. Silva, J. Cabral and C. Del Negro (2016a). Lava flow hazard modeling during the 2014–2015 Fogo eruption, Cape Verde, *Journal of Geophysical Research*, vol. 121 (4), pp. 2290–2303. doi:10.1002/2015JB012666.
- Cappello, A., A. H erault, G. Bilotta, G. Ganci, and C. Del Negro (2016b). MAGFLOW: A physics-based model for the dynamics of lava flow emplacement, in *Detecting, Modelling and Responding to Effusive Eruptions*, edited by A. Harris et al., Geological Society, London, Special Publications, 426, 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.
- Cole R. H. (1948). *Underwater Explosion*, (Princeton, NJ: Princeton University Press).
- Cordonnier B., E. Lev, F. Garel (2016). Benchmarking lava-flow models, in *Detecting, Modelling and Responding to Effusive Eruptions*, edited by A. Harris et al., in Geological Society, London, Special Publications, 426, 425–445, <http://doi.org/10.1144/SP426.7>.
- Costa A. and G. Macedonio (2005). Computational modelling of lava flows: a review. In: M. Manga, G. Ventura (eds) *Kinematics and Dynamics of Lava Flows*, Geological Society of America, Special Papers, 396, pp. 209–218. doi:10.1130/0-8137-2396-5.209.
- Del Negro C., A. Cappello, and G. Ganci (2016). Quantifying lava flow hazards in response to effusive eruption, *GSA Bulletin*; v. 128; no. 5/6; p. 752–763; doi:10.1130/B31364.1.
- Del Negro C., A. Cappello, M. Neri, G. Bilotta, A. H erault and G. Ganci (2013). Lava flow hazards at Mount Etna: constraints imposed by eruptive history and numerical simulations. *Scientific Reports*, vol. 3, 3493, doi:10.1038/srep03493.
- Del Negro, C., L. Fortuna, A. H erault, and A. Vicari (2008). Simulations of the 2004 lava flow at Etna volcano by the MAGFLOW Cellular Automata model, *Bull. Volcanol.*, 70, 805–812, doi:10.1007/s00445-007-0168-8.
- Ganci, G., A. Cappello, G. Bilotta, A. H erault, V. Zago, C. Del Negro (2018). Mapping Volcanic Deposits of the 2011–2015 Etna Eruptive Events Using Satellite Remote Sensing, *Frontiers in Earth Science*, <https://doi.org/10.3389/feart.2018.00083>.
- Ganci G., A. Cappello, G. Bilotta, A. H erault, V. Zago, C. Del Negro (2018). Mapping Volcano Deposits of the 2011–2015 Etna Eruptive Events using Satellite Remote Sensing, *Frontiers in Earth Science*. <https://doi.org/10.3398/feart.2018.00083>.
- Ganci G., A. Vicari, A. Cappello and C. Del Negro (2013). An emergent strategy for volcano hazard assessment: from thermal satellite monitoring to lava flow modelling, *Remote Sensing of Environment*, 119, pp. 197–207. doi:10.1016/j.rse.2011.12.021.
- Garel F., E. Kaminski, S. Tait, A. Limare (2012), An experimental study of the surface thermal signature of hot subaerial isoviscous gravity currents: Implications for thermal monitoring of lava flows and domes, *Journal of Geophysical Research*, vol. 117, B02205. doi:10.1029/2011JB008698.
- H erault A. (2008) Cr eation d’un syst eme d’information pour la gestion des risques volcaniques, PhD Thesis, Information Scientifique et Technique, Universit e de Paris-Est, France, <https://tel.archives-ouvertes.fr/tel-00470546>.
- H erault A., G. Bilotta, R. A. Dalrymple (2010). SPH on GPU with CUDA, *Journal of Hydraulic Research*, vol. 48 (special issue), pp. 74–79. doi:10.1080/00221686.2010.9641247.
- H erault A., G. Bilotta, A. Vicari, E. Rustico and C. Del Negro (2011). Numerical simulation of lava flow using a GPU SPH model, *Annals of Geophysics*, vol. 54, no. 5, 2011. doi:10.4401/ag-5343.
- H erault, A., A. Vicari, A. Cirauco, and C. Del Negro (2009). Forecasting lava flow hazard during the 2006 Etna eruption: using the MAGFLOW cellular automata model, *Comput. Geosci.*, 35, 1050–1060, doi:10.1016/j.cageo.2007.10.008.
- Hu X., N. Adams (2006). A multi-phase SPH method for macroscopic and mesoscopic flows. *Journal of*

- Computational Physics, 213(2), 844–861.
- Ihmsen, M., J. Cornelis, B. Solenthaler, C. Horvat, M. Teschner (2014). Implicit Incompressible SPH, *IEEE Transactions on Visualization and Computer Graphics*, Volume 20, Issue: 3, 2014, doi:10.1109/TVCG.2013.105.
- Kahan W. (1965). Further Remarks on Reducing Truncation Errors, *Communications of the ACM*, 8(1):40. doi:10.1145/363707.363723.
- Lister J. R. (1992). Viscous flows down an inclined plane from point and line sources, *Journal of Fluid Mechanics*, 242, pp. 631–65. doi:10.1017/S0022112092002520.
- Molteni D., A. Colagrossi (2009). A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH, *Computer Physics Communications*, 180, pp. 861–72, 2009. doi:10.1016/j.cpc.2008.12.004.
- Monaghan J. J. (1989). On the problem of penetration in particle methods, *Journal of Computational Physics*, Volume 82, Issue 1, May 1989, Pages 1-15. [https://doi.org/10.1016/0021-9991\(89\)90032-6](https://doi.org/10.1016/0021-9991(89)90032-6).
- Monaghan J. J. (2005). Smoothed Particle Hydrodynamics, *Reports on Progress in Physics*, vol. 68, pp. 1703–1759. doi:10.1088/00344885/68/8/R01.
- Monaghan J. J., A. Kos (1999). Solitary Waves on a Cretan Beach, *Journal of Waterway, Port, Coastal and Ocean Engineering*, 125, 145–155. [http://dx.doi.org/10.1061/\(ASCE\)0733-950X\(1999\)125:3\(145\)](http://dx.doi.org/10.1061/(ASCE)0733-950X(1999)125:3(145)).
- Morris J. P., P. J. Fox and Y. Zhu (1997). Modeling low Reynolds number incompressible flows using SPH, *Journal of Computational Physics*, vol. 136 (1), pp. 214–226. doi:10.1006/jcph.1997.5776.
- Rustico E., G. Bilotta, G. Gallo, A. Hérault, C. Del Negro (2012). Smoothed Particle Hydrodynamics Simulations on Multi-GPU Systems, 20th Euromicro International Conference on Parallel, Distributed and Network-based Processing, Garching, pp. 384–391. doi:10.1109/PDP.2012.21.
- Rustico E., G. Bilotta, A. Hérault, C. Del Negro, G. Gallo (2014). Advances in Multi-GPU Smoothed Particle Hydrodynamics Simulations, *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 43–52. doi:10.1109/TPDS.2012.340.
- Saramito P., C. Smutek, B. Cordonnier (2013). Numerical modeling of shallow non-Newtonian flows: Part I. The 1D horizontal dam break problem revisited, *International Journal of Numerical Analysis and Modeling, Series B*, 4, pp. 283–298.
- Scifoni, S., M. Coltelli, M. Marsella, C. Proietti, Q. Napoleoni, A. Vicari, and C. Del Negro (2010). Mitigation of lava flow invasion hazard through optimized barrier configuration aided by numerical simulation: The case of the 2001 Etna eruption, *J. Volcanol. Geotherm. Res.*, 192, 16–26, doi:10.1016/j.jvolgeores.2010.02.002.
- Vicari, A., A. Cirauda, C. Del Negro, A. Hérault, and L. Fortuna (2009). Lava flow simulations using effusion rates from thermal infrared satellite imagery during the 2006 Etna eruption, *Nat. Hazards*, 50, 539–550, doi:10.1007/s11069-008-9306-7.
- Wei Z., R. A. Dalrymple, E. Rustico, A. Hérault, G. Bilotta, and H. Yeh (2015). SPH modeling of dynamic impact of tsunami bore on bridge piers, *Coastal Engineering* 104, 26–42, <https://doi.org/10.1016/j.coastaleng.2015.06.008>.
- Wei Z., R. A. Dalrymple, E. Rustico, A. Hérault, and G. Bilotta (2016). Simulation of nearshore tsunami breaking by Smoothed Particle Hydrodynamics method, *Journal of Waterway, Port, Coastal, and Ocean Engineering* 142, [https://doi.org/10.1061/\(ASCE\)WW.1943-5460.0000334](https://doi.org/10.1061/(ASCE)WW.1943-5460.0000334).
- Wendland H. (1995). Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, *Advances in Computational Mathematics*, 4(1), 389–396.
- Zago V., G. Bilotta, A. Cappello, R. A. Dalrymple, L. Fortuna, G. Ganci, A. Hérault, C. Del Negro (2017). Simulating complex fluids with Smoothed Particle Hydrodynamics, *Annals of Geophysics*, 60(6), PH669, 2017. doi:10.4401/ag-7362.
- Zago V., G. Bilotta, A. Hérault, R. A. Dalrymple, L. Fortuna, A. Cappello, G. Ganci, C. Del Negro (2018). Semi-implicit 3D SPH on GPU for lava flows. *Journal of Computational Physics*, in press.

*CORRESPONDING AUTHOR: Vito ZAGO,

Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Catania,

Osservatorio Etneo, Catania, Italy

Dipartimento di Ingegneria Elettrica, Elettronica e Informatica

Università di Catania, Catania, Italy

email: vito.zago@ingv.it

© 2019 the Istituto Nazionale di Geofisica e Vulcanologia.

All rights reserved