Semi-implicit 3D SPH on GPU for lava flows

Ingeneria dei Sistemi, Energetica, informatica e delle telecomunicazioni XXXI ciclo

UNIVERSITÀ di CATANIA

DIEEI

a.y. 2016-2017

Student: ***Vito Zago***

UNICT tutor: prof. ***Luigi Fortuna***

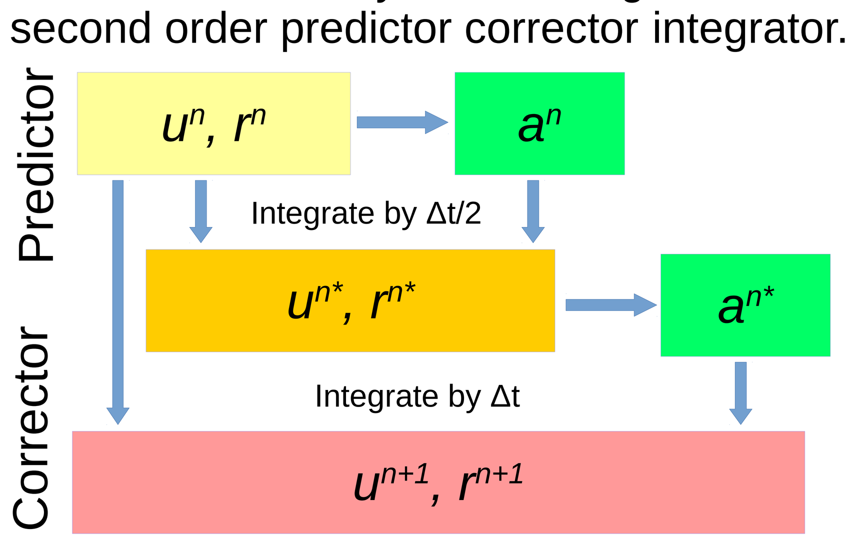
INGV tutor: dott. ***Ciro Del Negro***

INGV

Abstract: GPUSPH (1) is an implementation of Weakly-Compressible Smoothed Particle Hydrodynamics (WCSPH) that takes advantage of the embarrassingly parallel nature of the method to run on Graphic Processing Units (GPUs). Despite the massive speed-up granted by the use of GPUs, the application of GPUSPH to the simulation of highly viscous fluids is still problematic, due to the severe time-stepping restrictions imposed by the explicit integration scheme when the viscous term becomes dominant. This is an issue in the simulation of lava flows, where the thermal-dependent rheology can lead to kinematic viscosities in the order of 10^4 m^2 s^{-1} or more at low temperatures. To overcome this limitation, we present here a semi-implicit integration scheme (2, 3), where only the viscous part of the momentum equation is solved implicitly. We show that the proposed approach presents significant advantages in terms of simulation run times as well as better quality of the results over the fully explicit scheme.

Second order Integration scheme

To have a suitably small integration error, GPUSPH uses a second order predictor corrector integrator.



u: velocity
r: position
a: acceleration

In both explicit and semi implicit formulation the integration scheme have similar structure; the computation of *u* and *a* is different.

The issue with high viscosity

With a fully explicit integration scheme, CFL-like stability conditions on the time step are necessary. In the general case, separate conditions emerge from the acceleration magnitude, the sound speed, the viscous terms and the thermal equation, so that, for each particle β, we must have:

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

In most low-viscosity flows, the sound speed condition dominates (usually $\Delta t \sim 10^{-5}$s). For highly viscous flows the viscous term takes over; since the latter is quadratic in the resolution, it leads to very small time steps when using fine discretization ($\Delta t \sim 10^{-8}$s or lower). Such small time steps cause two main issues:

- Increase of the simulation time;
- Numerical precision affecting the integration process.

A semi-implicit integration scheme allows to disregard the viscous stability condition.

The Explicit integration scheme

1) Compute accelerations at instant *n*

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

2) Compute half-step intermediate positions and velocities:

a) $\mathbf{x}^{(n\star)} = \mathbf{x}^{(n)} + \mathbf{u}^{(n)}\frac{\Delta t}{2}$,
b) $\mathbf{u}^{(n\star)} = \mathbf{u}^{(n)} + \mathbf{a}^{(n)}\frac{\Delta t}{2}$

3) compute corrected accelerations:

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

4) compute new positions and velocities:

a) $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + (\mathbf{u}^{(n)} + \mathbf{a}^{(n\star)}\frac{\Delta t}{2})\Delta t$,
b) $\mathbf{u}^{(n+1)} = \mathbf{u}^{(n)} + \mathbf{a}^{(n\star)}\Delta t$.

The Semi-implicit integration scheme

1) compute the inviscid acceleration ā, i.e. the acceleration evaluated without considering the viscous term, at instant n:

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

2) compute half-step intermediate positions and velocities:

a) $\mathbf{x}^{(n\star)} = \mathbf{x}^{(n)} + \mathbf{u}^{(n)}\frac{\Delta t}{2}$,
b) solve the three linear systems $\mathbf{A}_{\frac{\Delta t}{2}}\mathbf{u}^{(n\star)} = \mathbf{u}^{(n)} + \bar{\mathbf{a}}^{(n)}\frac{\Delta t}{2}$

3) compute corrected inviscid accelerations:

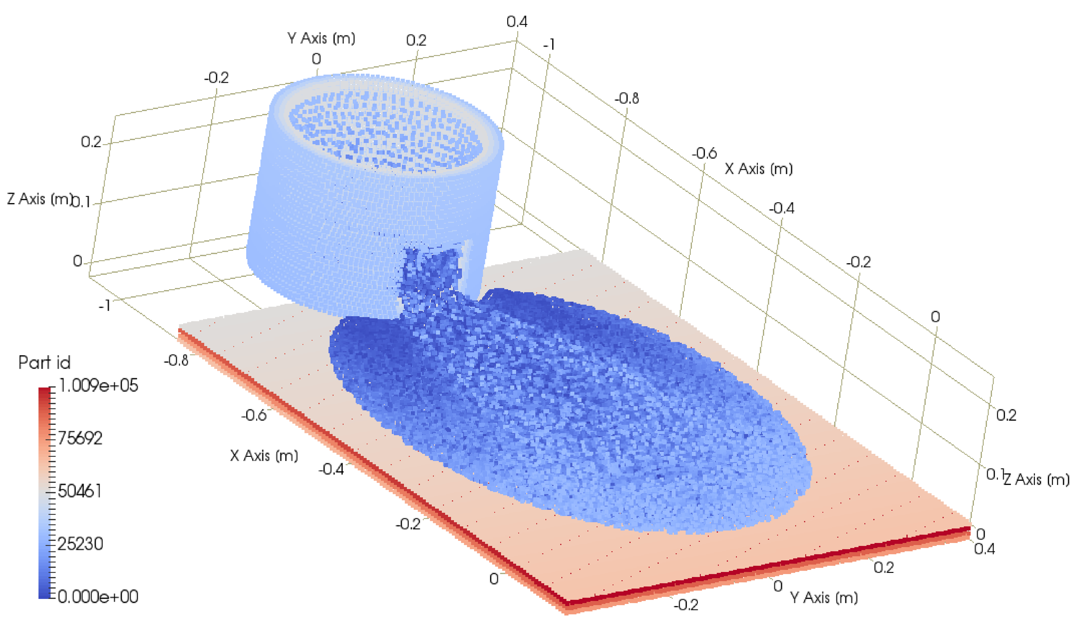
$$\bar{\mathbf{a}}^{(n\star)} = \mathbf{a}(\mathbf{x}^{(n\star)}, \mathbf{u}^{(n\star)}, \rho^{(n\star)}, T^{(n\star)}),$$

4) compute new positions and velocities:

a) compute the new velocity solving: $\mathbf{A}^\star\mathbf{u}^{(n+1)} = \mathbf{u}^{(n)} + \bar{\mathbf{a}}^{(n\star)}\Delta t$
b) compute the new position using the trapezoidal rule:

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + (\mathbf{u}^{(n)} + \mathbf{u}^{(n+1)})\Delta t/2.$$

Performances test

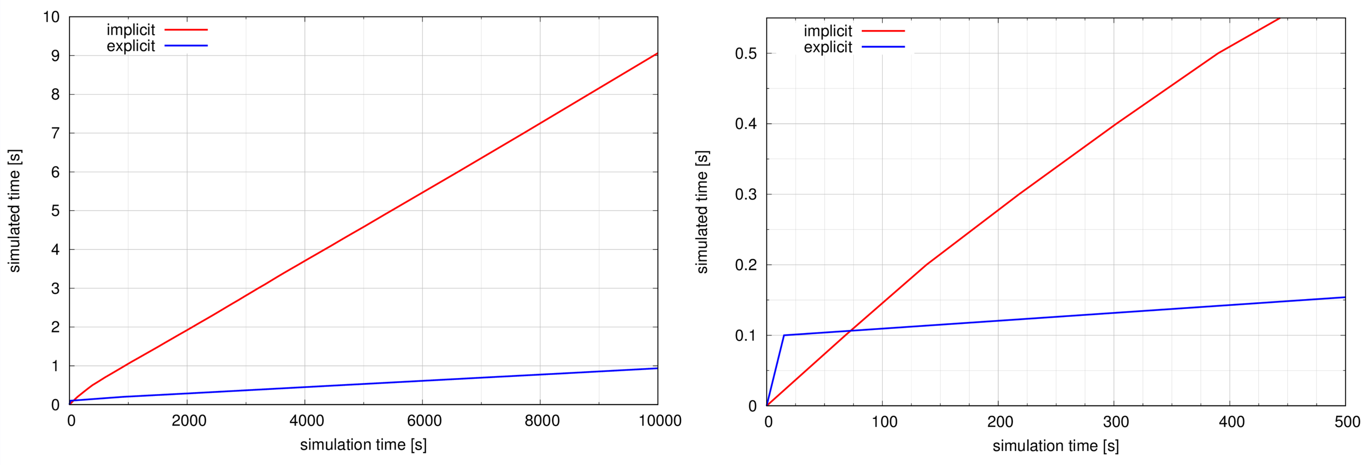


Problem description:
Lava spreading onto an inclined plane from a crucible.
- Domain size: 80cm x 123 cm x 55 cm
- Δp = 0.008m, 100,923 particles
- Thermal dependent viscosity [Pa s]:

$$\mu = 10\exp\left(-5.94 + \frac{5500}{T - 610}\right)$$

- density: ρ=2350 kg/m3
- max temperature: T0 = 1323K
- min viscosity: ν_{T0} = 0.0252 m²/s
- max (limited) viscosity: ν_{max} = 46.45 m²/s

Performance comparison between explicit and semi-implcit



Above: on the left, the comparison of simulation speed between explicit and semi-implicit formulations. On the right, a zoom on the initial region. At steady state the explicit simulates 1s in 12, 326s (about three and half an hour) while the semi-implicit in 1,124 s (about 19 minutes).

Simulations were run on an NVIDIA Titan X GPU. In the explicit case the simulation speed decreases once lava touches the cold ground (Δt=3.5 10^{-7}s) (4). The semi-implicit formulation gives an almost constant and much favorable speed (Δt=2.5 10^{-5}s). At steady state the gain in simulation time is about 11.

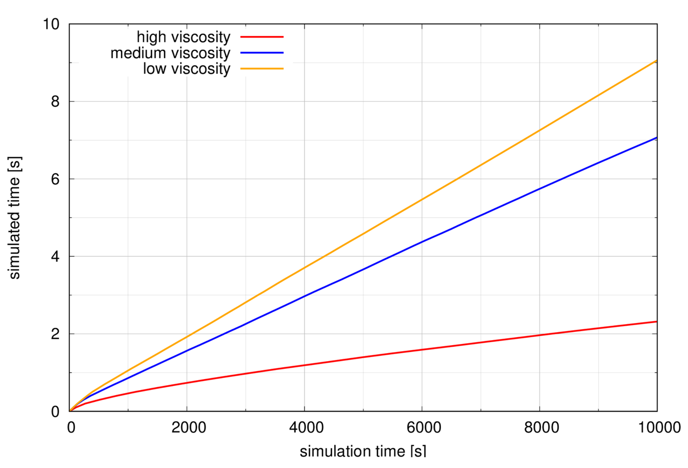
Influence of the main parameters on performances

1) Varying the maximum viscosity

The maximum viscosity over temperature is introduced to allow the simulation to run with the explicit scheme; Using the semi-implicit scheme, higher viscosity values can lead to a better realization of the viscous model.

ν_{medium} = 81.93 m²/s → 80% of initial performance.
ν_{high} = 67,258 m²/s → 27% of initial performance.

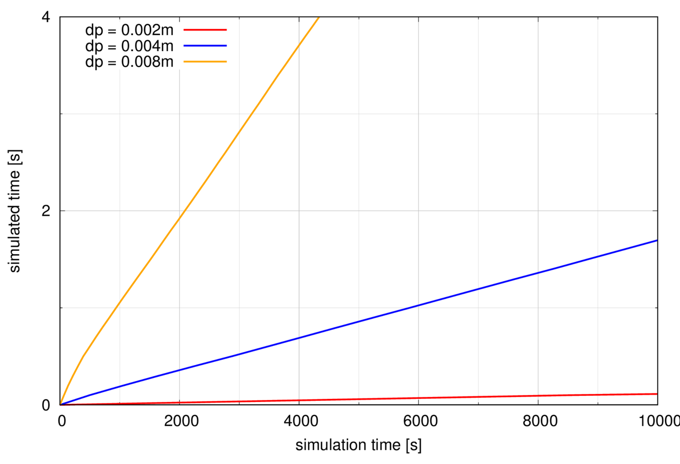
In the latter case, the viscous time-step would be in the order of 10^{-10}, impossible to hndle in single precision.



2) Varying the resolution of the spatial discretization

The resolution affects the simulation speed through the time-step and the dimension of the linear system.

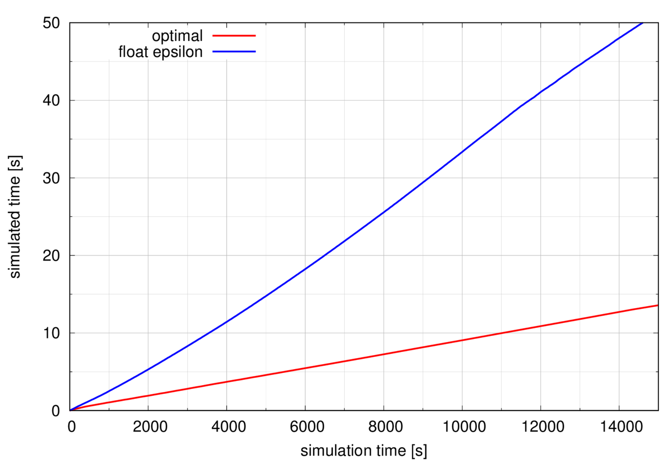
Using ν_{low}= 46.45 m²/s
Δp = 0.008m→ 100,923 parts → 1s in 1,124s
Δp = 0.004m→ 495,658 parts → 1s in 5,700s
Δp = 0.002m→ 2,781,568 parts → 1s in 116,000s

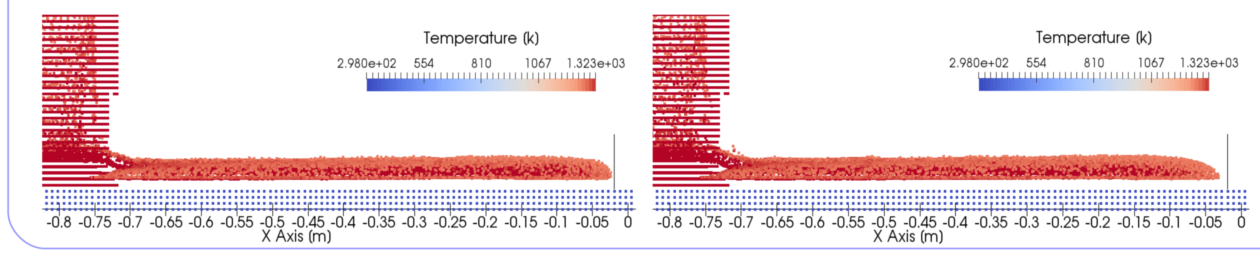


3) Varying the threshold of the system solver

Two thresholds are considered:
- optimal: ensures the smallest numerical residual
- customized: leaves a residual suited for the problem.

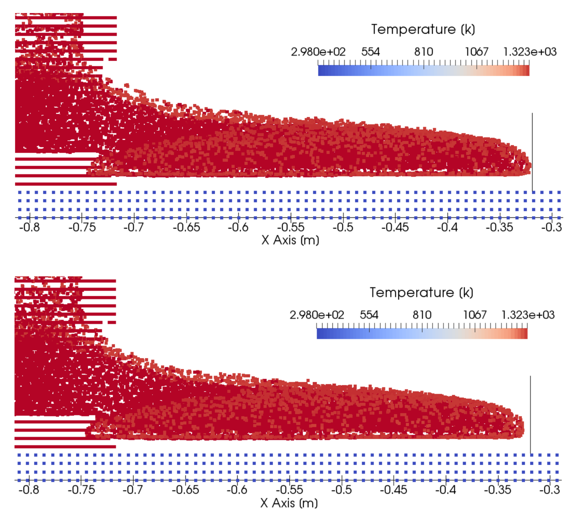
With the latter, chosen as the machine epsilon, the result (figure below) is acceptable and 1s is simulated in 287 s, 43 times faster than the explicit case.





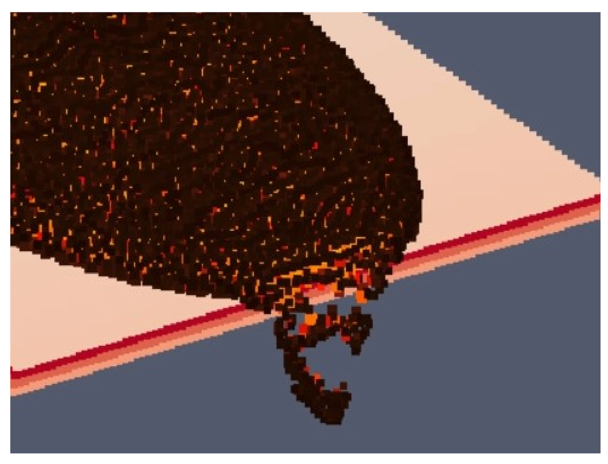
Above: Simulation speeds over variations of the main parameters.
On the left:
left: emplacement with optimal threshold, right: emplacement with customized threshold.

Advantages on the simulation results



Above: on the top, a lateral view at *t* = *2s* of the emplacement simulated using the explicit formulation. On the bottom, the result obtained using the semi-implicit formulation.

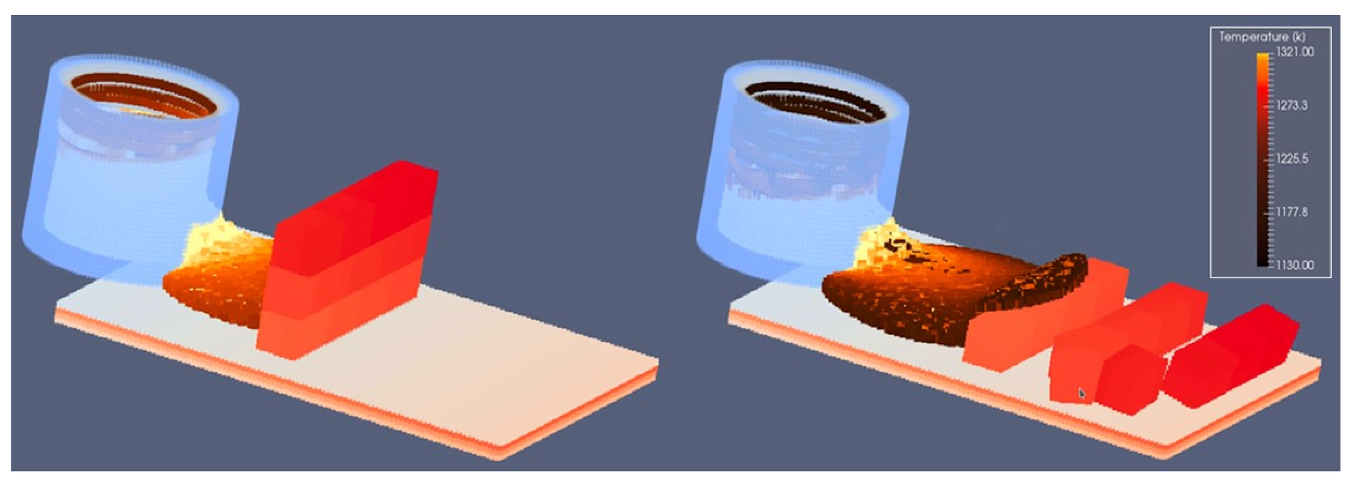
The larger time-step allows a better rounding error when integrating in single precision. Moreover the lower number of steps reduces the accumulation of numerical noise.
This can be seen in the lower amount of artifacts given by the semi-implicit formulation.
Differences on the emplacement length are due to the use of different velocity to compute the acceleration.



Above: Simulation of lava dropping.

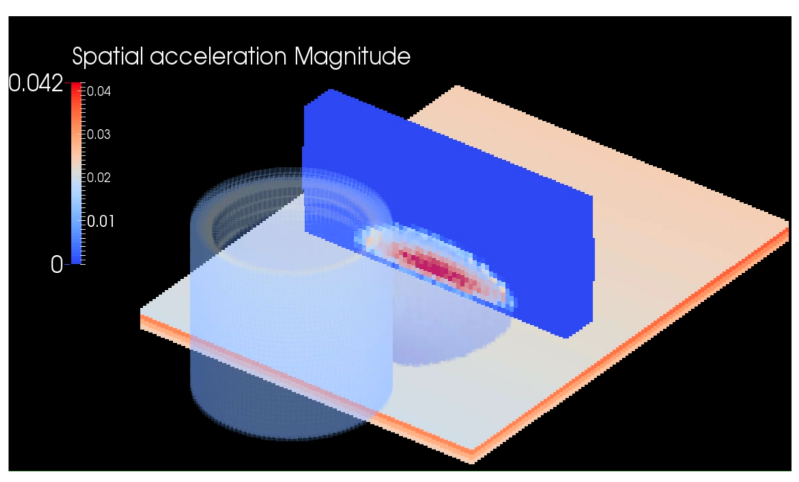
The semi-implicit scheme allows to use realistic viscosity values allowing a faithful reproduction of phenomena like dropping.

Immediate industrial applications



On the left: lava flowing toward a wall. The wall is modeled considering single tiles and using realistic dynamical properties.
Below:
Visualization of the distribution of force exerted by the lava onto the wall.

The semi-implicit model has been extended to manage floating objects and simulate the interaction of lava with a more realistic environment. As first case of study, a lava flow reaching a wall has been simulated . The study gives information on how the structure reacts to the action of the fluid and, on the other side, on what is the action exerted onto the structure by the luid. The latter aspect is useful for the design of buildings in volcanic environment.



Conclusions and future work

We have shown how a semi-implicit integration scheme in WCSPH allows the simulation of highly viscous fluids with larger time-steps, bringing in the best case to a reduction of the simulation time of about 43 times.

We have shown the effect that the main parameters have on the simulation time, and we saw that for relatively high viscosities or relatively fine resolutions, the semi-implicit approach is the only one able to run a simulation when working in single precision.

The largest downside to the semi-implicit scheme is the much higher computation cost for each integration step, so that performance-wise it only becomes advantageous for very large viscosities or very fine resolutions. A possible approach in this sense would be to dynamically switch between the explicit and the semi-implicit integration schemes depending on resolution and viscosity.

Accuracy of the results could also be improved by adopting more sophisticated boundary conditions, such as the Dummy Boundary model, at the cost of matrix simplicity.

Large simulation cannot run on a single GPU; this limitation could be avoided by introducing support for multiple GPUs in the semi-implicit scheme.

References:
[1] V. Zago et al. Simulating complex fluids with Smoothed Particle Hydrodynamics. Accepted by Annals of Geophysics, 2017.
[2] V. Zago et al., Implicit integration of the viscous term and GPU implementation in GPUSPH for lava flows. Proceedings of the 12th SPHERIC Conference, Ourense, Spain, 2017.
[3] V. Zago et al. Semi-implicit 3D SPH on GPU for lava flows. Submitted to Journal of Computational Physics, 2017.
[4] G. Ganci, V. Zago et al. Lava Cooling modelled with GPUSPH, European Geosciences Union (EGU) General Assembly 2017, Vienna, Austria, April 2017.