IL NUOVO CIMENTO **43 C** (2020) 122
DOI 10.1393/ncc/i2020-20122-4

COMMUNICATIONS: SIF Congress 2019

# Studying the rheology of geophysical flows with physical-mathematical models: An application of the GPUSPH particle engine

G. BILOTTA

*Istituto Nazionale di Geofisica e Vulcanologia, Osservatorio Etneo - Catania, Italy*

received 29 January 2020

**Summary.** — We present an application of GPUSPH, a particle engine based on the Smoothed Particle Hydrodynamics numerical method with implementation on Graphic Processing Units (GPUs) for high performance, to the study of the rheology of geophysical flows.

## 1. – Introduction

Most geophysical flows (*e.g.*, landslides, lahars, lava flows) can be classified as complex fluids, due to the presence of multiple phases (fluid/solid), non-Newtonian rheology, and/ or strong thermal effects (temperature-dependent rheological law, phase transition, etc.).

The study of the behavior of these flows is not only very interesting from a purely scientific perspective, but also of extreme importance in practice, given the strong impact they can have on populations living in or around relevant regions (volcanoes, geologically unstable areas, etc.). A reliable model of the behavior of the flow over time is therefore essential to be able to predict the evolution of such natural phenomena, for the purpose of scenario forecasting, hazard mitigation, and risk reduction.

The key aspect controlling the behavior of geophysical flows is their rheology. Rheology expresses the relation between shear stress $\tau$ and strain rate $\dot{\gamma}$ in fluids. Classical Newtonian fluids have a linear relationship $\tau = \mu\dot{\gamma}$, with the dynamic viscosity $\mu$ which is independent from $\dot{\gamma}$ (but may depend on other fluid properties such as temperature). Fluids that do not exhibit such a linear relationship are collectively classified as *non-Newtonian* fluids. These fluids can have arbitrarily constitutive laws, with arbitrary functional dependencies of the shear stress not only on the strain rate, but also on its evolution over time (which is the case, *e.g.*, for thixotropic and rheopectic fluids).

Within the large family of non-Newtonian fluids, of particular interest are the so-called "generalized Newotnian" fluids, for which it is still possible to define an *apparent* or *effective* viscosity $\mu(\dot{\gamma}, \dots)$ and express the shear stress as $\tau = \mu(\dot{\gamma}, \dots)\dot{\gamma}$. These

Creative Commons Attribution 4.0 License (http://creativecommons.org/licenses/by/4.0)

include, *e.g.*, Bingham fluids [1], that exhibit rigid-body behavior for stress below a *yield strength* $\tau_0$, and otherwise behave like Newtonian fluids $\tau = (\tau_0/\|\dot{\gamma}\| + \mu_0)\dot{\gamma}$, or power-law fluids that depend on some power of the second invariant of the shear rate tensor [2], *i.e.*, $\tau = \mu_n \|\dot{\gamma}\|^{n-1}\dot{\gamma}$, or Herschel-Bulkley fluids [3, 4] that generalize both.

The flexibility given by more sophisticated rheological laws comes at the cost of a higher number of parameters: while this makes fitting any specific data set easier, it also increases the likelihood of overfitting, making it harder to derive more general relationships for the parameters, especially when they may depend on fluid properties such as its composition or temperature.

Studying the rheology of geophysical flows as they naturally occur is further hindered by thing such as the speed of the phenomenon, the conditions under which it manifests (such as the extremely high temperatures (over 1200 K) reached by lava flows), as well as the possible evolution of the rheological law itself during the flow.

Laboratory (analog) experiments are essential to collect necessary data, but the extent of their effectiveness depends on the phenomenon being studied. While it is possible to reproduce landslides with different granularity distributions, for example, it is much harder to reproduce the wide range of physical characteristics and chemical compositions of lava flows: multiple experiments conducted on the same samples may generally produce different results.

Numerical methods for Computational Fluid Dynamics (CFD) can complement and help direct laboratory experiments, thanks to the possibility they offer to run arbitrary (virtual) test-cases, thus providing a (safe) test-bed for experimentation on the influence rheology has on flow emplacement and its evolution over time, and ultimately to validate the viscosity model.

The numerical approach to studying flow rheology is based on three key components. The first is a robust numerical method that allows discretization of the equations describing the physics of the flow, and its implementation as computer programs; as a plus, high-performance computing solutions can be adopted to speed up the execution of such computer programs, useful to simulate a wider range of test-cases in less time.

The second step is to *validate* the numerical code: this ensures that the code reproduces the expected results under known conditions, which may be simple cases where an analytical solution to the equations is known, or experimental data where all the relevant conditions and parameters are known.

Finally, the validated code can be used to run several test-cases with different functional laws for the rheology, as well as different functional dependencies of the rheological parameters on the fluid properties, and compare the results with field measurements or other experimental results, trying to find the laws that better fit the data.

We will present here some preliminary results concerning the application of this approach to the study of the rheology of lava flows, using the GPUSPH particle engine. We will begin with a presentation of Smoothed Particle Hydrodynamics (SPH), the numerical method underlying GPUSPH, followed by the results of the validation of the code, and finally illustrate an example where the discrepancy between the results of GPUSPH and those obtained in a laboratory experiment presents the opportunity for further investigation in the rheological law of lava flows.

## 2. – GPUSPH: SPH with GPU implementation

Simulating complex fluids must overcome not only the obstacles posed by the complexity of the laws describing their behavior, but also the challenges they pose to their discretization with traditional numerical methods: the free surface and natural topographies, internal surfaces such as solidification fronts, large deformations, heterogeneity of the physical properties are all aspects that require extreme care in classical Eulerian (such as finite differences or finite volumes) and Lagrangian (such as finite elements) mesh-based methods. This is the main reason behind our choice to rely on SPH, which is a Lagrangian, mesh-free method, and can handle many of these issues in a natural way.

**2**˙1. *Physical-mathematical model.* – Conservation of mass and momentum are the two fundamental laws from which the equations of the motion of a fluid can be derived.

Conservation of mass, in Lagrangian, differential form can be expressed as

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

where $D/Dt$ represents the Lagrangian (*i.e.*, total) derivative with respect to time, $\rho$ the physical density, and $\mathbf{u}$ the (Lagrangian) velocity. For incompressible flows, this reduces to $\nabla\cdot\mathbf{u} = 0$.

Conservation of momentum is expressed by the Navier-Stokes equations, which for incompressible flows with non-homogeneous viscosity can be written

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

where $P$ is the pressure, $\mu$ is the effective viscosity and $\mathbf{G} = \rho\mathbf{g}$ models external forces (typically, gravity).

For incompressible flows, the pressure can be derived from (2) by taking the divergence on both sides and leveraging the reduced $\nabla\cdot\mathbf{u} = 0$ equation of conservation of mass. However, with SPH a weakly compressible regime is usually assumed (WCSPH): in this case, the density evolves according to (1), but the density variation is assumed to be small (less than 10%) so that the incompressible form of the Navier-Stokes equations can still be used.

We use Cole's equation of state [5] to model the dependency of pressure on density:

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

with $\rho_0$ the reference density, $\gamma$ the polytropic constant, and $c_0$ the speed of sound. A weakly compressible regime is achieved when $c_0$ is at least an order of magnitude higher than the maximum velocity that can be experienced during the flow.

For a fluid with temperature-dependent rheology ($\mu = \mu(T)$), the conservation of mass and momentum must be coupled with the heat equation

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

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

**2˙2.** *Smoothed Particle Hydrodynamics.* – SPH is a Lagrangian, mesh-less method, originally devised for astrophysics, and then adopted and extended for general computational fluid dynamics [6, 7]. In SPH, the fluid is discretized as a set of particles, each representing a small volume of fluid. The particles' position and physical properties evolve over time according to a discretized version of eqs. (1), (2), (4) and the appropriate boundary conditions, that will be discussed in sect. **2**˙3. Discretization of the spatial derivatives is obtained weighting contributions from neighboring particles.

More formally, a family of *smoothing kernels* $W(\mathbf{r}, h)$ is defined, depending on a *smoothing length* $h$. These weighting functions should approximate Dirac's delta in the distribution sense, and thus in particular have unitary integral over the domain. Additionally, for practical reasons, they are chosen with radial symmetry and compact support, limiting the number of neighbors whose contribution must be considered. The neighborhood within the radius of influence of the kernel effectively takes the place of the stencil in traditional gridded methods, and the smoothing length can be considered the equivalent of the grid spacing: the method's consistency is assessed for the limit of $h \to 0$.

Due to the radial symmetry, $W(\cdot, h)$ only depends on the *norm* of the distance vector $r = |\mathbf{r}|$, *i.e.*, $W(\mathbf{r}, h) = \tilde{W}(r, h)$, and its gradient can be expressed as $\nabla_{\mathbf{r}} W(\mathbf{r}, h) = \mathbf{r} F(r, h)$ where $F(r, h) = (1/r)\partial\tilde{W}(r, h)/\partial r$.

Following the usual notation for SPH, we will write $\mathbf{r}_{\alpha\beta} = \mathbf{r}_\beta - \mathbf{r}_\alpha$ for the inter-particle distance with length $r_{\alpha\beta} = |\mathbf{r}_{\alpha\beta}|$, $\mathbf{u}_{\alpha\beta} = \mathbf{u}_\beta - \mathbf{u}_\alpha$ for the relative velocity, $W_{\alpha\beta} = W(\mathbf{r}_{\alpha\beta}, h)$ and $F_{\alpha\beta} = F(r_{\alpha\beta}, h)$.

We discretize the physical-mathematical equations modeling the fluid behavior as described in [8-10]. The mass continuity (1) takes the form

$$\frac{D\rho_\beta}{Dt} = \sum_\alpha m_\alpha \mathbf{u}_{\alpha\beta} \nabla_\beta W_{\alpha\beta}, \tag{5}$$

with $\alpha$ and $\beta$ being particles indices. For the momentum (2) and thermal equation (4), the Laplacian is discretized following [11], [12] and [13]. The Navier-Stokes momentum conservation equation then becomes

$$\frac{D\mathbf{u}_\beta}{Dt} = -\sum_\alpha \left(\frac{P_\alpha}{\rho_\alpha^2} + \frac{P_\beta}{\rho_\beta^2}\right) F_{\alpha\beta} m_\alpha \mathbf{r}_{\alpha\beta} + \sum_\alpha \frac{2\bar{\mu}_{\alpha\beta}}{\rho_\alpha \rho_\beta} F_{\alpha\beta} m_\alpha \mathbf{u}_{\alpha\beta} + \mathbf{g}, \tag{6}$$

where $\bar{\mu}_{\alpha\beta}$ is the harmonic mean of the particles' viscosities $\mu_\alpha$ and $\mu_\beta$, and $P_\alpha, P_\beta$ represent the particles' pressures, while the heat equation becomes

$$\frac{DT_\beta}{Dt} = \frac{1}{c_p} \sum_\alpha \frac{\bar{\kappa}_{\alpha\beta} T_{\alpha\beta}}{\rho_\alpha \rho_\beta} F_{\alpha\beta}, \tag{7}$$

again with $\bar{\kappa}_{\alpha\beta}$ the harmonic mean of $\kappa_\alpha$ and $\kappa_\beta$.

**2˙3.** *Boundary conditions.* – At the free surface, the momentum conditions are automatically satisfied by null pressure and stress, so no special treatment is needed. More sophisticated treatment however is needed for the thermal boundary terms and near solid boundaries.

**2**˙3.1. Free-surface thermal boundary conditions. For heat dissipation, the surface particles are detected according to [8, 9], and the particle surface is computed as in [10] to take into account the rarefaction of the flow: we compute the particle numerical volume [14] as

$$V_\beta = 1/\sum_\alpha W_{\alpha\beta},$$

from which we compute a numerical radius assuming a spherical shape, and then the numerical surface area $A_\beta$, assuming a circular shape. Temperature loss by thermal radiation is then computed using the Stefan-Boltzmann law

$$\left.\frac{DT_\beta}{Dt}\right|_{rad} = A_\beta \frac{K_B \kappa_\beta \epsilon}{m_\beta c_p}(T_\beta^4 - T_a^4) \tag{8}$$

and for temperature loss due to air convection we use

$$\left.\frac{DT_\beta}{Dt}\right|_{conv} = A_\beta \frac{\eta}{m_\beta c_p}(T_\beta - T_a). \tag{9}$$

$K_B$ is Stefan-Boltzmann's constant, $\epsilon$ the fluid emissivity, $T_a$ the ambient temperature and $\eta$ a convection coefficient.

**2**˙3.2. Wall boundary conditions. Since our fluids are viscous, we can assume no-slip boundary conditions between the fluid and solid boundaries (walls, ground). Analytically, this is achieved by imposing null normal velocity, and a tangential velocity matching the solid boundary velocity $\mathbf{v}_w$ (which in our cases will also be null).

Numerically, we realize these using the "dummy" boundary model proposed by Adami *et al.* [15]: solid walls are represented by multiple layers of boundary particles (enough to cover a full influence radius), and the no-slip boundary conditions are obtained by assigning pressure and velocity to the boundary particles in the following way.

The pressure must compensate the hydrostatic pressure of the adjacent fluid, so for each boundary particle we compute a Shepard-averaged pressure from the neighboring fluid, plus a hydrostatic correction term:

$$P_\beta = \left(\sum_{\alpha\in\mathcal{F}} P_\alpha W_{\alpha\beta} + \mathbf{g}\sum_{\alpha\in\mathcal{F}} \rho_\alpha \mathbf{x}_{\alpha\beta} W_{\alpha\beta}\right) \Bigg/ \left(\sum_{\alpha\in\mathcal{F}} W_{\alpha\beta}\right), \tag{10}$$

where $\mathcal{F}$ denotes the set of fluid neighbors.

The boundary particles' velocity is set to the wall velocity $\mathbf{v}_w$, except in the computation of the viscous term, for which the no-slip boundary condition is enforced by using a "viscous wall velocity" equal to

$$\mathbf{u}_\beta = \mathbf{u}_w - \left(\sum_{\alpha\in\mathcal{F}} \mathbf{u}_\alpha W_{\alpha\beta}\right) \Bigg/ \left(\sum_{\alpha\in\mathcal{F}} W_{\alpha\beta}\right). \tag{11}$$

For the heat equation, we adopt absorbing boundary conditions, implemented using a sponge layer approach: the temperature of the boundary particles is initially set equal

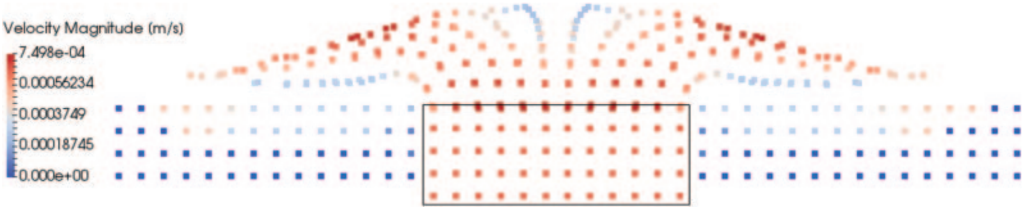

Fig. 1. – Cross-section lateral view of the GPUSPH inlet implementation. The frame highlights the buffer area. The snapshot is taken from the validation tests for BM3 mentioned in sect. **3**.

to a prescribed temperature $T_w$, and it evolves following the heat equation. Absorbing conditions are achieved if the boundary thickness $H_s$ is large enough to ensure that the temperature of the external layer (at a distance $H_s$ from the nearest fluid) remains at the prescribed temperature $T_w$ for the entire length of the simulation.

**2**˙3.3. Open boundary conditions. A complete lava flow simulation must also model the generation of new mass at the vent. This is achieved with inflow open boundary conditions. For implementation on SPH, this requires the generation of new particles, with a prescribed velocity and such that the overall mass flux matches the expected flow rate.

For particle generation, we adopt a “buffer” approach (fig. 1): this is a three-dimensional region with a cross-section matching the area of the vent, and with the same thickness as the solid walls. Particles in the buffer move according to the prescribed inflow velocity, and when a particle leaves the buffer area, a new one is generated on the opposite side. The pressure is computed using a three-dimensional extension of the Generalized Riemann Invariants approach described in [16] for the case of prescribed velocity.

**2**˙4. *Parallel implementation on GPU*. – One of the most interesting benefits of WC-SPH is that the acceleration, density change and temperature change can be computed independently for each particle, without solving linear systems. This makes WCSPH very easy to parallelize, and allows it to run efficiently on high-performance parallel computing hardware. GPUSPH([1]) was the first implementation of SPH that leverages this property of the method to run entirely on Graphic Processing Units (GPUs), gaining nearly two orders of magnitude in performance [17].

The engine is composed of two parts: a host part, running on CPU, takes care of the administrative tasks such as the initial setup of the particle system and loading/storing data. The actual computations needed for the evolution of the particle system run on GPU, by means of several *computational kernels* dedicated to the different steps of the simulation, such as *buildNeibs* that takes care of the construction of the neighbors list (needed to reduce the number of particle/particle interactions from $N^2$ to $NM$, where $N$ is the number of particles in the system and $M$ the number of neighbors that can be found within the influence radius of the smoothing kernel), *forces* that computes the time derivative of the velocity, density and temperature, from the discretized equations (5), (6), (7), and *euler* that integrates the particle properties given the time derivatives and time-step.

([1]) `https://gpusph.org`.

Since the first public release, designed for Newtonian fluids and the solution of the dynamics problem only (no temperature), GPUSPH has been extended to include support for the heat equation and a wide range of non-Newtonian rheologies, with the specific intent of supporting lava flows [8]. Moreover, the engine has been extended to allow the distribution of the computations across multiple GPUs [18], and even multiple nodes in a cluster, each equipped with multiple GPUs [19]. In these cases, the host part of GPUSPH also takes care of the data exchange between different GPUs (and between nodes), needed to ensure that computational devices have a consistent view of the current state of the particle system.

**2**˙5. *Semi-implicit integration.* – While the embarrassingly parallel nature of the standard WCSPH formulation is a benefit for implementation on parallel computing hardware, allowing the computation of a full time-step to be significantly speed up, this is paid in numerical terms by the need to adopt smaller time-steps: a fully explicit integration scheme that can exploit the parallel nature of the method is affected by stability conditions for the time-step.

For inviscid fluids, the dominant condition for the time-step is usually $\Delta t < C_s h/c_0$, where $c_0$ is the sound speed of the weakly compressible fluid, $h$ the influence radius of the kernel and $C_s < 1$ an appropriately chosen constant. This condition is necessary to ensure that sound waves propagate correctly through the fluid. Because of this restriction, the physical sound speed of the fluid is rarely used (being of the order of thousands of meters per second), and an artificial, lower sound speed is chosen instead, taken to be an order magnitude higher than the maximum possible flow velocity to guarantee the weak compressibility regime.

For viscous fluids, an *additional* restriction on the time-step is $\Delta t < C_v h^2/\mu$, where again $C_v < 1$ is a constant. This condition ensures that the diffusion effects of the viscosity propagate properly through the flow. An important thing to remark is that while the sound speed restriction scales *linearly* with the resolution, the viscous restriction scales *quadratically*: as a consequence, for any combination of sound speed and viscosity there is a resolution at which the viscous restriction becomes dominant.

Lava flow simulations, especially at lower temperatures, are dominated by the viscous restrictions, even at moderate and low resolutions; this can result in time-steps of the order of $10^{-8}$ or lower, leading not only to very long simulation times, but also to significant numerical issues when working in single precision, as is typically the case on GPUs.

To circumvent this, we have developed a semi-implicit integration scheme, that solves the inviscid part of the Navier-Stokes equation explicitly, whereas an implicit solver (requiring the resolution of a large, sparse linear system) is used for the viscous contribution. This eliminates the time-stepping restriction from the high viscosity, at some additional computation cost, with significant benefits for both performance and numerical stability [20]. The implementation of the Conjugate Gradient solver for the linear system is still parallel on GPU.

## 3. – Validation of GPUSPH for lava flows

Due to the complexity of the physical phenomenon, validating numerical codes for lava flow simulations is non-trivial. Cordonnier *et al.* [21] have proposed a set of progressively more complex *benchmarks* that numerical codes can be validated against. GPUSPH was validated against the first three test cases [10, 22]. The first three benchmarks (BM1 to

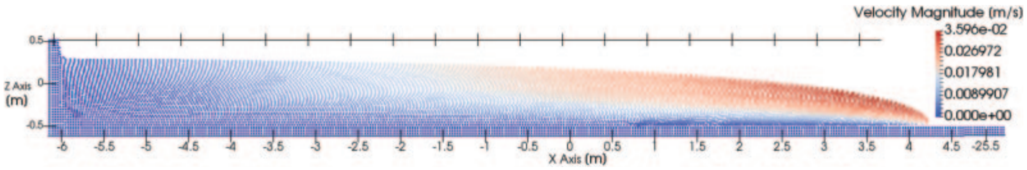

Fig. 2. – Cross-section lateral view of the BM1 test case simulated with GPUSPH.

BM3) posed no problems in GPUSPH, and the results are illustrated here. The fourth test case (BM4), the analog lava flow experiment, will be the topic of discussion for the possibility to use the engine as a tool to explore the rheological model of the flow.

BM1 is an isothermal dam break (fig. 2). For fluids with a high viscosity, semi-analytical solutions for the front progress and reservoir water level can be found. Convergence tests show that GPUSPH has quadratic convergence trends for the front, but only linear in the reservoir height [10, 22].

BM2 is an isothermal fluid injection from a point source on an inclined plane (fig. 3). Validation data is provided for the down-slope and cross-slope extent over time. Good results are obtained with GPUSPH, but only a linear convergence trend is observed. The main sources of error in this case are the finite dimension of the inlet (with a minimum size dictated by the SPH requirement that the minimum resolved length must be no less than the influence radius) and the rarefaction of the fluid during its spreading.

BM3 is an axisymmetric cooling and spreading test case (fig. 4). While this includes thermal effects, the viscosity of the fluid is constant, and temperature is only used as a passive tracker. Convergence is observed for the main part of the emplacement, although some difficulties were encountered in validating this test-case with GPUSPH [10, 22]: these were due partly to incomplete information about the thermal model in the description in the case [21], and partly due again to the rarefaction of the fluid in the outer rim of the emplacement.

## 4. – BM4: uncertainty in the viscosity model

The fourth benchmark proposed in [21] (BM4: split flow experiment) is a test case based on experimental data obtained at the Syracuse University Lava Project laboratory, melting fully degassed natural basalt. The flow hits a triangular obstacle and splits into two flows that merge again after the obstacle. The main data for the comparison with

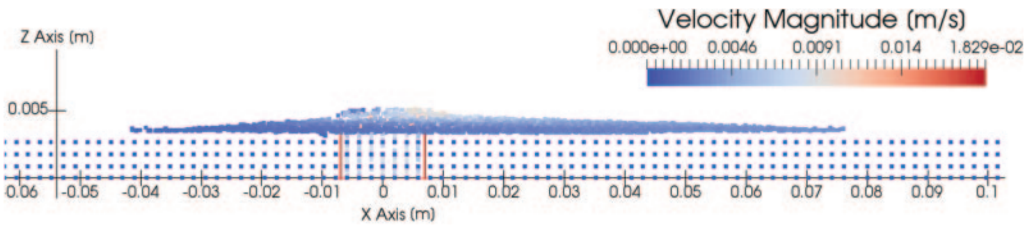

Fig. 3. – Cross-section lateral view of the BM2 test case simulated with GPUSPH. The floor appears flat because the slope is modelled by changing the angle of the gravity force vector, and the $x$ axis is aligned with the floor.

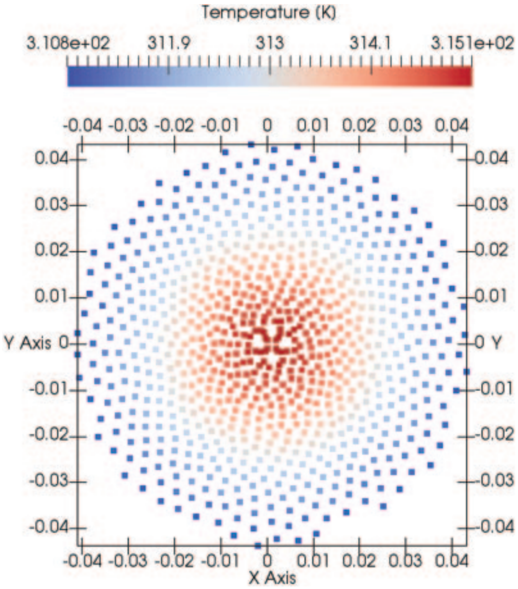

Fig. 4. – Top view of the BM3 test case simulated with GPUSPH showing the rarefaction of the particle distribution far from the injection point.

numerical models consists of: the progress of the flow front(s) before and after hitting the obstacle, the maximum width and the average flow surface temperature.

For the test case, the authors propose an empirically determined Newtonian rheology with temperature-dependent viscosity following the Vogel-Fulcher-Tammann equation $\log_{10}(\mu) = a + b/(T + c)$ where $T$ is the temperature expressed in kelvins.

With the proposed coefficients $a = -4.55, b = 5550, c = -610$, the BM4 viscosity law is

$$\log_{10}(\mu) = -4.55 + \frac{5550}{T - 610}. \tag{12}$$

Due to the high viscosities involved, we use the semi-implicit integration scheme [20]. Additionally, we clamp the maximum viscosity at the value for $T = 1190\,\mathrm{K}$, which with the law (12) corresponds to a kinematic viscosity $\nu = \mu/\rho = 44.45\,\mathrm{m}^2\mathrm{s}$. While this could be expected to lead to a faster flow at lower temperatures, the limit temperature is not actually reached by the flow during the simulation, and only boundary particles far from the flow are actually affected by the clamping, and they do not influence the flow behavior (further details about this approach can be found in [20]).

In fact, simulating BM4 with the law (12) results in a thicker, *slower* flow than in the analog experiments (fig. 5, right). This is confirmed by running convergence tests (at 64, 128, and 256 particles per meter) to try and isolate the model effect from the numerical effects.

However, using a slightly different rheological law as proposed in [23],

$$\log_{10}(\mu) = -5.94 + \frac{5550}{T - 610} \tag{13}$$

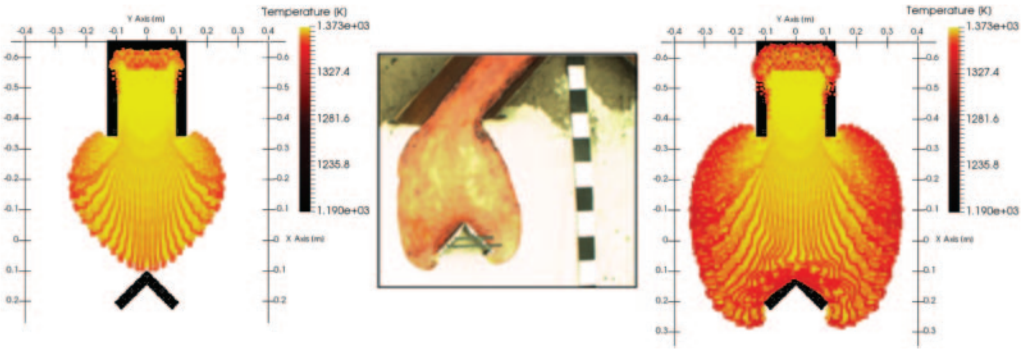

Fig. 5. – Top view comparison of the BM4 test case: analog results (center) and GPUSPH simulations with the originally proposed law (left) and the other proposal (right).

gives us numerical results that are closer to the experimental results (fig. 5, right).

## 5. – Discussion and conclusions

There are several effects that can contribute to the discrepancy between the experimental results and the numerical simulation.

On the numerical side, while GPUSPH has been validated for several test cases (including BM1, BM2 and BM3 from [21]), there may be aspects that affect this specific test case and are not fully captured by the other validation test cases. (It should be noted that this is independent of any discretization errors, since the discrepancy was consistently maintained even at higher resolutions.)

However, due to the analog nature of this experiment, it is also possible that the discrepancy is due to limitations in the preparation of the test case. In particular, the rheological law was determined experimentally, but no uncertainty or error bounds are reported for the value of the parameters (this is in contrast to, *e.g.*, the subsequent test case in [21], named CCTC1, where the range of uncertainty of the parameters are reported).

Indeed, the difference between (12) and (13) is only in the offset constant, with a relative change of less than 15% which, depending on the way the coefficient values were derived, may fall within the measurement uncertainty range, even with the consequent viscosity relative error being nearly 100%.

Additionally, the actual rheology of basaltic lava flows is non-Newtonian, with a Herschel-Bulkley model with coefficients depending on temperature, crystallinity and water content being most likely [24], even though as discussed in [23] for experiments similar to BM4, the non-linear exponent for the melts in these experiments is very close to 1, suggesting that a Newtonian approximation is possible.

A point of interest is that both (12) and (13) were proposed as rheological laws for the basaltic melt used in [23], with the former providing better fit for the unconfined case, and the latter for channel flow, and remarking that the channel flow case allows tighter control on the flow geometry and more accurate measurements: the second law can thus be considered more reliable than the one proposed in [21], and the numerical results of GPUSPH ultimately confirm this, while still stressing the need for further research, possibly leveraging the possibility for GPUSPH to model non-Newtonian fluids.

* * *

The author wishes to thank the Laboratory of Technologies for Volcanology (TechnoLab) of the Osservatorio Etneo, Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Catania, and in particular Vito Zago for the essential contributions to the research presented here.

## REFERENCES

[1] Bingham E. C., *Bull. Bureau Stds.*, **13** (1916) 309.
[2] Ostwald W., *Kolloid Z.*, **36** (1925) 99.
[3] Herschel W. H. and Bulkley R., *Kolloid Z.*, **39** (1926) 291.
[4] O'Brien J. S. and Julien P. Y., *J. Hydraul. Eng.*, **114** (1988) 877.
[5] Cole R. H., *Underwater Explosion* (Princeton University Press, Princeton, NJ) 1948.
[6] Gingold R. and Monaghan J. J., *Mon. Not. R. Astron. Soc.*, **181** (1977) 375.
[7] Monaghan J. J., *Rep. Prog. Phys.*, **68** (2005) 1703.
[8] Bilotta G., Hérault A., Cappello A., Ganci G. and Del Negro C., *Geol. Soc. London*, **SP426** (2016) 387.
[9] Hérault A., Bilotta G., Vicari A., Rustico E. and Del Negro C., *Ann. Geophys.*, **54** (2011) 600.
[10] Zago V., Bilotta G., Cappello A., Dalrymple R. A., Fortuna L., Ganci G., Hérault A. and Del Negro C., *Ann. Geophys.*, **62** (2019) VO224.
[11] Brookshaw L., *Proc. Astron. Soc. Aust.*, **6** (1985) 207.
[12] Cleary P. W. and Monaghan J. J., *J. Comput. Phys.*, **148** (1999) 227.
[13] Morris J. P., Fox P. J. and Zhu Y., *J. Comput. Phys.*, **136** (1997) 214.
[14] Hu X. Y. and Adams N. A., *J. Comput. Phys.*, **213** (2006) 844.
[15] Adami S., Hu X. Y. and Adams N. A., *J. Comput. Phys.*, **231** (2012) 7057.
[16] Ferrand M., Joly A., Kassiotis C., Violeau Damien, Leroy Agnès, Morel F.-X. and Rogers B., preprint hal-01376579.
[17] Hérault A., Bilotta G. and Dalrymple R. A., *J. Hydraul. Res.*, **48** (2010) 74.
[18] Rustico E., Bilotta G., Hérault A., Del Negro C. and Gallo G., *IEEE Trans. Parallel Distrib. Syst.*, **25** (2012) 43.
[19] Rustico E., Jankowski J., Hérault A., Bilotta G. and Del Negro C., in *Proceedings of the 9th SPHERIC Workshop* (National Research Council of Italy, Institute of Marine Engineering (CNR-INM)) 2014, p. 127.
[20] Zago V., Bilotta G., Hérault A., Dalrymple R. A., Fortuna L., Cappello A., Ganci G. and Del Negro C., *J. Comput. Phys.*, **375** (2018) 854.
[21] Cordonnier B., Lev E. and Garel F., *Geol. Soc. London*, **SP426** (2016) 425.
[22] Zago V., *Smoothed Particle Hydrodynamics method and flow dynamics: The case of lava numerical modeling and simulation*, PhD Thesis, Università di Catania, 2019.
[23] Lev E., Spiegelman M., Wysocki R. J. and Karson J. A., *J. Volcanol. Geotherm. Res.*, **247** (2012) 62.
[24] Spera F. J., Borgia A., Strimple J. and Feigenson M., *J. Geophys. Res.*, **93** (1988) 10273.