

MESH CREATION FOR REALISTIC TERRAIN CASES FOR SHALLOWFOAM - 2D OPENFOAM SOLVER

Hao Zeng, Luka Grbčić, Ivana Lučin & Lado Kranjčević



This Publication has to be referred as: Zeng, H[ao]; Grbcic, L[uka]; Lucin, I[vana] & Kranjcevic, L[ado] (2018). Mesh Creation for Realistic Terrain Cases for Shallowfoam - 2D OpenFOAM Solver, Proceedings of the 29th DAAAM International Symposium, pp.1065-1070, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-20-4, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/29th.daaam.proceedings.152

Abstract

shallowFoam is a 2D solver under the Open Source CFD framework - OpenFOAM. It solves Shallow Water Equations (SWE). A pseudo-2D mesh is used in OpenFOAM for 2D solvers, which means in the top view the shape of the 2D mesh can be arbitrarily, while in the side view, the vertical direction and all the cells have a constant uniform height. In this way, the bottom elevation information is not stored in the mesh but in the parameter field. How to set up a realistic terrain for shallowFoam cases is therefore an issue to be tackled. In this work, we present a way to create a 2D mesh with the elevation data stored in the parameter field using surface interpolation of real terrain data. The method is validated on the flooding simulation of the Malpasset dam break case, which is a good example of a problem with realistic terrain.

Keywords: mesh creation; 2D solver; OpenFOAM; radial basis function; flooding simulation

1. Introduction

The behaviour of a fluid in shallow water areas is governed by SWE, which has been obtained from the Navier Stokes equations by imposing several simplifying assumptions. These equations are widely used in large-scale open channel simulations and the results are turned out to be very close to reality. Several of CFD 2D solvers have been implemented for solving Shallow Water Equations, e.g. MIKE FLOOD 1D-2D [1], TUFLOW [2] and most implementations have parallel computing support like in [3].

Chair of Hydromechanics at Technical University of Munich develops the 2D solver shallowFoam under OpenFOAM environment [4]. It solves the transport equation for flow depth (1) and the momentum equation (2), where h is the flow depth, q_i is the specific discharge $h\bar{u}_i$, \bar{u}_i is the depth-averaged velocity, z_b is the bottom level, τ_{bi} is the bottom stresses via Manning-Strickler, and ν_{eff} is the effect viscosity via depth-averaged parabolic eddy viscosity model.

$$\frac{\partial h}{\partial t} + \frac{\partial q_i}{\partial x_i} = 0 \quad (1)$$

$$\frac{\partial q_i}{\partial t} + \overline{u_j} \frac{\partial q_i}{\partial x_j} = -gh \frac{\partial(z_b+h)}{\partial x_i} - \frac{\tau_{bi}}{\rho} + \frac{\partial}{\partial x_j} \left[v_{eff} \left(\frac{\partial q_i}{\partial x_j} + \frac{\partial q_j}{\partial x_i} \right) \right] \quad (2)$$

OpenFOAM, an Open Source software under the GNU General Public License, is a toolbox for CFD simulations including pre-processing, problem solving and post-processing. It can work on an unstructured mesh and the implementation is based on the cell centred FVM, which means that all information is stored at the cell centres and the cells, do not overlap.

The 2D mesh in OpenFOAM is a pseudo 2D mesh. The mesh only follows the geometry in the horizontal dimensions (x- and y- directions) and is always flat with a constant thickness in the vertical dimension (z-direction). Most of the numerical simulations read the geometry information through the mesh, such as the dimension of the simulation domain, the objects inside the domain and the bottom elevation. Unfortunately, no information in z-direction is available in the OpenFOAM 2D mesh. For this reason, another parameter S is introduced in shallowFoam to represent z_b in (2). Consequently, five simulation fields depict the SWE in shallowFoam: H, HU, S, kst, nut stand for flow depth, specific discharge, bottom elevation, Strickler value, turbulent viscosity respectively.

We can easily create a 2D mesh for a regular bottom (e.g. uniform slope bottom) and assign the bottom level value to each cell centre by using mesh manipulation tools in OpenFOAM (e.g. setField, funkySetField [5]). We can also use other mesh generators to produce the 2D mesh and convert it to the OpenFOAM format mesh. However, OpenFOAM mesh has its own mesh data structure that one can hardly distribute the bottom level data to the corresponding cell centre with the existing OpenFOAM mesh manipulation tools. Mesh creation and bottom data extraction of a realistic terrain for shallowFoam cases is our purpose in this work. Accurate 2D SWE simulations are of great importance since they could be applied to flooding problems. Good results provide useful information to flood crisis management systems like in [6], [7].

2. Materials and methods

2.1. Mesh creation

The default mesh generator in OpenFOAM is blockMesh. It reads a dictionary file of mesh, decomposes the given dimension of the domain geometry into a set of one or more three dimensional, hexahedral blocks and writes out the mesh data to points, faces, cells and boundary files [5]. For instance, a 2D mesh is a set of hexahedral cells with only one cell in z-direction.

blockMesh can get a regular rectangular mesh domain effortlessly. In the interest of complex geometry creation, mesh manipulation is possible with another utility in OpenFOAM, snappyHexMesh. The snappyHexMesh utility generates 3D meshes from triangulated surface geometries (i.e. STL or OBJ format file). The mesh approximately adjusts to the object surface by iteratively refining a starting mesh (i.e. the background mesh generated from blockMesh) and shifts the resulting hex or split-hex mesh to the surface. Additionally, snapping the resulting mesh to better fit the surface and adding layers to the boundary are possible as well [5].

Since snappyHexMesh constructs 3D meshes, one more step is needed with aim of creating 2D meshes. extrudeMesh utility is used to extrude the given mesh in a specific direction [5]. Thus, a 3D mesh turns into a pseudo 2D mesh, in another word; only one cell with a fixed height is displayed in z-direction. The flow chart of 2D meshes creation is shown in figure 1.

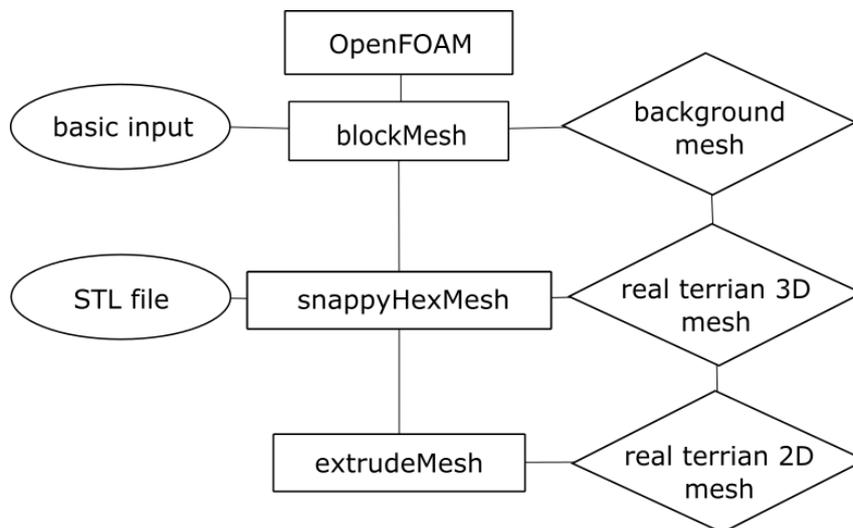


Fig. 1. Flow chart of 2D meshes creation

2.2. Surface interpolation of terrain data

Since the shallowFoam solver is based on the cell centred FVM it is necessary to store data at the cell centre in the mesh. Initial and boundary conditions for flow depth, specific discharge, Strickler value and turbulent viscosity are easily implemented at the cell centres using the dictionaries provided within the OpenFOAM environment. As explained in the previous section, a created 2D mesh is flat and it contains no information regarding bottom elevation and since the shallow water equations require bottom elevation it is necessary to store them at the cell centre for a simulation to work.

For cases that include river flow over a realistic terrain a problem arises since the elevation is usually not uniform and varies a lot throughout the domain. Usually, the data that describes a three-dimensional realistic terrain is initially an unstructured set of $X Y Z$ points which are obtained using three-dimensional aerial laser scanning methods. The unstructured set of points is firstly used to construct a bottom geometry in a STL format using triangulation methods. Besides using them for geometry creation, they are useful for creating interpolation surfaces, which could be used to describe the completely realistic terrain geometry. An interpolated surface could be made based on a set of unstructured points that describe a real terrain. For this case we use radial basis functions to approximate the surface of a real terrain. Creating an interpolated surface allows us to extract the elevation level for every X and Y coordinate of the mesh cell centre. In [8], [9] radial basis functions were used to obtain elevation in realistic surfaces. Radial basis interpolation is defined as:

$$z_e = \sum_{k=1}^n \lambda_k \phi(r_k) + b \quad (3)$$

Equation (3) describes a simple variant of the radial basis interpolation method. The term $\lambda_k \phi(r_k)$ is a weighted function of distance from the unstructured grid point to the input data point and the term b is the bias value (λ_k and b are estimated from the unstructured data points). The term z_e is the estimated value for the surface grid point and in this case it is the elevation. The function of distance or the radial basis function could be defined as a wide variety of different spline functions but in this case, it is defined as a *thin plate* spline function:

$$\phi(r) = r^2 \log(r) \quad (4)$$

Equations (3) and (4) combined form a linear system of equations that is solved to obtain the interpolated surface elevation data points. To apply the method of surface interpolation with radial basis functions, a set of unordered $X Y Z$ points that describe a geometry need to be used to create a surface interpolation function. In order to use the created surface interpolation function that is based on the original geometry, the extracted cell centres X_c and Y_c of the 2D mesh need to be evaluated. The output of this process is the corresponding elevation Z_c for every X_c and Y_c .

3. Results and discussion

Mesh creation method and cell centre points extraction method were tested on a flooding simulation case that has a realistic terrain. The case used was the Malpasset river dam break which was experimentally and numerically studied in [10], [11], [12]. The dimensions of the study domain are 17 500 x 9000 meters. Elevation of the bottom surface ranges from -20 meters above sea level to 100 meters above sea level. The terrain is described as a set of 13 451 measured points. Figure 2. represents the case domain with the location of the dam. The area to the left of the dam is the location of the reservoir. Water level in the reservoir is initially set to 100 meters above sea level.

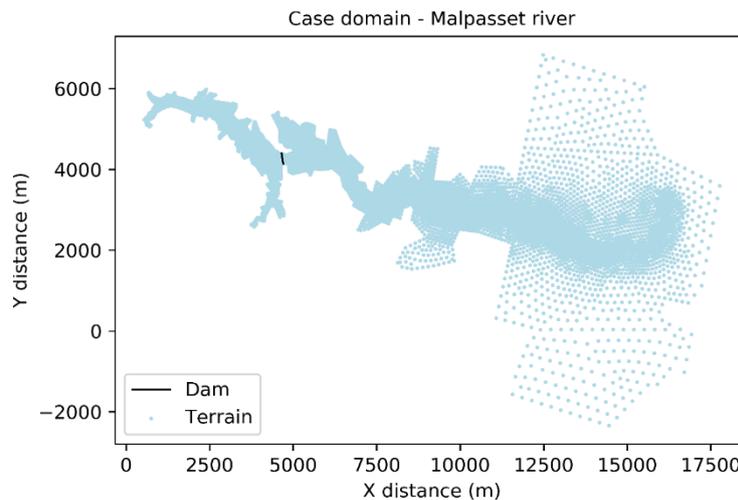


Fig. 2. Malpasset river terrain with dam location

In figure 3 the elevation levels of the original measured data can be seen and in figure 4 the interpolated data based on the cell centres of the 2D mesh.

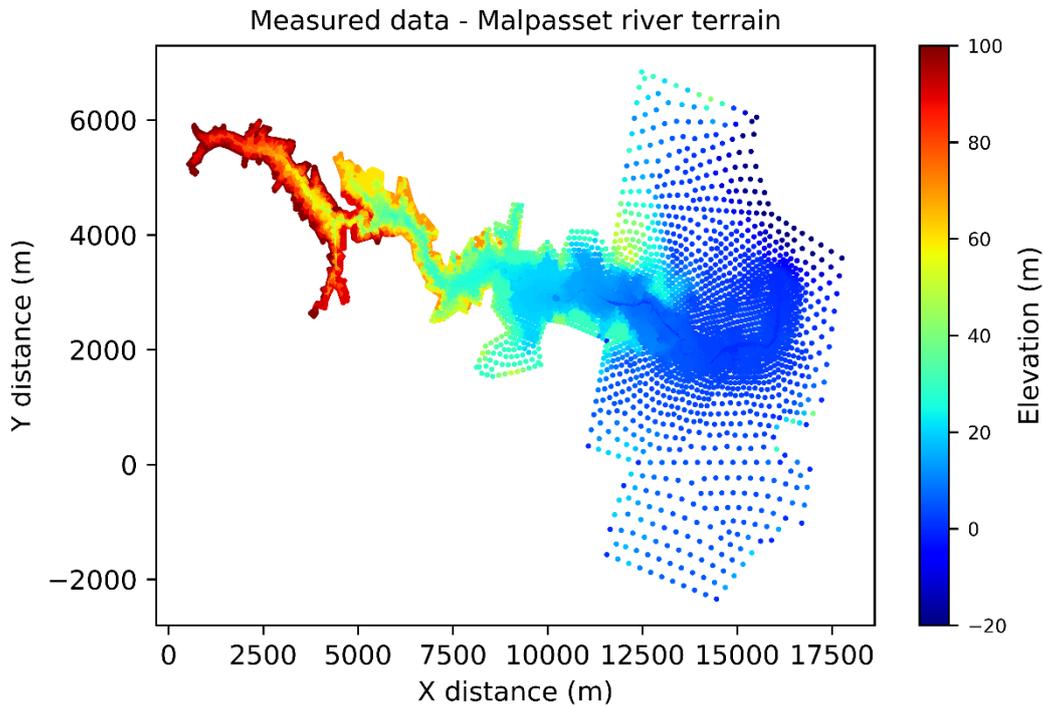


Fig. 3. Malpasset river terrain elevation based on measured data

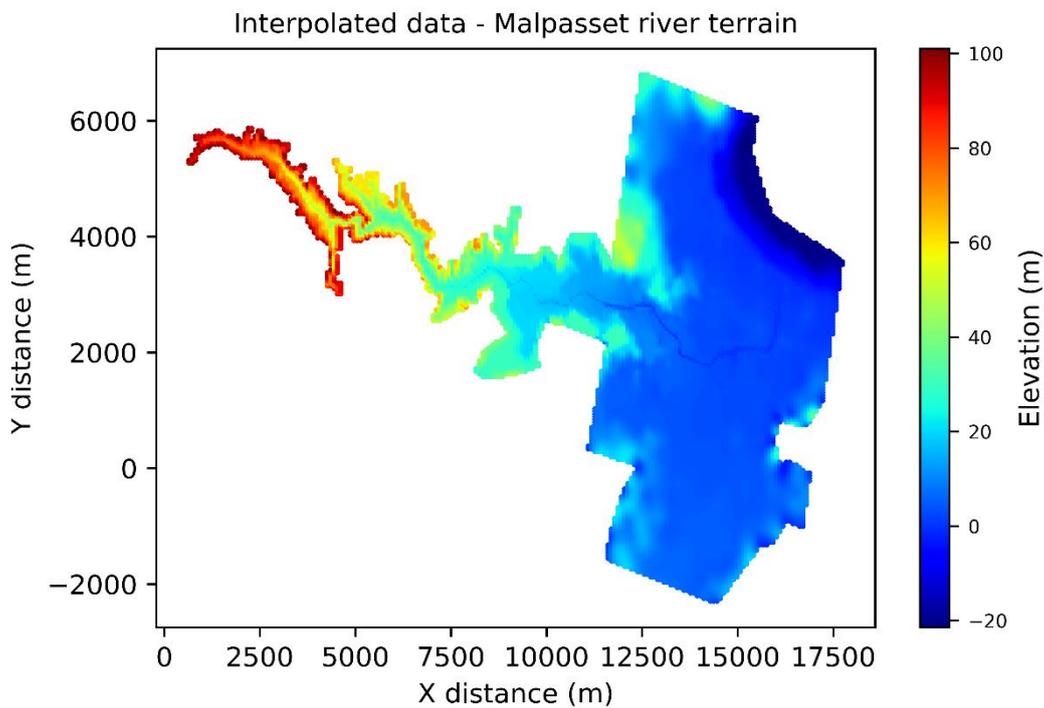


Fig. 4. Malpasset river terrain elevation based on interpolated data

The elevation values seen in figure 4 were used as the elevation levels stored at the cell centres. To test if this process is satisfactory, a simulation was run and the results were compared with the experimental results from [10]. The experimental results were obtained from a laboratory scaled model and were calibrated against real observations. In figure 5 the locations of the gauges can be seen and the results obtained from both experiment and simulation are the maximum observed water surface through the experiment/simulation.

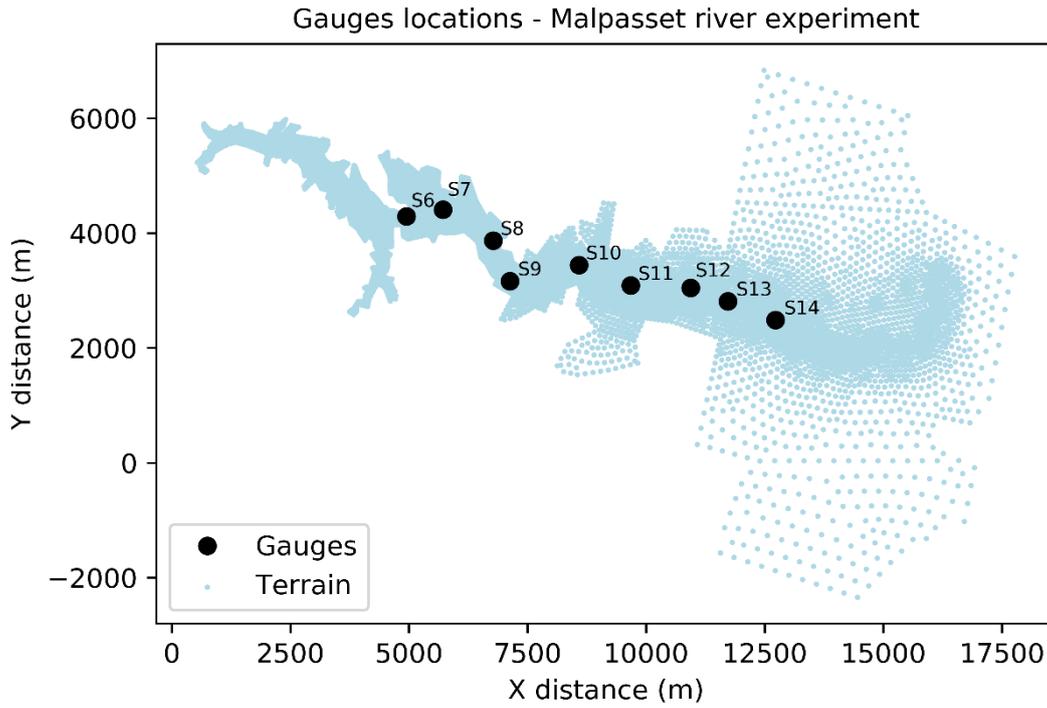


Fig. 5. Locations of the gauges from the physical model experiment

Simulation time was 2000 seconds and the used mesh contained around 80 000 elements. In figure 6 the comparison between the experimental data and the shallowFoam simulation data can be seen and it is clear that the agreement is satisfactory. That means that the shallowFoam 2D solver could be applied to realistic cases with expectation of accurate simulation results.

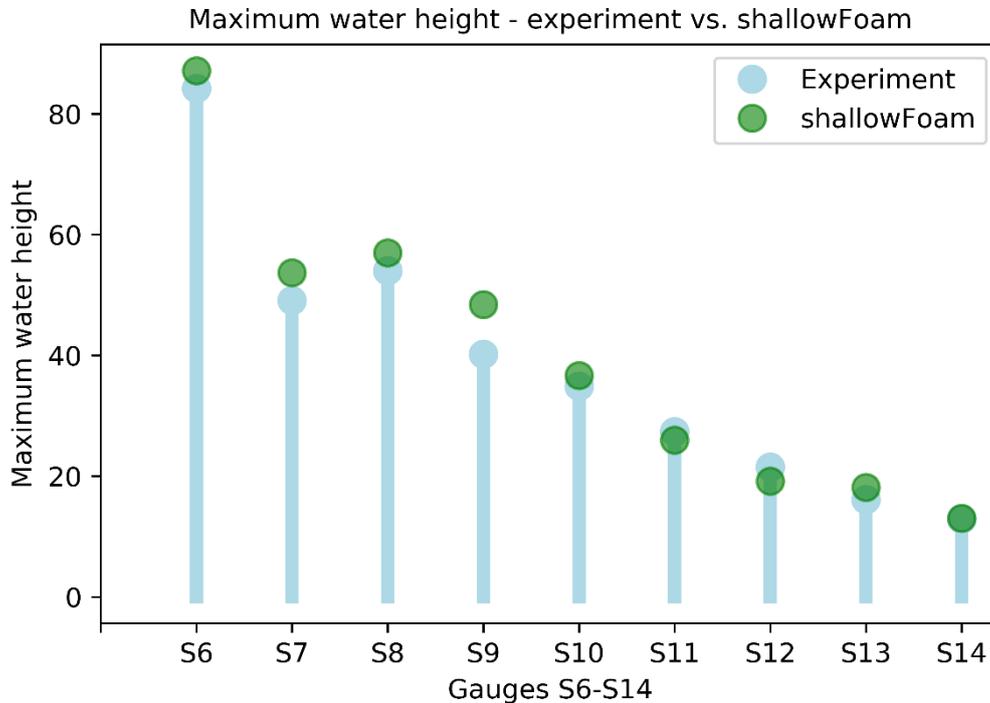


Fig. 6. Comparison of experimental data and shallowFoam

It is worth noting that methodology presented in chapter 2 of assigning interpolated values to cell centres of a 2D mesh enables users to obtain accurate results of realistic flooding simulations while using shallowFoam. Interpolating terrain elevation data is easily implemented but can be quite computationally costly if the number of terrain data points is large. Since shallowFoam is an open-source CFD solver this implementation can be useful for the broader community.

4. Conclusion

Disasters regarding floods are a great problem that is continuously being investigated. To improve civil protection, impacts of dam brakes and flooding must be researched with both experimental and numerical methods. Numerical analysis provides insight in fluid flow phenomena and predictions of flooding damage could provide guidelines for civil and structural protection.

Malpasset dam break case which has realistic terrain data was used for simulation procedure validation. In this paper a method of creating a computational mesh with all needed parameters stored in the cell centres for the OpenFOAM based SWE solver shallowFoam was presented. A 2D mesh was created from available triangulated data (STL) using the OpenFOAM meshing utility snappyHexMesh. Since bottom elevation data is required to solve SWE it was necessary to store it in the cell centres. This procedure was successfully completed using surface interpolation with radial basis functions.

Conducted numerical analysis showed good agreement with provided experimental data, which shows that this methodology can be used in other realistic flooding cases. Other flooding cases based on realistic terrain data should be used to further validate this procedure and point out further improvements in the meshing procedure, surface interpolation and simulation parameters in shallowFoam.

5. Acknowledgement

The authors thank the TUM International Graduate School of Science and Engineering (IGSSE) (project 9.11 - High Performance Interactive Flood Simulation - iFlood) and the German Academic Exchange Service (DAAD) (funding program Graduate School Scholarship Programm 2016, ID number 57437801) for their financial support.

6. References

- [1] DHI, "MIKE FLOOD 1D-2D Modelling User Manual" (2009). DHI Software
- [2] TUFLOW (2009). 'TUFLOW User Manual'
- [3] Kranjcevic, L., Druzeta, S., & Carija, Z. (2005). Parallel solution of implicit numerical scheme for open channel flow equations. *Annals of DAAAM & Proceedings*, 207-209.
- [4] Jud, M., Schwertfirm, F., Rapp, C., & Schilcher, M. (2012). Coupling of GIS and hydraulics – application at dornbirnerach, Austria. In P. Rutschmann, M. Grünzner, & S. Hötzl (Eds.), *Proceedings of the 2nd IAHR Europe congress*. München: Lehrstuhl u. Versuchsanstalt f. Wasserbau u. Wasserwirtschaft d. TU München.
- [5] OpenCFD Ltd. (2009). *OpenFOAM User's Guide (1.6 ed.)* [Computer software manual]. United Kingdom.
- [6] Blistanova, M., Katalinic, B., Kiss, I., & Wessely, E. (2014). Data preparation for logistic modeling of flood crisis management. *Procedia Engineering*, 69, pp.1529-1533.
- [7] Kiss, I., Wessely, E., & Blišťanová, M. (2014). Contribution to logistics of catastrophes in consequence of floods. *Procedia Engineering*, 69, pp.1475-1480.
- [8] Chaplot, V., Darboux, F., Bourennane, H., Leguédou, S., Silvera, N., & Phachomphon, K. (2006). Accuracy of interpolation techniques for the derivation of digital elevation models in relation to landform types and data density. *Geomorphology*, 77(1-2), pp. 126-141.
- [9] Erdogan, S. (2009). A comparison of interpolation methods for producing digital elevation models at the field scale. *Earth surface processes and landforms*, 34(3), pp. 366-376.
- [10] Alcrudo, F., & Gil, E. (1999). The Malpasset dam break case study. *Proceedings of the 4th Concerted Action on Dambreak Modelling Workshop* pp. 95-109.
- [11] Valiani, A., Caleffi, V., & Zanni, A. (2002). Case study: Malpasset dam-break simulation using a two-dimensional finite volume method. *Journal of Hydraulic Engineering*, 128(5), pp. 460-472.
- [12] Hervouet, J. M., & Petitjean, A. (1999). Malpasset dam-break revisited with two-dimensional computations. *Journal of Hydraulic Research*, 37(6), pp. 777-788.