Development of a GIS-GPU-based flood simulation model and its application to flood-risk assessment

Augusto Maidana, Julio García, Eugenio Oñate, Claudio Zinggerling and Miguel A. Celigueta

International Center for Numerical Methods in Engineering (CIMNE)
Technical University of Catalunya (UPC)
Gran Capitán s/n, 08034 Barcelona, Spain
maidana@cimne.upc.edu; julio@compassis.com; onate@cimne.upc.edu;
czingger@cimne.upc.edu; maceli@cimne.upc.edu, www.cimne.com

Summary
The aim of this paper is to present an efficient implementation of a high-resolution explicit scheme for the shallow water equations (SWE) based on a GPU (Graphics Processing Unit) solver which works with raster GIS (Geographic Information System) images. A number of applications of the numerical scheme to flood-risk assessment are presented.

Introduction

Perhaps due to climate change, natural disasters such as floodings induced by tsunamis, typhoons and heavy rains are occurring with more frequency around the world. Taking this into account, the Naval and Maritime Engineering research group of CIMNE has been developing in the last two years a new 2D free surface flow model to simulate the dynamics of gravitational surface waves to solve problems ranging from flood wave propagation to the propagation of a tsunami across the continental shelf.

In practical terms, a large spatial domain with high resolution is required so that source and run-up areas are adequately resolved. This requires using large scale discretization of the analysis domain that must be resolved in a quasi-real time manner so that the results are useful for practical purposes. For this reason, in this work the numerical scheme has been adapted to work on powerful GPU-based parallel computers. Additional ingredients of the new computational model are the efficient use of high-resolution images emanating from digital elevation models (DEMs) and the integration of Geographic Information System (GIS) data and software. The solution provided in this work makes direct use of the rasterized image of the terrain as provided by a GIS where the image pixels define the discretization of the analysis domain. This approach avoids the time consuming task of mesh generation and yields reduced computation times, increases data handling and analysis capability and simplifies the post-processing of the results.

This paper is organized as follows: in Section 1 the numerical model and the discretization scheme selected are described. In Section 2 the results of the validation and verification process are presented. Finally, in Section 3 the conclusions of the work are summarized.

1 Numerical Model

It is generally agreed that free surface effects of tsunamis can be described by depth-averaged equations as the Shallow Water Equations (SWE). Interest in these equations has arisen because comparisons with both large-scale laboratory data and field data have demonstrated a remarkable and perhaps surprising capability to accurately model complex evolution phenomena, such as the maximum run-up and inundation, over wide ranges of tsunami waves. The present work considers a 2D depth-averaged flow in the Cartesian coordinate system (x, y). The nonlinear shallow-water equations (NSWE) which include a continuity equation and two momentum equations in the x and y directions, define the fluid motion as a function of time t. The conservative form of the NSWE is written as

(1) $$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}=\mathbf{H}^1+\mathbf{H}^2$$

where $\mathbf{U}$ is the vector of conserved variables; $\mathbf{F}$ and $\mathbf{G}$ are the flux vectors; and $\mathbf{S}$ is the source term. These vectors are expressed in terms of the water depth h and the flow velocity (u, v) as

(2) $$\mathbf{U}=\begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \mathbf{F}=\begin{bmatrix} hu \\ hu^2+\frac{1}{2}gh^2 \\ huv \end{bmatrix}, \mathbf{G}=\begin{bmatrix} hv \\ huv \\ hv^2+\frac{1}{2}gh^2 \end{bmatrix}$$

in which g is the gravitational acceleration constant and hu and hv are the discharges along the x and y directions, respectively. Furthermore, the source terms in the equation (1), the bed slope and bottom shear stress terms can be written as

(3) $$\mathbf{H}^1=\begin{bmatrix} 0 \\ -ghS_{ox} \\ -ghS_{oy} \end{bmatrix}, \mathbf{H}^2=\begin{bmatrix} 0 \\ ghS_{fx} \\ ghS_{fy} \end{bmatrix}$$

where S_0 is the bottom topography measured from given data and S_f is the bottom friction term. The Manning's coefficient n for the surface roughness is employed in the following relations:

(4) $$S_{fx}=\frac{n^2u\sqrt{u^2+v^2}}{h^{\frac{4}{3}}} \quad ; \quad S_{fy}=\frac{n^2v\sqrt{u^2+v^2}}{h^{\frac{4}{3}}}$$

The set of equations (1) are solved by applying a Riemann solver [1] to each side of the finite volume cells discretizing the analysis domain. Following the procedure proposed by Toro [2] and treating the bottom slope of the source term according to Vázquez-Cendón [3], equations (1) are solved in time with a first order explicit scheme over homogeneous quadrilateral cells as:

(5) $$\begin{pmatrix} \mathrm{h} \\ \mathrm{hu} \\ \mathrm{hv} \end{pmatrix}_I^{n+1}=\begin{pmatrix} \mathrm{h} \\ \mathrm{hu} \\ \mathrm{hv} \end{pmatrix}_I^{n}-\frac{\Delta t}{2S_I}\begin{pmatrix} \mathrm{dh} \\ \mathrm{dhu} \\ \mathrm{dhv} \end{pmatrix}_I^{n}-\Delta t\begin{pmatrix} 0 \\ \mathrm{ghS}_{fx} \\ \mathrm{ghS}_{fy} \end{pmatrix}_I^{n}$$

where the upper index n denotes the time step number and the lower index I denotes the cell number or pixel.

The selected FV scheme is well balanced and supports wet/dry states [4]. The major advantage of employing this scheme is that it is particularly suitable for parallelization on graphics cards (GPUs) [5-9]. The "CPU + GPU" combination is a powerful computation tool; thus, while the CPU has several cores optimized for sequential computation, the GPU includes thousands of smaller cores designed for efficient parallel computation. The parts of the code that are executed in sequential mode are held on the CPU, while those that run in parallel are carried out on the GPU. From the perspective of the user, this scheme yields a better application performance that allows performing quasi real-time simulations. The simplicity of the model is largely compensated by the higher resolution of the raster image that can be employed.

A tool of this type will have a major impact on the simulation, management and assessment of flood-risk situations due to its clear advantages in the managing of the analysis data and the computational speed. We note that the implementation in GPUs, besides getting the parallel execution of the computer application, makes efficient use of device memory allowing real-time simulations of large scale problems.

2 Validations

In the paper we present validation results of the new algorithm for predicting different flooding situations. Standards and guidelines for validation and verification of the code have been used to avoid producing mathematically correct but physically unrealistic solutions of the inundation mapping. Following the standards with three types of validation data have been used for the validation work: analytical solutions, laboratory measurements and field measurements. In this paper, three validation cases will be presented where the first two types of validation data will be used: analytical solutions and laboratory measurements [10,11].
As a representative example, Fig. 1 depicts the flow base employed to estimate the celerity of a flooding wave gliding over the Ebro river in Spain. Fig. 2 shows the wave hydrograph between two gauging stations and the processes of translation and attenuation of the wave.

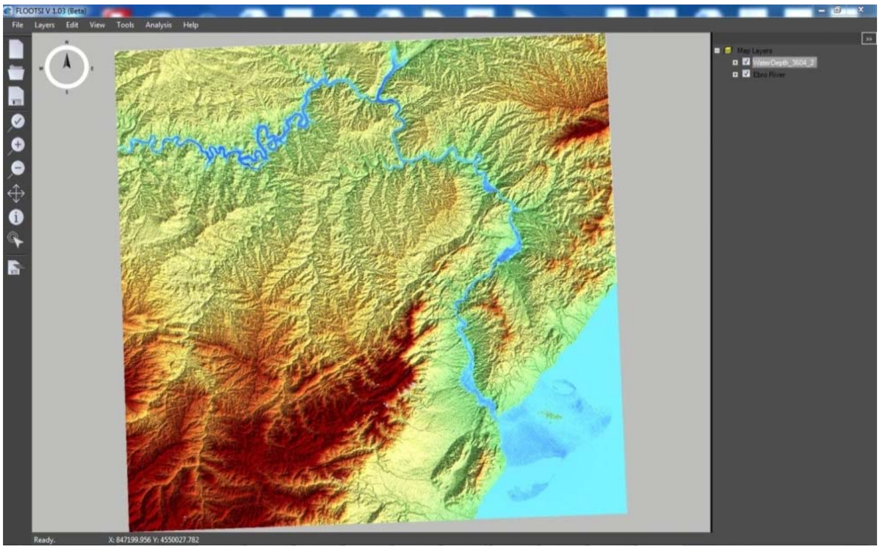

Figure 1: Flow base employed for analysis of the propagation of a flooding wave

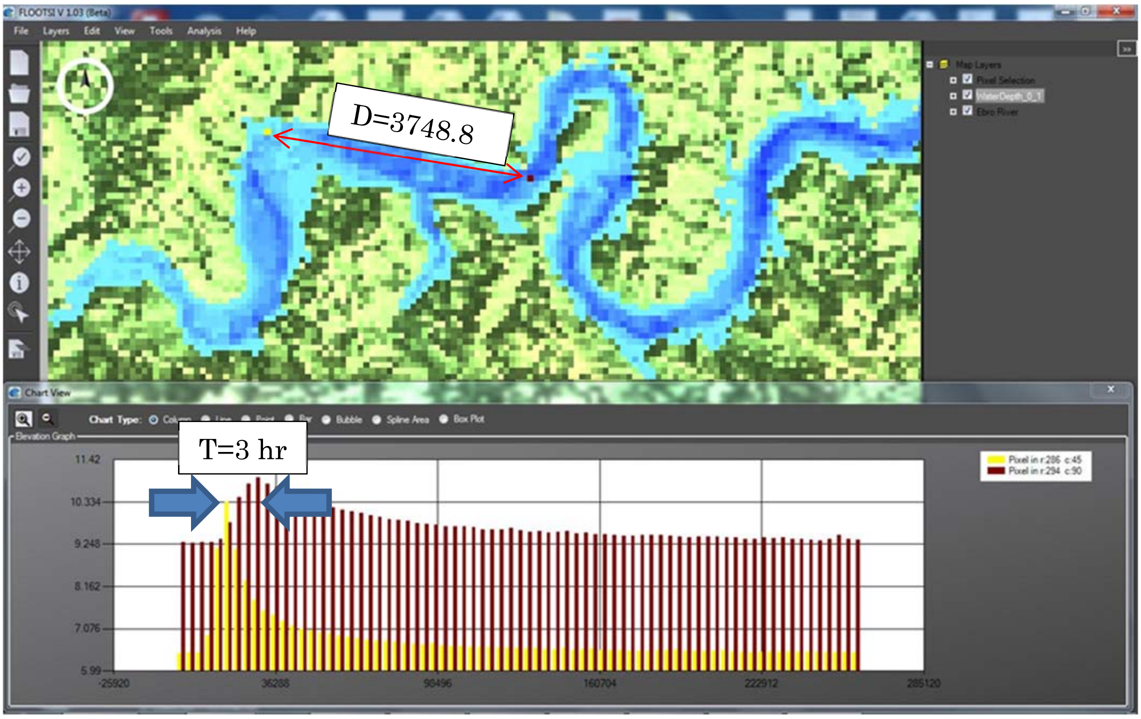

Figure 2: Hydrograph of the flooding wave

Conclusions

The method presented offers a powerful computational capability at low cost due to the use of graphics cards (GPU) and the seamless integration of GIS data where the terrain image pixels are taken as the basis for defining the discretization mesh. Computing in GPU ensures an excellent computer performance due to the highly parallel structure of the GPU. Numerical results have consistently shown that GPUs are more effective than general-purpose CPUs for algorithms such as the one presented in this work where processing of large blocks of data is done in parallel.

The developed model provides a highly versatile easy-to-use graphical user interface. This interface is used to manipulate images, prepare and launch simulations, giving also the opportunity to display and analyze in simple and efficient way the simulation outputs in the same tool, making it possible to simulate flood scenarios rapidly and accurately. The method presented will allow performing risk assessment and management of emergency measures due to flooding situations on low cost PC machines equipped with GPUs. This will enable risk consultants to design countermeasures based on realistic flood-disaster scenarios.

We conclude that the software is suitable to effectively assess flood risk in a quasi-real time manner. The accuracy of the simulation depends on the image resolution, i.e. the total number of pixels in the image. The implementation was verified and validated and showed that the use of efficient single precision hardware is sufficiently accurate for real-world simulations.

Acknowledgments: This work was partially supported by the SAFECON Advanced Grant project of the European Research Council. The support and advice of the members of the Flumen Institute of the Technical University of Catalonia (UPC) is acknowledged.

References

[1] P.L. Roe: Approximate Riemann solvers, parameter vectors, and difference schemes, *Journal of Computational Physics*, Vol. 43, pp. 357–372, 1981.

[2] E.F. Toro: Riemann solvers and numerical methods for fluid dynamics: a practical introduction, Springer Verlag, 2nd Edition, 1999.

[3] M.E. Vázquez-Cendón: Improved treatment of source terms in upwind schemes for the shallow waters equations in channels with irregular geometry, *Journal of Computational Physics*, Vol. 148, pp. 497–526, 1999.

[4] E. Bladé and M. Gómez: Modelación del flujo en lámina libre sobre cauces naturales. Análisis integrado en una y dos dimensiones, Monografía CIMNE Nº 97. Barcelona, 2006.

[5] J. Sanders and E. Kandrot: CUDA by Example: An introduction to general-purpose GPU Programming, Addison-Wesley, 2011.

[6] R.Farber: CUDA application design and development, Morgan Kaufmann, Elsevier, 2011.

[7] A.R. Brodtkorb, M.L. Sætra and M. Altinakar: Efficient shallow water simulations on GPUs: Implementation, visualization, verification, and validation, *Computers & Fluids*, Vol. 55, pp. 1–12, 2012.

[8] NVIDIA CUDA C Best Practices Guide, DG-05603-001v5.0 NVIDIA, May 2012.

[9] CUDA API reference manual, v5.0 NVIDIA, July 2012.

[10] O. Delestre, C. Lucas, P.-A. Ksinant, F. Darboux, C. Laguerre, T.N.T. Vo, F. James and S. Cordier, SWASHES: a compilation of Shallow Water Analytic Solutions for Hydraulic and Environmental Studies, International Journal for Numerical Methods in Fluids, Vol. 72(3): 269-300, 2013.

[11] C.E. Synolakis, E.N. Bernard, V.V. Titov, U. Kânoğlu and F.I. González: Standards, criteria, and procedures for NOAA evaluation of tsunami numerical models, NOAA Technical Memorandum OAR PMEL-135.