ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

EFFICIENT TWO-DIMENSIONAL SIMULATION MODELS FOR HYDRAULIC AND MORPHODYNAMIC TRANSIENTS

Pilar Garcia-Navarro[1], Javier Murillo[1], Mario Morales-Hernandez[1], Carmelo Juez[2], and Asier Lacasta[1]

[1](1) Computational Hydraulics Group. University of Zaragoza. Spain.
e-mail: pigar@unizar.es, jnurillo@unizar.es, mmorales@unizar.es, alacasta@unizar.es

[2] EPFL, Lausanne, Switzerland.
e-mail: carmelo.juez@epfl.ch

Keywords: CUDA; GPU; Landslides; Numerical modeling; Shallow Flow; Coulomb forces; bed load transport, Exner equation.

Abstract. *Recent advances in the simulation of shallow flows over mobile bed have shown that accurate and stable results in realistic problems can be provided if an appropriate coupling between the shallow water equations (SWE) and the Exner equation is performed. In this way the computational cost may become unaffordable in situations involving large time and space scales. Therefore, for restoring the numerical efficiency, the coupling technique is simplified, not decreasing the number of waves involved in the Riemann problem but simplifying their definitions. The effects of the approximations made are tested against experimental data which include transient problems over erodible bed. The simplified model is formulated under a general framework able to insert any desirable discharge solid load formula. Also, the movement of poorly sorted material over steep areas constitutes a hazardous environmental problem. Computational tools help in the understanding and predictions of such landslides. The main drawback is the high computational effort required for obtaining accurate numerical solutions due to the high number of cells involved. However, recent advances in massive parallelization techniques for 2D hydraulic models are able to reduce computer times by orders of magnitude making 2D applications competitive and practical for operational flood prediction in large river reaches. Moreover, high performance code development can take advantage of general purpose and inexpensive Graphical Processing Units (GPU), allowing to run 2D simulations more than 100 times faster than old generation 2D codes, in some cases.*

1 INTRODUCTION

Landslides play an important role in the evolution of landscape and constitute an important environmental topic. They can be responsible for dramatic civil damages and that is the reason why the building of defenses and barriers is required. The computational tools are a suitable partner for developing a careful design of such elements. Over recent years, reliable predictions of the spreading of granular material have been obtained [21, 22, 17, 3] and numerical results have been validated with respect to series of experiments based on granular dry flows [**?**, 8, 23, 14, 15]. In particular, [19, 9] have recently presented a robust finite volume upwind scheme which includes the presence of steep slopes leading to obtain promising results.

Under the presence of complex topography or the presence of hydraulic structures, the use of 2D or 3D hydrodynamic models may be required. 2D depth averaged models are widely accepted for most practical purposes in complex cases. These models provide predictions for the water depth and the two-dimensional, depth averaged, flow velocity field at the cost of a fine topographic representation. The bed evolution is frequently computed through the Exner equation. Asynchronous techniques are based on the assumption that morphodynamic time scales are not relevant enough for altering the hydrodynamic variables within the interval of a computational time step. Therefore, the fluid mass and momentum equations are solved apart from (decoupled of) the Exner equation. Conversely, synchronous procedures assume that changes in the morphodynamic and hydrodynamic quantities take place within the same time scale, i.e. equations for both phases are solved at the same time and with the same time restriction. As stated in [10], unsteady flows with a wide range of hydrodynamic and morphodynamic situations can only be properly tackled by means of a synchronous technique.

Once the forecasting capacity of the computational tool has been reached another important concern is the improvement of the efficiency in terms of the computational cost. This type of geophysical flow involves the study of huge domains where the accuracy of the results is tied to the resolution of the Digital Terrain Model considered. Hence, a high number of cells is usually needed in the simulation. For this reason, the hardware GPU emerges as a promising strategy for handling this environmental and up to date problem.

In terms of scientific computation, the last four decades have followed Moore's law [16] where the number of transistors on a chip increased exponentially. This integration allowed to obtain faster and faster applications just recompiling the code for this new processors. Unfortunately, power has become the primary design constraint for chip designers, where both energy and power dissipation create a technological barrier for the integration capacity [4]. Nevertheless, Multi-Core micro architecture together with an adequate programming model (OpenMP is one of the most extended) brings a chance to exploit the parallelism of some parts of the code [24]. Moreover, Multi-Core paradigm has a large power consumption rate when performing small tasks and, for these purposes the Many-Core systems appears to be a very interesting option [2]. Many-Core architectures are those composed of smaller and not so complex cores that, usually have special purposes. Industrial implementation of this solution has been obtained in the field of Graphical Processing, where several efforts have been devoted to make more powerful devices. Indeed, this technology has been historically oriented to a very particular task of performing shading operations when rendering graphics. The purpose of the present work is to apply this hardware to the simulation of hazardous and high time consuming geophysical flows.

2 MATHEMATICAL MODEL

2.1 Equations

The mathematical model considered for reproducing the landslides phenomenon is based on the shallow flow equations, where the general three-dimensional conservation laws are depth averaged. The pressure distribution is considered hydrostatic and as frictional terms, only Coulomb type friction forces are assumed so that the 2D hydrodynamic equations are written in global coordinates as follows:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} = \mathbf{S}_\tau + \mathbf{S}_b \tag{1}$$

where

$$\mathbf{U} = (h, hu, hv)^T \tag{2}$$

are the conserved variables with h representing granular material depth in the z coordinate and (u, v) the depth averaged components of the velocity vector. The fluxes are given by

$$\begin{aligned} \mathbf{F} &= \left(hu, hu^2 + \frac{1}{2} g_\psi h^2, huv\right)^T \\ \mathbf{G} &= \left(hv, huv, hv^2 + \frac{1}{2} g_\psi h^2\right)^T \end{aligned} \tag{3}$$

with $g_\psi = g \cos^2 \psi$ and ψ the direction cosine of the bed normal with respect to the vertical. The physical basis of this gravity projection is explained in [9] and it is of utmost importance for not ruining the numerical predictions when the simulation involves the presence of steep slopes.

The term $\mathbf{S}_\tau$ notes the frictional effects in the bed, and is defined as

$$\mathbf{S}_\tau = \left(0, -\frac{\tau_{b,x}}{\rho}, -\frac{\tau_{b,y}}{\rho}\right)^T \tag{4}$$

with $\tau_{b,x}, \tau_{b,y}$ the bed shear stress in the x and y directions respectively and ρ the density of the granular mass. Since the geophysical flows considered in this work are dense, the main rheological properties are governed by the frictional forces. These interactions between the sand grains are computed by means of the Manning and/or Coulomb laws depending on the case. The latter is based on the internal friction angle of the material, θ_b.On the other hand, the term $\mathbf{S}_b$ is defined for gathering the information relative to the pressure force exerted over the bottom.

Thanks to the hyperbolic character of (1) it is possible to obtain a Jacobian matrix, $\mathbf{J_n}$, which is built by means of the flux normal to a direction given by the unit vector $\mathbf{n}$, $\mathbf{E_n} = \mathbf{F}n_x + \mathbf{G}n_y$,

$$\mathbf{J_n} = \frac{\partial \mathbf{E_n}}{\partial \mathbf{U}} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} n_x + \frac{\partial \mathbf{G}}{\partial \mathbf{U}} n_y \tag{5}$$

whose components are

$$\mathbf{J}_n = \begin{pmatrix} 0 & n_x & n_y \\ (g_\psi h - u^2) n_x - uvn_y & vn_y + 2un_x & un_y \\ (g_\psi h - v^2) n_y - uvn_x & vn_x & un_x + 2vn_y \end{pmatrix} \tag{6}$$

The eigenvalues of this Jacobian matrix constitute the basis of the upwind technique which is detailed in the next subsection.

On the other hand, the bed evolution is modeled through the Exner equation, which is basically a movable bed continuity equation where the bed level time variations are due to the solid fluxes which cross the control volume. In this work the authors only focus on highly concentrated bed-load phenomena and, consequently, the 2D Exner equation is:

$$\frac{\partial z}{\partial t} + \xi \frac{\partial q_{s,x}}{\partial x} + \xi \frac{\partial q_{s,y}}{\partial y} = 0 \tag{7}$$

where $\xi = \frac{1}{1-p}$, p is the material porosity and $q_{s,x}$, $q_{s,y}$ are the solid fluxes. They are computed as a function of excess bed shear stress with respect to the critical value and taking into account the bed shear stress direction. This bedload transport is often expressed through the following dimensionless parameter:

$$\Phi = \frac{|\mathbf{q}_s|}{\sqrt{g_\psi (s-1) d_m^3}} \tag{8}$$

where $s = \rho_s / \rho_w$ is the ratio of solid material (ρ_s) over water (ρ_w) densities, and d_m is the grain median diameter. According to the numerical assessment performed in [9] the empirical Smart (1984) formula is chosen for computing the dimensionless bedload discharge as follows:

$$\Phi = 4 \ (d_{90}/d_{30})^{0.2} \ F \ S^{0.1} \theta^{1/2} (\theta - \theta_c^S) \tag{9}$$

where S is the velocity vector projected over the bed slope vector, as in [9], for distinguishing between positive and negative sloping beds. On the other hand d_{90} and d_{30} are grain diameter values for which 90% and 30% of the weight of a nonuniform sample is finer respectively. F is the Froude number, θ is the dimensionless shear stress and θ_c^S is the critical shear stress. This formula is only applied when the shear stress is larger than the critical shear stress. Otherwise there is no sediment transport.

2.2 Numerical method

The numerical scheme is constructed by defining an approximate Jacobian matrix $\widetilde{\mathbf{J}}$ at each k edge between neighboring cells defined through the normal flux $\mathbf{E_n}$ so that the volume integral in the cell at time t^{n+1} is expressed as:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \sum_{k=1}^{NE} \sum_{m=1}^{3} (\widetilde{\lambda}^- \widetilde{\alpha} - \widetilde{\beta}^-)_k^m \widetilde{\mathbf{e}}_k^m l_k \frac{\Delta t}{A_i} \tag{10}$$

The superscript minus in (10) implies that only the incoming waves are considered for updating the flow variables of each cell, defining $\widetilde{\lambda}^- = \frac{1}{2}\left(\widetilde{\lambda} - \left|\widetilde{\lambda}\right|\right)$. Further, special care is considered when calculating wet/dry fronts. The strategy proposed is based on enforcing positive values of interface discrete water depths coming from a detailed study of the Riemann problem [18]. When they become negative, the numerical values of the friction and bed slope source terms is reduced instead of diminishing the time step.

Equation (7) is also integrated in a grid cell Ω_i. Using Gauss theorem:

$$\frac{\partial}{\partial t} \int_{\Omega_i} z d\Omega + \oint_{\partial \Omega_i} q_{s\mathbf{n}} dl = 0 \tag{11}$$

where $q_{s\mathbf{n}} = (q_{s,x}n_x + q_{s,y}n_y)$.

Assuming a piecewise representation of the variable z and that the second integral can be written as the sum of fluxes across the cell edges, the bed level is updated as:

$$z_i^{n+1} = z_i^n - \sum_{k=1}^{NE} \xi q^*_{s\mathbf{n},k} \frac{\Delta t \; l_k}{A_i} \tag{12}$$

where:

$$q^*_{s\mathbf{n},k} = \begin{cases} q_{s\mathbf{n},i} & \text{if} \quad \widetilde{\lambda}_s > 0 \\ q_{s\mathbf{n},j} & \text{if} \quad \widetilde{\lambda}_s < 0 \end{cases} \tag{13}$$

where $q_{s\mathbf{n},i}$ and $q_{s\mathbf{n},j}$ are the bed load discharge computed at the neighboring cells i, j, and $\widetilde{\lambda}_s$ is the numerical bed celerity estimated as:

$$\widetilde{\lambda}_s = \frac{\delta q_{s\mathbf{n},k}}{\delta z_k} \tag{14}$$

with $\delta q_{s\mathbf{n},k} = q_{s\mathbf{n},j} - q_{s\mathbf{n},i}$ and $\delta(z_k) = z_j - z_i$.

2.2.1 Stability criteria

The explicitly updated conserved variables are defined through the fluxes obtained within each cell, so, the computational time step has to be chosen small enough for ensuring a stability region. Traditionally, the numerical stability has been controlled through a dimensionless parameter, CFL,

$$\Delta t = CFL \, \frac{\min(\chi)}{\max|\widetilde{\lambda}^m|} \qquad CFL \leq 0.5 \tag{15}$$

where χ is a relevant distance between neighboring cells [18] and $\widetilde{\lambda}^m$ are the hydrodynamic celerities. The stability criterion is revisited for including a discrete estimation of the bed celerity, $\widetilde{\lambda}_s$, as in [10],

$$\Delta t = CFL \, \frac{\min(\chi)}{\max|\widetilde{\lambda}^m, \widetilde{\lambda}_s|} \qquad CFL \leq 0.5 \tag{16}$$

With this numerical strategy, the stability condition takes into consideration the most restrictive numerical wave speed coming from the hydrodynamical and morphodynamical solvers. The resulting global time step is used for updating the whole set of conserved hydrodynamic and morphological variables in the system of equations.

3 NUMERICAL RESULTS

3.1 Landslide in a practical application

The test considered for the validation of the GPU implementation is based on a real topography of a catchment [5]. The Arnas catchment is located in the northern Spain Pyrenees, in the Borau valley, and has a surface of 2.84 km^2, ranging in altitude from around 900 to 1340 meters. Geologically, the catchment lies over Eocene flysch formations and has suffered land use and coverage changes in recent decades, generating a mixed vegetation cover which ranges from forest patches, dense and open shrubs, grassland cover and bare land. The assumption made by the authors is that the part of the bare terrain is composed by poorly sorted material and the idea

is to verify the maximum run out and potential consequences of a massive mobilization of that material. In Figure 1 is plotted the fixed bed rock and onto it the moving granular material. Due to the large dimensions of the catchment, the number of cells involved in the calculus is over 869000, making this type of phenomena a suitable candidate for GPU strategy.

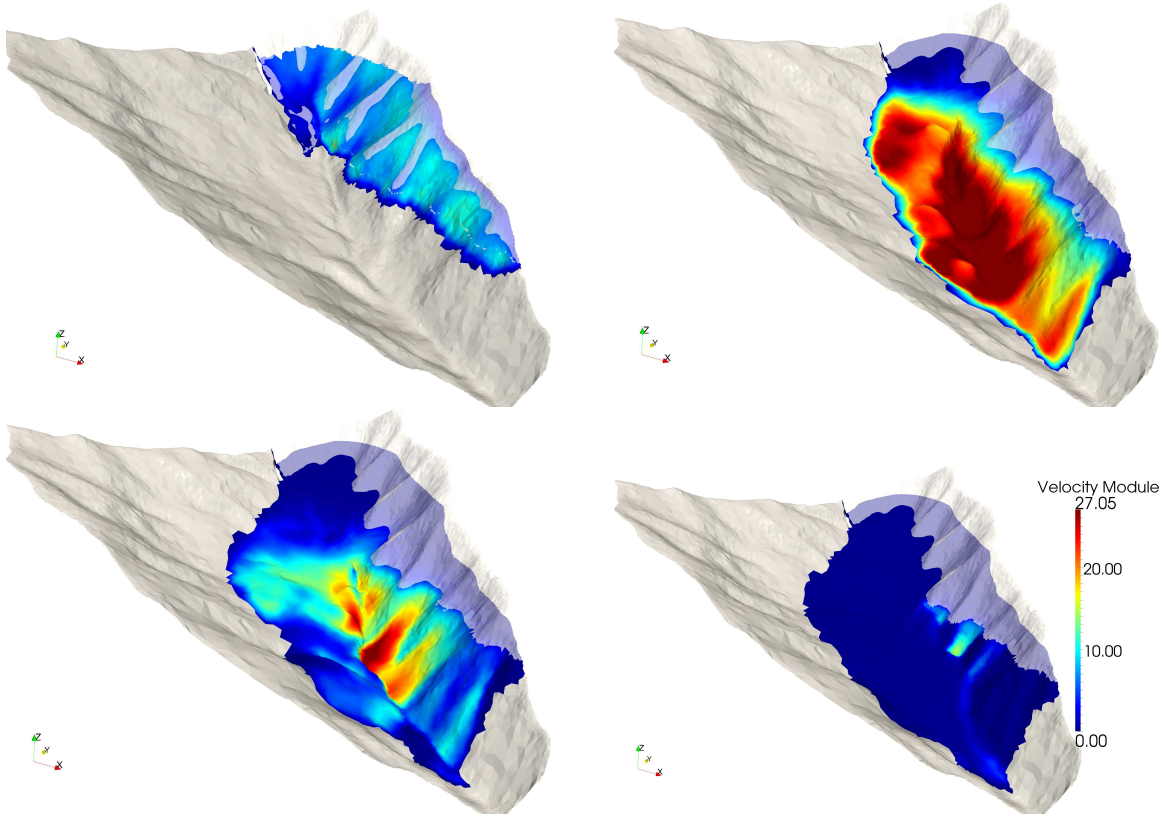

Figure 1: Evolution of the depth with the velocity (m/s) at time $t = 1s$. (top-left), $t = 2s$. (top-right), $t = 4s$. (bottom-left) and $t = 54s$. (bottom-right). The transparent element corresponds to the initial condition.

4 Computational load

The necessity of refined mesh for tests 1, 2 and 3 is justified by the important effect of the bed slope in the phenomena and then, an accurate representation of the topography is required. This means that the time-step length restricted by the CFL condition, is very low and then many time-steps are required to complete the simulation. Moreover, each time-step has a very high computational cost because of the number of cells. On the other hand, the last test case does not require such refinement because of the uncertainty of the data acquisition. Indeed, the mesh has been designed using the most accurate available LIDAR data for the topography, being the resolution of 5mx5m.

In order to analyze the computational load of the numerical engine as well as the gain obtained using both Single-Core and Multi-Core approaches, a sequential version using an *Intel Core i7 3770k@3.5 GHz* is compared against an OpenMP (4 Threads) parallel version and a GPU version without taking into account the improvements proposed on the paper (not optimized) and against an optimized version running on a *NVIDIA Tesla c2075* GPU. All the implementations have been tested in the previous four cases and the computational cost as well as the speed-ups of the parallel versions are highlighted in Table 1 and in Figure 2.

Taking into account both factors, small time-step size and high number of cells, the cost of the simulations for the sequential version of the numerical engine for tests cases 1 2 and 3 is three orders of magnitude larger than the real time. On the other hand, the parallel Multi-Core version, with 4 cores of the same CPU, can accelerate the computation between 2.25 and 2.65 times. The GPU improves the simulation cost between 34.88 and 49.40 times compared with the sequential version. Moreover, if the mesh is reordered, the computational cost is smaller and a speed-up between 49.96 and 59.85 is obtained.

The main reason for discrepancies on the speed-up of cases 1, 2 and 3 against test case 4 is that the number of calculations in the latter is higher compared with the number of calculations for the other cases, i.e. there are more cells that satisfy $h > 0$ in test 4. Therefore, the more cells are involved in the calculus, the larger speed-ups can be obtained with the GPU. This is also described in [13].

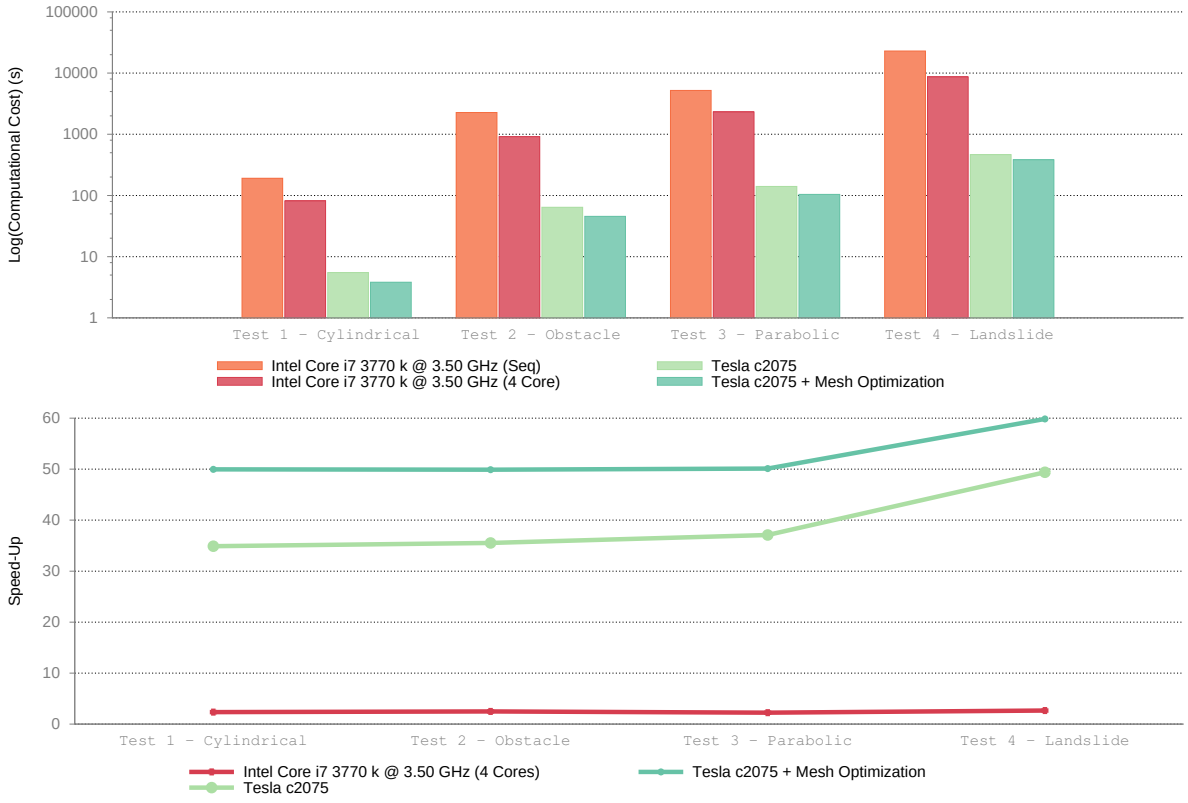

Figure 2: Computational time (upper) using logarithmic scale and speed-up (lower) for the compared implementations

		Seq.	4 Cores		GPU Std.		GPU Opt	
Case	n_{cells}	$t(s)$	$t(s)$	s_{up}	$t(s)$	s_{up}	$t(s)$	s_{up}
Test 1	319354	191.15	81.63	**2.34**	5.48	**34.88**	3.83	**49.96**
Test 2	458684	2274.13	912.94	**2.49**	64.03	**35.52**	45.59	**49.89**
Test 3	670940	5217.70	2319.48	**2.25**	140.68	**37.09**	104.12	**50.11**
Test 4	869149	22929.15	8657.59	**2.65**	464.16	**49.40**	383.13	**59.85**

Table 1: Detail of computational cost and speed-up for the compared implementations using the four cases.

4.1 TOUS DAM BREAK

To test large spatial domains that require a high number of cells for flood warning/hazard prediction the dam failure of Tous dam is proposed [1]. Tous dam is the last flood control structure of the Júcar River basin in the central part of the Mediterranean coast of Spain. During the 20th and the 21st October 1982 a particular meteorological condition led to extremely heavy rainfall. As a result the Júcar River basin suffered flooding all along and the Tous Dam failed with devastating effects downstream. The first affected town was Sumacárcel, about 5 km downstream of Tous Dam, lying at the toe of a hill on the right bank of Júcar river [1]. The terrain is moderately mountainous and most of the buildings lie on a slope that partially protected them from the flood. The ancient part of the village, however, is located closer to the river course and was completely flooded, with high water marks reaching between 6 m and 7 m.

The DTM model used in this work was generated by CEDEX in 1998 [1]. From this information a numerical mesh with $3 \cdot 10^5$ cells has been defined. This computational domain covers most of the original DTM, starting just after the dam location and finishing approximately 1 km downstream of Sumacárcel. The mesh has been refined in the dam area and in the village area (Fig. 3) for providing an adequate resolution for the hydraulic structures and the buildings. It is stressed that the decrease in the cell size leads to an increment in the simulation time since the stability criterion is more restrictive. This is also described in [11].

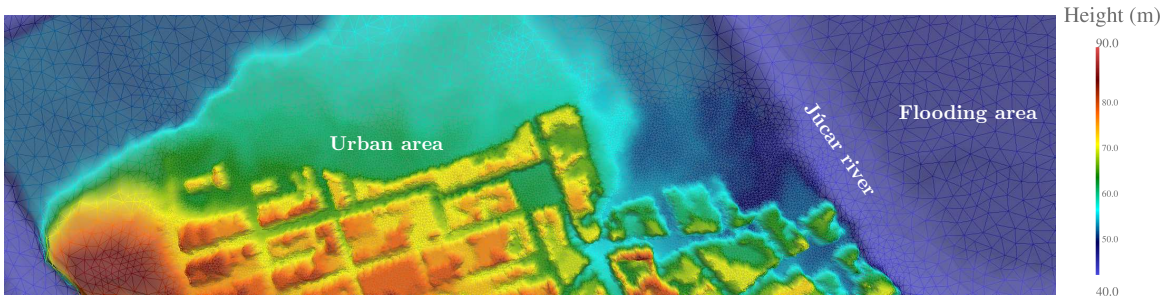

Figure 3: Detail of the simulation mesh at the village area nearby

The cause of the dam break was overtopping/dam-breaching, due to intense rainfall, and its later erosion and collapse. The height of the dam crest was 98.5 m and before reaching this level the discharge facilities of the dam were opened in order to evacuate the huge amount of incoming water. To reproduce this situation, the authors have considered the water elevation records together with the reservoir rating curves for simulating the spillway procedure, i.e. a water discharge of 3568 m^3s^{-1} is considered for obtaining the initial condition. Once the crest level is reached, a dam breach starts and it causes the erosion and collapse process. Hence, an outflow discharge emerging from the dam creates the traveling wave which is the responsible for the flooding event, i.e. it is the key information for the prediction of this event. In previous studies [1], since the morphodynamic change of the dam was not modeled, a tuning synthetic discharge, based on several assumptions, was estimated. Finally, at the outlet boundary, downstream of the domain, the flow was let to exit freely without imposing any conditions, as no information was provided.

On the other hand, following [1], a Manning coefficient of 0.030 $sm^{-1/3}$ has been set for the whole river bed reach and, additionally, an increased roughness coefficient of 0.1 $sm^{-1/3}$ has been defined in two zones close to the village with dense orange trees. The mean sediment

diameter involved in the erosion process has been set to 0.02 m. As the ground in the town area was fully paved with concrete the flood did not erode it. The real time simulated has been 11.1 hours from the beginning of the dam overtopping.

In Fig. 4 the breach evolution of the dam is plotted at several times. The flow overtopping causes the inception of the erosion at the front edge of the dam crest. As the breach increases in size the flow is accelerated and a severe erosion occurs. Consequently, the water discharge in the breach also augments. The earthfill material is grabbed by the flow and it is settled downstream the dam creating a sediment tongue which migrates towards the riverbed. At the end of the event the morphology of the dam area has changed completely and an important fraction of the dam has been completely removed, which is in agreement with the photos taken after the event and provided in [1].

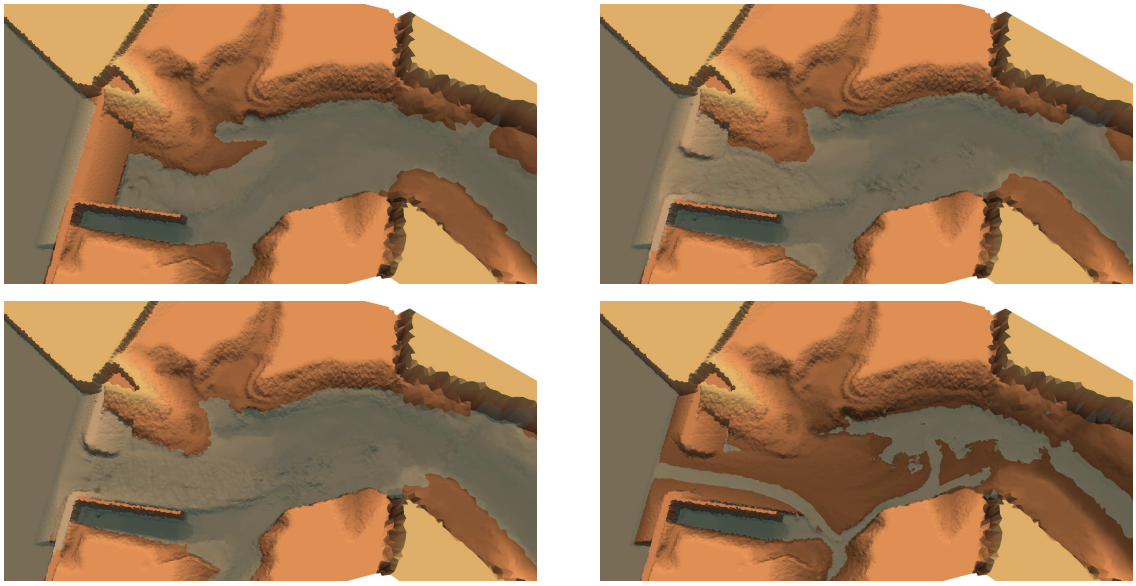

Figure 4: Initial condition (Top-Left) and evolution of the erosion process at t=1.3 hours (Top-Right), t=2.7 hours (Bottom-left) and at final stage (t=11.1 hours) (Bottom-right)

The evolution of the computed flooding can be seen in full plan view in Fig. 5 at times t =0, 1.3, 2.7 and 11.1 hours considering the time t=0 when the water surface level inside the reservoir has reached the dam crest and the overtopping is about to start. The flow advances towards the village filling the riverbed capacity and, consequently, inundating the floodplain areas nearby.

Thanks to the work described in [1], there are field data for the estimation of: (i) the maximum and minimum levels reached by the flood wave or (ii) a unique level for the water surface at different locations within the town, for evaluating the quality of the simulations. This estimation was performed considering a range of values within which it was completely ensured that the water reached that level. The location of the gauging points is shown in Fig. 6. Figure 7 displays the water depth recorded at several locations in Sumacárcel village together with the numerical predictions. There is a good agreement between the field data and the estimated depth, since most of the probes reach the range, between the maximum and minimum, estimated during the event. This agreement is attributed to the adequate simulation of the erosion process at the Tous dam.

It is also important to highlight that, by coupling the hydrodynamic and the breach erosion phenomena, less assumptions are required. This may be relevant in practical applications but is

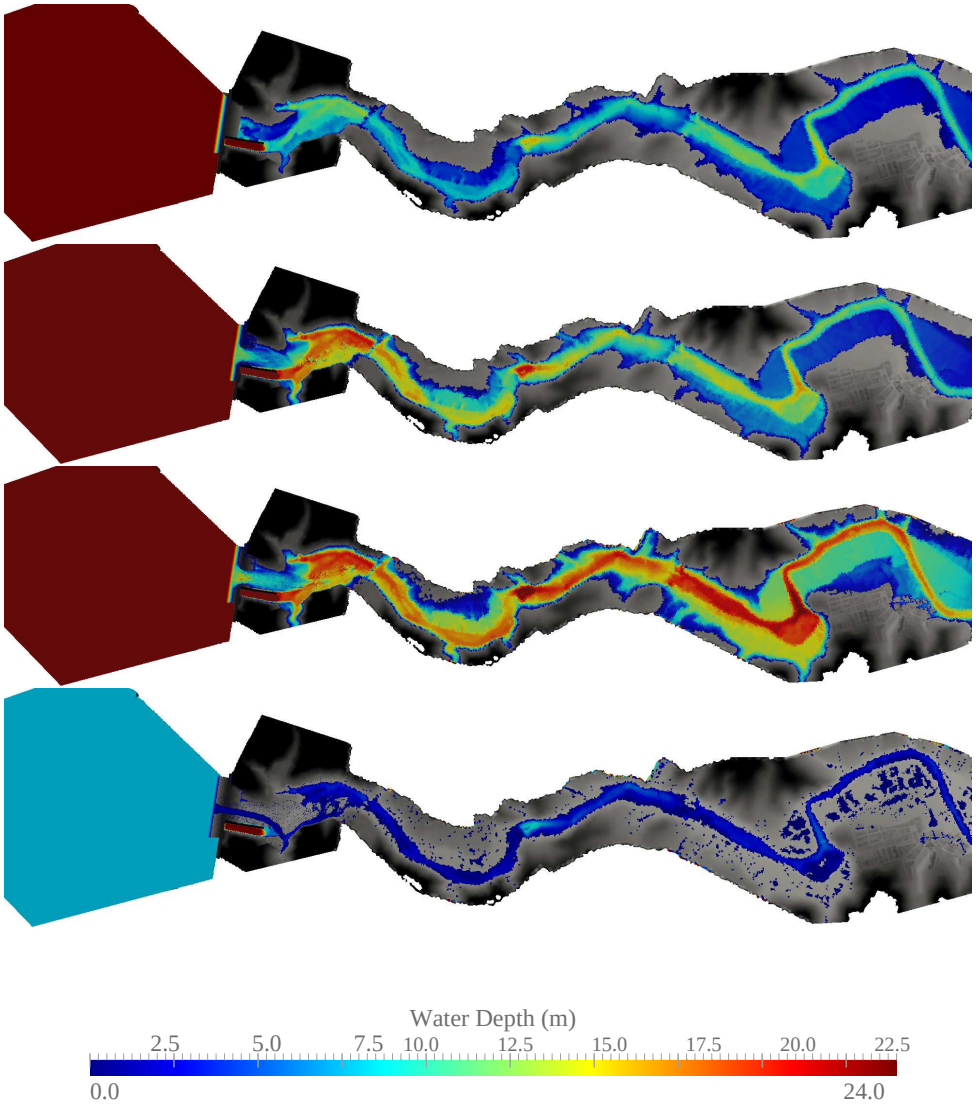

Figure 5: Water depth evolution along the valley at times t=0, 1.3, 2.7, 11.1 hours from top to bottom

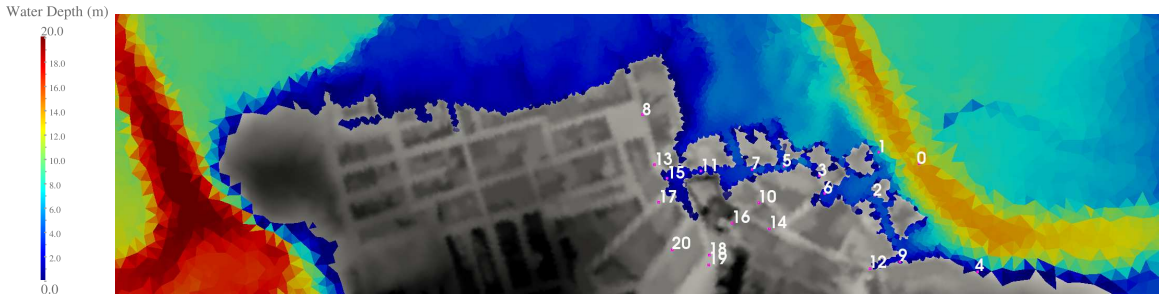

Figure 6: Detail of the location of the gauging points

costly in computational terms. For instance, in [1] a synthetic hydrograph based on a detailed analysis of how the dam failed was proposed. However, thanks to the GPU capabilities it is possible to couple the hydrodynamics and the dam erosion for obtaining directly the hydrograph which is the responsible for the later flooding event. In Fig. 8 both hydrographs, the synthetic

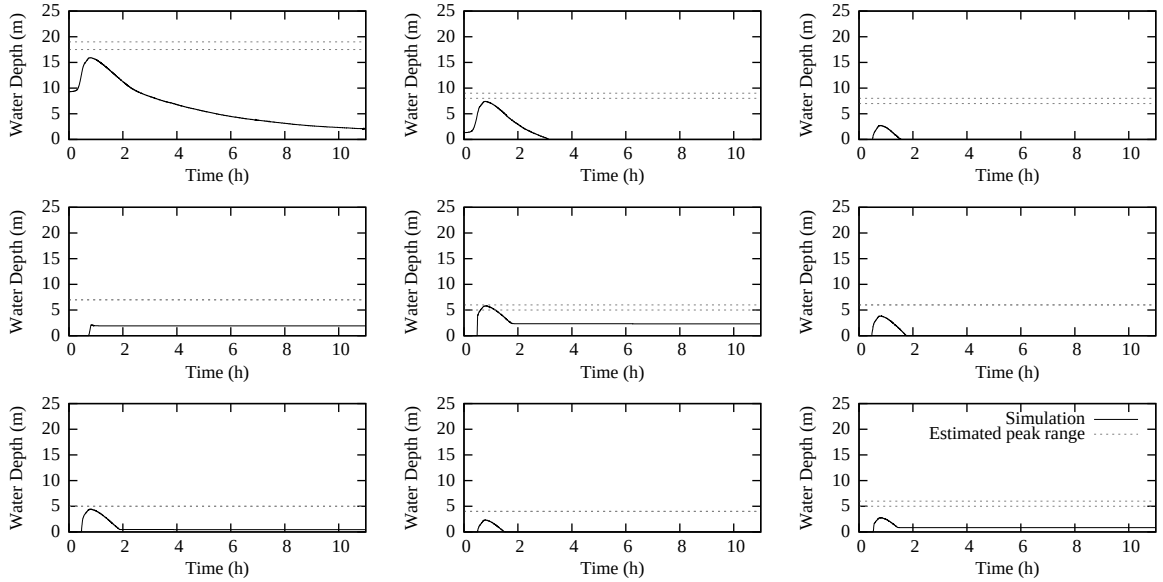

Figure 7: Water depth numerical predictions at several locations in Sumacárcel village and estimated range provided in [1] for gauges 1, 2, 3, 4, 6, 7, 11, 12 and 15 (from top to bottom and from left to right), see Fig. 6

and the computed one in the dam-breach, are plotted. It is remarkable that the peak discharge observed by means of the simulation, Q_{peak}= 14568.09 m^3s^{-1}, is very close to the peak discharge estimated in [1], where Q_{peak}= 15000 m^3s^{-1}. Conversely, the computed discharge is less sustained in time. This difference is probably because the inlet tributaries of the reservoir have been neglected. Since this effect has not been taken into account, in [1] there is not a fair estimation of the magnitude of these inlet tributaries, only the water contained in the reservoir at the beginning of the event is allowed to outflow in the simulation.

The evolution of the dam-breach is also plotted in Fig. 8 using the same cross section used to evaluate the discharge. It can be observed that most of the process has occurred within the first 1500 s, i.e. during the peak discharge. After t=1500 s changes in bed morphology are less violent.

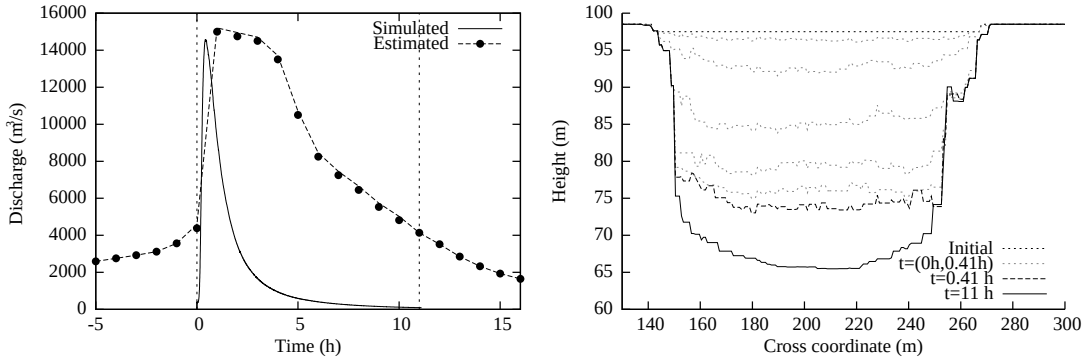

Figure 8: (Left) Comparison of the hydrograph generated due to the dam failure using the presented implementation against the hydrograph estimated in [1]. Simulated window is highlighted considering the time interval between $t = 0$ and $t = 11$ hours. (Right) Evolution of the dam-breach from $t = 0$ to $t = 0.41$ hours (peak discharge) each 0.07 hour and $t = 11.0$ hours (final state)

The execution time is summarized in Table 2. In this case, only the parallel CPU version has been benchmarked due to the huge execution time required for the single-core CPU version.

Table 2: Detail of execution time and speed-up for the compared implementations

4 Cores	GPU	
t	t	S_{up}
207 h 7 min	8h 7 min	25.25

The GPU reduces the simulation effort 25 times compared with the 4-Core version allowing an efficient simulation and accurate prediction. It is important to take into account that, in this case, the improvement has been increased compared against the previous cases where the GPU accelerates the computation of the OpenMP solution in a 20 factor. This effect has been previously reported in hydrodynamic simulation in [12] and it is due to the large number of elements included in the calculation. Thanks to the GPU capabilities it has been affordable to locally refine the mesh in the breach area and provide an adequate design for the initial breach which provides the dam-breaching discharge. Therefore, several possibilities can be addressed in the same day which is a noticeable advance when comparing with the computational effort based on CPU.

5 ACKNOWLEDGEMENTS

This work was partially supported and funded by the Spanish Ministry of Science and Technology under research projects CGL2015-66114-R, and by Diputación General de Aragón, DGA, through FEDER funds.

6 CONCLUSIONS

- The new opportunities given by the GPU implementation have been described for the analysis of several situations where the morphodynamic effects are relevant.
- For this purpose, the shallow water equations in combination with the Exner equation have been discretized in Finite Volumes and the numerical schemes implemented to run on a GPU card.
- This model allows to properly represent the propagation of bed and surface waves over realistic bathymetries in affordable computation time even when considering large domains and retaining a high level of accuracy.
- Unstructured meshes have been considered, since this grid topology is the only one which avoid misleading preferential flow directions.
- The computational times have been compared with those obtained when considering Single-Core and Multi-Core processors. The GPU implementation and specially, the application of cell and wall ordering algorithms, have driven to obtain noticeable improvements in the speed-up of the test cases. The GPU implementation provides a peak speedup of 50.
- This saving of time allows to address large-number-of-cells, large-time and large-space scenarios, strengthening preventive measures and enhancing response capacities.
- This opens the possibility of facing the sediment transport analysis in a particular location for several years or the geomorphological changes in domains of a regional-size.

REFERENCES

[1] F. Alcrudo and J. Mulet. Description of the Tous dam break case study (Spain). *Journal of Hydraulic Research*, 45(Extra Issue):45–57,2007.

[2] S. Borkar. Thousand core chips: A technology perspective. In *Proceedings of the 44th Annual Design Automation Conference*, DAC '07, pages 746–749, New York, NY, USA, 2007. ACM.

[3] F. Bouchut, E. D. Fernández-Nieto, A. Mangeney, and P. Y. Lagrée. On new erosion models of Savage-Hutter type for avalanches. *Acta Mechanica*, 199(1-4):181–208, 2008.

[4] R.G. Dreslinski, M. Wieckowski, D. Blaauw, D Sylvester, and T. Mudge. Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits. *Proceedings of the IEEE*, 98(2):253–266, Feb 2010.

[5] J. García-Ruiz, J. Arnaez, S. Beguería, M. Seeger, C. Marti-Bono, D. Regues, N. Lana-Renault, and S. White. Runoff generation in an intensively disturbed, abandoned farmland catchment, Central Spanish Pyrenees. *Catena*, 59:79–92, 2005.

[6] Peter N Glaskowsky. Nvidia's fermi: The first complete GPU computing architecture. *A white paper prepared under contract with NVIDIA Corporation*, (September):126, 2009.

[7] J.M.N.T. Gray, K. Wieland, and K. Hutter. Gravity-driven free surface flow of granular avalanches over complex basal topography. *Proc. Royal Soc. London*, Ser. A, 455:1841, 1999.

[8] R.M. Iverson and R.P. Denlinger. Flow of variably fluidized granular masses across three-dimensional terrain. A Coulomb mixture theory. *Journal of Geophysical Research*, 106(B1):537–552, 2001.

[9] C. Juez, J. Murillo, and P. García-Navarro. 2D simulation of granular flow over irregular steep slopes using global and local coordinates. *Journal of Computational Physics*, 255:166–204, 2013.

[10] C. Juez, J. Murillo and P. García-Navarro. A 2D weakly-coupled and efficient numerical model for transient shallow flow and movable bed. *Advances in Water Resources*, 71:93–109,2014.

[11] C. Juez, A. Lacasta, J. Murillo and P. García-Navarro. An efficient GPU implementation for a faster simulation of unsteady bed-load transport. *Journal of Hydraulic Research*, 1–14,2016.

[12] A. Lacasta, M. Morales-Hernández, J. Murillo, and P. García-Navarro. An optimized gpu implementation of a 2d free surface simulation model on unstructured meshes. *Advances in Engineering Software*, 78:1–15, 2014.

[13] A. Lacasta, C. Juez, J. Murillo, and P. García-Navarro. An efficient solution for hazardous geophysical flows simulation using GPUs. *Computers & Geosciences* 78:63–72, 2015.

[14] E. Lajeunesse, A. Mangeney-Castelnau, and J. P. Villote. Spreading of a granular mass on a horizontal plane. *Physics of Fluids*, 16:2371–2381, 2004.

[15] A. Mangeney, O. Roche, O. Hungr, N. Mangold, G. Faccanoni, and A. Lucas. Erosion and mobility in granular collapse over sloping beds. *Journal of Geophysical Research*, 115:F03040, 2010.

[16] G.E. Moore. No exponential is forever: but "forever" can be delayed! [semiconductor industry]. In *Solid-State Circuits Conference, 2003. Digest of Technical Papers. ISSCC. 2003 IEEE International*, 1:20–23, 2003.

[17] L. Moretti, A. Mangeney, Y. Capdeville, E. Stutzmann, C. Huggel, D. Schneider, and F. Bouchut. Numerical modeling of the Mount Steller landslide flow history and of the generated long period seismic waves. *Geophysical Research Letters*, 39:L16402, 2012.

[18] J. Murillo and P. García-Navarro. Weak solutions for partial differential equations with source terms: Application to the shallow water equations. *Journal of Computational Physics*, 229:4327–4368, 2010.

[19] J. Murillo and P. García-Navarro. Wave Riemann description of friction terms in unsteady shallow flows: Application to water and mud/debris floods. *Journal of Computational Physics*, 231:1963–2001, 2012.

[20] NVIDIA. *NVIDIA CUDA C Programming Guide*, Jun 2011.

[21] M. Pirulli, M. Bristeau, A. Mangeney-Castelnau, and C. Scavia. The effect of the earth pressure coefficients on the runout of granular material. *Environmental Modelling and Software*, 22:1437–1454, 2007.

[22] M. Pirulli and A. Mangeney. Results of Back-Analysis of the Propagation of Rock Avalanches as a Function of the Assumed Rheology. *Rock Mechanics and Rock Engineering*, 41:59–84, 2008.

[23] O. Pouliquen and Y. Forterre. Friction law for dense granular flows: application to the motion of a mass down a rough inclined plane. *Journal of Fluid Mechanics*, 453:133–151, 2002.

[24] S. Sharma and K. Gupta. Parallel performance of numerical algorithms on multi-core system using openmp. In Natarajan Meghanathan, Dhinaharan Nagamalai, and Nabendu Chaki, editors, *Advances in Computing and Information Technology*, volume 177 of *Advances in Intelligent Systems and Computing*, pages 279–288. Springer Berlin Heidelberg, 2013.