

Adaptive Meshing Methodology Based on Topometric Data for Dambreak Inundation Assessments

by

Mahdi MOSLEMI

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE IN
PARTIAL FULFILLMENT FOR A MASTER'S DEGREE WITH THESIS IN
MECHANICAL ENGINEERING
M.A.Sc.

MONTREAL, 8TH FEBRUARY, 2019

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

© Copyright 2019 reserved by Mahdi Moslemi

© Copyright reserved

It is forbidden to reproduce, save or share the content of this document either in whole or in parts. The reader who wishes to print or save this document on any media must first get the permission of the author.

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Azzeddine Soulaïmani, Thesis Supervisor
Département de génie mécanique à l'École de technologie supérieure

Mr. Georges W. Tchamen, Co-Supervisor
Conseiller Hydraulique, Unité Expertise en Barrages, Hydro-Québec

Mr. Mohammad Jahazi, President of the Board of Examiners
Département de génie mécanique à l'École de technologie supérieure

Mr. Pierre Dupuis, External Evaluator
Spécialiste hydraulicien - bureau de projet de gestion des risques d'inondation – Direction
générale, Communauté métropolitaine de Montréal

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC

18 JANUARY 2019

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

This work has been supported by Hydro-Québec and NSERC (Natural Science and Engineering Research Council of Canada).

I wish to express my gratitude to my supervisor, Prof. Azzeddine Soulaïmani, whose expertise and patience added considerably to my graduate experience. Thanks for your abundant help, and assistance. It was also a great opportunity for me to be connected with Hydro-Québec and Dr. G.W. Tchamen. Thanks for your invaluable support, help, encouragement, and insightful comments.

I have to express my love and gratitude to my beloved family, specially my parents; for their understanding and endless love, and their support through the duration of my studies. Without their love, encouragement, and editing assistance, I would not have finished this thesis. I would also like to thank my sister, Parisa, who helped me in this stage of life with her great experience and provided me valuable inspiration.

Méthodologie de maillage adaptatif basée sur des données topométriques pour les évaluations d'inondation de rupture de barrage

Mahdi MOSLEMI

RÉSUMÉ

Les cartes d'inondation constituent le produit final des études de rupture de barrage exigées par le Règlement pour la Sécurité des Barrages (RSB). La limite d'inondation qui représente l'enveloppe maximale du terrain atteint par l'onde de crue provient généralement du résultat d'un scénario de rupture simulé par un modèle numérique hydraulique. Toutefois, un modèle numérique n'utilise qu'une portion limitée des données topométriques disponibles pour bâtir le modèle de terrain (maillage 2D plus élévation topométrique aux noeuds). Notamment dans les cas où la donnée topométrique a été relevée par le Lidar, cette donnée s'évalue en plusieurs millions de points. Les modèles numériques hydrauliques les plus étendus excèdent rarement des centaines de milliers de noeuds notamment en raison des contraintes informatiques et de temps associés à l'exploitation de ces modèles. La production de la carte d'inondation finale nécessite d'assurer la cohérence entre les niveaux projetés et les élévations pour tous les points de la carte. Cette vérification peut être pénible en raison de la grandeur du domaine ou d'apparition de plusieurs petites vallées secondaires de cours d'eau tributaires qui n'ont pas été représentées par le modèle numérique hydraulique original.

Le présent travail propose une stratégie automatique de remaillage qui utilise l'enveloppe des dimensions maximales atteintes par le modèle original couplé avec les données LIDAR disponibles pour produire un maillage amélioré qui réussira à capturer avec une grande précision les fronts humide ou sec et les débordements dans les vallées secondaires. Ce modèle permet de considérer la pente maximale à l'intérieure de chaque élément basé sur des données réelles, au lieu de contrôler la pente pour ne pas avoir de profondeur négative ou pour contrôler la vitesse. Par ailleurs, l'algorithme est basé sur quelques étapes présentées comme suit : (i) trouver les éléments coupés par l'enveloppe des interfaces humides et sèches, (ii) projeter les points topo-métriques sur les éléments coupés, (iii) si ces points sont très proches de l'interface, s'ils se trouvent dans une vallée ou s'ils sont plus élevés que les éléments coupés correspondants, alors ces points sont ajoutés aux nœuds précédents pour qu'ils soient inclus dans l'étape suivante de triangulation, (iv) finalement, relancer la simulation sur un nouveau maillage. Cet algorithme a été implanté et validé dans un écoulement de rupture de barrage avec une topographie complexe pour les rivières Eastmain et Romaine-Puyjalon au Québec.

Mots Clés: Flux de rupture de barrage, cartographie d'inondation, raffinement automatique de maillage, écoulements d'eau peu profonde, transitions états mouillés-secs.

Adaptive Meshing Methodology Based on Topometric Data for Dambreak Inundation Assessments

Mahdi MOSLEMI

ABSTRACT

Flood maps are the final products of dam failure studies that are required by dam safety regulations. A flood limit, which represents the maximum envelope reached by flood waves, is generally the result of a dam-break scenario simulated by a hydraulic numerical model. However, the numerical model uses only a limited portion of the available bathymetry data to build the terrain model (2D mesh plus topometric elevation at nodes). This is particularly so in the cases where the topo-metric data recorded by LIDAR was estimated in several million points. But the hydraulic numerical models rarely exceed hundreds of thousands of nodes, in particular because of the computer constraints and time associated with the operation of these models. The production of the final flood map requires consistency between projected levels and elevations for all points on the map. This verification may be tedious for a large area with several small secondary valleys of tributary streams that have not been represented by the original hydraulic numerical model. The aim of this work is to propose an automatic re-meshing strategy that uses the envelope of the maximum dimensions reached by the original model coupled with the available LIDAR data to produce an improved mesh that can accurately capture the wet/dry fronts and the overflows of the secondary valleys. This model helps us to consider the maximum slope inside each element on the basis of the real data, instead of controlling the slope for not having negative depth or controlling the velocity. The algorithm is based on a few basic steps: (i) find the elements cut by the envelope of the wet/dry interfaces; (ii) project the topometric points onto the cut elements; (iii) if these points are very close to the interface, if they are found in a valley, or if they are more elevated than the corresponding cut elements, then these points will be added to the previous nodes and included in a subsequent triangulation step; and (iv) re-run the simulation on the new mesh. This algorithm has been implemented and validated in the study of a dambreak flow with a complex river topography on the Eastmain River and the Romaine-Puyjalon River.

Keywords: dambreak flows, flood mapping, automatic mesh refinement, shallow water flows, wet-dry transitions.

TABLE OF CONTENTS

INTRODUCTION.....	1
CHAPTER 1 LITERATURE REVIEW	7
1.1 Shallow Water Equations (SWE)	7
1.1.1 Eulerian Form.....	7
1.1.2 Free Surface Flow Equations	8
1.2 Derivation of SWE	8
1.3 Conservative variables in SWE.....	13
1.4 Shallow Water Wave Equation (SWWE).....	14
1.4.1 Progressive wave.....	14
1.4.2 Standing Waves.....	15
1.5 Public domain software used to implement the TMA method.....	16
1.6 Numerical difficulty in simulating Shallow Water Equations	17
1.6.1 Discontinuous Galerkin Method	17
1.6.2 Positivity-preserving Method.....	18
1.7 Review of adaptive meshing methodology.....	19
CHAPTER 2 METHODOLOGY FOR DEVELOPING THE TOPO MESH ADAPTOR METHOD	27
2.1 Generating mesh with Blue Kenue.....	27
2.2 Boundary Conditions	27
2.3 Topo Mesh Adaptor (TMA).....	28
2.4 Simulation over dry beds	29
2.5 Cut-elements.....	31
2.6 Interpolation	32
2.6.1 Linear Interpolation.....	32
2.6.2 Inverse Distance Weight Function Interpolation (IDW)	32
2.6.3 Nearest Neighbour Interpolation.....	33
2.6.4 Linear 2D Interpolation using T3 elements	34
2.7 Avoiding Creation of Elements That Are Too Small	36
2.8 First Algorithm of TOPO MESH ADAPTOR	36
2.8.1 Preprocessing	37
2.8.2 Serial Programming on First Algorithm of TMA	37
2.9 Description of Boundary Conditions.....	38
2.10 Case File required as input for Telemac	38
2.11 Input File for the First Algorithm of TMA	43
2.12 Local Mesh Refinement	44
2.13 Parallel Mode of the First Algorithm on TMA	45
2.14 The Second Algorithm of TOPO MESH ADAPTOR.....	47
2.14.1 Preprocessing	47
2.14.2 Serial mode of the Second Algorithm in TMA.....	49
2.14.3 Parallel Mode of TMA Second Algorithm	49
2.14.4 Modifying the weight factor for netter convergence.....	51
2.15 Utilities for TMA.....	54

CHAPTER 3	APPLICATIONS TO DAMBREAK FLOW STUDIES ON TWO RIVERS IN QUEBEC	56
3.1	Test Case 1: A dambreak study on the Eastmain River	56
3.1.1	Topographic Data	56
3.1.2	Computational Domain	58
3.1.3	Analysis.....	61
3.1.4	Eastmain Refinement Results with TMA	63
3.1.5	Change of Bathymetry with TMA.....	63
3.1.6	Error File	66
3.1.7	Improvement of Inundation Lines after Refinements.....	68
3.1.8	Correction of Water Depth Representation Along the Shoreline.....	70
3.1.9	Modifying the Weight Factor	73
3.1.10	Conclusion.....	80
3.2	Test Case 2: Romaine-Puyjalon River	80
3.2.1	Boundary Condition and Initial Condition of the Romaine-Puyjalon River	81
3.2.2	Romaine-Puyjalon Mesh Refinement Results with TMA	85
3.3	Conclusion.....	95
CONCLUSION	96
FUTURE WORK	99
APPENDIX I	USER GUIDE OF TOPO MESH ADAPTOR.....	101
REFERENCES	129

LIST OF TABLES

		Page
Table 0.1	Dam failures in the world. This table is adapted from (failure, 2018).....	3
Table 3.1	Number of Elements and Nodes in different Meshes.....	66
Table 3.2	Comparison of the entering volume and exiting volume in the simulation with the different meshes.....	79
Table 3.3	Comparison of total volume, entering volume, exiting volume, and errors between different meshes on the Romaine-Puyjalon River.....	92

LIST OF FIGURES

		Page
Figure 1-1	Bed topography sketch for SWE.....	10
Figure 1-2	Positive and Negative Progressive waves. a) at $t=0$, b) at $t = t + dt$. This has been programmed with Visual Basic.....	15
Figure 1-3	standing wave profile made with superposition of two progressive 1 meter wave height in negative and positive directions.	16
Figure 1-4	Possibility of intermediate element in linear and discrete bathymetry (captured from (Tchamen & Kahawita, 1998).....	24
Figure 2-1	Illustration of Cut-elements. The Blue line is the border between the wet and dry elements. For each cut-element, we do the mesh refinement by adding two dry points with the maximum and minimum elevations.	31
Figure 2-2	Illustrating the effect of points in the inverse distance method.	33
Figure 2-3	Dividing the triangular figure into 3 parts with interior point.	35
Figure 2-4	First algorithm of TOPO MESH ADAPTOR.....	37
Figure 2-5	Illustrating discharge hydrograph over time for an assumed.....	38
Figure 2-6	Case File for Eastmain River.	40
Figure 2-7	Case File for Romain-Puyjalon River.	41
Figure 2-8	An example of a parameter.txt file.....	43
Figure 2-9	Comparison of initial mesh and the mesh after first refinements. New mesh is finer in cut-elements.	44
Figure 2-10	Illustration of the new added points along the inundation line ($H_{\text{countour}} = 0.05$). The projection of these data is UTM NAD83 Zone 33.....	44
Figure 2-11	Example of log file generated by the first algorithm.....	45
Figure 2-12	Example of parameters file for running parallel programming on first algorithm.....	46
Figure 2-13	GUI for collecting the nodes of mesh generated by Blue Kenue.....	48
Figure 2-14	Mesh refinement with second algorithm.	49

Figure 2-15 The word “yes” in the last three lines of a parameter.txt file would mean that the user wants to choose multiple points. If the user is satisfied with the results without choosing multiple points, he should write “No” in front of the last three lines. 50

Figure 2-16 Superposition of different contour lines for maximum water depth on the error_6_7.t3s. Black line is a contour line for $H > 0.05$, Yellow line is a contour line for $H > 1$, and Red line is a contour line for $H > 2$ 51

Figure 2-17 a) Cross-section of a channel with discontinuity in gradient of slop, b) Plane view of an element which is larger than the cross-section. 52

Figure 2-18 Location of new points after running the second algorithm of TMA. The projection of these data is UTM NAD83 Zone 33. 53

Figure 3-1 Official map of Eastmain River..... 57

Figure 3-2 Example of LIDAR data for the Eastmain River illustrated in Blue Kenue. 57

Figure 3-3 Terrain domain of the Eastmain River..... 58

Figure 3-4 Contour lines for topographic data of the Eastmain River..... 59

Figure 3-5 Boundary conditions of the Eastmain River. 59

Figure 3-6 Difference in computational domain of the Eastmain River: a) coarse mesh, b) finer mesh with LIDAR data done “manually” by Hydro-Quebec. The arrow points to the La Pêche River..... 60

Figure 3-7 Difference in inundation lines using coarse and finer meshes (the fine mesh was done manually by Hydro-Quebec around La Pêche River). 61

Figure 3-8 Illustration of a) free surface and b) water depth for the initial conditions of the Eastmain River..... 62

Figure 3-9 a) Location of new points in order to refine the cut-elements. These points are shown in green. b) Mesh refinement is based on new points around the inundation line. 64

Figure 3-10 An example of change of bathymetry with different refinements..... 65

Figure 3-11 Error file generated by Blue Kenue for comparing the water depth of the first mesh and the initial mesh..... 67

Figure 3-12 Error file generated by Blue Kenue for comparing the water depth of the 5th and 6th mesh refinements..... 67

Figure 3-13	Manual mesh refinement by comparing the LIDAR data and error_0_1.t3s.	68
Figure 3-14	Improvement of the inundation lines with different refinements.....	69
Figure 3-15	Visualization of inundation lines on topometric data.....	69
Figure 3-16	The change of the momentum equation in these regions was studied.	70
Figure 3-17	Change of the free surface max after the refinement with TMA in the cross- section of region 1: part a) before the mesh refinement; part b) after the mesh refinement.	71
Figure 3-18	Change of the free surface after the refinement with TMA in the cross-section of region 2: part a) before the mesh refinement; part b) after the mesh refinement.	72
Figure 3-19	Change of La Pêche River bathymetry with the different refinements.	73
Figure 3-20	Difference in altitude of La Pêche River in the 4th refinement with the different weight factors.....	73
Figure 3-21	Because of the uncertainty in the geometry, approximation of the elevation for point B created a wide channel, but in reality it is a narrow channel. .	74
Figure 3-22	Change of the bottom and the maximum free surface after mesh refinement with TMA at the beginning of La Pêche River in the first and third mesh refinements: a) the first mesh refinement; b) the third mesh refinement.	76
Figure 3-23	Change of the bottom and the maximum free surface at one-third inundation of La Pêche River: a) the first mesh refinement; b) the third mesh refinement; c) the fifth mesh refinement; d) the seventh mesh refinement.	77
Figure 3-24	Change of the bottom and the maximum free surface two-thirds inundation of La Pêche River in the different refinements: a) the first mesh refinement; b) the third mesh refinement; c) the fifth mesh refinement; d) the seventh mesh refinement. As can be seen, up until the third refinement, this section was dry.	78
Figure 3-25	Change of the bottom and the maximum free surface at the end of La Pêche River in the different refinements: a) the fifth mesh refinement; b) the seventh mesh refinement. As can be seen, until the fifth refinement, this section was dry.	79
Figure 3-26	Map of Romaine-Puyjalon River, domain of interest.	81

Figure 3-27 Initial conditions of the Romaine-Puyjalon River (first version)..... 82

Figure 3-28 Imposed boundary conditions for the Romaine- Puyjalon River domain. 83

Figure 3-29 Imposed dam break Boundary Condition for the Romaine-Puyjalon River.
..... 84

Figure 3-30 Terrain model of the Romaine-Puyjalon River. 85

Figure 3-31 Numerical model of the Romaine-Puyjalon River after refinement with
TMA..... 85

Figure 3-32 Comparison of inundation lines after the mesh refinement with TMA.... 86

Figure 3-33 Change of the bottom and maximum free surface at the extreme part of the
inundation line in dry zone 1..... 87

Figure 3-34 Change of the bottom and maximum free surface at the extreme part of the
inundation line in dry zone 2..... 88

Figure 3-35 Initial Condition of the Romaine-Puyjalon River (second version). 89

Figure 3-36 Initial numerical model of the Romaine-Puyjalon River. 90

Figure 3-37 Numerical model of the Romaine-Puyjalon River after two mesh
refinements with TMA..... 90

Figure 3-38 Error.t3s file generated between the first refinement and the second
refinement..... 91

Figure 3-39 Change of the inundation line after the mesh refinement with TMA. The
red line is the inundation line of the initial mesh refinement, and the blue
line is the inundation line after two further mesh refinements..... 91

Figure 3-40 Difference in the inundation line of the Romaine-Puyjalon River with
different meshes, prepared with Hydro-Quebec..... 92

Figure 3-41 Comparing the inundation limit for the mesh refined with TMA and the
finer mesh with LIDAR. 93

Figure 3-42 The bottom and maximum free surface along the Puyjalon branch before
the refinement..... 94

Figure 3-43 The bottom and maximum free surface along the Puyjalon branch after
mesh refinement with TMA. 94

LIST OF ABBREVIATIONS

SWE	Shallow Water Equation
SWWE	Shallow Water Wave Equation
TMA	Topo Mesh Adaptor
CFL	Courant Friedrich-Lewy
IDW	Inverse Distance Weight Interpolation
LIDAR	Light Detection And Ranging
DG	Discontinues Galerkin Method
GUI	Graphic User Interface

LIST OF SYMBOLS

BASIC UNITS

m	meter (Unit of length)
s	seconde (Unit of time)

LENGTH UNITS

m	meter
km	kilometer

AREA UNITS

m^2	square meter
km^2	square kilometer

TIME UNITS

h	hours
min	minute
s	second

REGULAR SYMBOLS

A	Area
E	Energy
c	Wave celerity
H	total water depth wave
h	average water level
ζ	sea level
z	bottom elevation
t	time
ρ	density
p	pressure
u	velocity
k	body force

μ	dynamic viscosity
λ	secondary viscosity
τ	stress
δ	displacement
u	velocity component
v	velocity component
g	gravity
ω	vertical velocity
S_0	friction bottom
S_f	friction coefficient
Q	discharge
Fr	Froude Number
C	Courant Number
h_l	Head lost
y_0	Normal Depth
y_c	Critical Depth

GREEK LETTERS

δ	delta
ϵ	epsilon
η	eta
λ	lambda
μ	mu
ν	nu
ω	omega
ϕ	phi
π	pi
θ	theta
ζ	zeta

INTRODUCTION

Spring snowmelt followed by rainfall is the major cause of flooding and inundations in Quebec, especially in the Ottawa River basin and the Montreal region. For the lower Ottawa River, there might be two snowmelt peaks. One is the snowmelt runoff from northeast basins, and the other one is from southern tributaries. Therefore, the Montreal region is sensitive to the most damaging floods located below the junction of the Ottawa River and the St. Lawrence River. Ice jams, which are the reason of breakup of ice cover caused by rapid change of weather with sensitive rainfall during the spring or midwinter are a major cause of floods in the St. Lawrence River. History shows that an inaccurate analysis of flood maps caused significant damage to cities. As an example, in 1957, rainfall of more than 250 millimeters in six hours on the area near Thetford Mines caused damage of about \$2 million. The St. Lawrence River, by flooding low-lying areas between Quebec City and Montreal, causes damage to shore properties. Ice jams develop in several sections of the river during the ice season. Then, during the spring freshet, water levels rise and the previously stable ice cover breaks up and major jams develop. In 1886 an ice jam flood on the St. Lawrence caused several millions dollars damage. In 1965, 20 people died in Montreal during the breakup period (Canada, 2010).

In 2017 it was also seen that rising water levels because of large amounts of rainfall closed several streets. A state of emergency was declared in La Pêche, schools were closed, and basements were flooded (floods, 2017). These floods could be a result of wave propagation from river inundations or because of ice-jam dambreaks happening upstream in a river and failure to adjust the outlet flow discharge from dam gates.

0.1. Dam breaks in the rest of the world

The following figures show the La Grande Hydroelectric Complex, a series of dams and power stations in Northern Quebec.

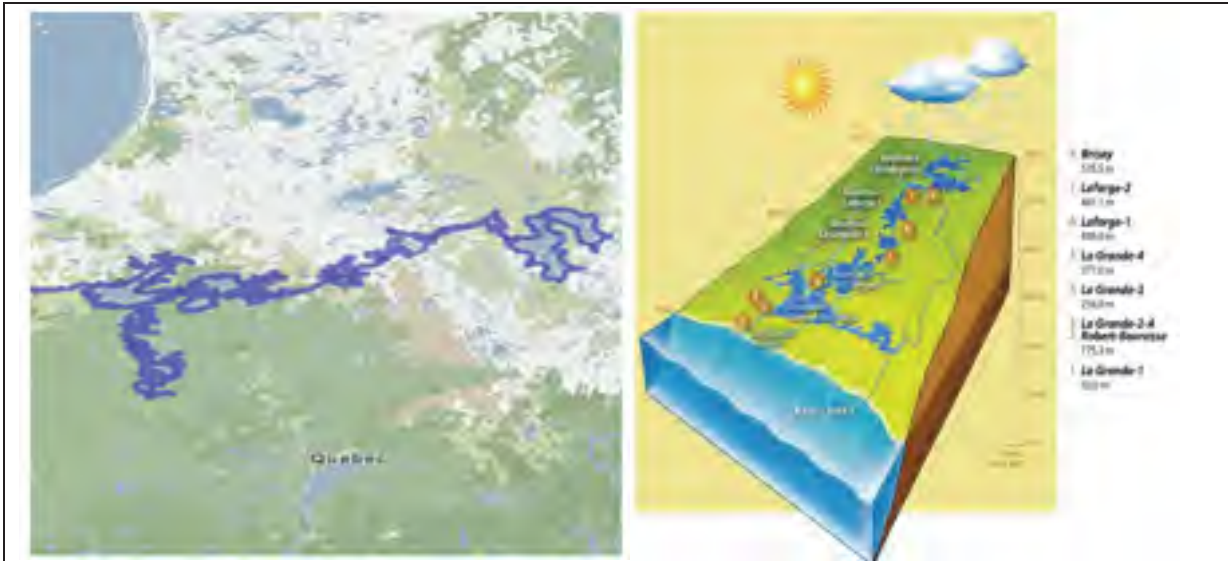


Figure 0.1 La Grande Rivière, one of the largest hydroelectric complex. Captured from (Mapfrappe, 2012).

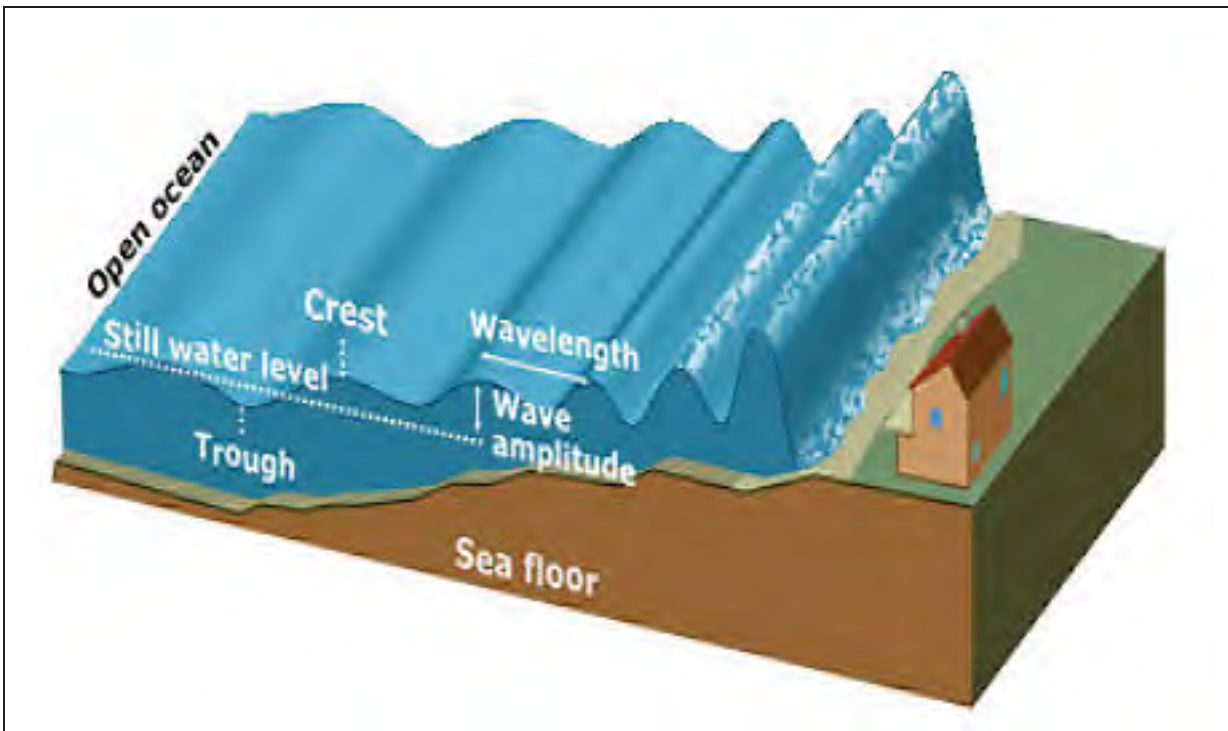


Figure 0.2 Increase in amplitude of waves after a tsunami or dam break, captured from (Tennessee).

A dam can be seen as a blockage in the way of flowing water which controls or slows down the outflow by creating a reservoir or impoundments. In certain dams the water flows continuously or intermittently through a special hydraulic work like a weir or a spillway. Dams have a large possible impact on destroying the environment and civilian populations. However, dam failures are very rare; but they usually cause loss of life when they happen. In 1975, dam failures in China killed more than 170,000 people and millions of people became homeless (failure, 2018). So, risk analyses over dam failures are one of the interests of this research.

Here are some of the main reasons for dam failures:

- Employing bad construction materials or techniques;
- Geological instability caused by rapid water level changes in the reservoir;
- Extreme inflow;
- Earthquakes;
- Human and design error.

In the following table, some examples of dam failures due to extreme inflow or heavy rain are shown.

Table 0.1 Dam failures in the world. This table is adapted from (failure, 2018).

Saguenay Flood	1996	Quebec, Canada	Problems started after two weeks of constant rain, which severely engorged soils, rivers, and reservoirs. Post-flood enquiries discovered that the network of dikes and dams protecting the city was poorly maintained.
Lake Toxaway Dam	1916	Transylvania County, United States	Heavy rains caused the dam to give way. Dam was later rebuilt in the 1960s.

Table 0.1 (continued).

Walnut Grove Dam	1890	Wickenburg, United States	Heavy snow and rain causing failure after public calls by the dam's chief engineer to strengthen the earthen structure.
Taisho Lake Dam	1995	Ide, Kyoto Prefecture, Japan	Under the influence of heavy rain, outburst from the Ninotani Lake Dam.
Testalinda Dam	2010-06-1	Oliver, Canada	Heavy rain, low maintenance. Destroyed at least 5 homes.

0.2. Dambreak flow modeling

Dam breaks analysis can be categorized by three different modeling techniques (Steininger, 2014). The first one is doing analysis by using the dam failure data, such as an outflow hydrograph and the dam's geometry. The second technique is to make some predictions based on physical characteristics that can cause dam failure. The third category is to do computational analysis through a numerical model of dam failure, overtopping, and flood wave propagation. Over the years, researchers have developed different flow models for analyzing and simulating dam breaks with the help of numerical solvers of the shallow water equations based on the finite difference, finite volume, and finite element methods. One of the most used softwares for solving the shallow water equations, especially for dambreak, in two dimensions is TELEMAC (Electricité de France). TELEMAC is a powerful open source software that can analyze the numerical model extremely fast; however, the validation of the results still takes some time. Often, the numerical model is generated using other software like Blue Kenue (Gardin, 2017). The preprocessors provide interfaces useful for integrating geospatial data and for post-processing the results obtained with TELEMAC.

0.3. Motivation and Objective

Numerical results of dambreak flows might have some uncertainties because of uncertain input parameters (such as the friction coefficient and the bathymetry or the topography of the river). Usually, the validation of the results is done by comparing the simulation results with real measurements. A calibration procedure is followed that consists of modifying the bed friction coefficient to control the errors. Identifying the uncertainty sources and how they might affect the results can also improve the computational model (Zagonjoli, 2007).

Generally, we can categorize the uncertainties as Input Uncertainty, Model Uncertainty, and Completeness Uncertainty. The roughness coefficient and the Digital Elevation Model (DEM) or aerial images data used for generating the numerical models which are considered the Input Uncertainty, can affect the scale of flooding in terms of measurement and time. (W. Lai & Khan, 2014) tried to use Galerkin's method for a nonlinear approach in a one-dimension shallow water equation. They studied how to avoid unphysical flows caused by improper treatment of the bottom slope. Advanced modeling of complex real cases is usually based on the two-dimensional shallow water equations.

OBJECTIVE: In this thesis, we propose an adaptive meshing methodology to improve the precision of flood maps for secondary valleys when rich topographic data of the LIDAR type on a complex bathymetry is available, for example, for the Eastmain River and the Romain-Puyjalon River. A code called TOPO MESH ADAPTOR (TMA) was implemented written in Python.

0.4. Thesis plan

In the **Introduction**, a short description of dambreak flows resulting from heavy rainfall is described. The general objective of the thesis is set out.

In **Chapter 1**, after discussing the different methods for describing the fluid motion, the development of shallow water equations (SWE) is explained. An introduction on wave movement and wave propagation in free surface flows, which is the main reason for floods in dry zones, is discussed. In this chapter, we also present some results related to studying the

effect of sudden changes of bathymetry in the appearance of discontinuous flows, such as hydraulic jumps. In **Chapter 2**, after a review of the software used in this thesis, Blue Kenue and TELEMAC, used for dambreak simulations, we present the methodology (TMA) to implement the mesh refinement along the inundation lines separating the wet and dry zones. Finally, **Chapter 3** shows the numerical tests performed to validate TMA that results in a better approximation for the cross-section of the rivers.

A user guide on how to run the code TMA is presented in **Appendix I**. It explains how to install and run TOPO MESH ADAPTOR on Windows, LINUX, or MAC along with a detailed example.

CHAPTER 1

LITERATURE REVIEW

To describe fluid motion, one should know the variation of physical quantities over time and space, for example, density, velocity, stress, pressure, etc., on the basis of specific spatial places. We can describe fluid motion in two ways. One is called the *Lagrangian* method, in which all fluid particles should be followed and the variations around each particle are described during the fluid motion. The other one is the *Eulerian* method, in which the variations as functions of time are described at all fixed positions. In other words, at different time, different particles pass through the particular location (Munson, Young, & Okiishi, 1990).

1.1 Shallow Water Equations (SWE)

In this section, we present the derivation of the Shallow Water Equations from the Navier-Stokes equations.

1.1.1 Eulerian Form

The Navier-Stokes equation is (Edom, 2008):

$$\rho \frac{Du_i}{Dt} = \rho k_i + \frac{\partial}{\partial x_i} \left\{ -p + \lambda^* \frac{\partial u_k}{\partial x_k} \right\} + \frac{\partial}{\partial x_j} \left\{ \mu \left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right] \right\} \quad (1.1)$$

and the material derivative is:

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u_i \frac{\partial}{\partial x_i} \quad (1.2)$$

where \vec{k} is body force, λ^* is second viscosity, p is pressure and μ is dynamic viscosity.

If we neglect the effect of body force, heat conduction, and viscous stresses, the Navier-Stokes equation produces:

$$\rho \frac{Du_i}{Dt} = \rho k_i - \frac{\partial p}{\partial x_i} \quad (1.3)$$

or

$$\rho \frac{D\vec{u}}{Dt} = \rho \vec{k} - \nabla p \quad (1.4)$$

1.1.2 Free Surface Flow Equations

If we consider that the fluid is incompressible, non-heat-conducting, non-viscous and the gravity is considered as the body force in z direction, we have:

$$\frac{Du_i}{Dt} = \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} - g\delta_{i3} \quad (1.5)$$

For incompressible flow, the mass equation or continuity equation is given by:

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (1.6)$$

1.2 Derivation of SWE

Assuming the velocity field in z direction is neglected, the vertical acceleration is also negligible compared to the gravity; thus, we have:

$$\frac{Dw}{Dt} = -\frac{1}{\rho} \frac{\partial p}{\partial z} - g = 0 \quad (1.7)$$

From this equation, we can obtain the following well-known equation:

$$p = \rho gh = \rho g(\eta - z) \quad (1.8)$$

where $\eta(t, x, y)$ is the free surface elevation. This equation shows that the pressure is purely hydrostatic. As we have atmospheric pressure at the free surface, for convenience sake we consider it to be zero.

$$P_{z=\eta} = P_{atm} = 0 \quad (1.9)$$

Equation 1.8 shows that the pressure is independent of the velocity components v and u . Differentiating of pressure with respect to x or y gives:

$$\begin{cases} \frac{\partial p}{\partial x} = \rho g \frac{\partial \eta}{\partial x} \\ \frac{\partial p}{\partial y} = \rho g \frac{\partial \eta}{\partial y} \end{cases} \quad (1.10)$$

These equations have no dependency on z . Thus, the motion equation with constant density becomes:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -g \frac{\partial \eta}{\partial x} \quad (1.11a)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -g \frac{\partial \eta}{\partial y} \quad (1.11b)$$

By performing an integration of the continuity equation with respect to the coordinate in z direction between a fixed bottom from $z_b = z_b(x, y)$ to $\eta = \eta(t, x, y)$, as shown in figure 1.1, we obtain:

$$\int_{z_b}^{\eta} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) dz = 0 \quad (1.12)$$

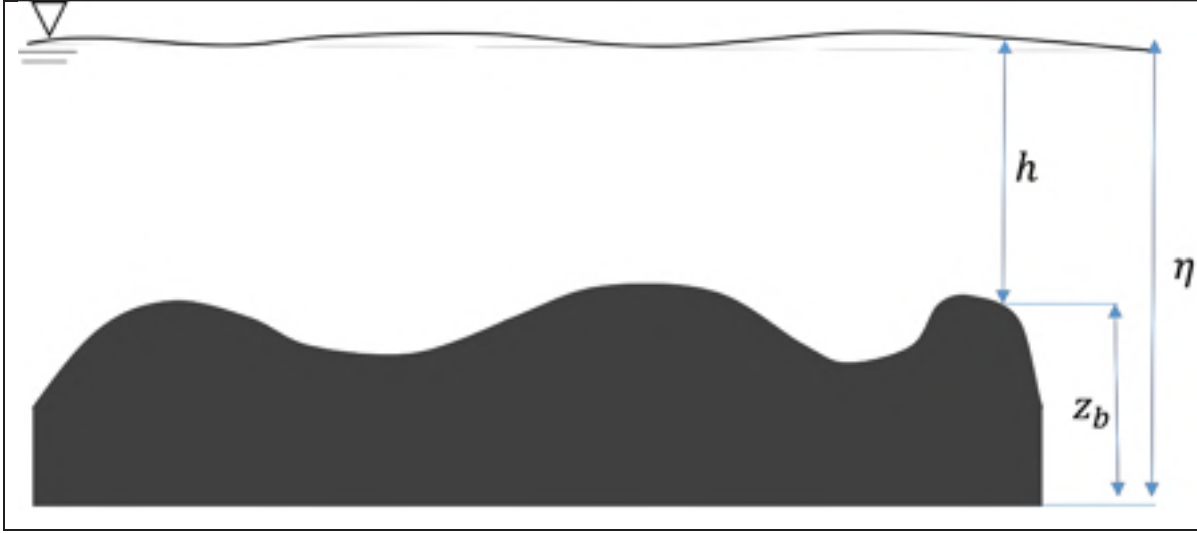


Figure 1-1 Bed topography sketch for SWE

which leads to:

$$w_{z=\eta} - w_{z=z_b} + \int_{z_b}^{\eta} \frac{\partial u}{\partial x} dz + \int_{z_b}^{\eta} \frac{\partial v}{\partial y} dz = 0 \quad (1.13)$$

With the help of the free surface derivative and Leibniz's theorem (Hossaini, 2003), applying the boundary conditions yields:

$$\left(\frac{\partial \eta}{\partial t} + u \frac{\partial \eta}{\partial x} + v \frac{\partial \eta}{\partial y} - w \right)_{z=\eta} = 0 \quad (1.14)$$

$$\left(\frac{\partial z_b}{\partial t} + u \frac{\partial z_b}{\partial x} + v \frac{\partial z_b}{\partial y} - w \right)_{z=z_b} = 0 \quad (1.15)$$

Therefore:

$$w_{z=\eta} = \left(\frac{\partial \eta}{\partial t} + u \frac{\partial \eta}{\partial x} + v \frac{\partial \eta}{\partial y} \right)_{z=\eta} \quad (1.16)$$

With $\frac{\partial z_b}{\partial t} = 0$ (for non-moving bed), we have:

$$w_{z=z_b} = \left(u \frac{\partial z_b}{\partial x} + v \frac{\partial z_b}{\partial y} \right)_{z=z_b} \quad (1.17)$$

For the two integral parts of equation 1.13 with partial integration, we have:

$$\int_{z_b}^{\eta} \frac{\partial u}{\partial x} dz = \frac{\partial}{\partial x} \int_{z_b}^{\eta} u. dz - u|_{z=\eta} \frac{\partial \eta}{\partial x} - u|_{z=z_b} \frac{\partial z_b}{\partial x} \quad (1.18)$$

$$\int_{z_b}^{\eta} \frac{\partial v}{\partial y} dz = \frac{\partial}{\partial y} \int_{z_b}^{\eta} v. dz - v|_{z=\eta} \frac{\partial \eta}{\partial y} - v|_{z=z_b} \frac{\partial z_b}{\partial y} \quad (1.19)$$

And by applying equations 1.16, 1.17, 1.18, 1.19 into 1.13, the result is:

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x} \int_{z_b}^{\eta} u. dz + \frac{\partial}{\partial y} \int_{z_b}^{\eta} v. dz = 0 \quad (1.20)$$

The average velocity components are defined as $U = \frac{\frac{\partial}{\partial x} \int_{z_b}^{\eta} u. dz}{h}$ and $V = \frac{\frac{\partial}{\partial y} \int_{z_b}^{\eta} v. dz}{h}$, where $h = \eta - z_b$ is the water depth. The depth averaged continuity equations for a non-moving bed reads:

$$\frac{\partial h}{\partial t} + \frac{\partial U h}{\partial x} + \frac{\partial V h}{\partial y} = 0 \quad (1.21)$$

Using the same integration over the depth and using Leibniz's integration rule, the momentum equations after ignoring the viscosity terms become:

$$\frac{\partial U h}{\partial t} + \frac{\partial U^2 h}{\partial x} + \frac{\partial U V h}{\partial y} = -g h \frac{\partial \eta}{\partial x} \quad (1.22)$$

and

$$\frac{\partial V h}{\partial t} + \frac{\partial U V h}{\partial x} + \frac{\partial V^2 h}{\partial y} = -g h \frac{\partial \eta}{\partial y} \quad (1.23)$$

As is clear, the momentum equations are dependent on the gradient of the free surface.

However, the right-hand term can be modified using:

$$\eta = h + z_b, \quad h \frac{\partial h}{\partial x} = \frac{\partial}{\partial x} \left(\frac{1}{2} h^2 \right) \quad (1.24)$$

This yields:

$$-gh \frac{\partial \eta}{\partial x} = -\frac{1}{2} g \frac{\partial h^2}{\partial x} - gh \frac{\partial z_b}{\partial x} \quad (1.25)$$

so we can rewrite the momentum equation in two-dimensions as follows:

$$\frac{\partial U h}{\partial t} + \frac{\partial}{\partial x} \left(U^2 h + \frac{1}{2} g h^2 \right) + \frac{\partial U V h}{\partial y} = -gh \frac{\partial z_b}{\partial x} \quad (1.26)$$

$$\frac{\partial V h}{\partial t} + \frac{\partial U V h}{\partial x} + \frac{\partial}{\partial y} \left(U^2 h + \frac{1}{2} g h^2 \right) = -gh \frac{\partial z_b}{\partial y} \quad (1.27)$$

In the one-dimension case, the flow discharge is equal to hu . Then the conservation equations read:

Continuity Equation:

$$\frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} = 0 \quad (1.28)$$

Momentum Equation:

$$\frac{\partial q}{\partial t} + \frac{\partial}{\partial x} \left(\frac{q^2}{h} + \frac{1}{2} g h^2 \right) = gh(S_0 - S_f) \quad (1.29)$$

The bed slope is:

$$\frac{\partial z_b}{\partial x} = -S_0 \quad (1.30)$$

and the friction slope is:

$$S_f = \frac{q^2 n^2}{h^{\frac{10}{3}}} \quad (1.31)$$

where n is the Manning friction coefficient.

1.3 Conservative variables in SWE

Different variables can be chosen as a way to describe the flow. One possible way is to consider the velocity and the water depth or the water elevation as the primary unknown variables. Another way is to choose the so-called “conservative” variables, which are the discharge competent hu and hv along with the water depth h .

The conservative Saint-Venant equations can be written as:

$$U_t + F_x = 0 \quad (1.32)$$

where $U = U(x, t)$ and $F = F(U)$,

with the conservative vector:

$$U = \begin{Bmatrix} h \\ hu \end{Bmatrix} \quad (1.33)$$

and the flux:

$$F(U) = \begin{Bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{Bmatrix} \quad (1.34)$$

The conservative equilibrium equations can admit a discontinuous solution, such as shock waves and hydraulic jumps. This equation in a quasilinear fashion can be written as:

$$U_t + F'(U)U_x = 0 \quad (1.35)$$

where $F'(U)$ is the Jacobian matrix:

$$F'(q) = \begin{bmatrix} 0 & 1 \\ -u^2 + gh & 2u \end{bmatrix} \quad (1.36)$$

This equation is hyperbolic since the Jacobian matrix has real distinct eigenvalues:

$$\lambda_1 = u - \sqrt{gh} \quad (1.37a)$$

$$\lambda_2 = u + \sqrt{gh} \quad (1.37b)$$

On the basis of the method of characteristic of SWE, the interaction of two waves C_1 moving downstream and C_2 with the velocity of $V - C$ which can move either upstream or downstream according to the sign of C , a subcritical or supercritical type of wave will be declared. If wave C_2 travels upstream ($V < C$) the flow is called subcritical, and if wave C_2 travels downstream ($V > C$) the flow is called supercritical (Litrico & Fromion, 2009).

1.4 Shallow Water Wave Equation (SWWE)

In this part, we survey the mathematical properties and characteristic structure of shallow water wave equation (SWEE). For a better understanding of the analysis of the wave equation, it is necessary to explain the wave criteria.

1.4.1 Progressive wave

Waves transfer energy from one point to another point. Progressive waves, usually known as sinus waves, move through a medium height and a displacement of particles in the same direction of waves' motions. We can describe the wave profile for a free surface flow as a combination of several progressive waves, such as:

$$\zeta(x, t) = a. \sin(k_1 x - \omega_1 t) + b. \sin(k_2 x - \omega_2 t) \quad (1.38)$$

$$H(x, t) = h(x, y) + \zeta(x, t), \quad \text{total water depth} \quad (1.39)$$

where for each progressive wave, we have $k = \frac{2\pi}{\lambda}$, $\omega = \frac{2\pi}{\tau}$, λ as the wavelength and τ as the wave period. The wave speed is defined as $c = \frac{\lambda}{\tau} = \sqrt{g * h}$, where g is the gravity constant.

$$\zeta(x, t) = a. \sin(k_1 x - \omega_1 t) \quad \text{Positive Progressive wave} \quad (1.40)$$

$$\zeta(x, t) = a. \sin(k_1 x + \omega_1 t) \quad \text{Negative Progressive wave} \quad (1.41)$$

As an example, for $h = 10.2$ cm, $\tau = 12$ and $a = 2$, we can show the progressive wave in figure 1-2 with the help of PDE.

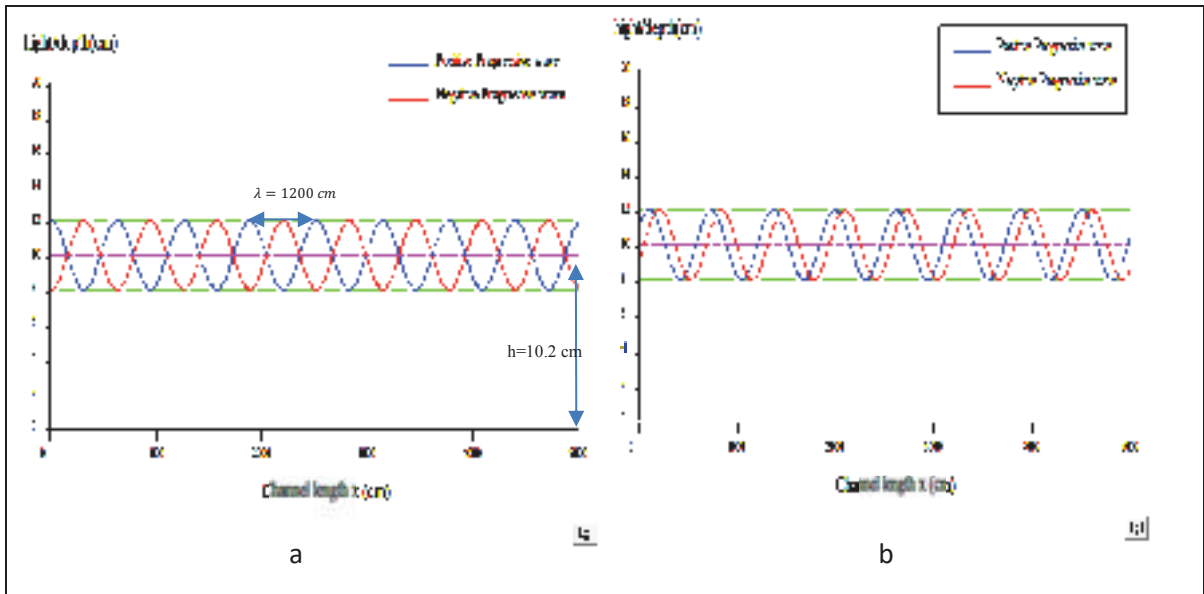


Figure 1-2 Positive and Negative Progressive waves. a) at $t=0$, b) at $t = t + dt$. This has been programmed with Visual Basic.

1.4.2 Standing Waves

Standing waves are the kinds of wave that have a vibration within the medium to minimum or maximum height of the wave. Actually, they do not have any movement in longitude, but they vibrate in an up-and-down direction. One of the ways that standing waves might be created is with the superposition of a positive progressive wave and a negative one with the same profile. For finding the standing wave with the same amplitude of a progressive wave, the wave amplitude equation becomes:

$$\zeta(x, t) = \frac{1}{2} (a \cdot \sin(k_1 x - \omega_1 t) + a \cdot \sin(k_2 x + \omega_2 t)) \tag{1.42}$$

Also, it should be considered that in some cases, where the wave motions are not both supercritical or subcritical we might see the mixture of fluid regime, like when the flow hit the wall or the adverse slope at the end of streams.

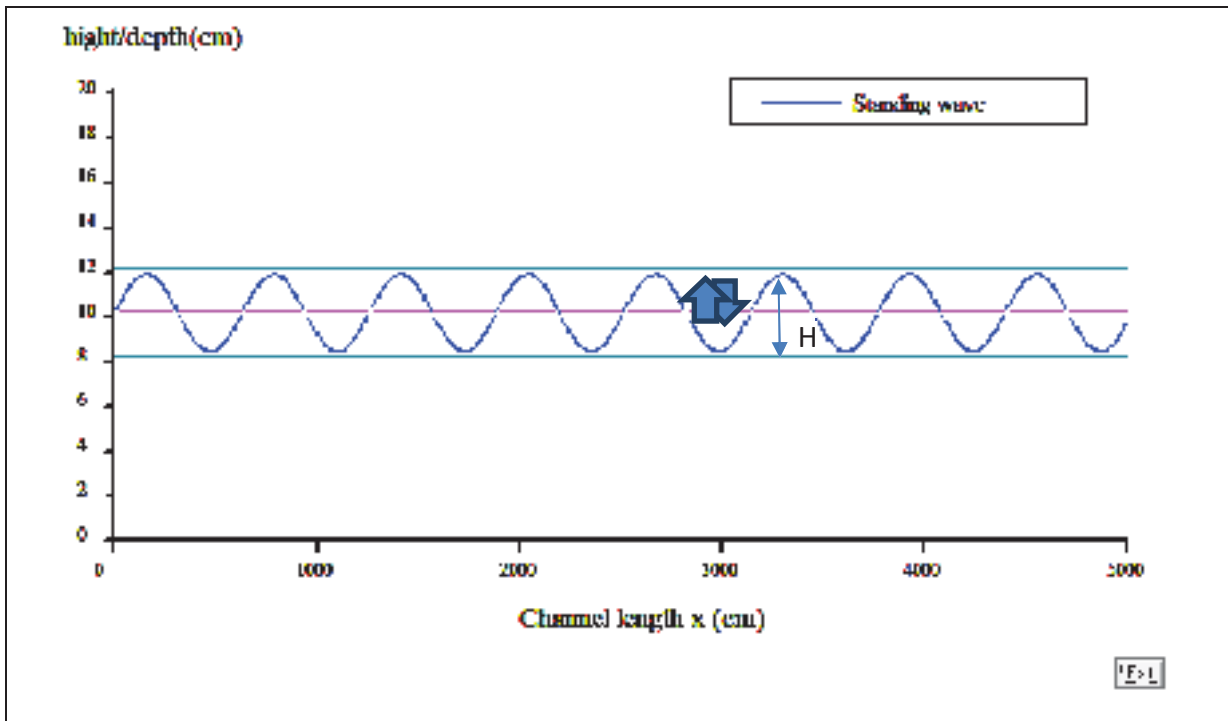


Figure 1-3 standing wave profile made with superposition of two progressive 1 meter wave height in negative and positive directions.

Figure 1-3 shows the profile of a standing wave.

1.5 Public domain software used to implement the TMA method

An in-house software based on the Python programming language was created for refining the mesh files generated with Blue Kenue (Blue Kenue is an open source for pre-processing and post-processing of 2D simulations with the finite element method). This new mesh is then used for future analysis with TELEMAC (EDF), and the process is repeated until the optention of final optimized mesh. This software is named TOPO-MESH-ADAPTOR (TMA) and can perform mesh refinement using appropriate aerial images or LIDAR data, where there are high changes in water level. Blue Kenue (Gardin, 2017-04-26) is mainly used for data preparation for hydraulic modeling but, it can also do pre-processing and post-processing with direct import/export from/to TELEMAC. One can have access the generated mesh files and boundary conditions prescribed by Blue Kenue by opening specific text files.

TELEMAC (EDF) is an open source program that has been used for different hydraulic applications. This code employs unstructured triangular meshes and the finite element method (FEM) for solving shallow-water equations. It also has a module for using the finite volume method (FVM).

1.6 Numerical difficulty in simulating Shallow Water Equations

Usually two types of difficulties can occur in the resolution of shallow water equations. The first is related to the preservation of the steady-state solutions for a volume of water initially at rest, and the second is related to the preservation of water height positivity. The first problem is related to the treatment of the source term as due to non-flat bathymetry or topography. Recently, the well-balanced numerical approach was designed specifically for calculating the steady-state condition with comparatively coarse meshes (Glowinski, Du, Hintermüller, & Suli, 2017; Zokagoa & Soulaïmani, 2010).

The other problem usually occurs when we want to solve SWEs over dry regions. This will happen in a simulation of flood waves, dam breaks and run-up phenomena with tsunamis on the coast. We should pay attention to the wet/dry front so that water height is always positive; otherwise, we may observe a nonphysical negative water height in the results. That will create a problem for calculating the eigenvalues $u \pm \sqrt{gh}$ used for determining the size of the time step Δt (Glowinski et al., 2017; Tchamen & Kahawita, 1998; Zokagoa & Soulaïmani, 2010).

1.6.1 Discontinuous Galerkin Method

Discontinuous Galerkin (DG) method employs discontinuous basic function, usually as the polynomials. The flexibility in DG method is that the degrees of polynomial in neighbors' elements can change independently (P-adaptivity). DG method can also apply to unstructured grids, with hanging nodes and refine the elements (h-adaptivity). Parallel calculation is very efficient in these cases. However, one of the most disadvantages of DG method is that it requires a high number of degrees of freedom (Blain & Massey, 2005). The DG method is

obtained by multiplying the test function $v(x, y)$ which is a polynomial in degree of m into the conservative form of equations (Jakeman, 2006)

1.6.2 Positivity-preserving Method

One of the difficulties in SWE analysis is applying the numerical method near the wet/dry front since there is no water near the wet/dry elements, and the SWEs are defined only in the wet region. With the Positivity-preserving Method using an adaptive mesh methodology, we can modify the mesh. Adaptive mesh methodology can be very accurate but usually it is expensive in computation, especially in 2D. That is why it is very important to do mesh refinement in a specified part of a dry zone. For example, doing mesh refinement in a wet/dry element is of great interest. Thus, the focus is on this particular aspect in this thesis. The first approach in the Positivity-preserving Method is the thin-layer approach (Glowinski et al., 2017). A thin layer on the dry zone is imposed (in our case 1 cm) for computation to avoid having a nonphysical negative value for water height. Another approach is to compute the velocity in the dry zone by height and discharge. In this method, velocity at a very low water height ($h \ll 1$ m) should not be seen (Glowinski et al., 2017). DG methods was previously applied to the Positivity-preserving Method (Bunya, Kubatko, Westerink, & Dawson, 2009; Kesserwani & Liang, 2012) mainly by modifying the slope in order to avoid negative water height. This is something common in both the DG method and finite volume method. Most high order methods for wet and dry treatment are done by post-processing work on reconstructing the data obtained after each numerical simulation. However, this might not guarantee the absence of a negative water height in the results with just a one-time reconstruction of the data, though it makes a big improvement in the results (Glowinski et al., 2017).

The general instability in the simulation shows itself in the form of a nonphysical negative water height that can cause a run-time error in the computation of the celerity and/or the calculation of the friction slope (Tchamen & Kahawita, 1998). This problem can be explained in the following ways:

- Using an inappropriate numerical scheme for mixed regime flows (complex flows condition);

- A large and sudden change in topography or wrong treatment of the friction source;
- Mishandling of elements that are partially wet or partially dry.

One of the challenges we are faced with in the numerical approach is that we cannot determine exactly which dry zone will be affected by the flow upon completion. In other words, we have moving boundaries in the domain. For this problem, two solutions exist: Following a Lagrangian approach or using the Eulerian grid. In this thesis we focus on the Eulerian grid.

1.7 Review of adaptive meshing methodology

The first approach of applying adaptive mesh methodology on fluid flow was by (Berger & Olinger, 1984) and (Berger & Colella, 1989) on a structured grid with the finite difference method. More recently, (K. George & Thomas, 2014) also did adaptive mesh methodology on ground water where there was a change in hydraulic head. They then optimized the results according to the error of the previous mesh to the new mesh.

Adaptive mesh refinement provides a powerful tool for solving fluid flow problems. In the past decades, many adaptive techniques have proposed a variety of computational solutions for problems involving a free surface. However, many of them are very complicated and difficult to implement.

The calculated mesh should be adapted to the displacement of the wet-dry front within specific physical areas. It can be adjusted to the shape of an area by applying Lagrangian formulations that take into consideration large displacements (Belytschko, Liu, Moran, & Elkhodary, 2013) or Arbitrary Lagrangian–Eulerian formulations (Donea, Huerta, Ponthot, & Rodriguez-Ferran, 2004). However, for such a large deformation, the mesh adaption can incur element folding or distortion.

To solve the problem of distortion, several methodologies were developed. One of the methodologies is to generate a mesh that covers the deformed part and estimate the result of that part in the newly created mesh (Askes & Sluys, 2000; Díez & Huerta, 1999; Mestreau, Löhner, & Aita, 1993). However, to implement this method an external software is needed to perform the meshing of the deformed parts in parallel, and it results in high complexity and

cost in time. Other methods, which are known as meshless methods, can approximate the overlapping space functions instead of the mesh calculations (Nguyen, Rabczuk, Bordas, & Dufloot, 2008). In contrast, particle finite element methods can build a Lagrangian configuration in which each node of the finite element mesh is tracked over time (Oñate, Idelsohn, Del Pin, & Aubry, 2004).

(Egelja, Schäfer, & Durst, 1998) studied an approach of applying adaptive grid Eulerian methods to solve mass and Navier-Stokes equations using a finite volume method with the help of a pressure correction method.

(Tsubaki & Fujita, 2010) studied the inundation flow structure in the basin of Shin-minato River and discussed the effect of topographic features on flow characteristic. (Hoppe & Krömker, 2009) and (Rath & Pasche, 2004) explained the general challenges for mesh generation and modeling flood maps with high resolution LiDar data.

Fixed or immersed mesh approaches are other methods in which the boundary of the mesh does not have to be matched with a physical boundary. In these methods, variations in finite element equations for the part of the mesh involved in the physical area do not have any meaning. Also, instead of assuming a computational mesh boundary, a function determining the boundary of the flow is imposed (Codina, Houzeaux, Coppola-Owen, & Baiges, 2009). The fixed meshed approaches are classified according to how they impose the boundary conditions and the relations between time and space over the meshes (Mittal & Iaccarino, 2005). For example, they include the approaches imposing the Dirichlet boundary conditions, where a penalty term is used in the immersed boundary method (M.-C. Lai & Peskin, 2000; Peskin, 1972); Approaches employing Lagrange multipliers, need additional calculations for the flows on the Dirichlet boundary (Barbosa & Hughes, 1991; Barrenechea & Chouly, 2012; Glowinski et al., 2017); Nitsche's approach, is commonly used and provides stable formulations that apply a limited estimated penalty term (Burman & Hansbo, 2010; Codina & Baiges, 2009; Hansbo & Larson, 2002; Nitsche, 1971).

Approaches can also be classified according to the relations between time and space. For example, in the immersed boundary method and the fictitious domain method, the part of the mesh that is not covered by the physical area is considered part of the same material as the physical boundary (Glowinski, Pan, Hesla, Joseph, & Périaux, 1999; Peskin, 1972). These

methods are simple but yield error results from calculating the temporal derivatives at the nodes close to the boundary on the basis of the non-physical area (Codina et al., 2009; Schott & Wall, 2014). For this reason, the fixed-mesh arbitrary Lagrangian–Eulerian method or isogeometric analysis has been developed to increase the accuracy of the temporal derivative on those nodes close to the boundary.

(Soulaimani & Saad, 1998) observed that the Arbitrary Lagrangian-Eulerian Kinematic Scheme is suitable to be applied on free boundary problems in fluid and solid mechanics. They improved the solution of the Navier-Stokes equation with an implicit method on the fluid and mesh motion. Using a Galerkin formulation and GMRES algorithm, they proposed a good method for 3D problem simulation. (Soulaimani, Fortin, Dhatt, & Ouellet, 1991) applied the Arbitrary Lagrangian kinematic description to express the conservation equation on the moving domain. They obtained a good approximation of a variational model by linearizing the referential motion.

(Dai & Schmidt, 2005) developed a moving mesh algorithm to simulate large deformation of a free surface flow with an unstructured three-dimensional grid. They employed a smoothing mesh algorithm where the interface was deformed. The edge swapping algorithm relocates the cell in order to improve its minimum quality to maximize it. The local mesh refinement with this method provides good mesh quality while avoiding global re-meshing.

The edge swapping algorithm uses a 2D Delaunay triangulation. This method is applicable only when the two-tetrahedral elements are non-convex on all common edges.

(Legat & Oden, 1995) studied the adaptive hp – finite element method on incompressible flows, mainly by adjusting h and p (h is mesh size and p is the degree of the polynomial). In this method, the computational error should be controlled by an adaptive scheme. The adaptive strategy is based on the following three steps: 1) a coarse mesh which considers a cheap approximation but gives important information for the next refinements, 2) the intermediate mesh produced by only h adaptivity of the first mesh, and 3) the final mesh adapted by modifying the degrees of freedom enrichments.

(Fondelli, Andreini, & Facchini, 2015) explain that the Volume of Fluid (VOF) method, which was introduced by (Hirt & Nichols, 1981), is generally used for multiphase problems. This method tries to capture the interface by calculating the volume fraction of the computational

cell with respect to one of the fluid phases. So, if the cell volume's fraction is zero or unity, it contains no interface; but if the volume fraction has a value, the cell has an interface. In this method the solution-adaptive mesh feature implemented in ANSYS-Fluent (commercial software) is employed to refine meshes at the water-air interface. Globally, using an adaptive meshing methodology in VOF reduces computational errors.

Using a numerical model for a large-size domain, such as flooded areas over a large plain, may result in less accurate solutions. Adaptive methods usually refine the mesh, estimate the errors, and construct new meshes (Berger & Olinger, 1984). Unstructured mesh flexibility gives us the option to investigate anisotropic mesh adaptation. The aim of adaptivity is to optimize the ratio between the accuracy of the results and the time consumption of the simulation. Its benefits usually derive from the flow's physical features in a hyperbolic system where there are strong anisotropic components. Therefore, this means that the coarse mesh usually is not accurate enough (Glowinski et al., 2017).

Anisotropic mesh adaption methodology is globally based on:

- A way to estimate errors;
- A method to determine the desired element size and direction;
- A mesh modification operator for generating the anisotropic mesh.

One of the best known methods is to use a Metric Tensor in Adaptation. For more information about this method, see (Glowinski et al., 2017).

Usually in the adaptive method some of the following efforts will be made to handle the wet and dry elements (Tchamen & Kahawita, 1998):

- Applying the water depth algorithm as follows:

Identifying all of the wet and dry elements. For determining the wet or dry elements, first we should determine all the nodes that are wet, $h > \varepsilon$ where ε is the limiting minimum depth, which is accepted as an agreement to determine that a node is wet. So, the element is dry if all the nodes are dry, the element is wet if all the nodes are wet, and an intermediate element (wet/dry element) is an element in which some of the nodes are dry and some are wet;

- When a numerical approach is performed on a whole grid. Some post-correction could be needed;
- Evaluating the depth at all dry nodes. Sometimes we see that the water level at one node is greater than the bottom elevation of its two neighboring nodes $(z_{nb} + h_{nb}) > z_p$. (like when we have the mixture of fluid regime in dry bed by wave motion). These parts globally do not respect the conservation law in the wet area. That is why a conservation of mass equation will usually have to be checked;
- Modifying the momentum equation by considering the velocity in the dry zone or intermediate elements to be zero or modifying the bottom slope term in the momentum equation.

As discussed above, the mesh adaptivity can be done in two ways: the Lagrangian approach or using the fixed Eulerian grid. In the Lagrangian approach, the mesh grids are generated continuously to find the best mesh. We need to have a complex algorithm to work with 2D problems, especially with complex topography. So, it is hard to apply this method for real cases. (Landry, Soulaïmani, Luke, & Ali, 2016) state that the moving mesh algorithms are mainly in four categories: Pseudo Structural Method, Inverse Distance Weight Function, Moving Submesh Approach, Geometric Element Transformation Method.

We can also use a fixed Eulerian grid. In this method the scheme is usually able to guess the region that will be affected by the flow and refine the mesh so that it is appropriate for the modeller. In this case, the problems usually arise for the elements that are partially wet (Tchamen & Kahawita, 1998). Tchamen and Kahawita explain that partially wet or dry elements happen when the bathymetry representation changes linearly (see figure 1-4).

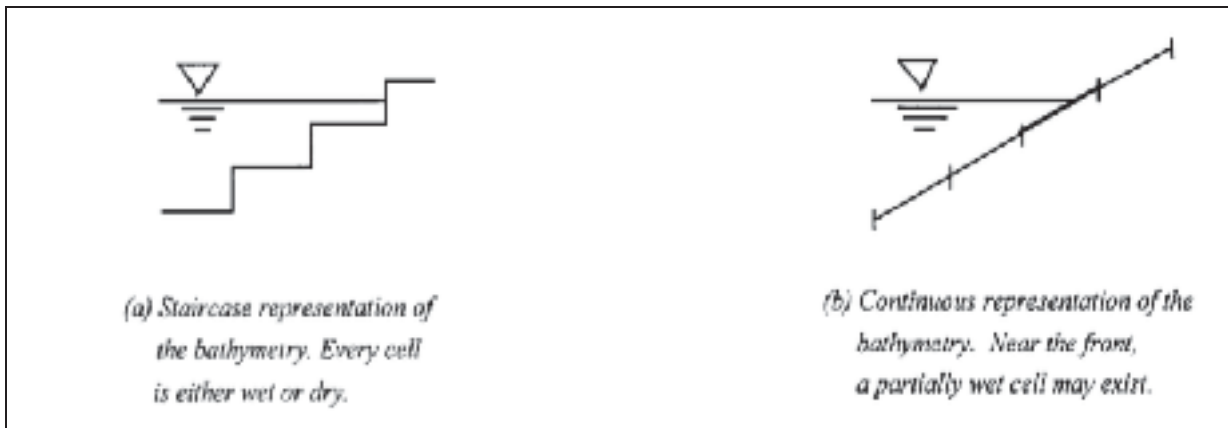


Figure 1-4 Possibility of intermediate element in linear and discrete bathymetry (captured from (Tchamen & Kahawita, 1998).

(Zokagoa & Soulaïmani, 2010) observe that steep slopes can produce undesirable instability in the numerical simulation by producing infinite or negative water depth or generating non-realistic velocity. A way to solve this problem is discussed in the following.

One use of adaptive mesh methodology with a fixed Eulerian grid is to apply it on partially wet elements on the basis of the velocity prediction within the cell. As the velocity in these cells is sometimes very large, it is essential to control it, for high velocities create instabilities. The non-physical negative value for water depth can be the result of an improper scheme in supercritical flow condition, or the occurrence of a large velocity can be the result of division by a previously evaluated depth that was very small in the numerical calculation (Tchamen & Kahawita, 1998). For more details on the existence of non-physical negative water depth, refer to (D. George, 2013).

For each mesh modification, there are the following three approaches (Tchamen & Kahawita, 1998):

- 1) *Mesh refinement/derefinement*: New nodes will be inserted in the elements when the error is too large, to make a finer grid. We might delete some nodes if the error around them is acceptable. We repeat this process until we reach to desired result.
- 2) *Mesh movement*: This approach is quite similar to the Lagrangian approach. In it the number of elements and nodes remains fixed, but the position of nodes will change. For this method, node movements are based on local velocities.

- 3) *Interpolation with remeshing*: This method is a combination of the previous two methods. This model generates a new mesh by adaptive process each time until it reaches the desired error distribution.

Some adaptive methods, such as the adaptive Newton-Raphson method and the adaptive Runge-Kutta method, have been applied on gradually varied flows by introducing a methodology to control the errors in the analysis at each time step, considering Δx a suitable step size. The idea is to use a small time step when the solution changes rapidly and to increase the time step when smooth areas are reached to keep errors within a desirable tolerance. The main object in doing mesh refinement is to use a smaller number of nodes while still obtaining more accurate results (Jahandar Malekabadi & Kalateh, 2017).

CHAPTER 2

METHODOLOGY FOR DEVELOPING THE TOPO MESH ADAPTOR METHOD

The methodology for developing TOPO MESH ADAPTOR method (TMA), which is used to do mesh refinements, is presented in this chapter.

2.1 Generating mesh with Blue Kenue

Blue Kenue is a powerful free software for mesh generation as well as for post processing results from hydrodynamic numerical models obtained using Telemac.

Blue Kenue can provide access to generated mesh files and imposed boundary conditions in text files. To generate a mesh with Blue Kenue for dambreak analysis one has to:

- 1) Define the limits of the domain by drawing the outline and imposing the boundary conditions on parts of the outline;
- 2) Use a subset of LIDAR data for generating the triangular mesh;
- 3) Generate the SELAFIN file (input data file in a particular format);
- 4) Impose the boundary conditions;
- 5) Generate the case file;
- 6) Run TELEMAC on the generated SELAFIN file with respect to the boundary conditions.

2.2 Boundary Conditions

Blue Kenue generates the text files that contain the mesh properties and the boundary conditions. Different boundary conditions can be specified in Blue Kenue using the following codes:

- | | |
|--|-----|
| 1) Boundary condition based on Q (discharge) | 455 |
| 2) Boundary condition based on H (water surface level) | 544 |
| 3) Boundary condition based on Q and H | 555 |
| 4) Boundary condition based on H and UV (velocity) | 566 |

5) Boundary condition based on UV	466
6) Boundary with incident waves	111

The boundary conditions that have been used for channel flows are of the Dirichlet type and can be characterized as follows (Litrico & Fromion, 2009):

- 1) When we have discharge at both boundaries, as when the channel is closed and there are two pumps in the boundaries, or when a hydraulic structure, such as a gate, produces a discharge;
- 2) When there is water depth at both boundaries. This will happen when the body of water is connected with a large body of water at each end, for example, a river in deltaic zone connected to the sea at one end and a lake at the other end;
- 3) When there is a discharge upstream and water depth downstream, as when there is a dam upstream or a known flow-water level relationship in a river that controls the discharge downstream so that it ends up in a river or a lake with known water elevation; This is the most common case;
- 4) When there is constant water depth upstream and discharge downstream. This situation occurs, for example, when there is a hydroelectric power plant that controls the downstream discharge of a river leaving from a very large reservoir with the same inflow into the lake.

Treatment of side walls:

- Most of the time, the software considers the slip-condition, by cancelling the velocity components, which are in a perpendicular direction to the wall. In other words, the section is considered wide enough to neglect the wall; otherwise, the wall friction should be considered by changing the hydraulic radius of the nodes in contact with the wall (Marqués).

2.3 Topo Mesh Adaptor (TMA)

LIDAR data or topometric data usually consist of millions of points. As they increase the time of simulation, it is not possible to use all these points to build the terrain model (i.e.

representation of the river bed for the hydraulic numerical model). So, one usually uses a subset of these points to build the numerical terrain model. Using a subset of topometric data may introduce an uncertainty in the geometry that in some cases may cause important inaccuracies in the hydraulic results (i.e. water levels) which are often post-processed when drawing flood maps. To reduce these uncertainties, a methodology is proposed that is based on adaptive refinements in which the terrain representation is enhanced iteratively with successive hydraulic simulations.

The flood fronts are usually defined by the positions where the water depth is almost zero. In order to obtain these refined meshes, we propose two mesh-refinement algorithms based on automatic mesh adaptive methodologies.

2.4 Simulation over dry beds

As discussed, the dambreak hydraulic analysis very often faces three major challenges:

- High water and velocity gradients in the propagation front;
- Stability problems on very rough beds near the shoreline (with very low depth);
- Wet-dry elements.

Imposing a large friction coefficient in numerical analysis and increasing the friction term can cause some instability, with results showing a number of oscillations. These oscillations generate very large and very small values during the numerical analysis (on water depth) that might kill the program or create negative values for water depths that are not physically acceptable.

For the wet-dry elements also, there is a discontinuity inside the shore elements. This causes difficulties in the numerical methods for integrating more accurately the conservation equations. One approach to enhance the accuracy is to refine the wet-dry elements.

By choosing more points, errors can be controlled for the cross-section or area of streams by decreasing the spatial step.

Recently a number of researchers have been working on how to use the discontinuous Galerkin method in one-dimension (W. Lai & Khan, 2014) or two dimensions, especially for higher order differential equations for discontinuous flow. The preferred method to avoid such discontinuity is to refine the mesh and increase the degrees of freedom at the location of the discontinuity. That is why engineers, in order to develop flood mapping, try to compare manually the obtained results of the maximum water depth at each node with respect to the LIDAR data or the available topometric data for that node and the neighbouring nodes. The most common reason of this type of re-meshing is when water depth should increase ($\frac{dy}{dx} > 0$), but steep adverse slopes or a dike in the way of the water acts like a wall. One method to solve this problem is the two-points cell refining method.

The following are the benefits of refining meshes using the two points with the maximum and minimum value of elevation:

- Refining elements with the maximum and minimum values of elevation obtained from the lidar data set helps us to consider the elements with the maximum slope that have the highest effect in the numerical simulation. In this way the numerical model indirectly considers the slope limit.
- Considering maximum and minimum values within an element also makes an approximation of the hydraulic channel cross-section easier. This means obtaining better information about the wet area in the channels, which improves the simulation, especially for cases with complex bathymetry.
- Mesh refinement in wet/dry elements makes the spatial step smaller. Increasing the degrees of freedom by considering more points in wet/dry elements also improves the accuracy of simulation.

The main objective is to define an automatic algorithm that can generate a new mesh with local mesh refinement of the elements with discontinuity or in regions characterized by a high-water depth gradient.

Before describing the algorithms, it is useful present some definitions that are of importance to better understand the concepts.

2.5 Cut-elements

The wet/dry elements are elements that are partially wet so that they are also partially dry. In these wet-dry elements there are some kinks in the water depth level.

In the proposed algorithm the wet-dry elements based on the maximum zone that are affected with water are known as “cut-elements”, and we try to do refine the mesh inside these elements. The element is considered completely wet if the minimum water depths at all nodes of that element are more than a specific value. We define this specific value according to two parameters, ε and α :

$$H_{wet} = \alpha * \varepsilon \quad (2.1)$$

where ε is a minimum water depth based on certain agreements for defining the wet or dry elements ($\varepsilon = 0.05 \text{ m}$) and α is a weight factor (safety factor). The weight factor will be discussed in more detail later.

The code that detects the cut-elements on the basis of post-processing of the results is called “`cut_elements.py`” in TMA.

Figure 2.1 shows an example of the mesh refinement on a selected cut-element.

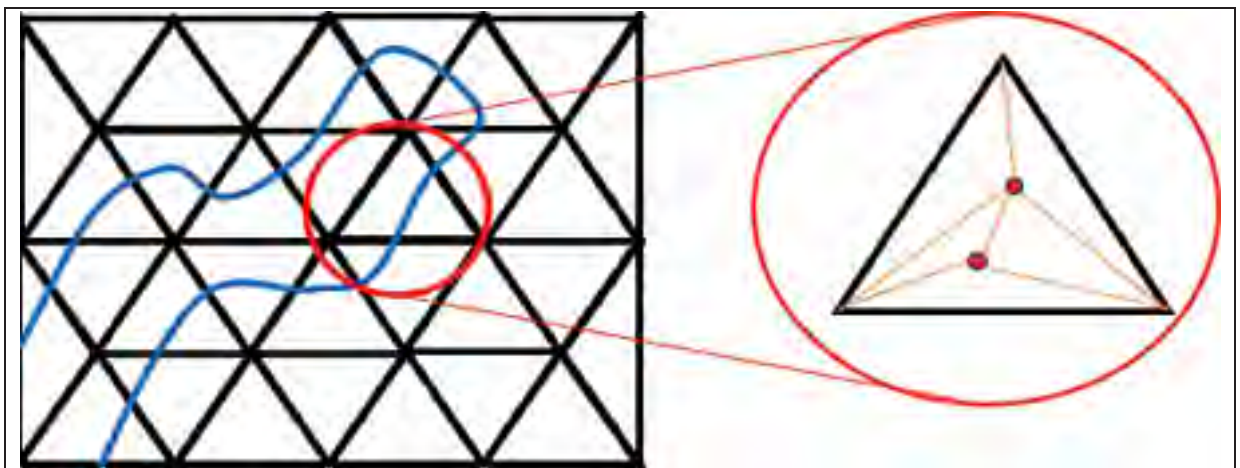


Figure 2-1 Illustration of Cut-elements. The Blue line is the border between the wet and dry elements. For each cut-element, we do the mesh refinement by adding two dry points with the maximum and minimum elevations.

2.6 Interpolation

Before explaining the type of interpolation, we use in this algorithm, it would be interesting to review different interpolation approaches. Interpolation in a numerical method serves to define an unknown value from a set of points with known values. Generally, the most important interpolation methods are:

1. Linear interpolation;
2. Lagrange Polynomial Interpolation;
3. Inverse Distance interpolation;
4. Nearest Neighbour Interpolation.

2.6.1 Linear Interpolation

The equation for linear interpolation between the two points with the coordinates of (x_1, y_1) and (x_2, y_2) for any point of (x, y) is:

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1} \quad (2.2)$$

The result is a continuous curve with discontinuous derivatives, so it is considered as C^0 class.

2.6.2 Inverse Distance Weight Function Interpolation (IDW)

Inverse Distance Interpolation (IDW) and Nearest Neighbour Interpolation are used quite often in geophysical analysis. IDW tries to approximate the results with the help of weight functions by assuming that the points closer to the position of a node with a fixed known value have more influence (weight) than those that are far off. So, the weight functions are based on a measured valued determined by the distance of the points from that position. The measured value of each point has a local influence that is distinguished by its distance from the position. When the points are closer, it has a stronger weight function; and when the points are farther off, the weight function is distinguished. For example, in the following picture the red points have more effect than the blue points for calculating the value of a point in the centre of the

circle. In figure 2.2 the effect of the distance weight function on the centre of the circle is shown.

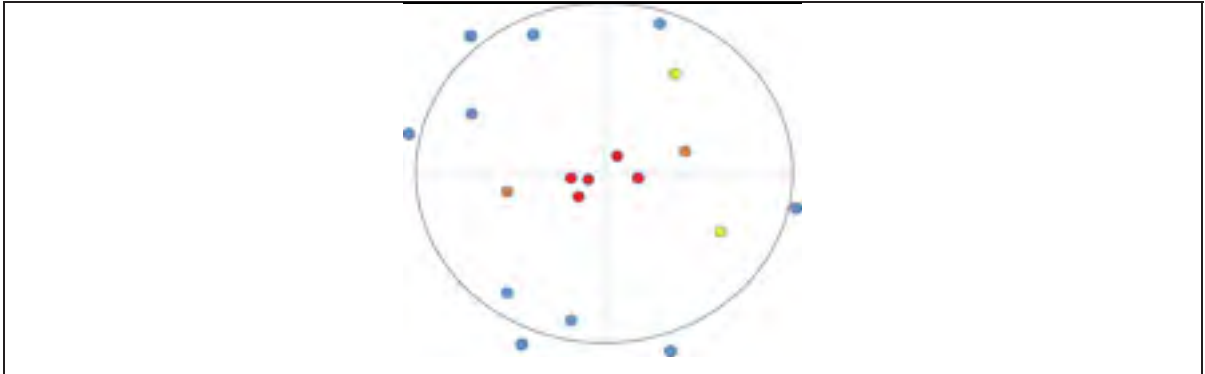


Figure 2-2 Illustrating the effect of points in the inverse distance method.

Generally, we can state the IDW as follows:

$$\hat{v}_1 = \frac{\sum_{i=1}^n \frac{1}{d_i^p} v_i}{\sum_{i=1}^n \frac{1}{d_i^p}} \quad (2.3)$$

where \hat{v}_1 is the value to be estimated, \hat{v}_i are the known values, and d_1^p, \dots, d_n^p are the distances from n points to the power of p .

2.6.3 Nearest Neighbour Interpolation

Nearest Neighbour Interpolation, or proximal interpolation, is a method of approximating the value of a function for each unknown point when the values of the surrounding points are known. With the nearest neighbour algorithm, we select the point value of the nearest point without considering the value of the other points. This methodology is used when producing a Voronoi diagram. In the Voronoi diagram, we decompose the space into different cells, all points within a cell being closer to the point associated with this cell. for each point. Therefore, we have discredited values for each cell in the space.

2.6.4 Linear 2D Interpolation using T3 elements

The methodology we used in our algorithm is based on Linear interpolation applied on triangular elements (T3). Bilinear interpolation is an extension of a linear interpolation in 2D that uses the three nearest node values to estimate a new node value.

When the shape functions for T3 elements (based on geometry) are derived in the next part, we can say for any value with the dependent node position (x, y) , such as water depth or elevation of topo-metric data, we have:

$$\phi(x, y) = N_1(x, y) \phi_1 + N_2(x, y) \phi_2 + N_3(x, y) \phi_3 \quad (2.4)$$

The code that does the interpolation in TMA is called “**interpolation.py**”.

In order to do the mesh refinement, we propose an adaptive mesh methodology that relies on LIDAR data. It is important to find all the points inside each cut-element as the first step. To evaluate the values of the points inside the elements, we interpolate between the vertices of elements using the shape functions. As the mesh generated by Blue Kenue uses linear triangular elements (T3), for each triangle we can employ the following FEM interpolation scheme:

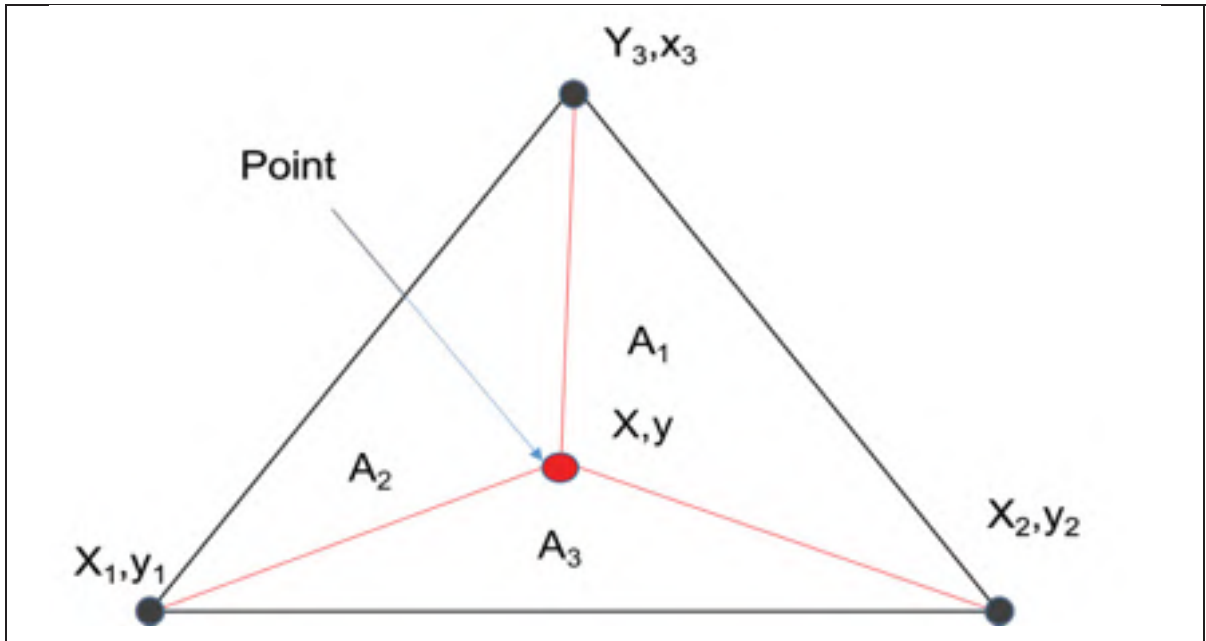


Figure 2-3 Dividing the triangular figure into 3 parts with interior point.

Suppose that we divide the area of triangle A into three triangular regions of A_1 , A_2 , and A_3 . It is clear that for any point inside the triangle we have:

$$A = A_1 + A_2 + A_3 \quad (2.5)$$

We can define the two-dimension linear interpolation shape functions N_1 , N_2 , and N_3 as:

$$N_1 = \frac{A_1}{A}, \quad N_2 = \frac{A_2}{A}, \quad N_3 = \frac{A_3}{A} \quad (2.6)$$

For any triangular element we have:

$$2A = (x_2y_3 - x_3y_2) - (x_1y_3 - x_3y_1) + (x_1y_2 - x_2y_1) \quad (2.7)$$

By using the above formula for each triangular region, we can calculate the shape functions:

$$N_1 = \frac{1}{2A}(x_2y_3 - x_3y_2 + (y_2 - y_3)x + (x_3 - x_2)y) \quad (2.8a)$$

$$N_2 = \frac{1}{2A}(x_3y_1 - x_1y_3 + (y_3 - y_1)x + (x_1 - x_3)y) \quad (2.8b)$$

$$N_3 = \frac{1}{2A}(x_1y_2 - x_2y_1 + (y_1 - y_2)x + (x_2 - x_1)y) \quad (2.8c)$$

Therefore, any point with the coordinate of (x, y) is inside the element if $N_1 + N_2 + N_3 = 1$. If any element of the shape function is zero, it means that the point is on an edge of the element, which will cause a problem in regenerating the mesh with Blue Kenue. In order to avoid having nodes on the edges of elements, only the points that are completely inside the element are chosen (not those on the edges), i.e.:

$$N_1, N_2, N_3 \neq 0 \quad (2.9)$$

The code snippet for identifying whether a point is inside an element or not in TMA is called “**inside_polygone.py**”.

2.7 Avoiding Creation of Elements That Are Too Small

In creating the algorithm, selecting the points that are too close to the edges of element, as they might produce very small elements with invalid shape factor ratios and can cause numerical problem in our analysis is avoided.

2.8 First Algorithm of TOPO MESH ADAPTOR

In the proposed method, we used the first algorithm only for the first refinement around the cut-elements. In the subsequent refinements, as it is not necessary to repeat the refinement for all elements, a second algorithm is proposed that does the refinement on the basis of other criteria. These criteria are discussed in the next part.

In the first algorithm, after running TELEMAC on the initial mesh, we find the two dry points with the maximum and minimum bottom elevations on the wet-dry elements; then Blue Kenue is used to generate the new mesh with the enriched list of nodes by applying a triangulation method (Delaunay) with the same outline. The result will be a new mesh with some refinements for the wet-dry elements. Figure 2.4 shows the first algorithm of Topo Mesh Adaptor.

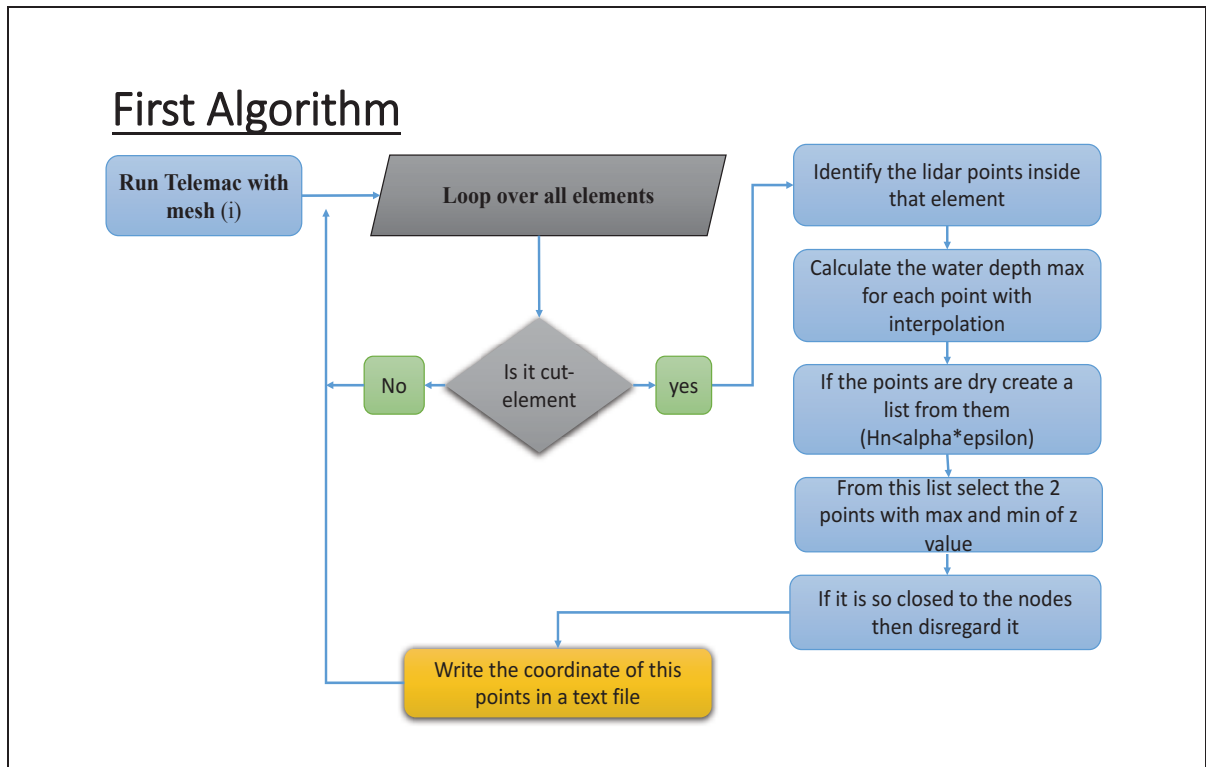


Figure 2-4 First algorithm of TOPO MESH ADAPTOR.

2.8.1 Preprocessing

We applied this algorithm on a case file of the Eastmain River in James Bay. In this case a large portion of the flow from the Eastmain River is diverted to the La Grande River. We observed significant changes in the topometric data in some regions, especially where there are dry beds. It demonstrates that working with coarse meshes cannot help to predict accurately the boundary of lands affected by the water.

2.8.2 Serial Programming on First Algorithm of TMA

The serial programming for the first algorithm of TMA can be applied on small rivers or when the user does not have access to the message passing standards, such as Message Passing Interface (MPI), for using the parallel mode. This code is called as “**TMA_Serial.py**”.

2.9 Description of Boundary Conditions

In our test case (Eastmain-1), after a dam break happens, the water moves downstream toward the Bay. The upstream conditions are based on the discharge and the downstream condition is based on a constant free water surface level (FS = 1.15 meter). For the discharge, we impose a dambreak hydrograph. This data has been provided by Hydro-Quebec, Quebec, Canada. In figure 2.5, the hydrograph of the discharge for the upstream boundary condition is shown.

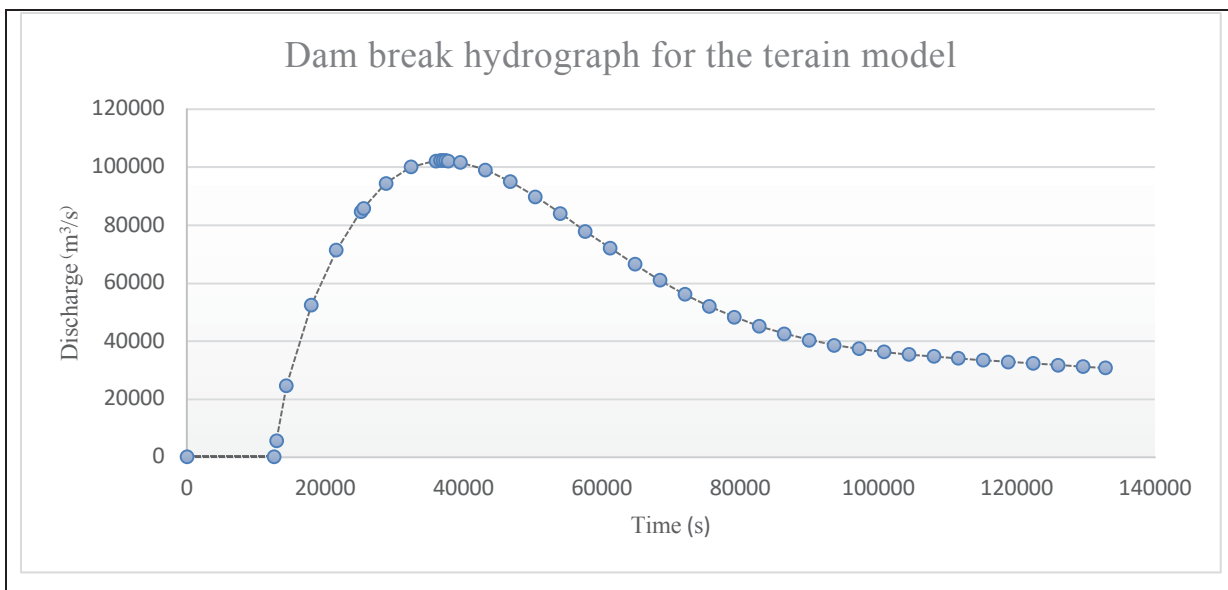


Figure 2-5 Illustrating discharge hydrograph over time for an assumed dam break for Eastmain river.

2.10 Case File required as input for Telemac

Telemac software requires a CAS text file as input to perform a simulation. CAS files contain identified properties and necessary variables to perform a simulation. For example, for this test case simulation of the Eastmain River dambreak case, we used a Strickler coefficient with a constant value of 30.

Below are two examples of case files (CAS) that were generated for running TELEMAC.

Figure 2-6 shows a case file for the Eastmain River, and figure 2-7 shows a case file for the Romaine-Puyjalon River. Note that all lines that begin with the slash character is considered as a comment.

```

CAS_EASTM.TXT - Ed 1992
TITRE = 'TELEMAC 2D : RIVIERE EASTMAIN: TOPOGRAPHIE LIDAR'
/
PROCESSEURS PARALLELES : 24
/
/
FICHER DES CONDITIONS AUX LIMITES : CL_GEO_EASTM.TXT
FICHER DE GEOMETRIE : GEO_EASTM.SLF
FICHER DES FRONTIERES LIQUIDES : EASTM_DEBIT.TXT
FICHER FORTRAN : EASTM_CDINI.FOR
FICHER DES RESULTATS : RES_EASTM.SLF
/
/----- POSTPROCESSING -----/
/
SUITE DE CALCUL = NON
VARIABLES POUR LES SORTIES GRAPHIQUES = U,V,H,S,B,W
DUREE DU CALCUL : 06400.
PAS DE TEMPS VARIABLE : OUI
NOMBRE DE COURANT SOUHAITE : 1.0
/NOMBRE DE PAS DE TEMPS = 28800 PAS DE TEMPS = 3.
PERIODE POUR LES SORTIES GRAPHIQUES = 300
PERIODE DE SORTIE LISTING = 300
INFORMATIONS SUR LE SOLVEUR : OUI BILAN DE MASSE : OUI
/----- CONDITIONS INITIALES -----/
CONDITIONS INITIALES : 'PARTICULIERES'
/FICHER DU CALCUL PRECEDENT : './CDINI_EASTM.SLF'
/----- CONDITIONS AUX LIMITES -----/
DEBITS IMPOSES : 0.0; 150. COTES IMPOSEES : 1.15; 0.0
/----- PARAMETRAGE NUMERIQUE -----/
PRECONDITIONNEMENT = 2
BANCS DECOUVRANTS = VRAI
FORME DE LA CONVECTION = 1;5
OPTION DE SUPG = 0;0
MAXIMUM D'ITERATIONS POUR LE SOLVEUR = 200
SOLVEUR = 7 OPTION DU SOLVEUR = 3 PRECISION DU SOLVEUR = 0.0001
STOCKAGE DES MATRICES : 3 PRODUIT MATRICE-VECTEUR : 1
IMPLICITATION POUR LA HAUTEUR = 0.6 IMPLICITATION POUR LA VITESSE = 0.6
MASS-LUMPING SUR H = 1.
CLIPPING DE H = NON LOI DE FROTTEMENT SUR LE FOND = 3
COEFFICIENT DE FROTTEMENT : 30.
MODELE DE TURBULENCE = 1 COEFFICIENT DE DIFFUSION DES VITESSES = 1.
EQUATIONS : 'SAINT-VENANT EF'
OPTION DE TRAITEMENT DES BANCS DECOUVRANTS : 1
TRAITEMENT DU SYSTEME LINEAIRE : 1
DISCRETISATIONS EN ESPACE : 11;11
/----- DISCRETISATIONS EN ESPACE : 12 ; 11 -----/
/----- FORME DE LA CONVECTION : 6;5 -----/
/----- OPTION DE SUPG : 1;2 -----/
/----- LOI DE FROTTEMENT SUR LE FOND = 3 -----/
/----- COEFFICIENT DE FROTTEMENT = 40. -----/
/----- PROPAGATION = OUI -----/
/----- COEFFICIENT DE DIFFUSION DES VITESSES = 0.005 -----/
/----- IMPLICITATION POUR LA HAUTEUR = 0.6 -----/
/----- IMPLICITATION POUR LA VITESSE = 0.6 -----/
/----- SOLVEUR : 1 -----/
/----- PRECONDITIONNEMENT : 2 -----/
/----- ORDRE DU TIR INITIAL POUR H : 1 -----/
/----- PRECISION DU SOLVEUR = 1.E-5 -----/
/----- OPTION DU SOLVEUR : 1 -----/
&FIN

```

Figure 2-6 Case File for Eastmain River.


```

/-----/
/          TELEMAC-2D Version 6.1 - OCT 2017
/          ROMAINE-1
/          CARTES INONDATION
/-----/
/          GW TCHAMEN
/          HYDRO-QUEBEC OCTOBRE 2017
/-----/
/ FICHER FORTRAN           : './T2D_CHDRUM.FOR'
/ NOMBRE MAXIMUM DE FRONTIERES : 99
/ FICHER DES CONDITIONS AUX LIMITES : './CL_FMSH.TXT'
/ FICHER DE GEOMETRIE       : './GEO_FMSH.SLF'
/ FICHER DES FRONTIERES LIQUIDES : './FL_ROM1.TXT'
/ FICHER DES RESULTATS      : './RES_ROM1_FMSH_V2.SLF'
/ LISSAGES DU FOND         : 1
/-----/
/                   OPTIONS GENERALES
/-----/
/ TITRE : 'TELEMAC-2D - ROMAINE-1 '
/
/ VARIABLES POUR LES SORTIES GRAPHIQUES : 'U,V,S,B,H,W,I,J'
/ PAS DE TEMPS                          : 1.0
/ PAS DE TEMPS VARIABLE                  : OUI
/ NOMBRE DE COURANT SOUHAITE             : 1.
/ DUREE DU CALCUL                       : 86400.0   /72 HRS
/ PERIODE POUR LES SORTIES GRAPHIQUES    : 300       /approx 5 min
/ PERIODE DE SORTIE LISTING              : 300       /approx 5 min
/ BILAN DE MASSE                         : OUI
/ INFORMATIONS SUR LE SOLVEUR            : OUI
/-----/
/                   MODELE DE TURBULENCE
/-----/
/ LOI DE FROTTEMENT SUR LE FOND : 4 /3=STRICKLER, 4=MANNING
/ COEFFICIENT DE FROTTEMENT      : 0.020
/ COEFFICIENT DE RUGOSITE DES BORDS: 11
/
/ MODELE DE TURBULENCE           : 3 /K-EPSILON
/ REGIME DE TURBULENCE POUR LES PAROIS : 2 /REGIME RUGUEUX
/ MODELE DE TURBULENCE           : 1
/ COEFFICIENT DE DIFFUSION DES VITESSES : 1.E-4
/-----/
/                   CONDITIONS INITIALES
/-----/
/ REMISE A ZERO DU TEMPS          : OUI
/ CONDITIONS INITIALES           : 'PARTICULIERES'
/
/ SUITE DE CALCUL                 : OUI
/ FICHER DU CALCUL PRECEDENT      : './FMSH_INI_V2.SLF'
/-----/
/                   CONDITIONS AUX LIMITES

```

Figure 2-7 Case File for Romain-Puyjalon River.

```

/-----
/
/                      CONDITIONS AUX LIMITES
/-----
/
/Frontiere 2: AVAL MODELE  455: 2 A 58
/Frontiere 1: AMONT MODELE 544: 2303 A 2291
/
COTES IMPOSEES :
0.0; 12.0
DEBITS IMPOSES :
2000.0; 0.0

/PROFILS DE VITESSE: 0; 1; 1; 0; 0; 4; 1; 0
//

/-----
/
/                      OPTIONS NUMERIQUES
/-----
/
/EQUATIONS : 'SAINT-VENANT VF'
/ 0 : ROE 1 : CINETIQUE ORDRE 1  2 : CINETIQUE ORDRE 2
/SCHEMA EN VOLUMES FINIS           : 0
BANCS DECOUVRANTS                   : OUI
CLIPPING DE H                       : NON
/VALEUR MINIMUM DE H                 : 0.001
TRAITEMENT DU SYSTEME LINEAIRE      : 1
/COMPATIBILITE DU GRADIENT DE SURFACE LIBRE : 0.9
STOCKAGE DES MATRICES                : 3
CORRECTION DE CONTINUITE            : OUI
/
SOLVEUR                              : 7
OPTION DU SOLVEUR                     : 3
PRECISION DU SOLVEUR                  : 1.E-4
MAXIMUM D'ITERATIONS POUR LE SOLVEUR : 100
/
DISCRETISATIONS EN ESPACE            : 12;11
FORME DE LA CONVECTION                : 1;5
OPTION DE SUPG                        : 1;2
NOMBRE DE SOUS-ITERATIONS POUR LES NON-LINEARITES : 2
MASS-LUMPING SUR H                    : 1
MASS-LUMPING SUR LA VITESSE           : 1
IMPLICITATION POUR LA HAUTEUR        : 0.6
IMPLICITATION POUR LA VITESSE       : 0.6
/
/
/-----
/
/                      CALCUL PARALLELE
/-----
/
PROCESSEURS PARALLELES                : 24
/NOMBRE DE TABLEAUX PRIVES           : 5
/
/-----

```

Figure 2-7 Case File for Romain-Puyjalon River (Continued).

2.11 Input File for the First Algorithm of TMA

In order to distinguish the cut-elements, it is necessary to create a file for the “maximum water depths” Blue Kenue.

To be able to run the first algorithm, it is necessary to prepare the following files from the initial mesh as the input files:

1. “the maximum water depth” of the initial mesh;
2. the geometry of the initial mesh, that is a mesh file with the point values of the elevation;
3. LIDAR or topometric files.

A “parameter.txt” file must be created that specify the input and output files names that are required. The detailed explanation on creating the parameters file and running the first algorithm of TOPO MESH ADAPTOR can be found in Appendix I.

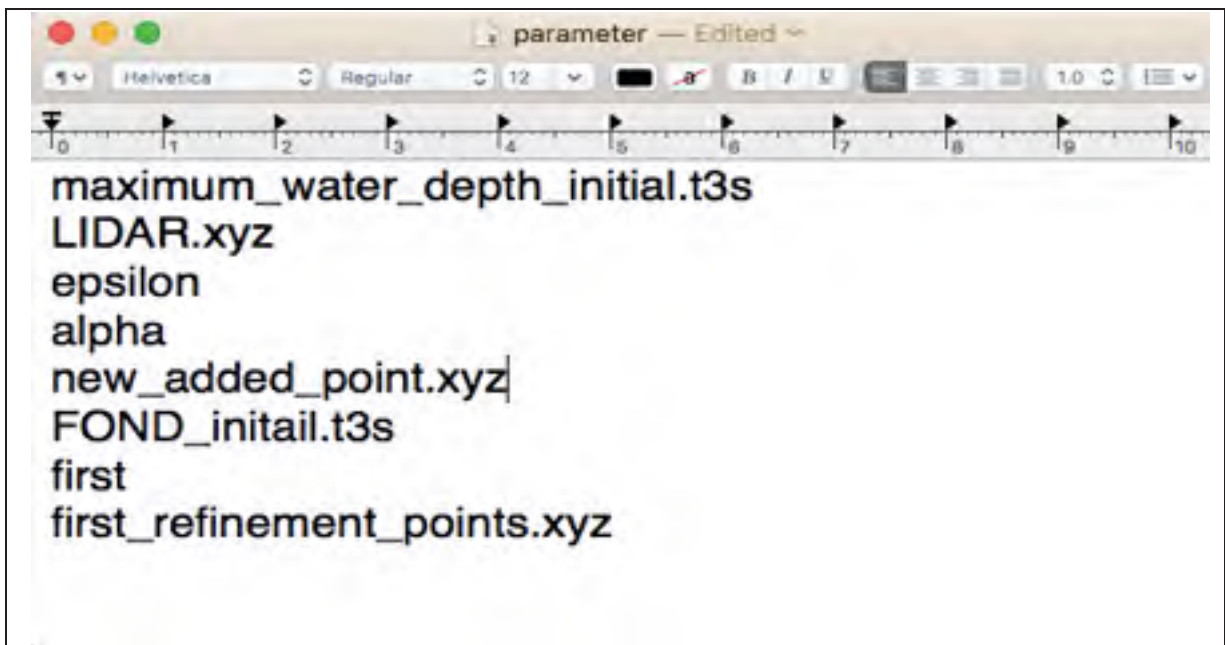


Figure 2-8 An example of a parameter.txt file.

2.12 Local Mesh Refinement

The following illustration shows the superposition of the initial mesh and the mesh after the first refinement. The tiny elements drawn in blue are related to the new mesh.

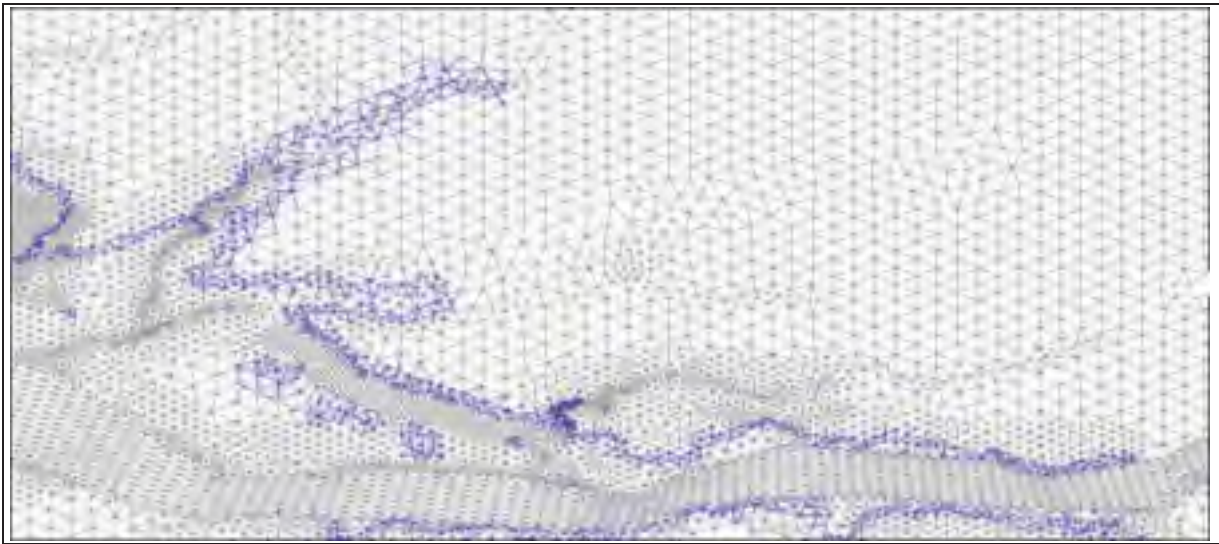


Figure 2-9 Comparison of initial mesh and the mesh after first refinements. New mesh is finer in cut-elements.

The refinement is applied to the cut-elements according to the value of maximum water depth observed during the whole duration of the simulation (figure 2.10).

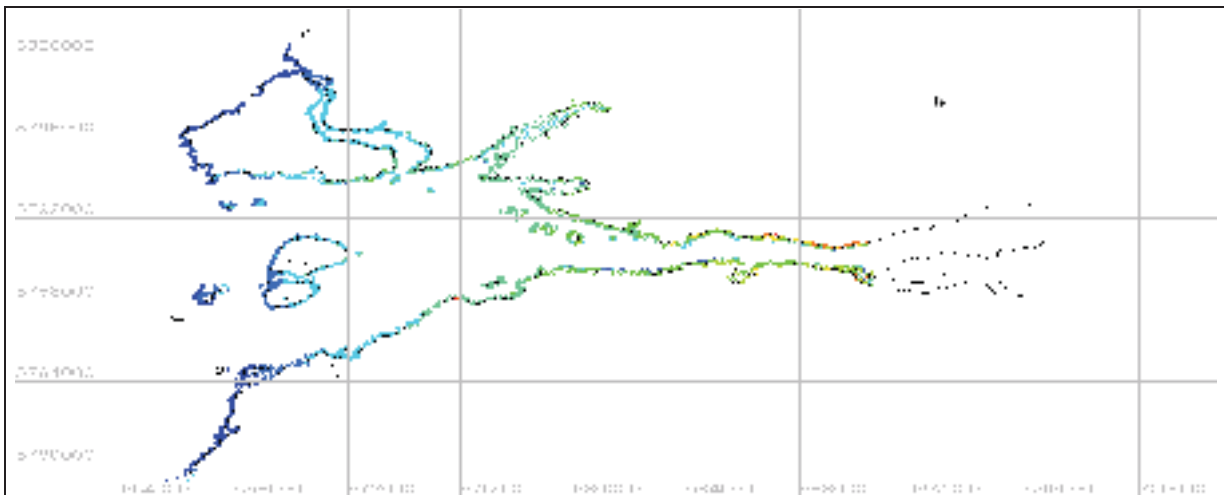
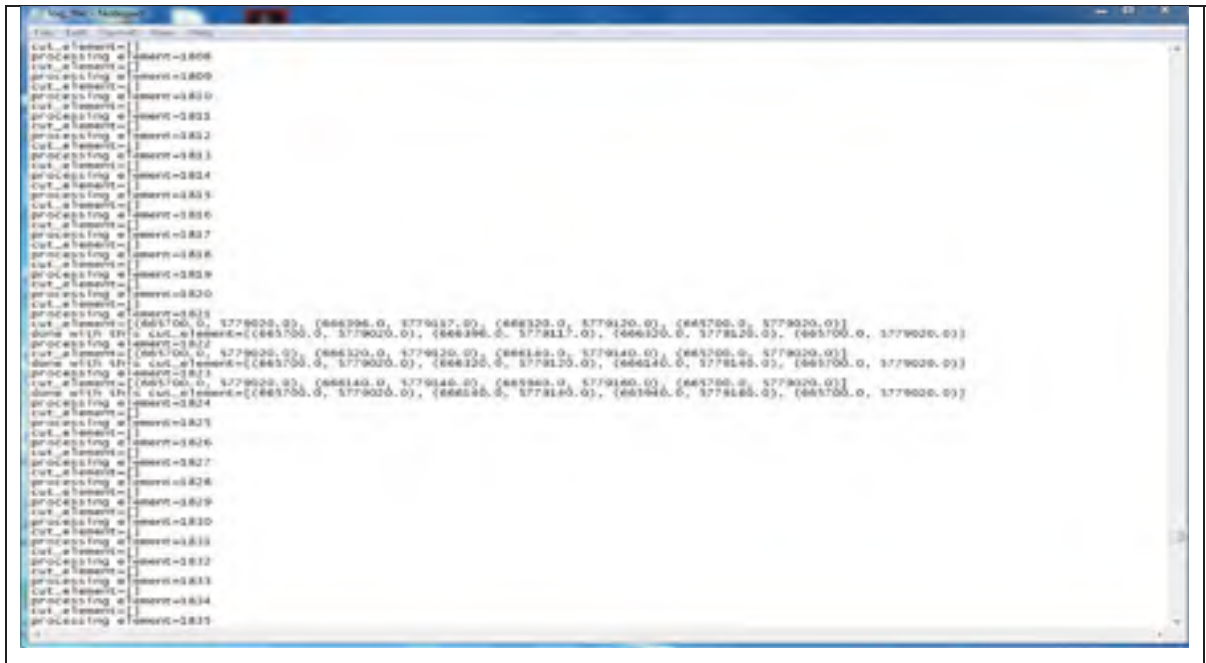


Figure 2-10 Illustration of the new added points along the inundation line ($H_{countour} = 0.05$). The projection of these data is UTM NAD83 Zone 33.

After running the first algorithm successfully, the code generates log files in which information about the cut-elements can be found that helps to validate the results drawn in figure 2.11.



```

CutL_#Element=1 | element=-0.006
prncwspk long # |
CutL_#Element=2 | element=-0.009
prncwspk long # |
CutL_#Element=3 | element=-0.010
prncwspk long # |
CutL_#Element=4 | element=-0.011
prncwspk long # |
CutL_#Element=5 | element=-0.012
prncwspk long # |
CutL_#Element=6 | element=-0.013
prncwspk long # |
CutL_#Element=7 | element=-0.014
prncwspk long # |
CutL_#Element=8 | element=-0.015
prncwspk long # |
CutL_#Element=9 | element=-0.016
prncwspk long # |
CutL_#Element=10 | element=-0.017
prncwspk long # |
CutL_#Element=11 | element=-0.018
prncwspk long # |
CutL_#Element=12 | element=-0.019
prncwspk long # |
CutL_#Element=13 | element=-0.020
prncwspk long # |
CutL_#Element=14 | element=[(665700.0, 3779020.0), (666396.0, 3779137.0), (666120.0, 3779120.0), (666700.0, 3779020.0)]
prncwspk long # |
CutL_#Element=15 | element=[(665700.0, 3779020.0), (666396.0, 3779137.0), (666120.0, 3779120.0), (666700.0, 3779020.0)]
prncwspk long # |
CutL_#Element=16 | element=[(665700.0, 3779020.0), (666396.0, 3779137.0), (666120.0, 3779120.0), (666700.0, 3779020.0)]
prncwspk long # |
CutL_#Element=17 | element=[(665700.0, 3779020.0), (666396.0, 3779137.0), (666120.0, 3779120.0), (666700.0, 3779020.0)]
prncwspk long # |
CutL_#Element=18 | element=[(665700.0, 3779020.0), (666396.0, 3779137.0), (666120.0, 3779120.0), (666700.0, 3779020.0)]
prncwspk long # |
CutL_#Element=19 | element=-0.021
prncwspk long # |
CutL_#Element=20 | element=-0.022
prncwspk long # |
CutL_#Element=21 | element=-0.023
prncwspk long # |
CutL_#Element=22 | element=-0.024
prncwspk long # |
CutL_#Element=23 | element=-0.025
prncwspk long # |
CutL_#Element=24 | element=-0.026
prncwspk long # |
CutL_#Element=25 | element=-0.027
prncwspk long # |
CutL_#Element=26 | element=-0.028
prncwspk long # |
CutL_#Element=27 | element=-0.029
prncwspk long # |
CutL_#Element=28 | element=-0.030
prncwspk long # |
CutL_#Element=29 | element=-0.031
prncwspk long # |
CutL_#Element=30 | element=-0.032
prncwspk long # |
CutL_#Element=31 | element=-0.033
prncwspk long # |
CutL_#Element=32 | element=-0.034
prncwspk long # |
CutL_#Element=33 | element=-0.035
prncwspk long # |

```

Figure 2-11 Example of log file generated by the first algorithm.

2.13 Parallel Mode of the First Algorithm on TMA

A parallel version of the code was developed using the library MPI4Py. This code helped to get the results in less than six hours using six processors, while the serial code took more than 2 days. The code for running the first algorithm of TAM in parallel is called “TMA_FA_parallel.py”.

The parallel programming is based on decomposing the LIDAR data into a number of parts, each of which is assigned to a processor where the first algorithm is applied (i.e. using the Single Program Multiple Data paradigm) of LIDAR files, which are imposed on the parameters file. If the main LIDAR data can be divided into 6 files, the following command should be written in Terminal (Mac or Linux) or Windows command line to run the code in parallel:

mpiexec - np 6 python MTMA_FA_parallel.py

In the parallel mode, each processor running the first algorithm is using one of the LIDAR files.

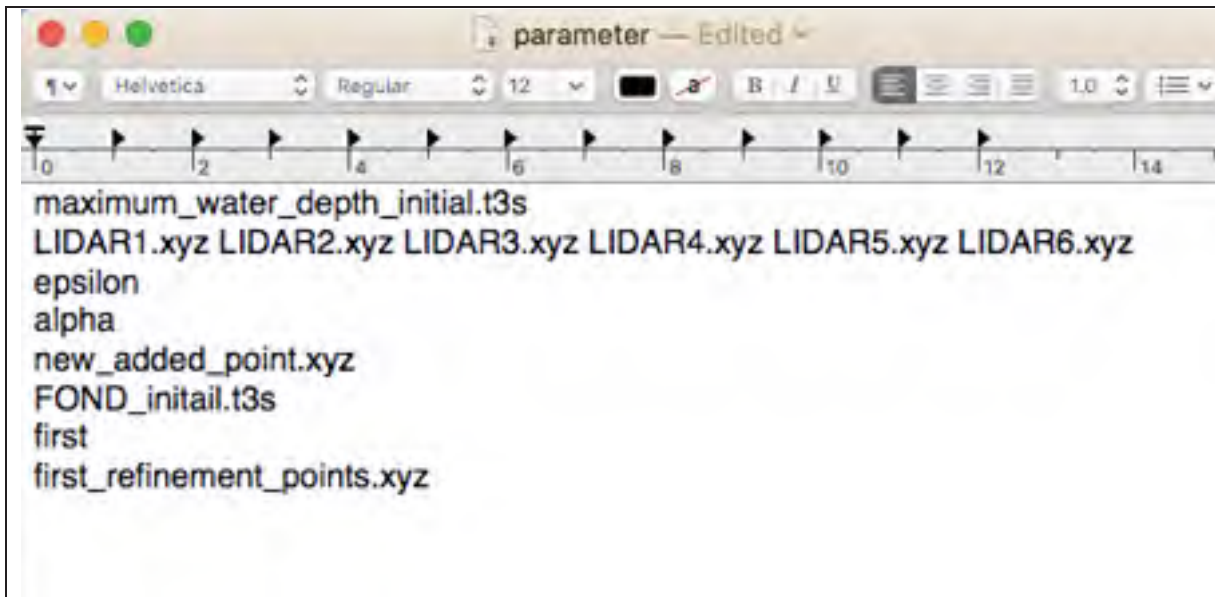


Figure 2-12 Example of parameters file for running parallel programming on first algorithm.

In this case, we have a separated log file for each processor running the code. After generating the new mesh with the first algorithm, the second algorithm should be employed for further refinements.

In the second algorithm, by comparing the results obtained using the initial (coarse) mesh and the first enriched mesh, we define new criteria that enable us to make the mesh refinement better.

The last part in the TMA algorithms is to create a *.xyz file that has the complete list of new points necessary to create the next mesh. In TMA a code named “sum_points.py” collects the nodes of the previous mesh and the new added points in one file.

2.14 The Second Algorithm of TOPO MESH ADAPTOR

In the second algorithm, we first compute the difference between the solutions of the maximum water depths obtained using the first refinement (last mesh) and the initial mesh (previous mesh), respectively. This difference is computed following the projection of the coarse mesh solution onto the refined mesh (by interpolation). Then, the cut-elements that have significant difference (this defines a mesh indicator or ‘error criterion’) are processed for refinement. In the second algorithm, we have the ability to select more points inside the elements to obtain finer elements. This option allows for a better representation of the bathymetry and a better approximation of the cross-sections of the streams.

2.14.1 Preprocessing

Before running the code for the second algorithm we should 1) generate the error file and 2) interpolate the maximum water depth obtained from the previous result on the newly generated numerical model as follows.

2.14.1.1 Generating the Error File

- 1) By interpolating the “maximum water depths” of the initial mesh (mesh (0)) on the first mesh refinement (mesh (1)) we can create a file named “water_depth_max_IP_0_1”. We do this step using the Blue Kenue software.
 - Blue Kenue has built-in option to do this interpolation; it uses “the inverse distance interpolation” or “the nearest neighbour interpolation” methods.
- 2) We generate a file using Blue Kenue named “error_0_1.t3s” created by taking an absolute value of the subtraction of “water_depth_max_IP_0_1” minus “water_depth_max” of the first mesh refinement (or mesh (i)) (i.e. $\text{abs}(\text{maxH}(\text{mesh}(i-1)) - \text{maxH}(\text{mesh}(i)))$).

2.14.1.2 Interpolating the “Maximum Water Depth” of mesh (i-1) on Mesh (i)

Interpolating a list of the points on a mesh is easily performed using Blue Kenue. So, if we want to interpolate mesh (i-1) on mesh (i), we need to have the nodes of the previous mesh. We can collect the nodes of the mesh manually from a text file or by using our generated graphic user interface GUI. In figure 2.13 this code is shown.

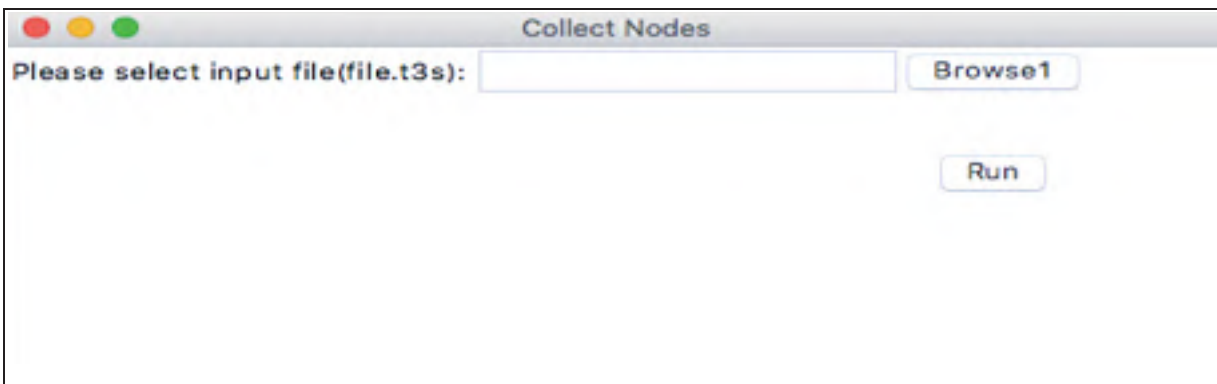


Figure 2-13 GUI for collecting the nodes of mesh generated by Blue Kenue.

After getting the nodes, we should use Blue Kenue for:

- 1) creating a “new interpolation” with the generated list of nodes;
- 2) using a map object to interpolate the “new interpolation” of topographic values into the desired mesh.

The methodology in the second algorithm is applied to (1) construct a list from all of the elements which have significant errors, (2) find the two points with the maximum and minimum values of elevation (z) for the cut-elements in the list. On this basis, we have defined a new parameter in our code:

$$\Delta H = \max(\text{abs}(\text{error})) \quad \text{for each element} \quad (2.10)$$

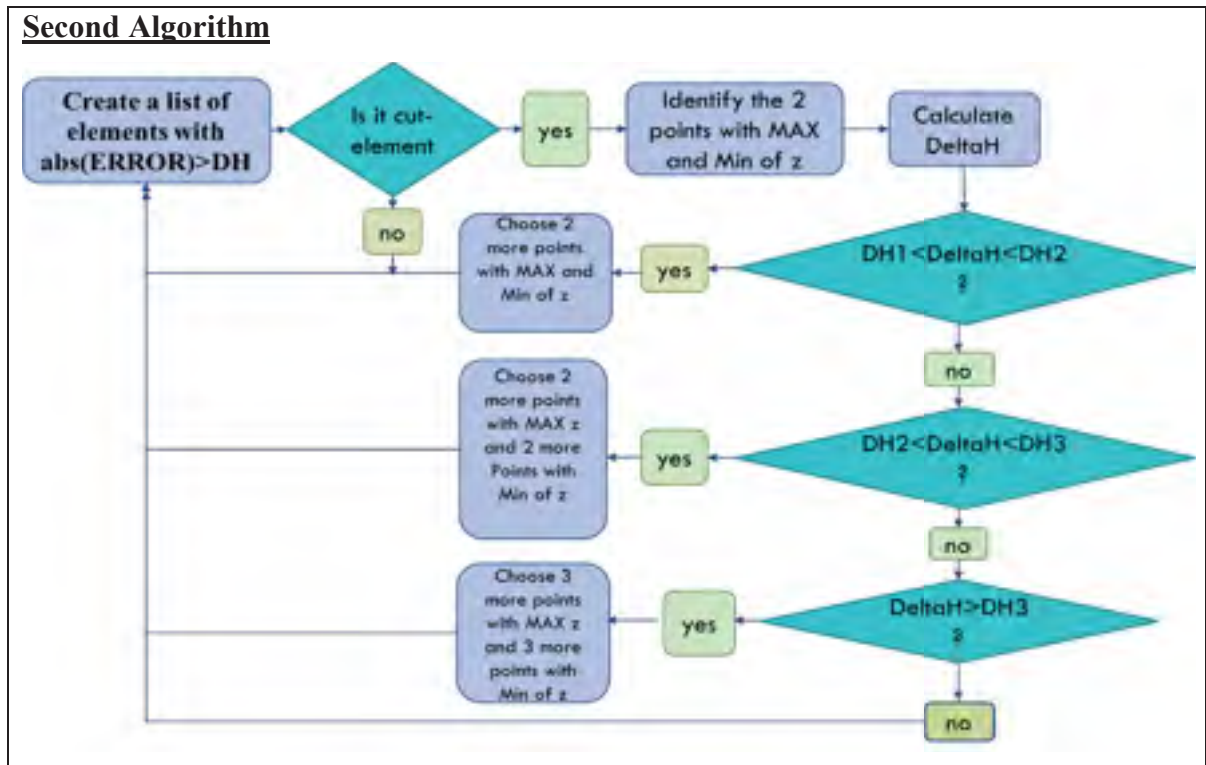


Figure 2-14 Mesh refinement with second algorithm.

The values of DH, DH1, DH2, DH3 can be defined in a “parameter.txt” file by the user along different inundation lines (shore line or inside the domain).

2.14.2 Serial mode of the Second Algorithm in TMA

As with the first algorithm, the second algorithm of TMA can be achieved by running the “TMA_Serial.py”, but in this case we should create a parameters.txt file that is related to the second algorithm. For more details, please see Appendix I.

2.14.3 Parallel Mode of TMA Second Algorithm

In this part, creating the text file for the parameters and modifying the weight factor are taken into account.

2.14.3.1 Creating parameter.txt file

The “parameter.txt” file for the second algorithm is similar to the one for the first algorithm except for some small changes:

- 1) Specify the name of an error file as “error_0_1.t3s”;
- 2) Consider a value for DH, DH1, DH2, DH3;
- 3) Choose whether or not you want to pick more points inside elements by writing “yes” in front of the last three lines.

In Figure 2.15 an example of a parameter.txt file is shown.



Figure 2-15 The word “yes” in the last three lines of a parameter.txt file would mean that the user wants to choose multiple points. If the user is satisfied with the results without choosing multiple points, he should write “No” in front of the last three lines.

Choosing multiple points enables us to get a better approximation of the cross-section. It also reduces the number of remeshings to obtain the final converged results.

To run the second algorithm in parallel mode, the following command should be written in the Terminal (Mac or Linux) or command line of Windows.

2.14.4 Modifying the weight factor for better convergence

Generally, in the first algorithm we consider the weight factor $alpha = 2$, but to calculate finer meshes in other parts of the domain (as when we work on a very small narrow stream), we can consider a large value for $alpha$ for further refinement. We suggest two approaches for estimating the value of α . The first is by drawing the inundation line with the highest value of water depth that passes through all the cut-elements. For example, if the contour line for the maximum water depth of $H_{contour} = 20 \text{ cm}$ across most of the cut-elements with a significant error (more than 10 cm), the weight factor will be:

$$\alpha = \frac{H_{contour}}{\varepsilon} = \frac{20}{5} = 4 \quad (2.11)$$

The priority is always to check whether or not the inundation lines pass through most of the cut-elements. It is possible sometimes to see that different contour lines for a specific region are almost the same. This means we should consider a larger weight factor for that region. Choosing the weight factor is by the user. The user should perform a sensitivity analysis on this parameter in order to fix the value to obtain finer and smoother meshes (by figuring out the high gradient of water depth as a result of mixture of fluid regime near the dikes or by comparing the cross section of channel in reality and the one exist in the numerical model).

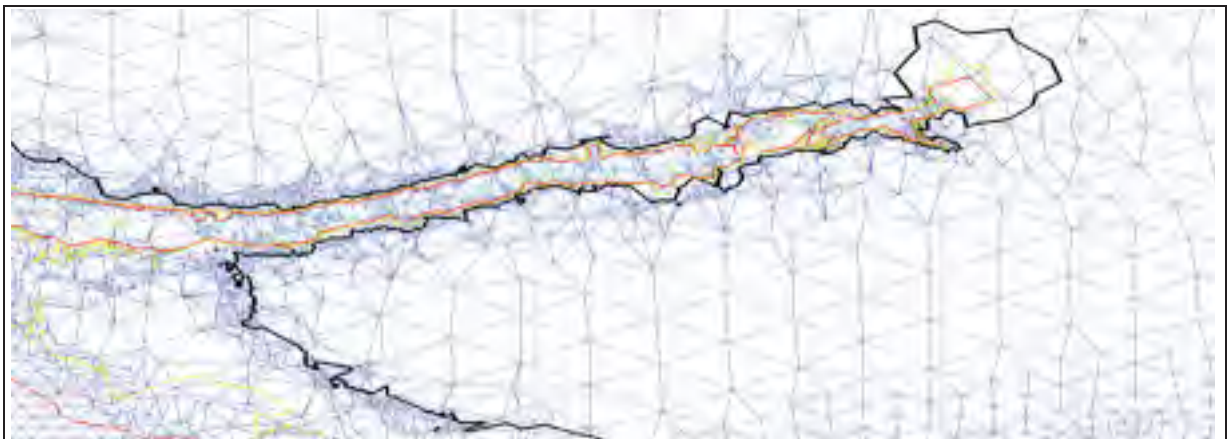


Figure 2-16 Superposition of different contour lines for maximum water depth on the error_6_7.t3s. Black line is a contour line for $H > 0.05$, Yellow line is a contour line for $H > 1$, and Red line is a contour line for $H > 2$.

As can be seen for the specific region of Rivière à la Pêche, the contour lines pass through most of the elements with a relatively high error rate (errors are shown by the coloured elements in figure 2.16) and reach to the end of the stream. Consequently, to do mesh refinement just in this area, we can consider the weight factor as equal to 40 ($2/0.05$) because the inundation line with the maximum value of 20 has passed through the cut-elements.

Another method for approximating the value of weight factor is based on the geometry of the cross-section. Imagine the channel has the following cross-section:

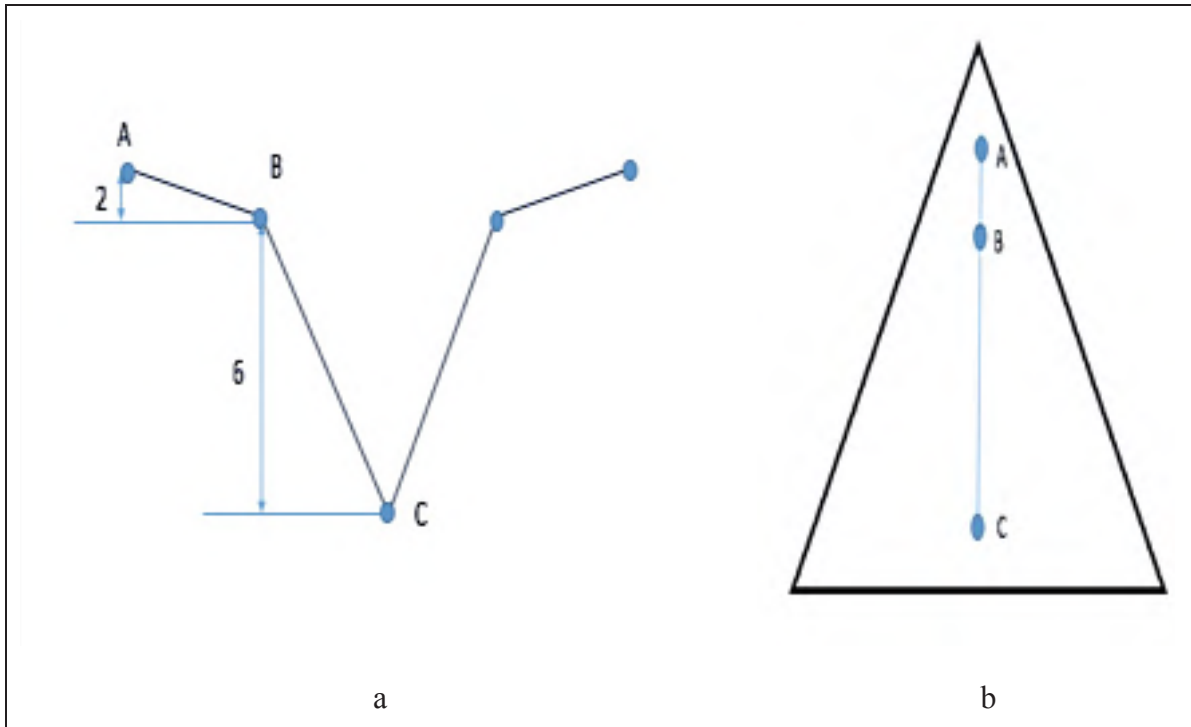


Figure 2-17 a) Cross-section of a channel with discontinuity in gradient of slope, b) Plane view of an element which is larger than the cross-section.

As can be seen in Figure 2.17, the slope of the cross section of the channel changes twice. If we consider the water depth at point A to B equal to zero, the maximum value at point B according to the shape of the cross-section would be 2 meters and at point C the maximum value of the water depth would be 8 meters. This means that on the basis of the cross-section the changes of water depth in a coarse mesh can be between 0 and 2 meters, 2 to 8 meters, or

even 0 to 8 meters in meshes with larger elements. The mesh refinements should be able to find point B and consider it for the next numerical model as we might have the mixture of fluid regime there. Therefore, in this case, considering the following weight factor can be helpful:

$$\frac{\text{maximum water depth Point B}}{\varepsilon} = \frac{2}{0.05} = 40 \quad (2.12)$$

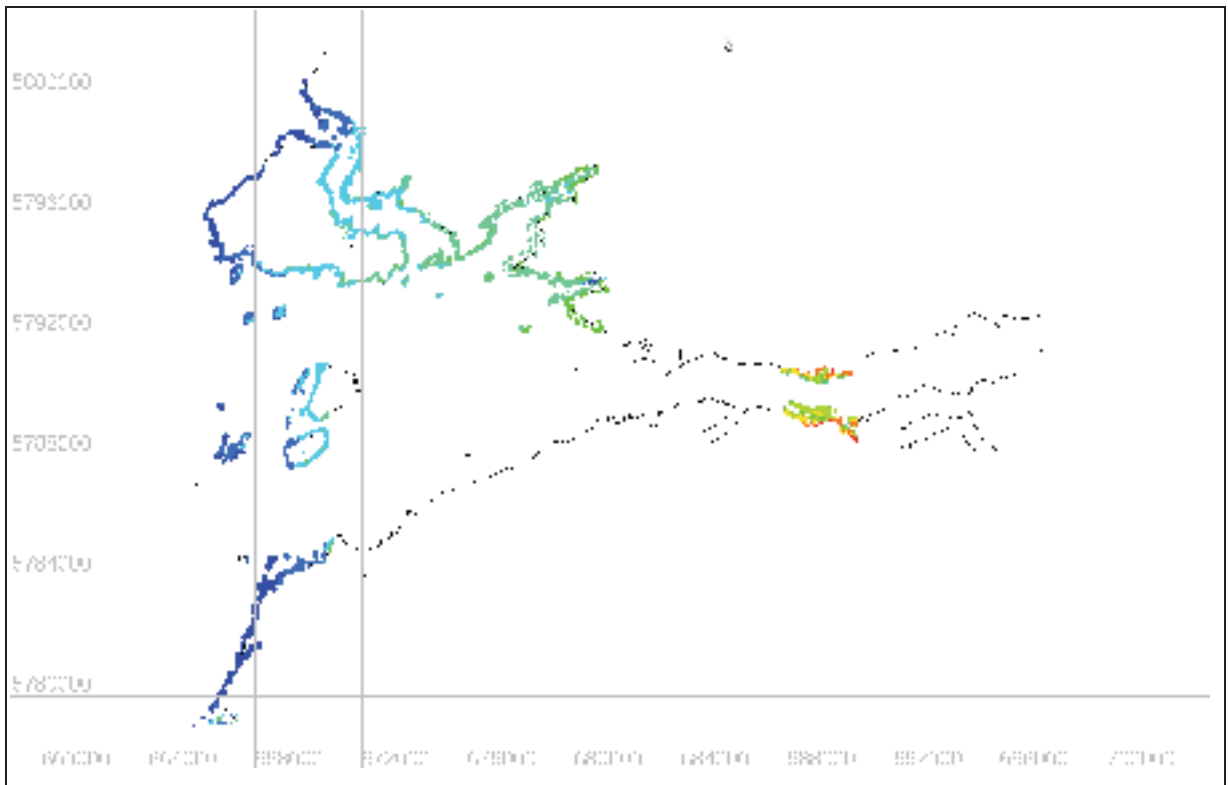


Figure 2-18 Location of new points after running the second algorithm of TMA. The projection of these data is UTM NAD83 Zone 33.

Two other versions of the second algorithm have been developed which do the mesh refinement by considering random points inside the cut-elements and choosing the points in the middle of the edge elements. For the first version, we attempted to use smaller spatial discretization in the cut-elements, but it was observed that the main effect is on the points with the minimum and maximum bottom elevation. Choosing the points in the middle of the edges was also not suitable as it might generate some hanging nodes that Blue Kenue would not be

able to ignore. Further explanations relating to these versions of the second algorithm are given in Appendix I.

2.15 Utilities for TMA

In the utilities part of the TMA, four code modules were programmed that can be used whenever deemed necessary. More details are accessible in the user guide of TMA in Appendix I. These codes are:

- **CollectNodes.py**
which can be used to collect the nodes of different mesh files generated by Blue Kenue;
- **MergeMesh.py**
To add or subtract from each other the nodes of different mesh files generated by Blue Kenue;
- **Divided_line_by_line.py** and **Divided_rectangular.py**
To be used as a cluster for partitioning LIDAR data before using TMA in the parallel mode.

CHAPTER 3

APPLICATIONS TO DAMBREAK FLOW STUDIES ON TWO RIVERS IN QUEBEC

In this chapter two test cases are presented. The first concerns a dambreak study on the Eastmain River (Quebec) and the second is on the Romain-Puyjalon River bouth in Quebec, Canada. The first test was used as a benchmark during the process of development and verification of the TMA code. The final results are comparable to those obtained previously by Hydro-Quebec but use a finer mesh along with LIDAR data. The second test case is used for the purpose of validation, i.e. the procedure learned from the first case is applied without any changes and the final results are compared to those also obtained previously by Hydro-Quebec but with the use of a very fine mesh.

3.1 Test Case 1: A dambreak study on the Eastmain River

The Eastmain-1 hydroelectric development was constructed on the Eastmain River upstream of the Opinaca reservoir in the James Bay region during the period 2002-2006. This project consisted of diverting a large portion of the flow from the Eastmain River to La Grande River through the Opinaca-Sarcelles reservoir. The project's available data are in the relative to the UTM NAD83 Zone 33 projection system (figure 3.1).

3.1.1 Topographic Data

LIDAR data, or light detection and ranging data, is a method used to measure the topographic data with pulsed laser light and to measure its reflection by sensors. The LIDAR data covers the last 25 km of the plain around the estuary of this river in the James Bay area (figure 3.2). These LIDAR data for the Eastmain River are available at two resolution level with constant distances of 20 meters and 5 meters following post-treatment. These data were prepared and validated by Hydro-Quebec.



Figure 3-1 Official map of Eastmain River.

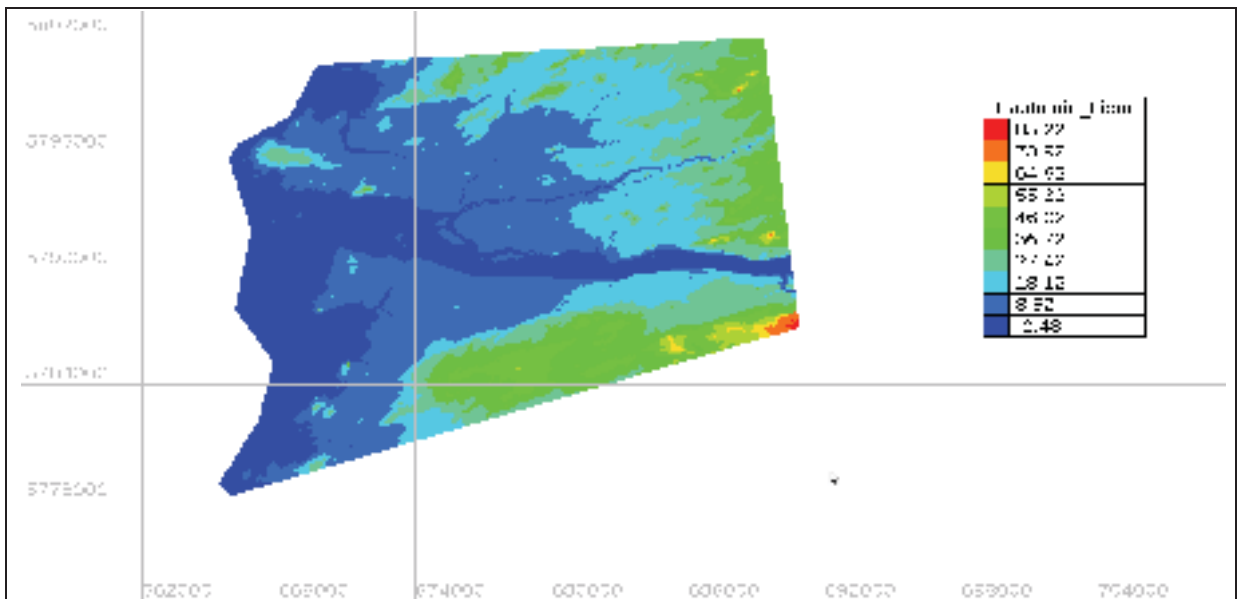


Figure 3-2 Example of LIDAR data for the Eastmain River illustrated in Blue Kenue.

3.1.2 Computational Domain

The computational domain of the Eastmain River is illustrated in figure 3.3. Topometric contour lines for this river are shown in figure 3.4. This dambreak study was done with a hydrograph discharge imposed at the inlet and constant water level of $H = 1.15$ m at the outlet (figure 3.5).

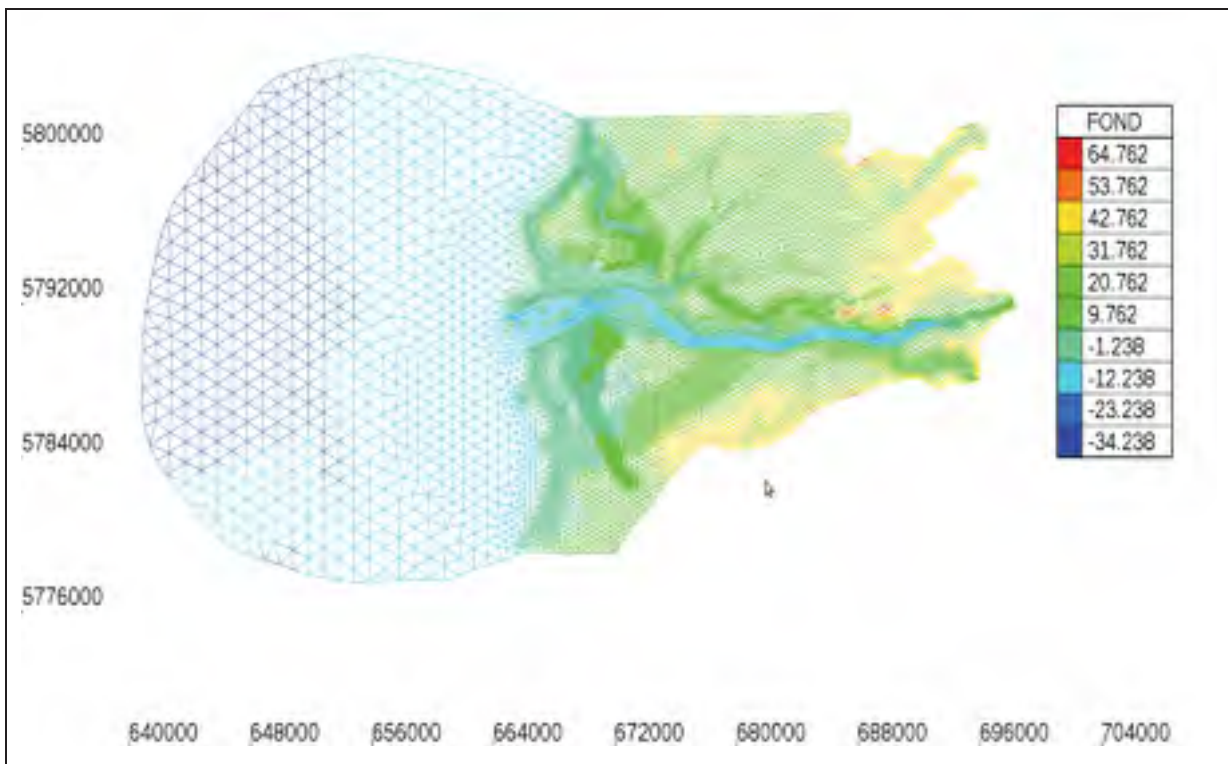


Figure 3-3 Terrain domain of the Eastmain River.

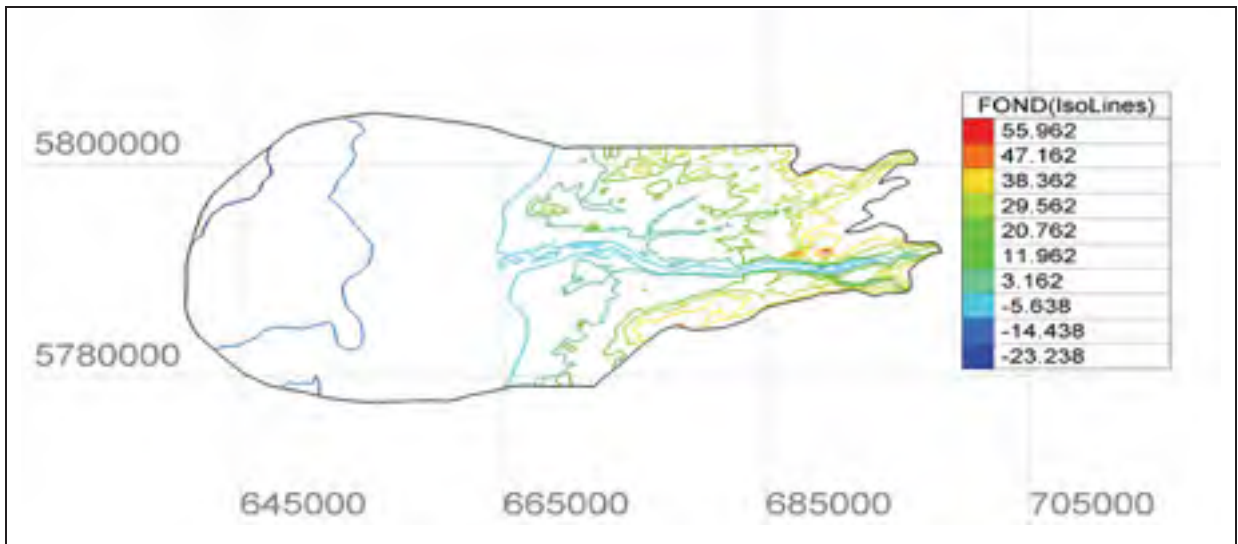


Figure 3-4 Contour lines for topographic data of the Eastmain River.

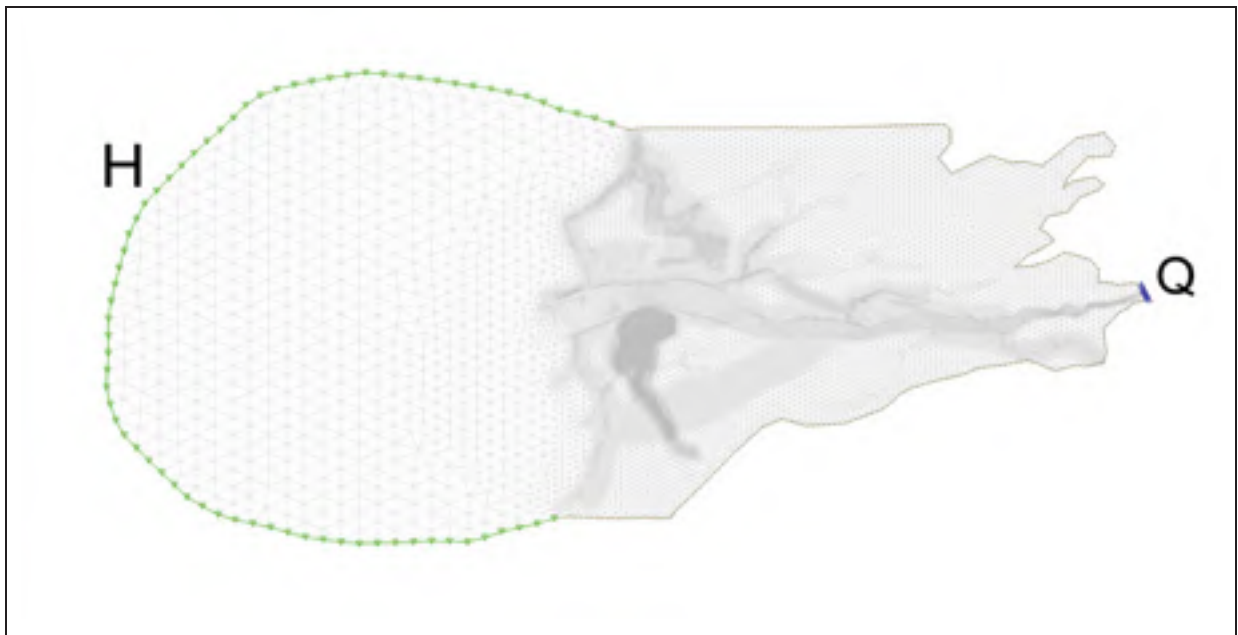


Figure 3-5 Boundary conditions of the Eastmain River.

In figure 3.6 the differences between the coarse and finer meshes around La Pêche River are shown. Figure 3.6.a is a coarse mesh and figure 3.6.b is a mesh refined by Hydro-Quebec “manually” around the La Pêche River. It is clear from figure 3.7 that for the la Pêche River the proper inundation limit wave was not well captured by the crude mesh.

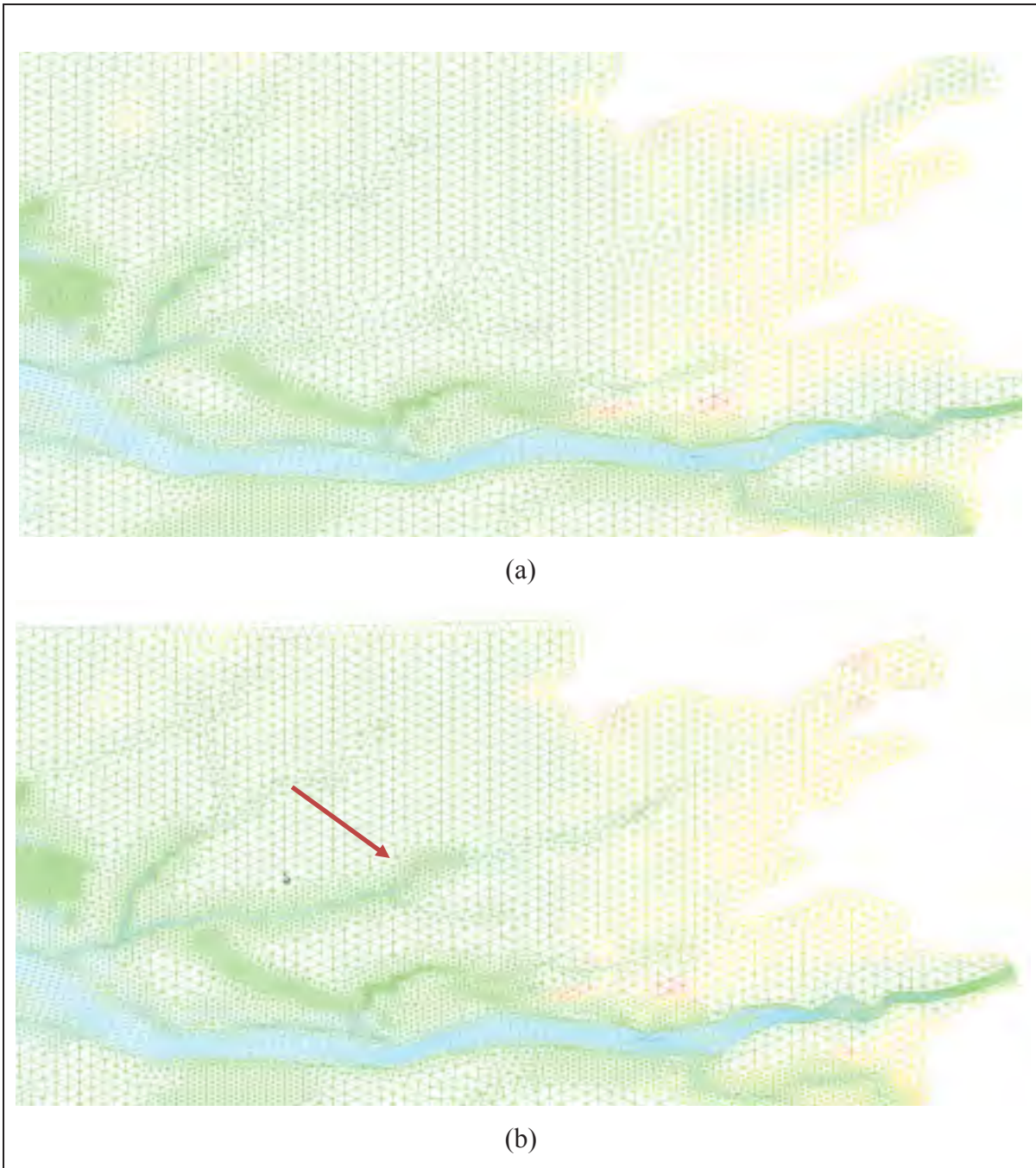


Figure 3-6 Difference in computational domain of the Eastmain River: a) coarse mesh, b) finer mesh with LIDAR data done “manually” by Hydro-Quebec. The arrow points to the La Pêche River.

Fig 3.7 shows the difference in the inundation limit of coarse mesh and finer mesh with LIDAR.

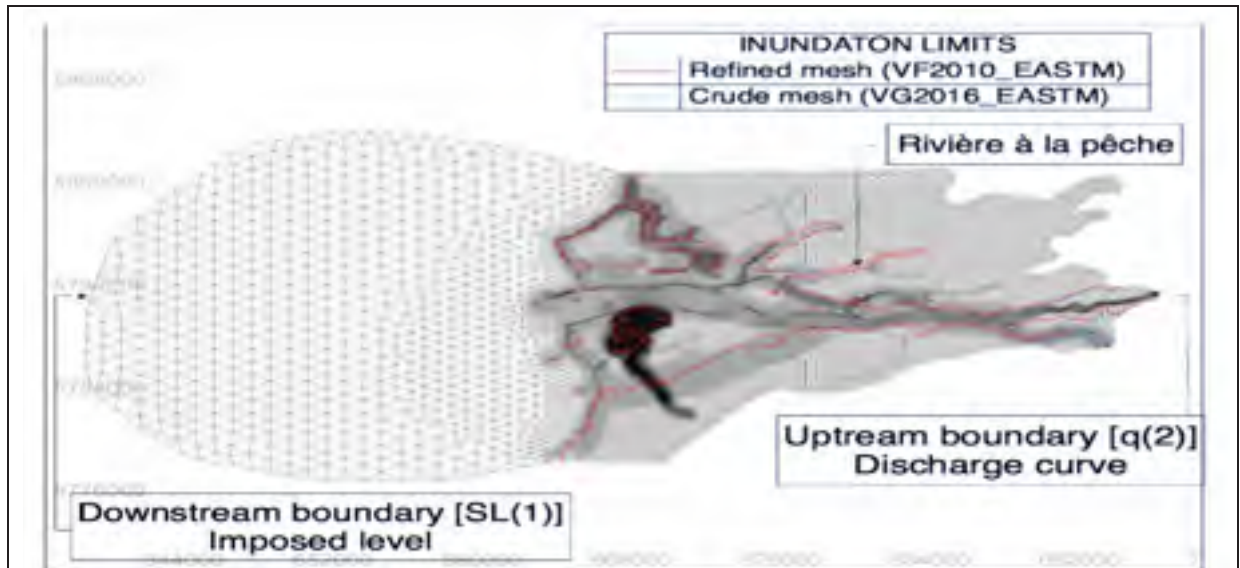


Figure 3-7 Difference in inundation lines using coarse and finer meshes (the fine mesh was done manually by Hydro-Quebec around La Pêche River).

3.1.3 Analysis

TMA was applied to do the mesh refinement for the computational domain. For the first step, TELEMAC ran the case file provided in chapter 2 with the initial conditions shown in figure 3.8. These initial conditions are related to the steady state of the river before the dam break occurs ($Q=150 \text{ m}^3$).

In this simulation, the water depth in dry zones is considered to be 1 cm, as the thin layer method was applied.

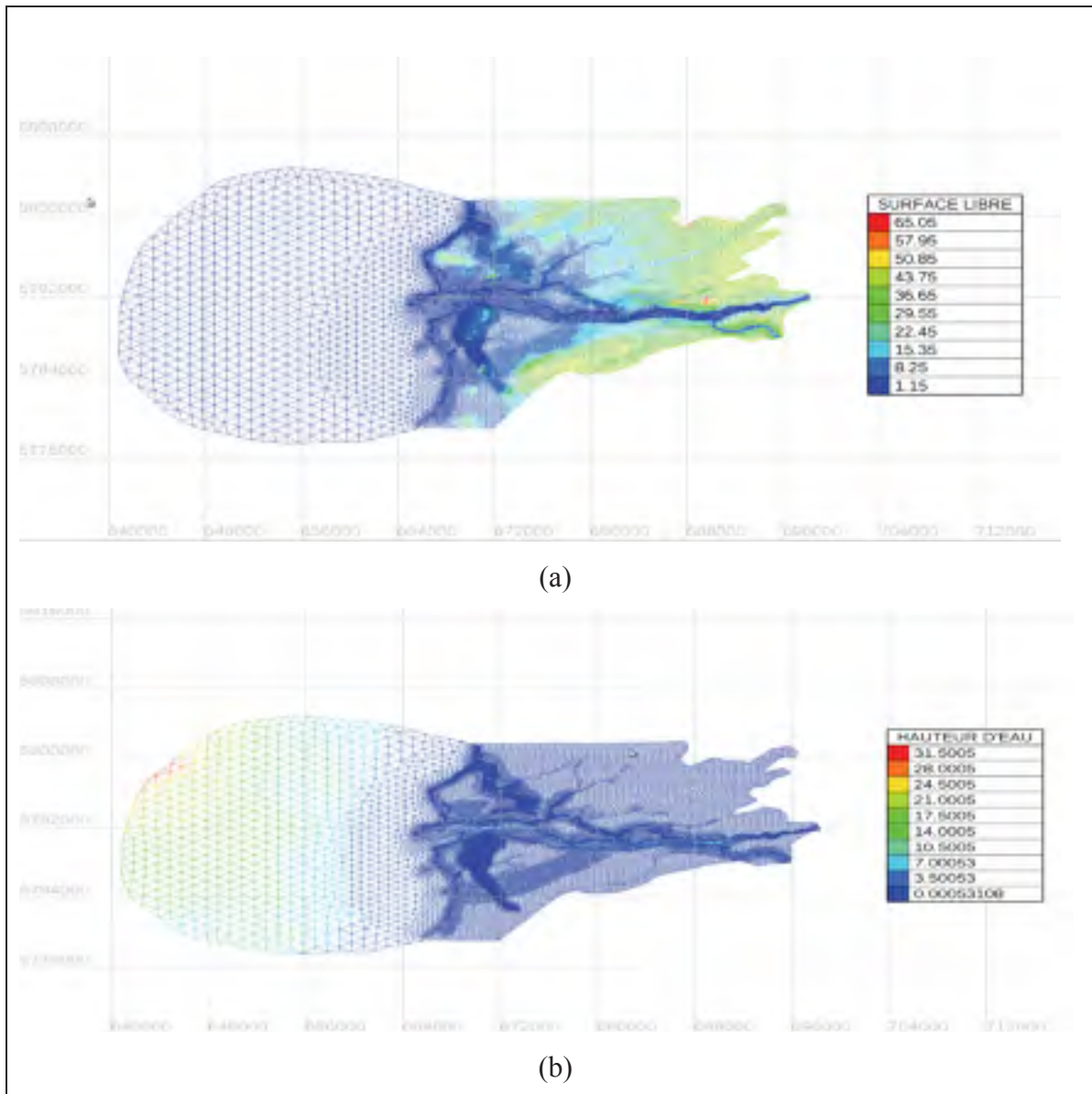


Figure 3-8 Illustration of a) free surface and b) water depth for the initial conditions of the Eastmain River.

The following steps in each mesh refinement were followed:

- As was explained in chapter 2, one of the output nodal variables of interest is $H_i = \max(H(i, t))$, which can be obtained by post-processing the results with Blue Kenue, where H is the water depth for node “ i ” at time “ t ”. The maximum is taken over the total time steps of the simulation (one day for the Eastmain River test case).

An element is considered partially wet if some (but not all) nodes of the element are dry and:

$$H^e > \varepsilon \quad (3.1)$$

where H^e is $\max(H_i)$ for the three nodes of each triangular element and ε is defined by the user. The default value for ε is 5 cm.

- Run TMA in order to do the first mesh refinement with the first algorithm for partially wet elements.
- For subsequence refinements, an error measure is defined as:

$$d_i = |H_i^n - H_i^{n+1}| \quad (3.2)$$

where H_i^n is the maximum water depth for node “i” and mesh “n”.

- An error for element (e) is defined as:

$$Err(e) = \max(d_i) \quad (3.3)$$

- A partially wet element is refined if $Err(e) > \alpha \varepsilon$. The factor α is then used to identify those elements that will be refined by adding at least two points corresponding to the maximum and minimum of the bed elevation.

3.1.4 Eastmain Refinement Results with TMA

In fig 3.9.a, the location of the new points that were employed for generation of a new mesh in figure 3.9.b is shown.

3.1.5 Change of Bathymetry with TMA

La Pêche River has a complex bathymetry. Therefore, a coarse mesh may lead to significant changes in the representation of the bathymetry. Changes of the bathymetry in La Pêche River with different refinements can be seen in the following figure. Figure 3.10 shows the bathymetry representation using both the coarse mesh and the first refined mesh.

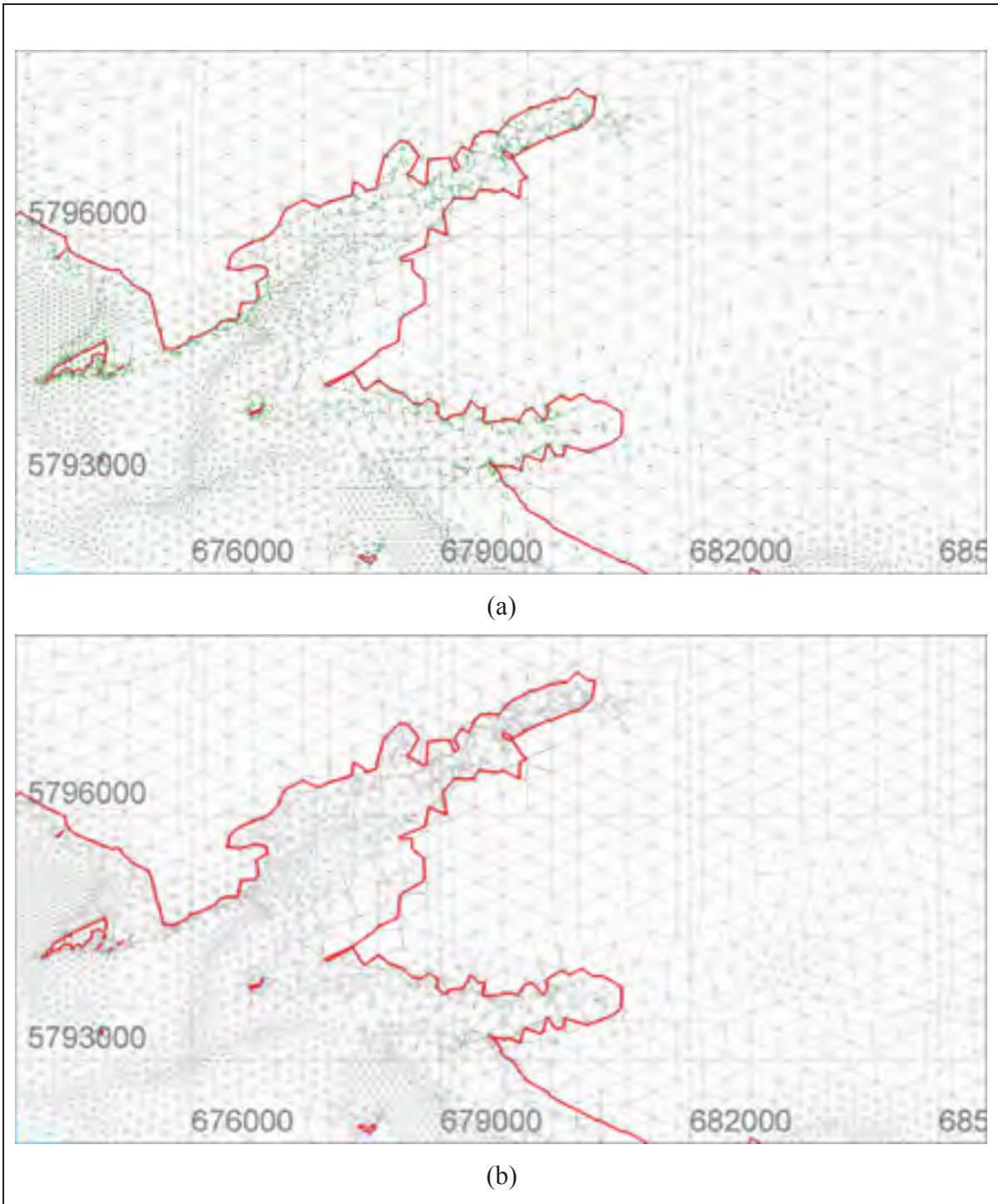


Figure 3-9 a) Location of new points in order to refine the cut-elements. These points are shown in green. b) Mesh refinement is based on new points around the inundation line.

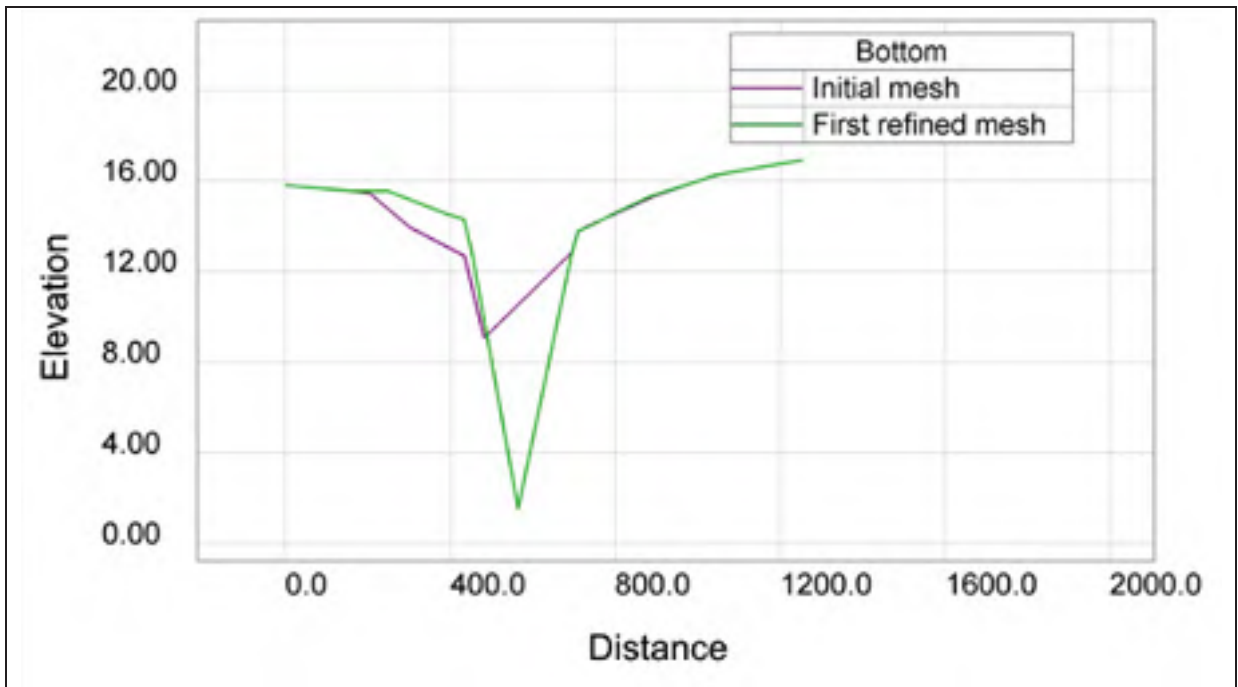


Figure 3-10 An example of change of bathymetry with different refinements.

Important remark: Mesh refinement not only controls the errors by reducing element size, but it also improves the bathymetry representation (i.e. bed slope and water depth, which are variables governing the shallow water) by projecting the new points onto the LIDAR map.

In Table 3.1, the number of elements that have been added to the initial mesh with each refinement can be calculated. This table illustrates that with fewer than 50,000 nodes we could obtain quite optimal results. After the 7th refinement, the results did not change significantly.

Table 3.1 Number of Elements and Nodes in different Meshes.

	Number of Nodes	Number of Elements	Epsilon(cm)	alpha	DH1 (m)	DH2 (m)	DH3 (m)
Initial Mesh	25323	50251	-	-	-	-	-
1st Refinement	31643	62891	5	2	-	-	-
2nd Refinement	38319	76241	5	2	1	5	15
3rd Refinement	39425	78507	5	20	1	5	15
4th Refinement	40795	81193	5	40	1	5	10
5th Refinement	46450	92500	5	100	1	2	5
6th Refinement	47278	94156	5	100	1	2	5
7th Refinement	48410	96420	5	2	1	2	3

3.1.6 Error File

In figure 3.11 the error d_i between the initial mesh and the first refined mesh is shown.

In the following figures, an example of error.t3s files that show a significant error along the progress of the inundation lines is illustrated. One or two points near the input boundary have large errors, but chiefly the error at the end of La Pêche River is around 10 meters in comparison to the previous simulation.

In figure 3.12 the error_5_6.t3s which contains the values of d_i is shown using the 5th refined mesh and the 6th refined mesh.

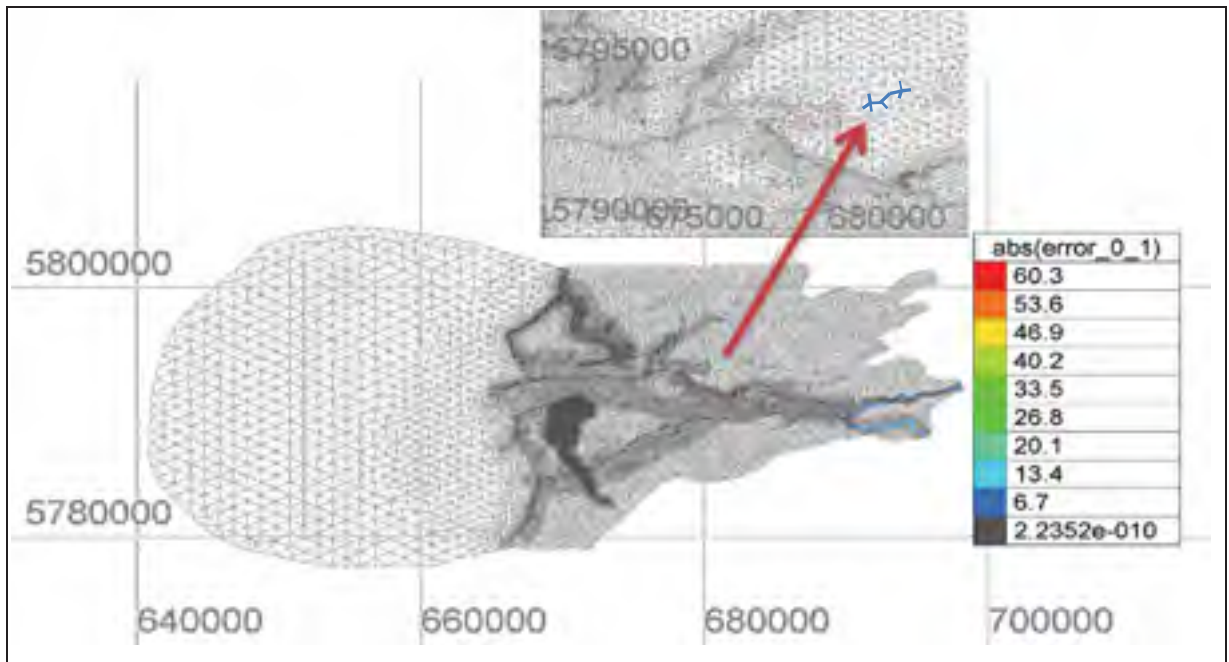


Figure 3-11 Error file generated by Blue Kenue for comparing the water depth of the first mesh and the initial mesh.

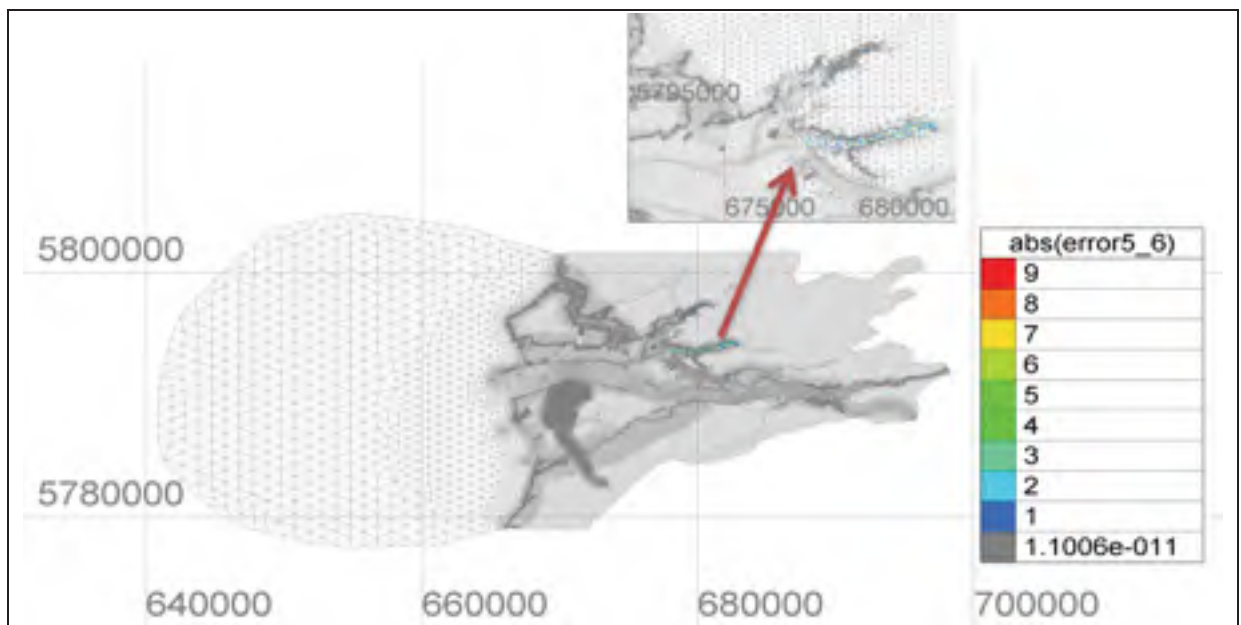


Figure 3-12 Error file generated by Blue Kenue for comparing the water depth of the 5th and 6th mesh refinements.

Generating error.t3s file helps to achieve a better understanding of where meshes need to be refined. For example, using error_0_1.t3s with the LIDAR data gives us the hint to modify the mesh along the La Pêche River. Figure 3-13 is an example of a mesh refined locally but “manually” using the errors indicator.

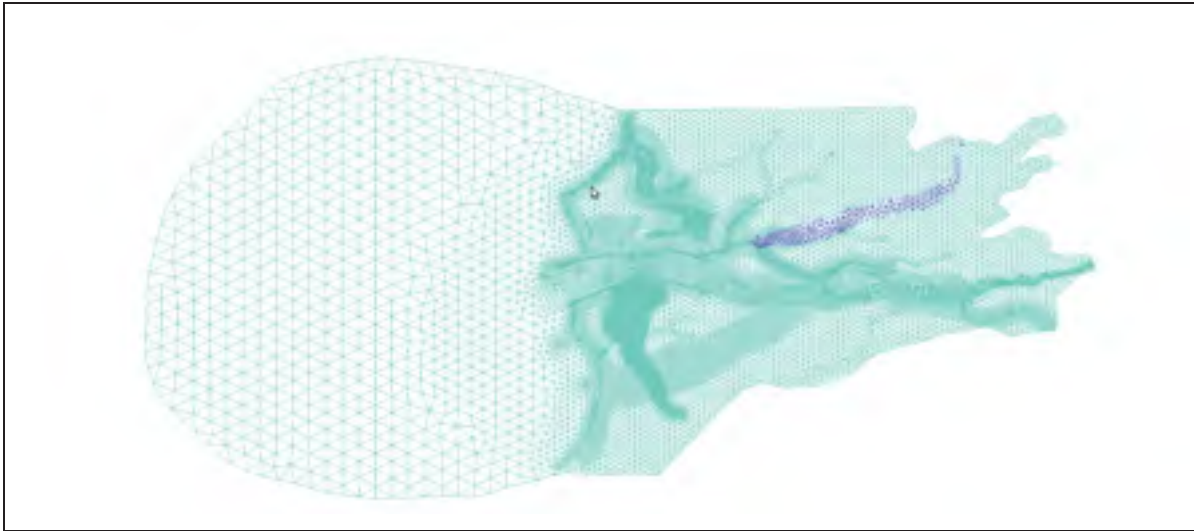


Figure 3-13 Manual mesh refinement by comparing the LIDAR data and error_0_1.t3s.

In TMA, the user, by employing error.t3s, can select the LIDAR data around the elements that have large errors to save computational time in running TELEMAC.

3.1.7 Improvement of Inundation Lines after Refinements

Figure 3.14 shows the improvement in inundation limits by using different mesh refinements. Mesh refinements were done along the shoreline to better consider the effect of the sidewalls of the channel in the dry zones and also to see whether or not the dikes that are in the way of the flow would be submerged.

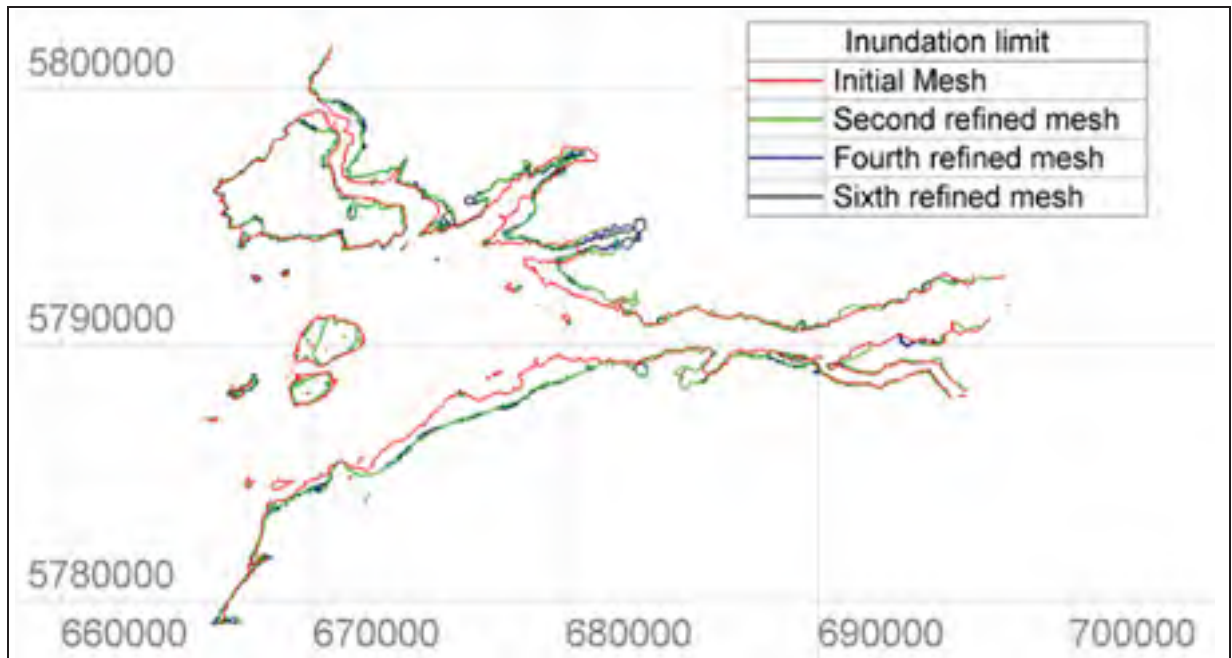


Figure 3-14 Improvement of the inundation lines with different refinements.

In fig 3-15 the inundation limit of the 7th refined mesh is illustrated with the topographic map.

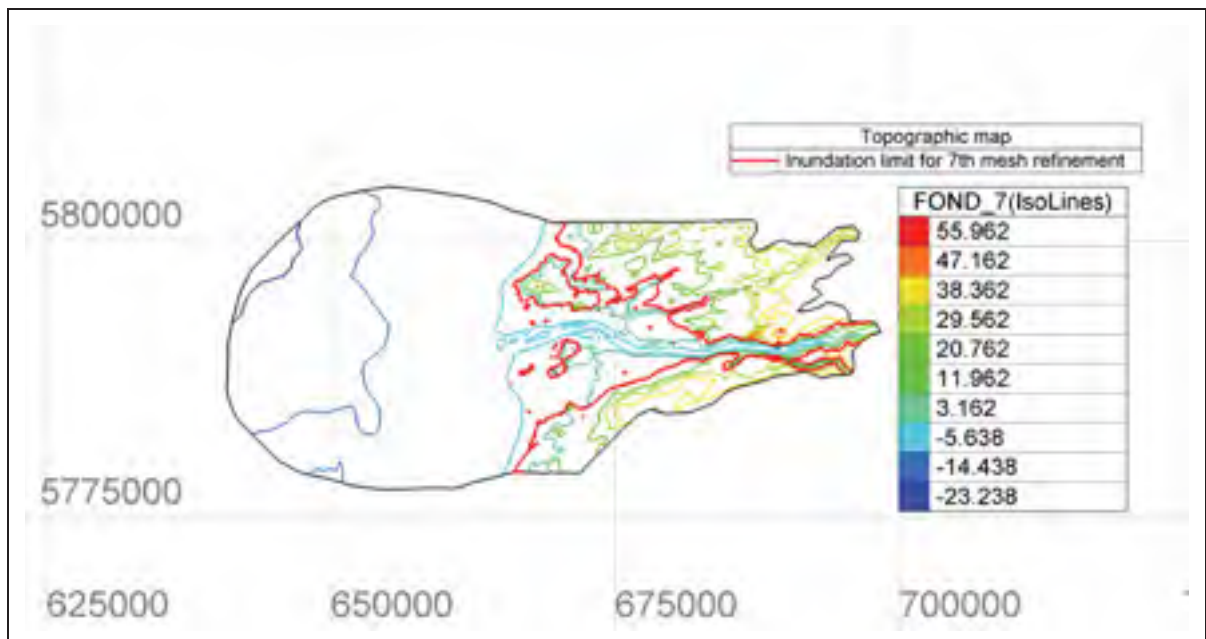


Figure 3-15 Visualization of inundation lines on topometric data.

3.1.8 Correction of Water Depth Representation Along the Shoreline

The effect of mesh refinement with TMA is shown in three regions (figure 3.16). In the initial mesh of region 3 it can be seen that the hydraulic structures, such as weirs or dikes, can act like a wall and prevent the dry zone from being affected by the water: however, after the mesh refinement, it is seen that the water can overflow the dike and move forward. In some other areas, large valleys in the way of the water could reduce the energy of the water, but after mesh refinement it can be seen that it was a small valley and that the water moved forward. Indeed, the mesh refinement in cut-elements helps the prediction of water that was stuck in the dry zone from exiting the outlet.

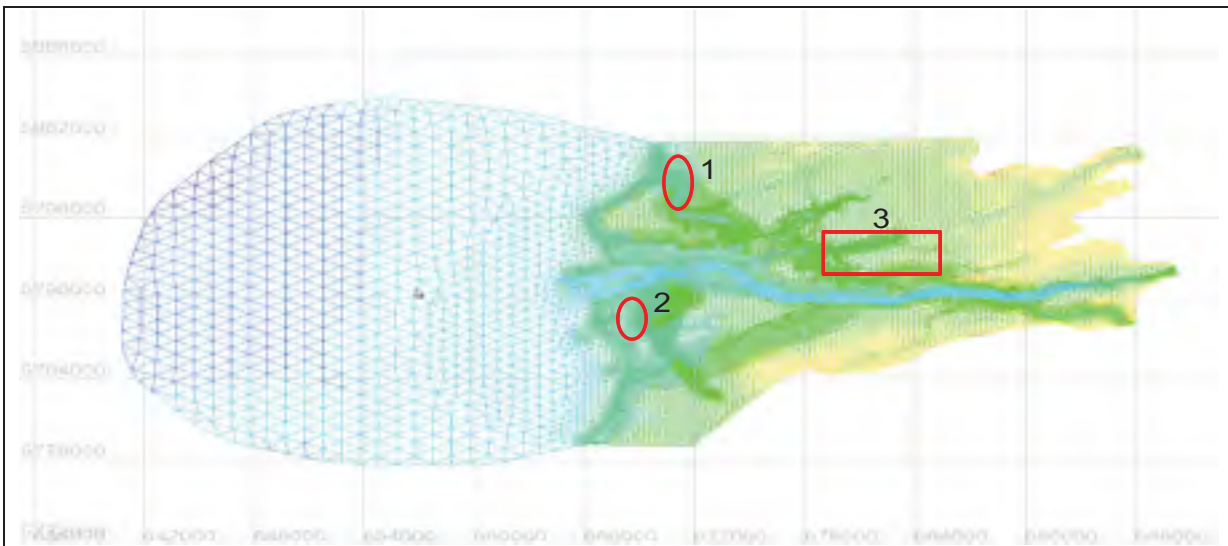


Figure 3-16 The change of the momentum equation in these regions was studied.

In the following figure the cross section of region 1 is studied, and it is understood that the steep bank causes the water level to increase. Refining the mesh along the shoreline helps the shallow water equation to better recognize the effect of water on the shoreline by considering a real representation of the topometric data.

After the mesh refinement, it can be seen that this valley was merely a result of uncertainty in the topometric data and was actually not a big valley at all. So, considering the correct slope for that element improved the simulation.

In figure 3-17, the increase of water depth level after mesh refinement is shown.

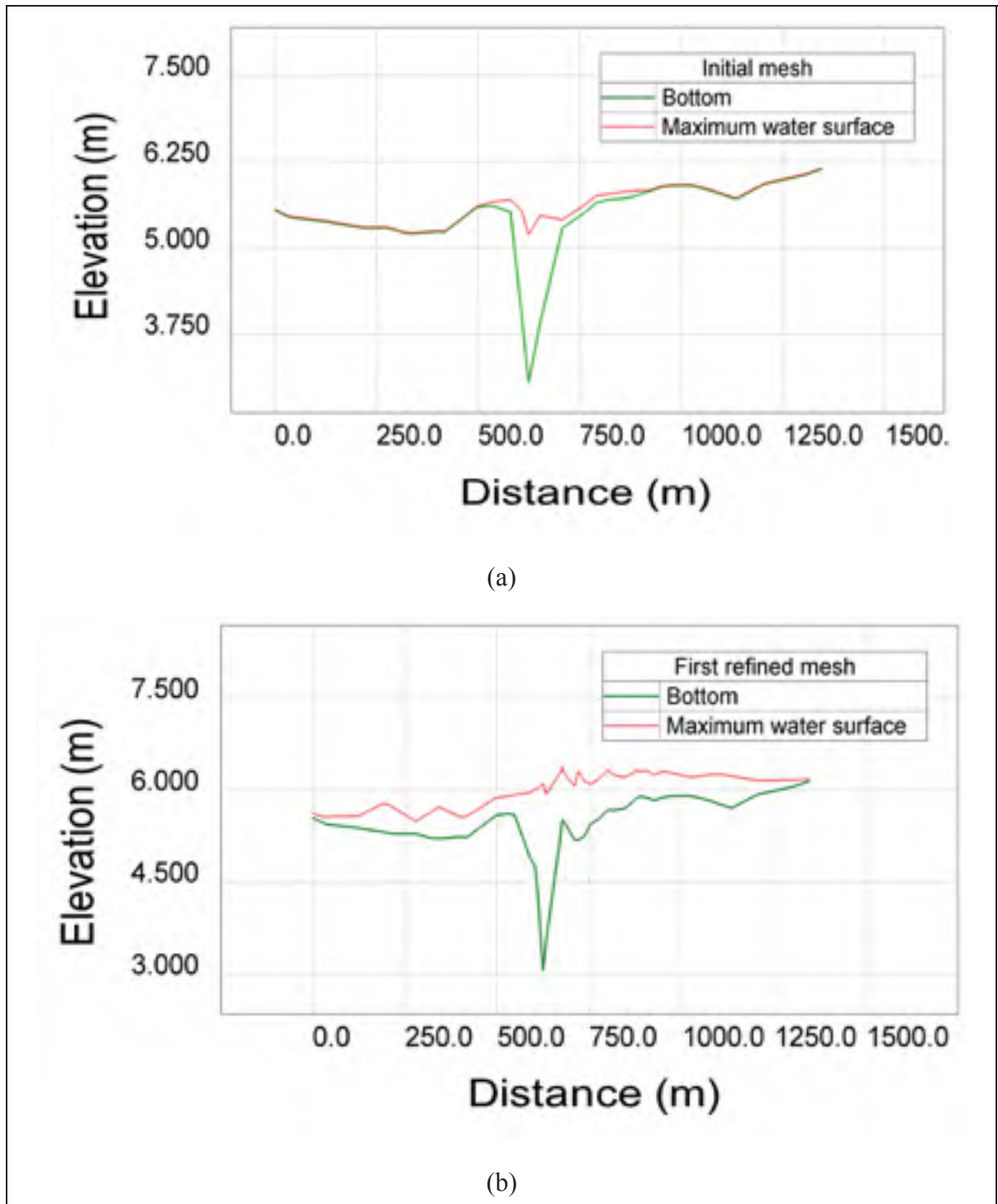


Figure 3-17 Change of the free surface max after the refinement with TMA in the cross-section of region 1: part a) before the mesh refinement; part b) after the mesh refinement.

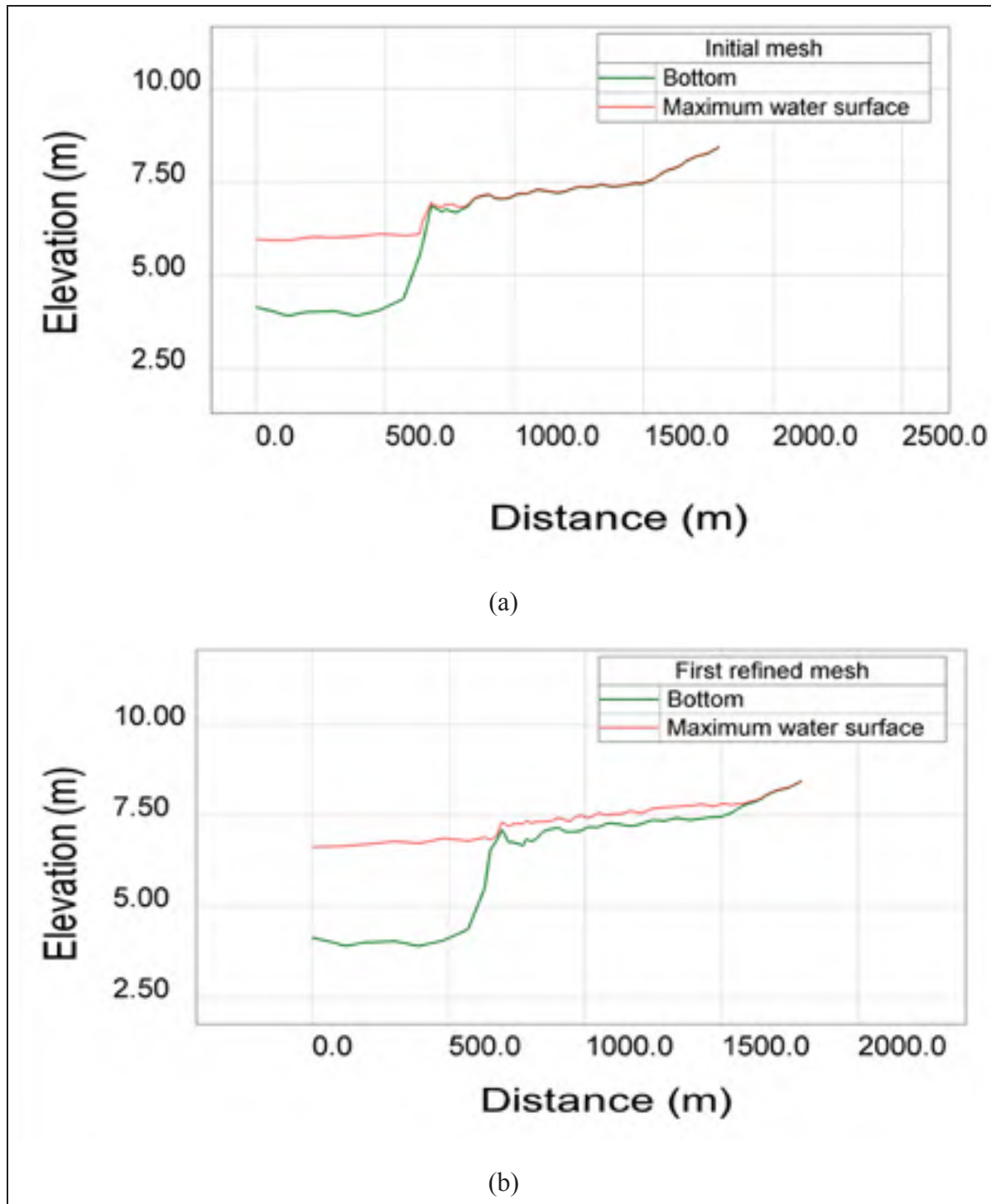


Figure 3-18 Change of the free surface after the refinement with TMA in the cross-section of region 2: part a) before the mesh refinement; part b) after the mesh refinement.

3.1.9 Modifying the Weight Factor

La Pêche River has a complex bathymetry. For this reason, having a finer mesh is necessary. That is why we should apply TMA several times.

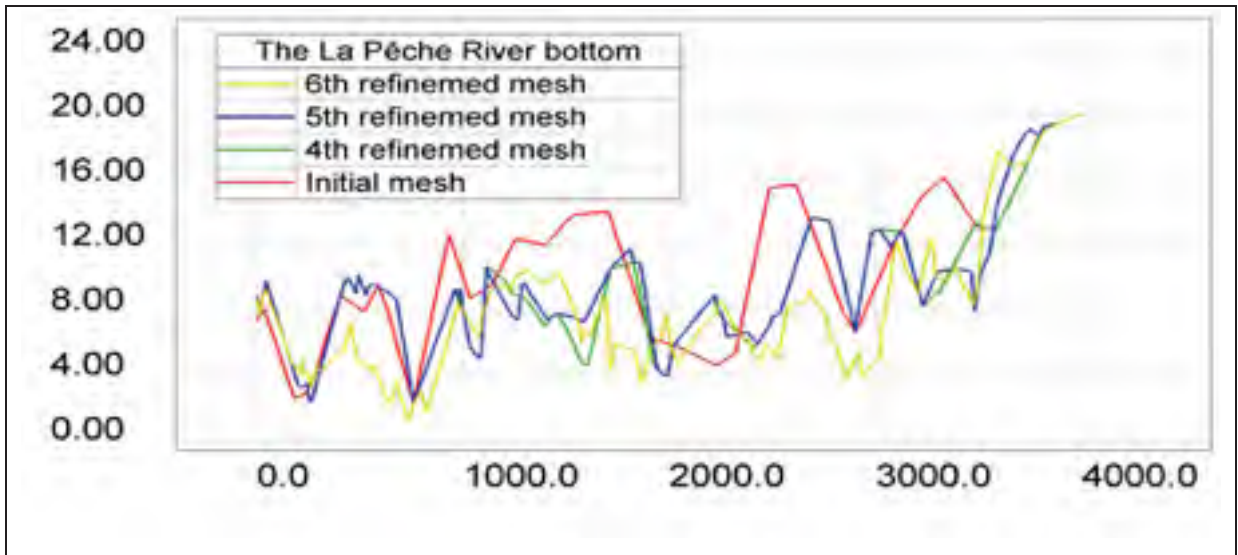


Figure 3-19 Change of La Pêche River bathymetry with the different refinements.

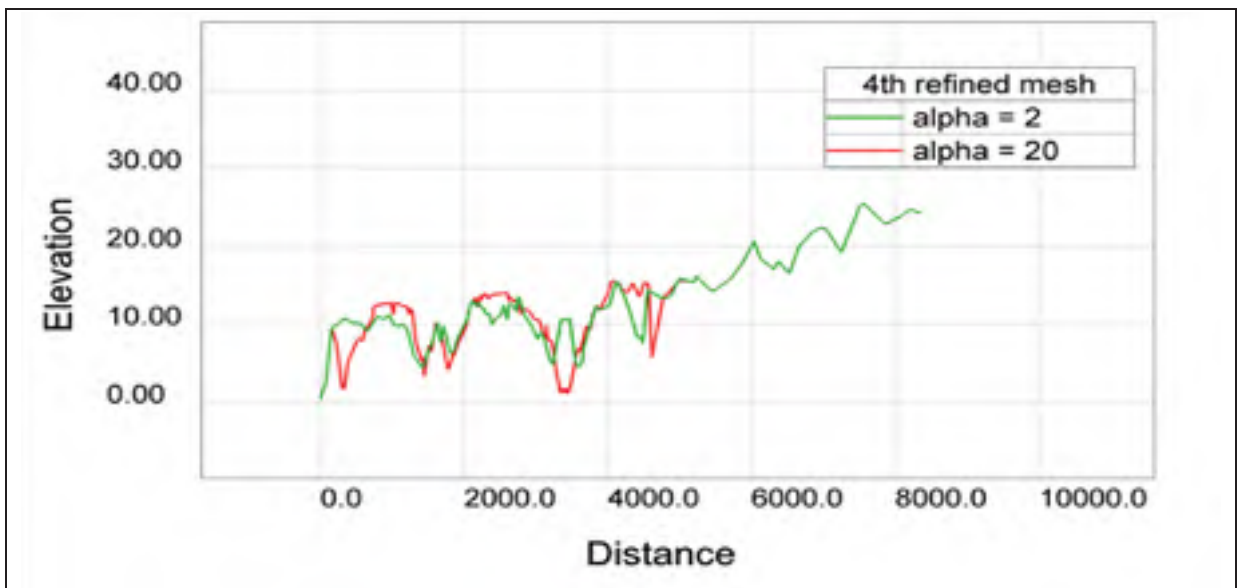


Figure 3-20 Difference in altitude of La Pêche River in the 4th refinement with the different weight factors.

Considering a higher weight factor can help us to get the next secondary valleys representation sooner. Choosing the next secondary valleys is possible with the option provided in the TMA. When the weight factor is small these points are near each other, and usually there is not a significant change in the geometry of the next mesh; however, when a higher weight factor is considered, these changes are better observed.

The idea of considering a higher weight factor was also used for the elements where there might be a change in the flow regime. However, as these elements are not cut-elements, the tolerance between the maximum water depth of the nodes of the element might be too high (i.e. the water depth might vary between 6 meters and 0.5 meters in these elements).

In the following figure (figure 3-21), we try to show how considering a higher weight factor can help the simulation to reduce the uncertainty produced by the topometric data. As can be seen, having a finer mesh helps the numerical model to have a better approximation of the cross-section. Narrow channels can produce a standing wave, which can pass the flow from the high elevated bumps in its way.

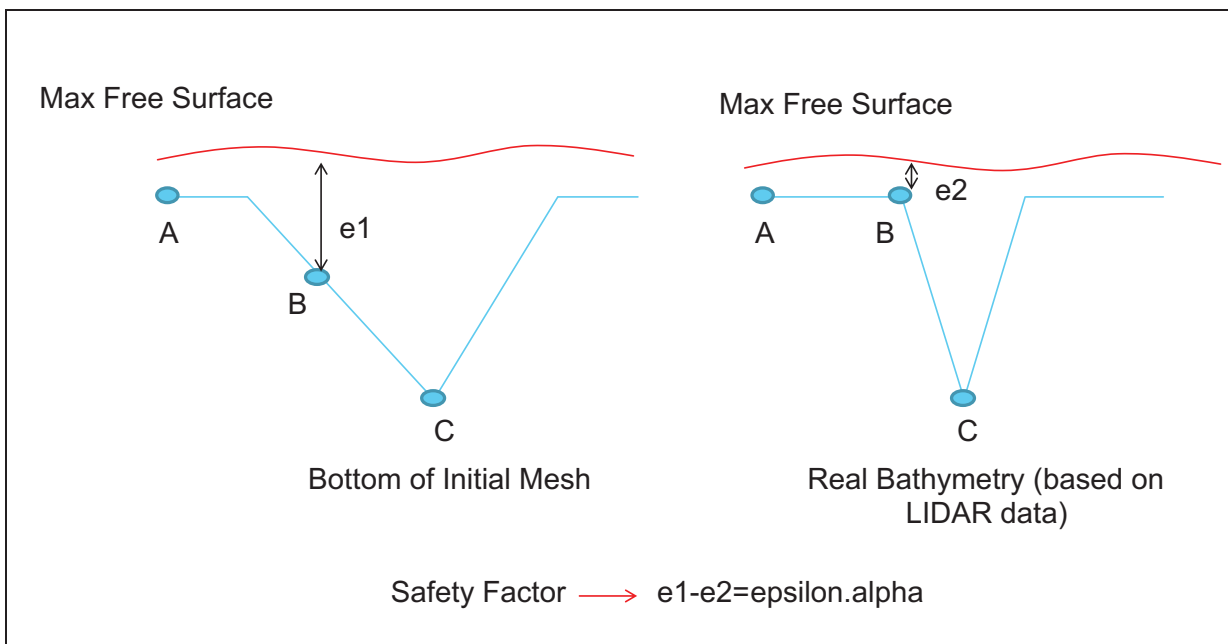


Figure 3-21 Because of the uncertainty in the geometry, approximation of the elevation for point B created a wide channel, but in reality it is a narrow channel.

It can be observed that the inundation line for the 4th and 5th refinement with $\alpha=2$ did not show any change. As the cross-section of La Pêche River in the middle of the stream is narrower than the cross-section at the beginning of it, a higher weight factor is considered ($\alpha = 40$). A narrow cross-section produces standing waves; thus, the water depth increases and the water can pass the high elevated bump that was in its way. Considering a higher weight factor adds more points for water projection, which help us to have fewer mesh refinements. But using a correct weight factor requires a good understanding of the domain under study.

As shown in figure 3.22, 3.23, 3.24, 3.25, after each refinement the water surface value becomes smoother and that makes the momentum equations easier to solve. The maximum free surface and the bottom of La Pêche River at the end of the inundation line for different mesh refinements are shown in these figures.

Table 3-2 also compares the volumes of water which entered the domain, exited the domain, or got lost during the simulation for different refinements.

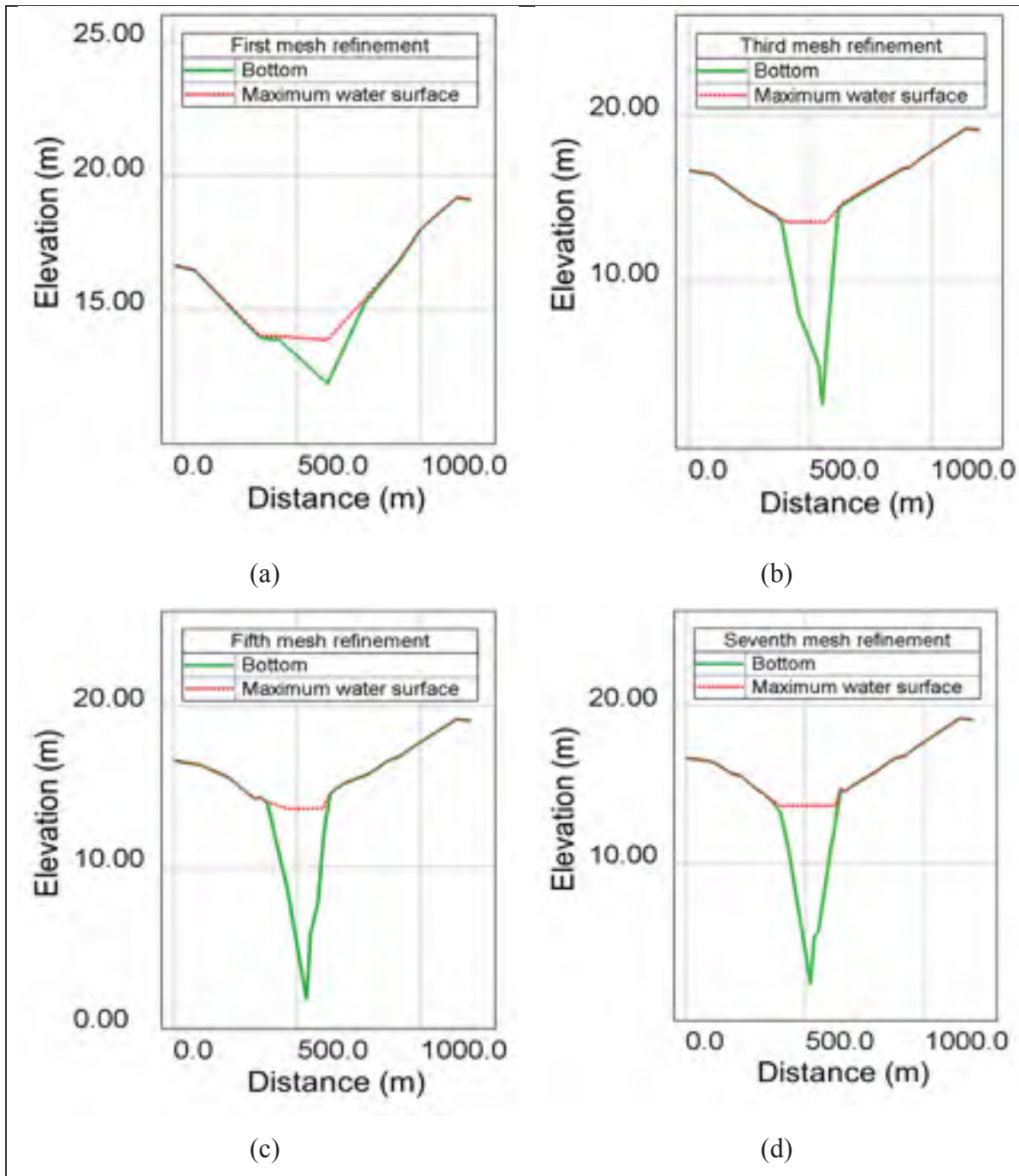


Figure 3-22 Change of the bottom and the maximum free surface after mesh refinement with TMA at the beginning of La Pêche River in the first and third mesh refinements: a) the first mesh refinement; b) the third mesh refinement.

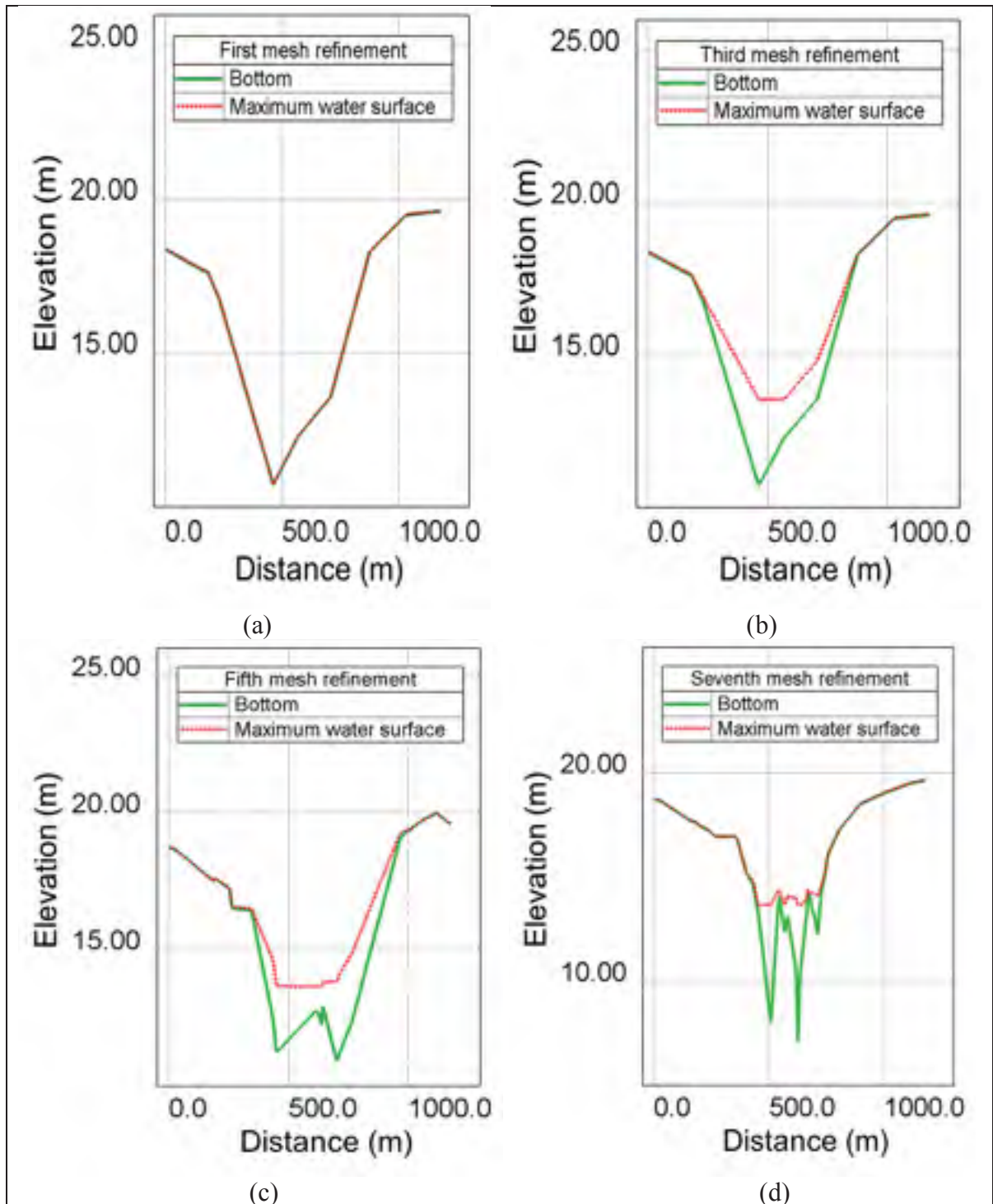


Figure 3-23 Change of the bottom and the maximum free surface at one-third inundation of La Pêche River: a) the first mesh refinement; b) the third mesh refinement; c) the fifth mesh refinement; d) the seventh mesh refinement.

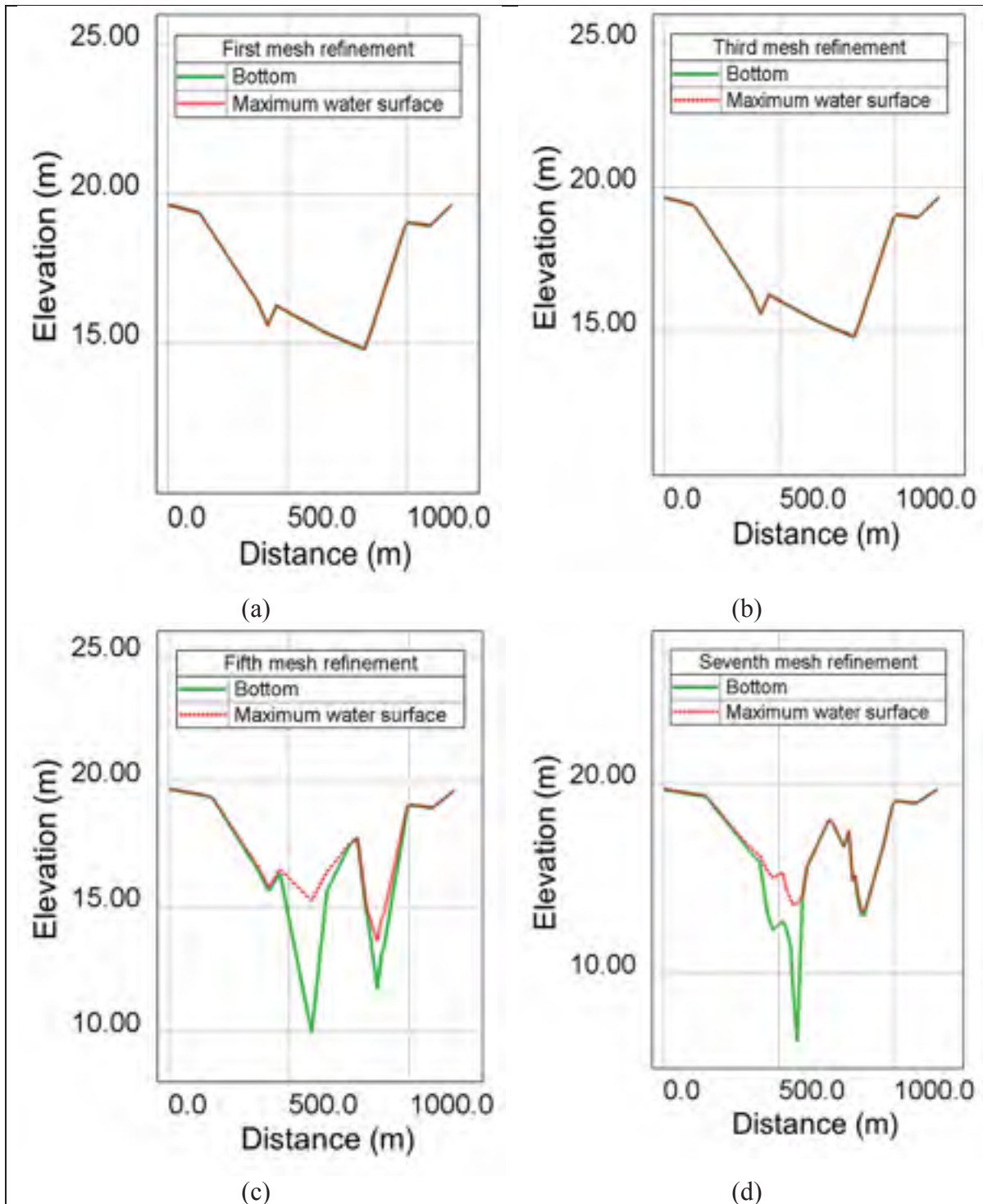


Figure 3-24 Change of the bottom and the maximum free surface two-thirds inundation of La Pêche River in the different refinements: a) the first mesh refinement; b) the third mesh refinement; c) the fifth mesh refinement; d) the seventh mesh refinement. As can be seen, up until the third refinement, this section was dry.

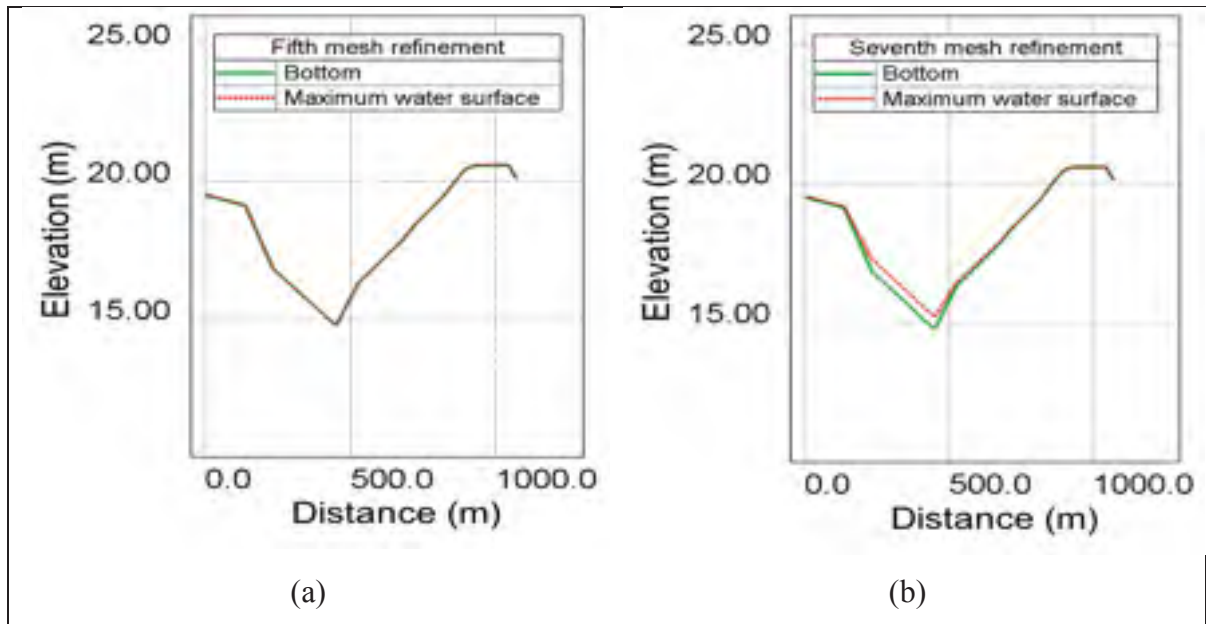


Figure 3-25 Change of the bottom and the maximum free surface at the end of La Pêche River in the different refinements: a) the fifth mesh refinement; b) the seventh mesh refinement. As can be seen, until the fifth refinement, this section was dry.

Table 3.2 Comparison of the entering volume and exiting volume in the simulation with the different meshes.

	Initial Volume M^3	Total Volume M^3	Entering Volume M^3	Exiting Volume M^3	Error Relative (Volume conservation)
1 st Refinement	0.733823E+10	0.7793070E+10	0.501837E+09	0.47006E+08	0.603155E-02
3 rd Refinement	0.733088E+10	0.7785185E+10	0.505787E+09	0.51590E+08	0.662675E-02
5 th Refinement	0.732724E+10	0.7780788E+10	0.517513E+09	0.63971E+08	0.822172E-02
7 th Refinement	0.732816E+10	0.7781320E+10	0.5101057E+9	0.56948E+08	0.731861E-02

As can be seen, in all of these meshes the relative error in the volume conservation is less than 0.01 percent. Mesh refinement with TMA for varied flows made a big improvement in the exit

volume. But when the flow captured further complex zones, the lost in volume increased, which point to the fact that the water was stacked between the very sharp valleys at the end of the stream where the velocity is small.

3.1.10 Conclusion

In this section, it is shown how TMA was applied to dambreak analysis on the Eastmain River. In this case, the boundary condition considered was the hydrograph discharge at the inlet and a constant free surface at the outlet. The adaptive meshing methodology was applied using LIDAR data after post-processing the simulation results. For this case, modifying the slope near the shoreline by using the adaptive meshing methodology based on topometric data was employed. This method helps to analyze the flow in the cut-elements and in the interfaces where the velocity is very small.

3.2 Test Case 2: Romaine-Puyjalon River

The Romaine River project was developed by Hydro-Québec for hydroelectric power generation. Construction started in 2009 on a new hydroelectric plant, along with four rock-filled dams and a 150 kilometer (93 mile) long access road. It took 11 years to build at an estimated cost of CAN \$6.5 billion. The LIDAR data for this case are expressed in geographic system Quebec MTM NAD83 zone 5 (figure 3.26).

Another study was done for the Romaine-Puyjalon river. For this case, we tried to do the mesh refinement at the inundation line in the dry zones and where there were valleys in the way of the fluid. In this case, the LIDAR data were available in small partitions and gave us the option to do the mesh refinement wherever necessary. Figure 3.27 shows the initial conditions. In figure 3.28 the location of imposed boundary conditions is shown; and in figure 3.29 to 3.31, where the mesh refinement has been applied, the changes in inundation limits are shown.



Figure 3-26 Map of Romaine-Puyjalon River, domain of interest.

3.2.1 Boundary Condition and Initial Condition of the Romaine-Puyjalon River

In the following figures the initial condition and the boundary condition of the Romaine-Puyjalon River are shown. The study of the Romaine-Puyjalon River was done with two different initial conditions.

For this new case, the friction coefficient near the shoreline was modified to better take into account the effect of the side walls of the river.

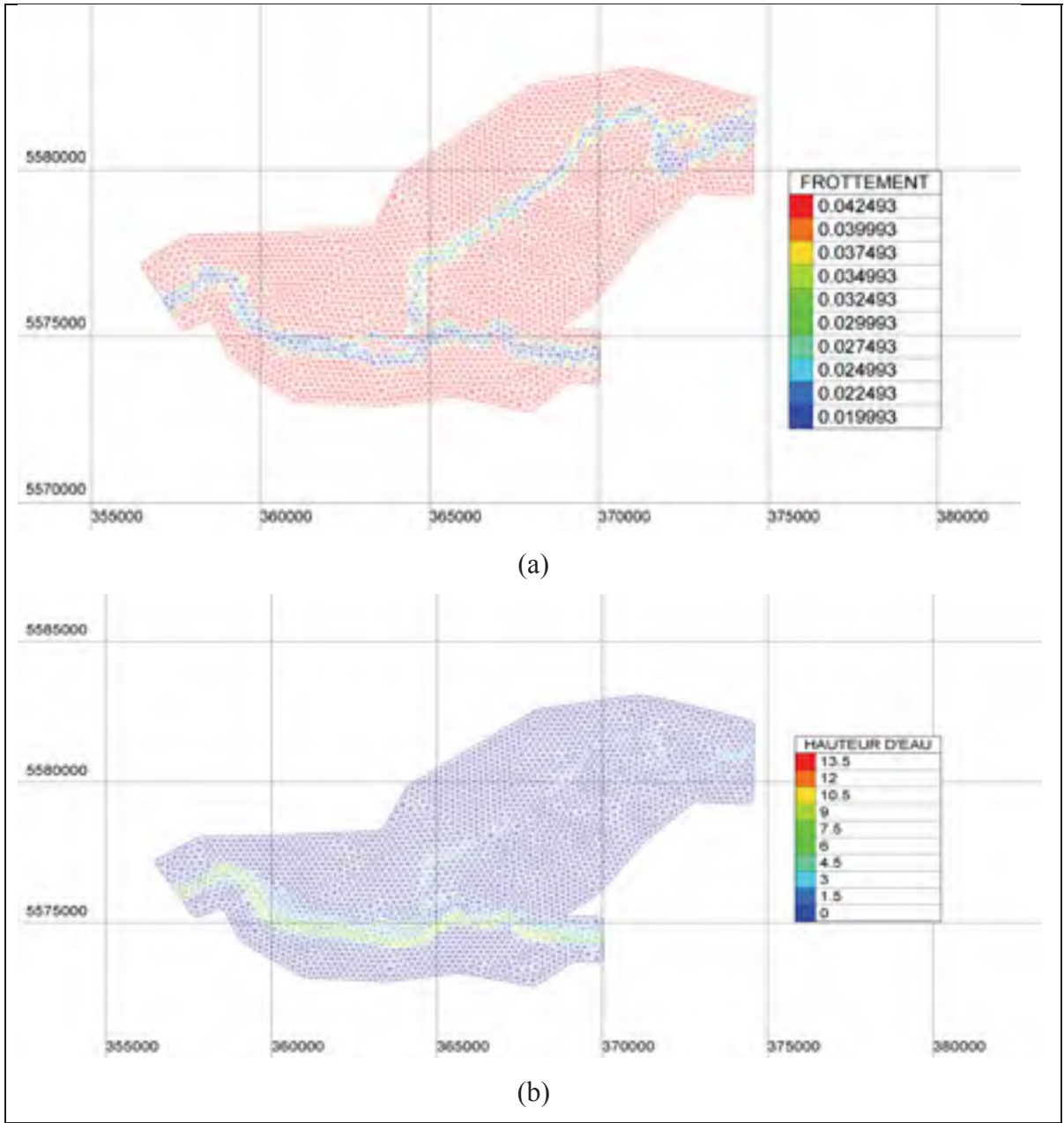


Figure 3-27 Initial conditions of the Romaine-Puyjalon River (first version).

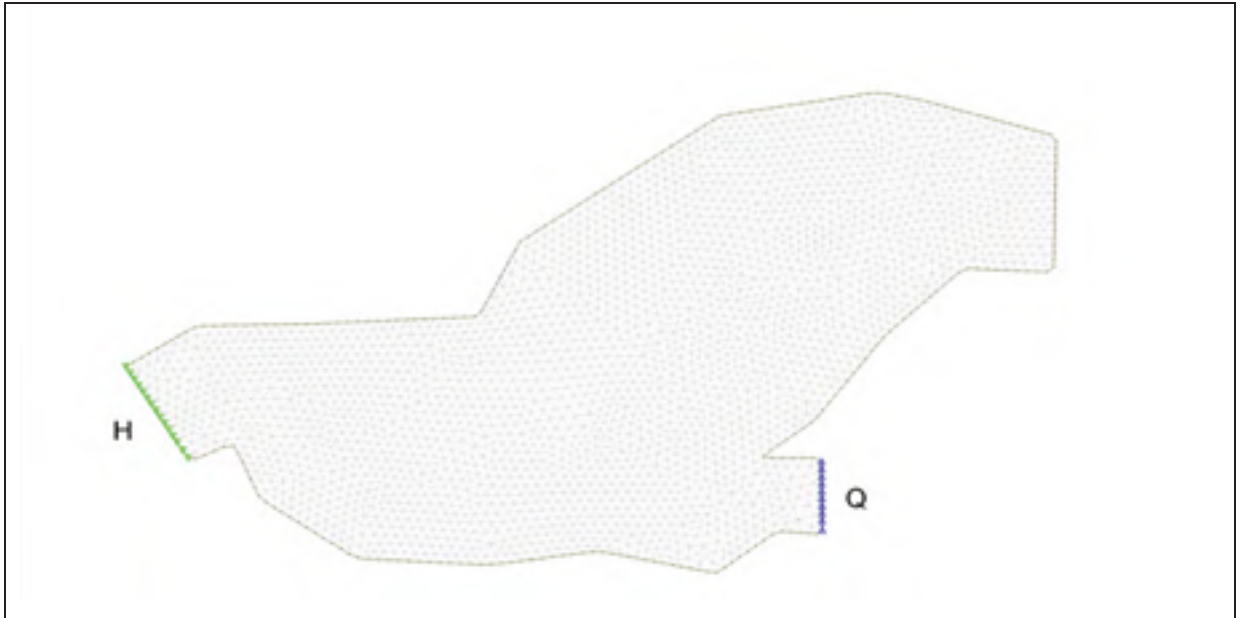


Figure 3-28 Imposed boundary conditions for the Romaine- Puyjalon River domain.

After generating the initial condition file and the boundary condition file, the case file shown in figure 2.8 was employed to run the TELEMAC for the Romaine-Puyjalon River.

The dam break hydrograph hydrograph for discharge is shown in fig 3.29. The water level in outlet also changed over time.

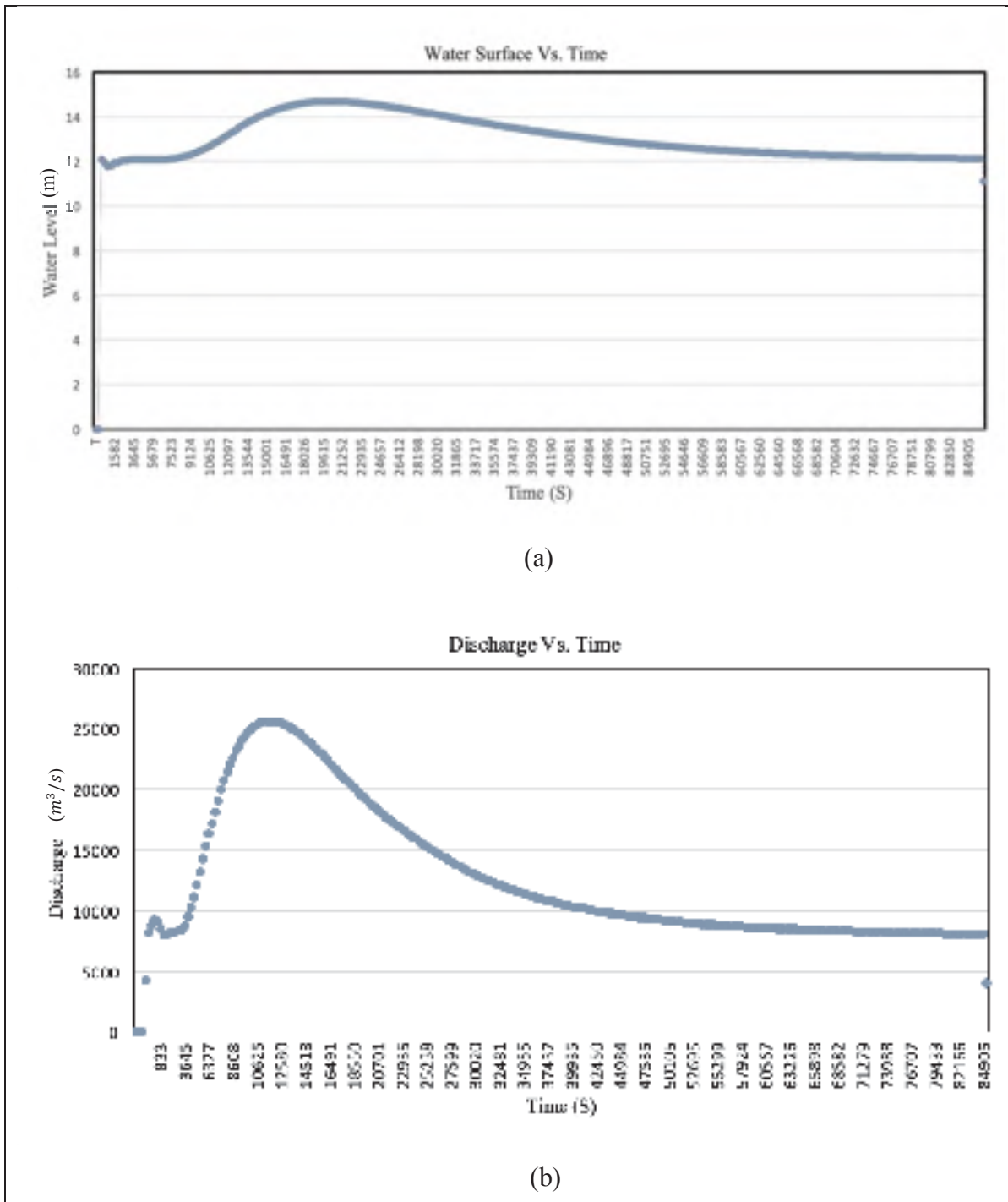


Figure 3-29 Imposed dam break Boundary Condition for the Romaine-Puyjalon River.

3.2.2 Romaine-Puyjalon Mesh Refinement Results with TMA

In Figure 3-30 and Figure 3-31, changes in the computational domain and inundation lines after the mesh refinement are shown:

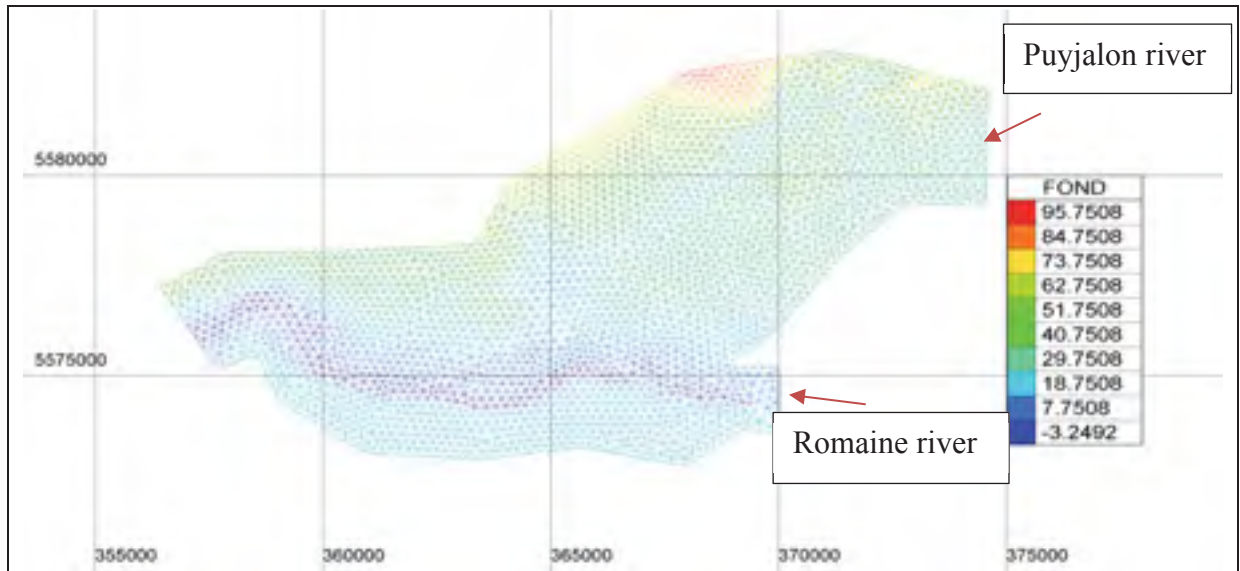


Figure 3-30 Terrain model of the Romaine-Puyjalon River.

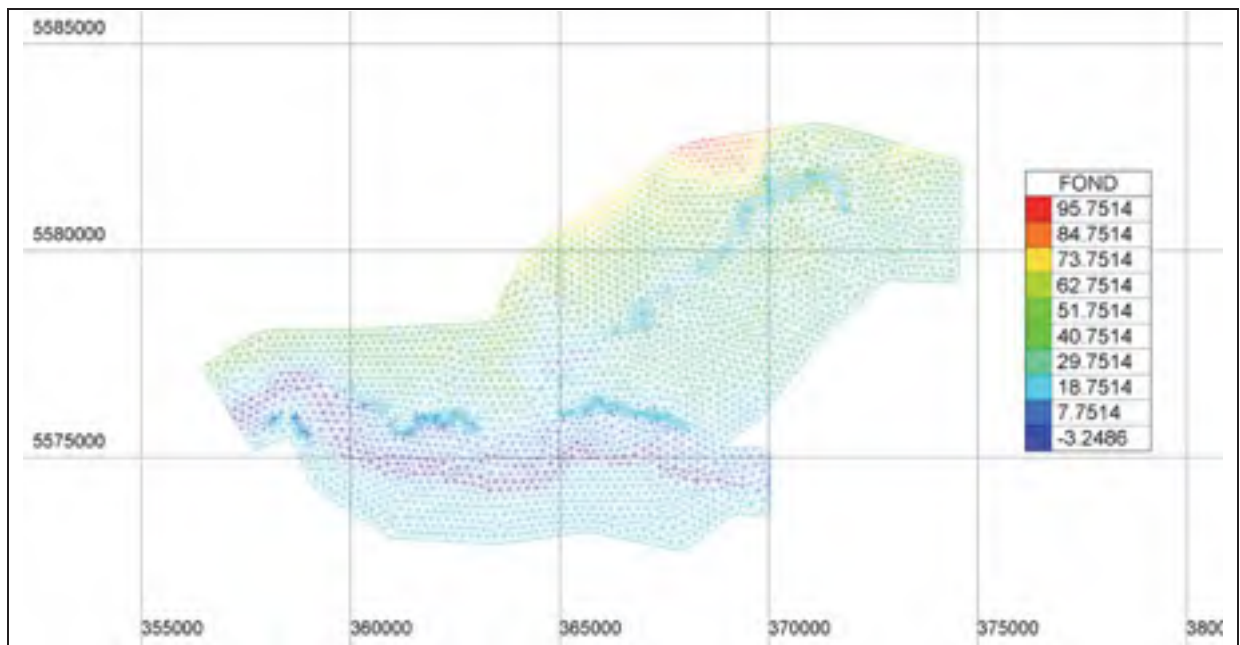


Figure 3-31 Numerical model of the Romaine-Puyjalon River after refinement with TMA.

The inundation lines do not show significant changes after the refinement (figure 3.32); however, changes in the bathymetry are evident.

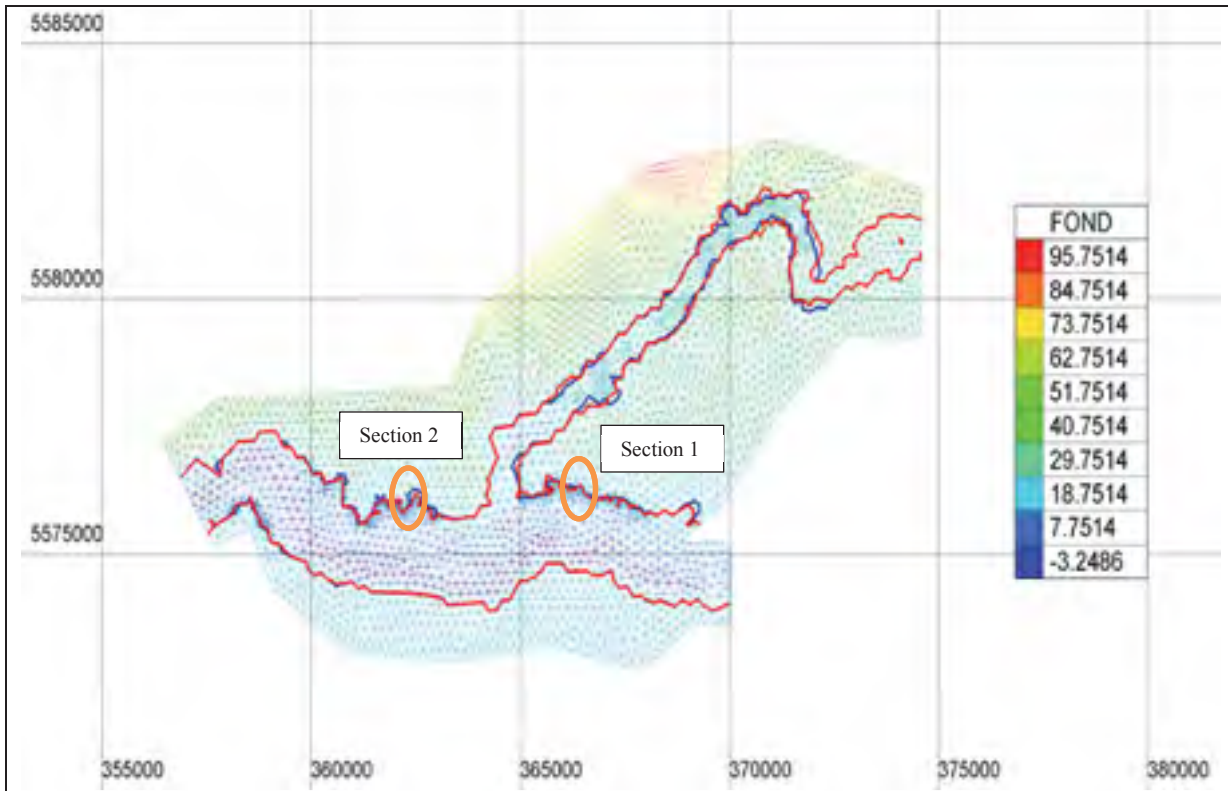


Figure 3-32 Comparison of inundation lines after the mesh refinement with TMA.

In the following figures the effect of mesh refinement on two sections is demonstrated.

Section1:

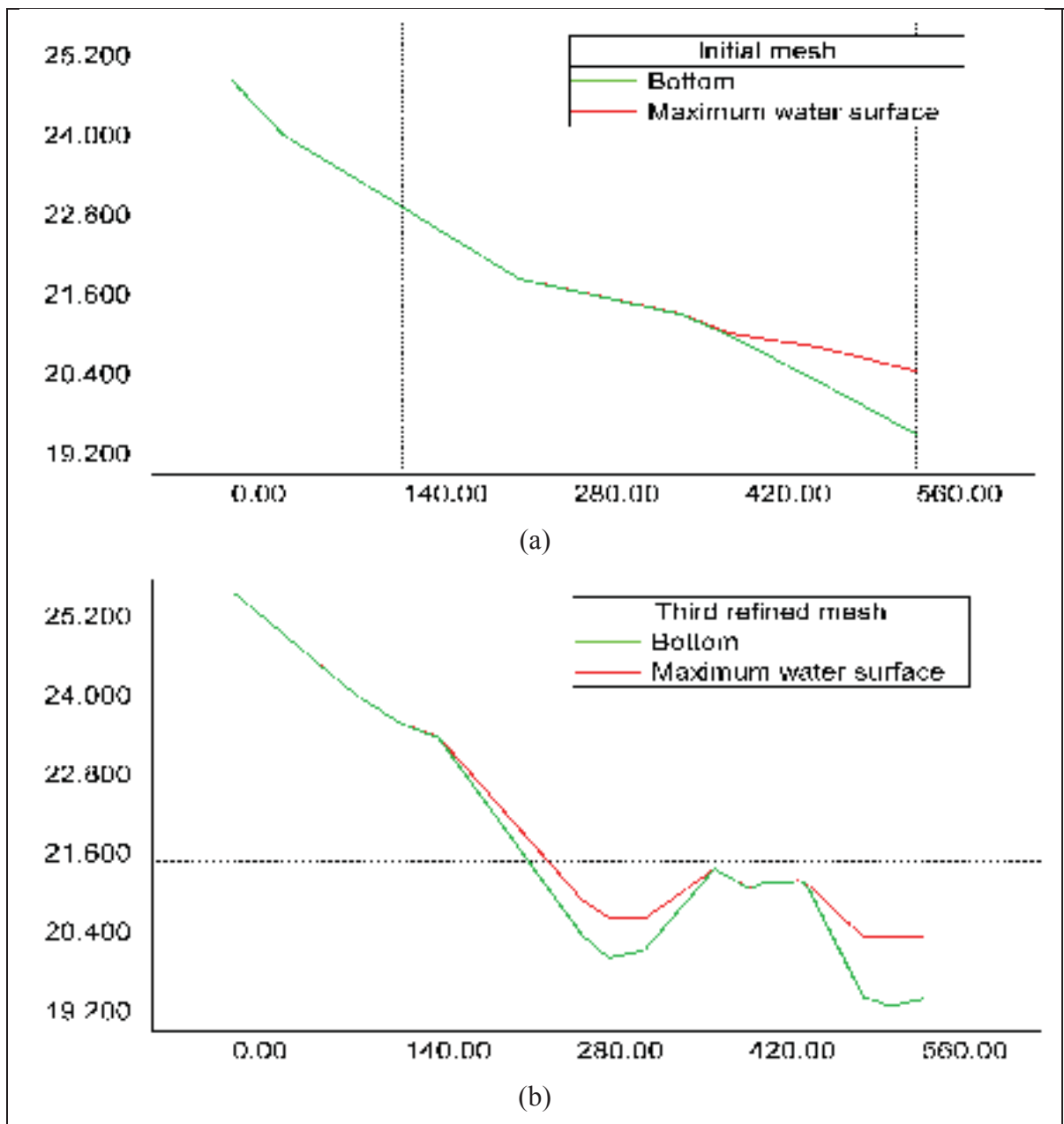


Figure 3-33 Change of the bottom and maximum free surface at the extreme part of the inundation line in dry zone 1.

Section 2:

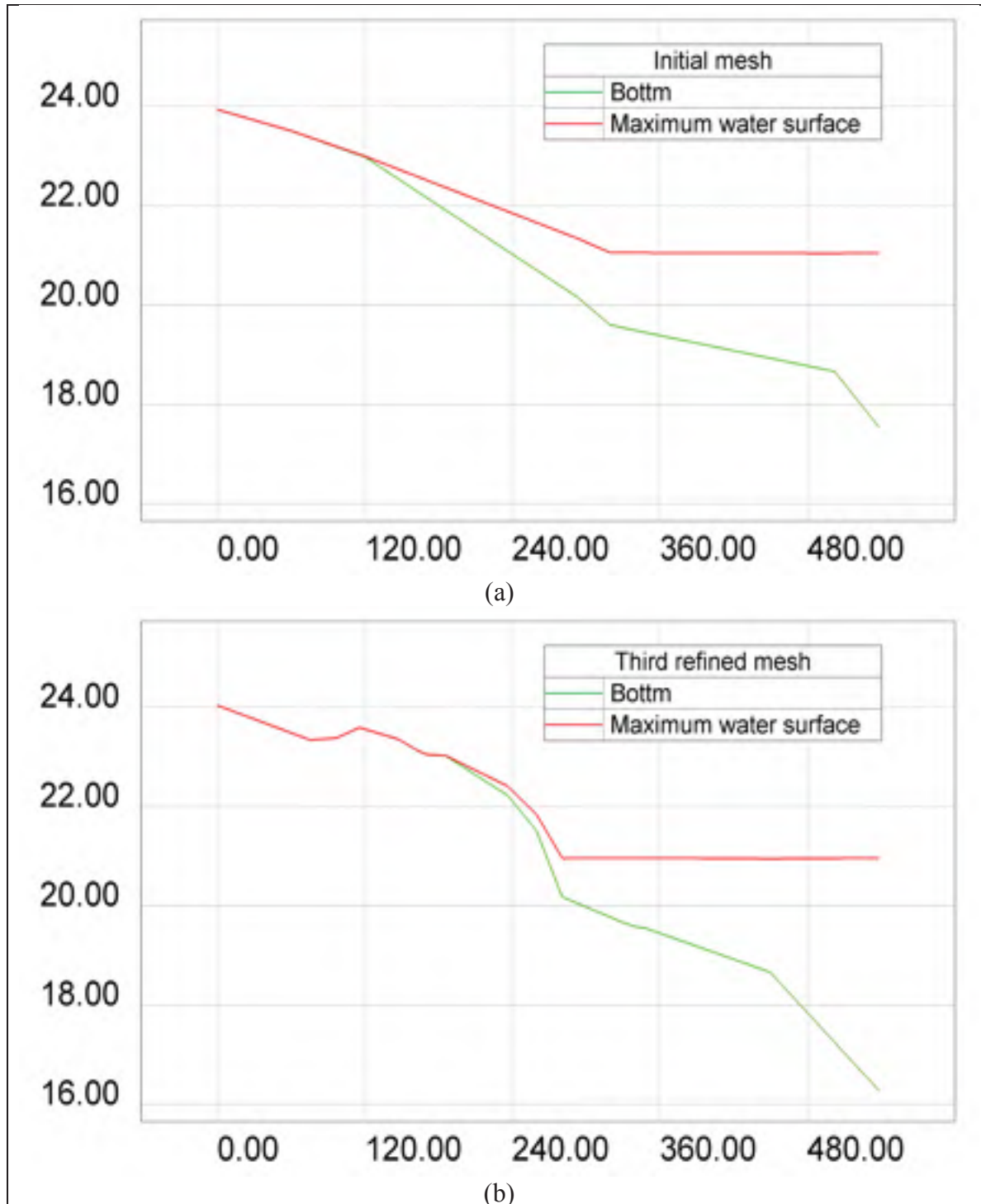


Figure 3-34 Change of the bottom and maximum free surface at the extreme part of the inundation line in dry zone 2.

In a second study, the mesh refinement with TMA was performed on the same river with different initial conditions. In this case, the inflow came only from the Romaine reservoir, and the volume of water coming from the Romaine to the Puyjalon branch was examined. In figure 3.35, the initial condition for this case is shown. Figure 3.36 and 3.37 shows the initial mesh and the refined mesh with TMA.

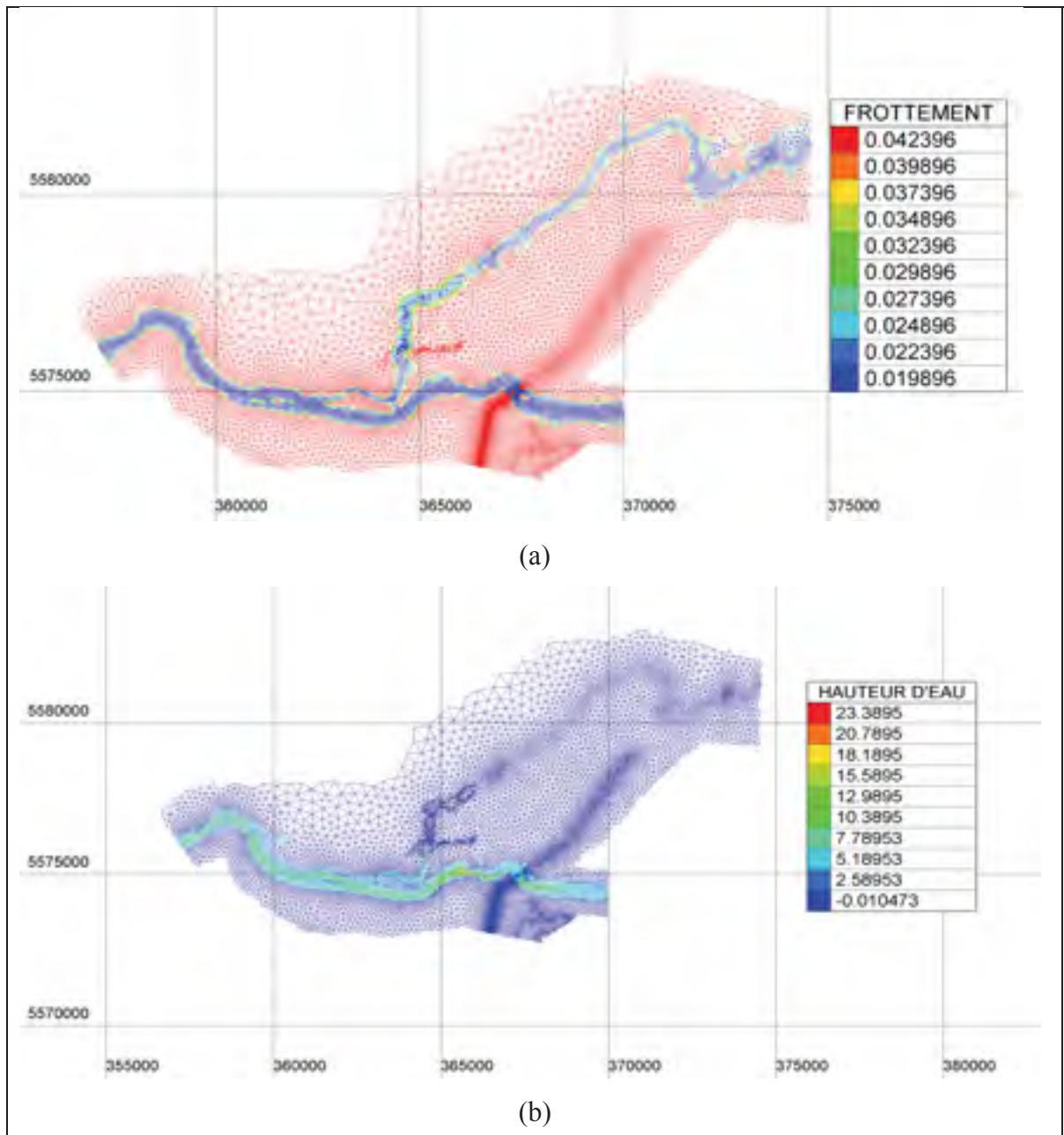


Figure 3-35 Initial Condition of the Romaine-Puyjalon River (second version).

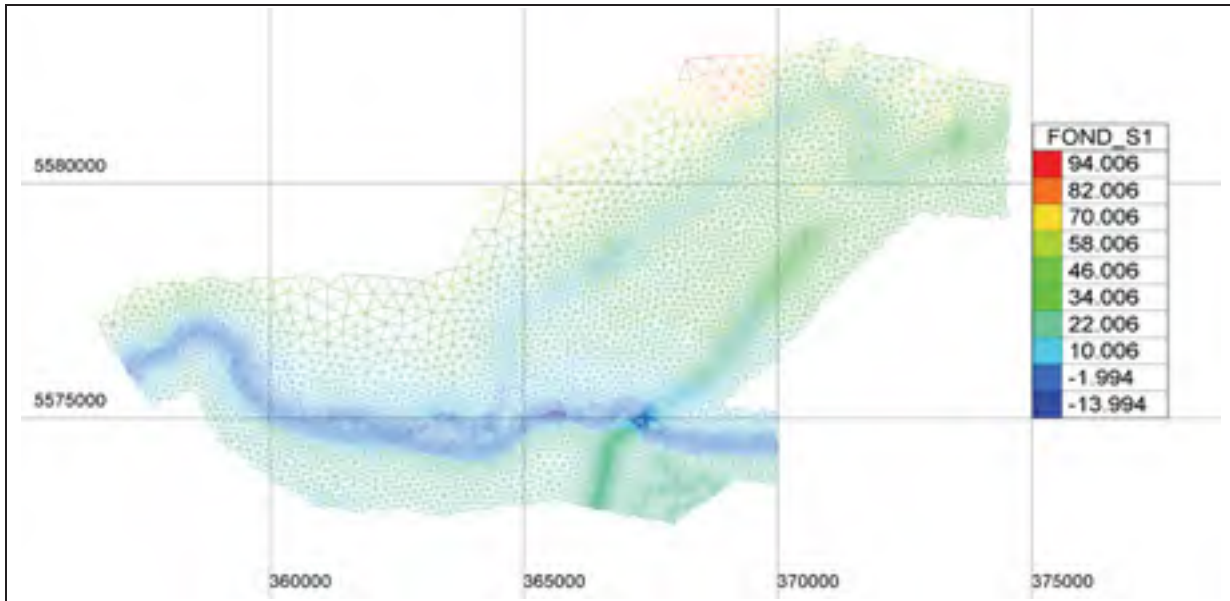


Figure 3-36 Initial numerical model of the Romaine-Puyjalon River.

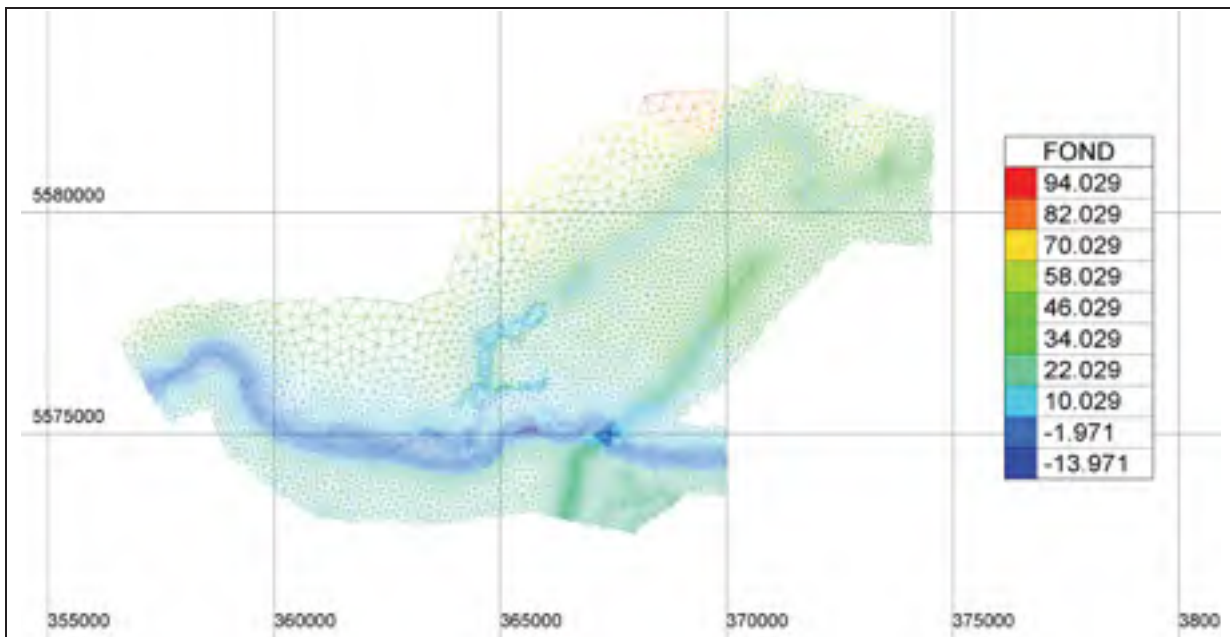


Figure 3-37 Numerical model of the Romaine-Puyjalon River after two mesh refinements with TMA.

As the LIDAR data were huge for this case, mesh refinement for other parts was considered, but only the LIDAR data along the Puyjalon River were selected.

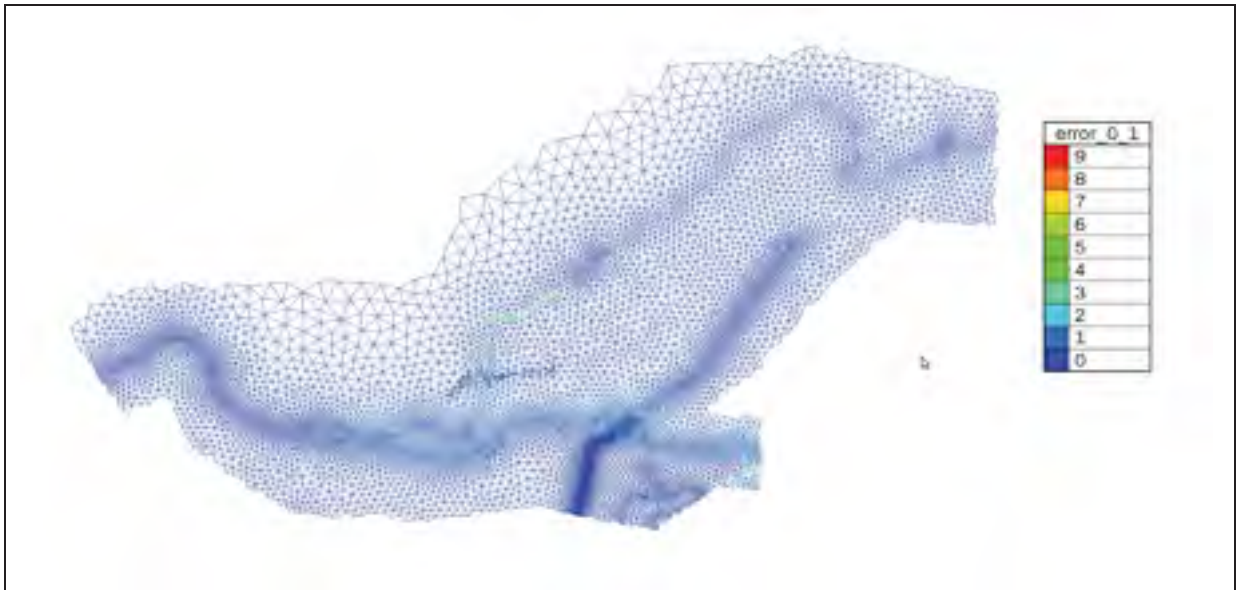


Figure 3-38 Error.t3s file generated between the first refinement and the second refinement.

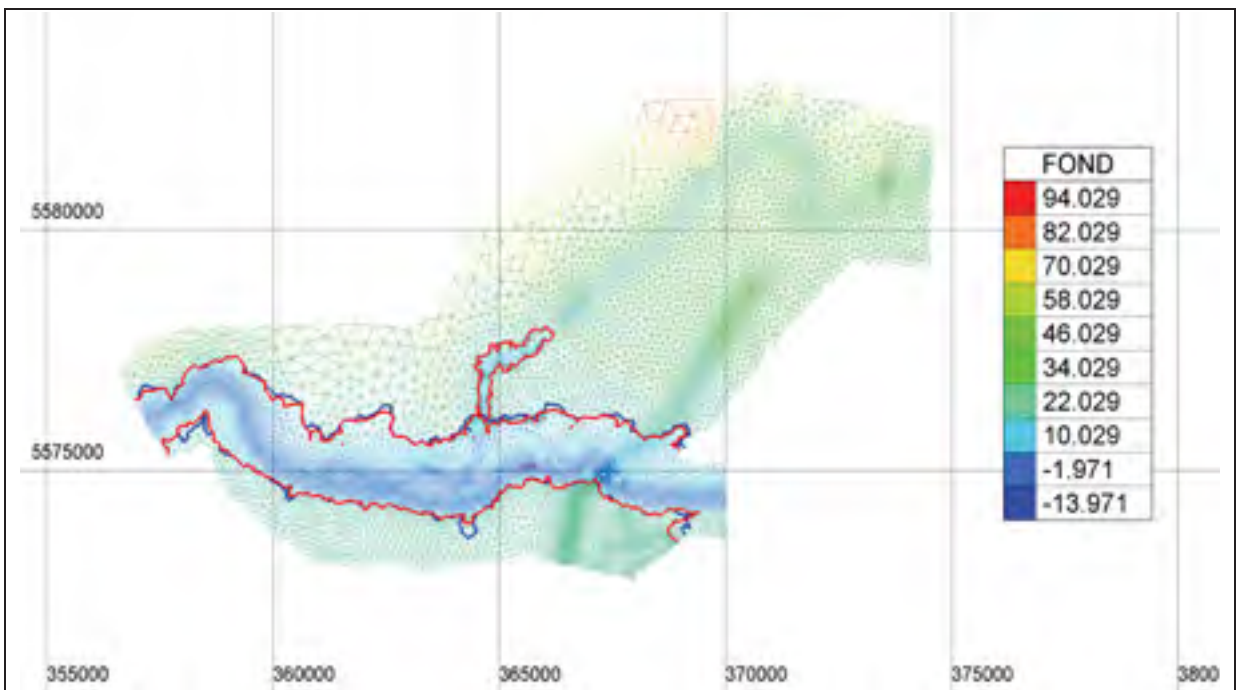


Figure 3-39 Change of the inundation line after the mesh refinement with TMA. The red line is the inundation line of the initial mesh refinement, and the blue line is the inundation line after two further mesh refinements.

Table 3.3 Comparison of total volume, entering volume, exiting volume, and errors between different meshes on the Romaine-Puyjalon River.

	Initial Volume (M^3)	Total Volume (M^3)	Entering Volume (M^3)	Exiting Volume (M^3)	Error (M^3)
Initial Mesh	0.6505984E+08	0.9268691E+08	0.2762690E+08	-159.9145	0.1725319E-05
1 st Refinement	0.6506278E+08	0.8305782E+08	0.1799370E+08	-1340.556	0.1614004E-04
2 nd Refinement	0.6506289E+08	0.9274181E+08	0.2767791E+08	-1006.263	0.1085015E-04
Finer Mesh	0.6609483E+08	0.9260096E+08	0.2650530E+08	-840.4796	0.9076359E-05

In figure 3.40, a comparison of the inundation line with different LIDAR data is made.

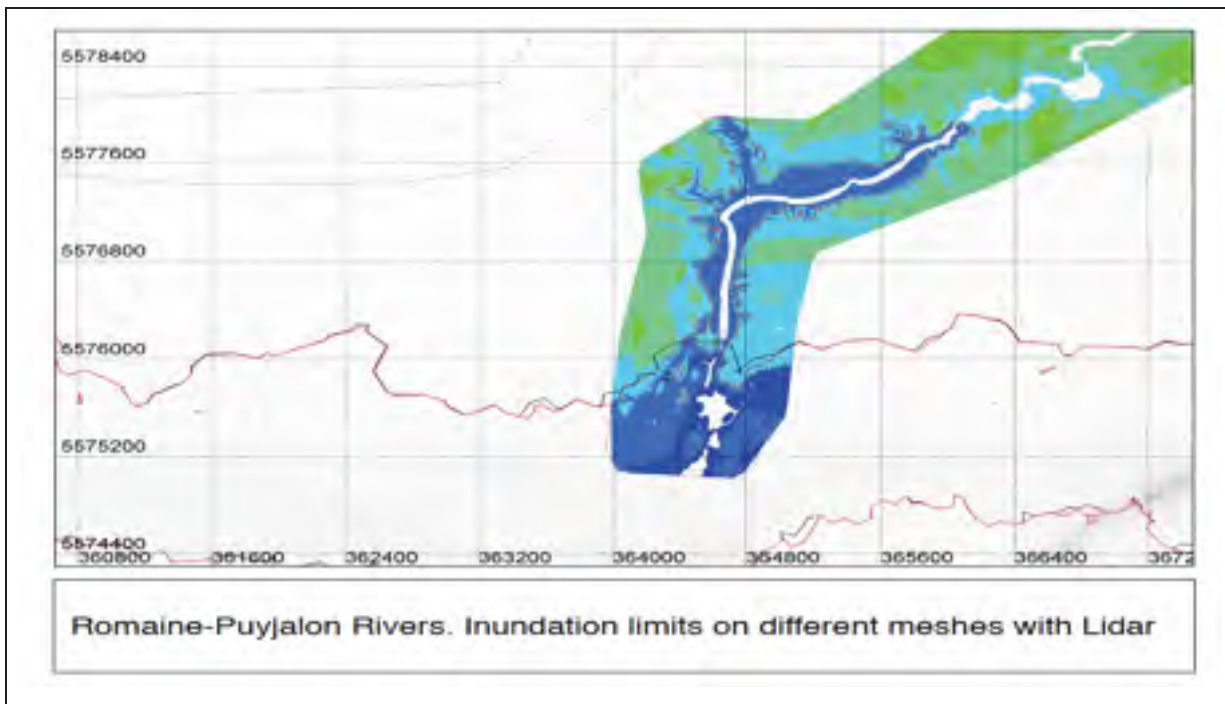


Figure 3-40 Difference in the inundation line of the Romaine-Puyjalon River with different meshes, prepared with Hydro-Quebec.

In the following figure the inundation limit for the finer mesh (with LIDAR) and the mesh refined with TMA is illustrated.

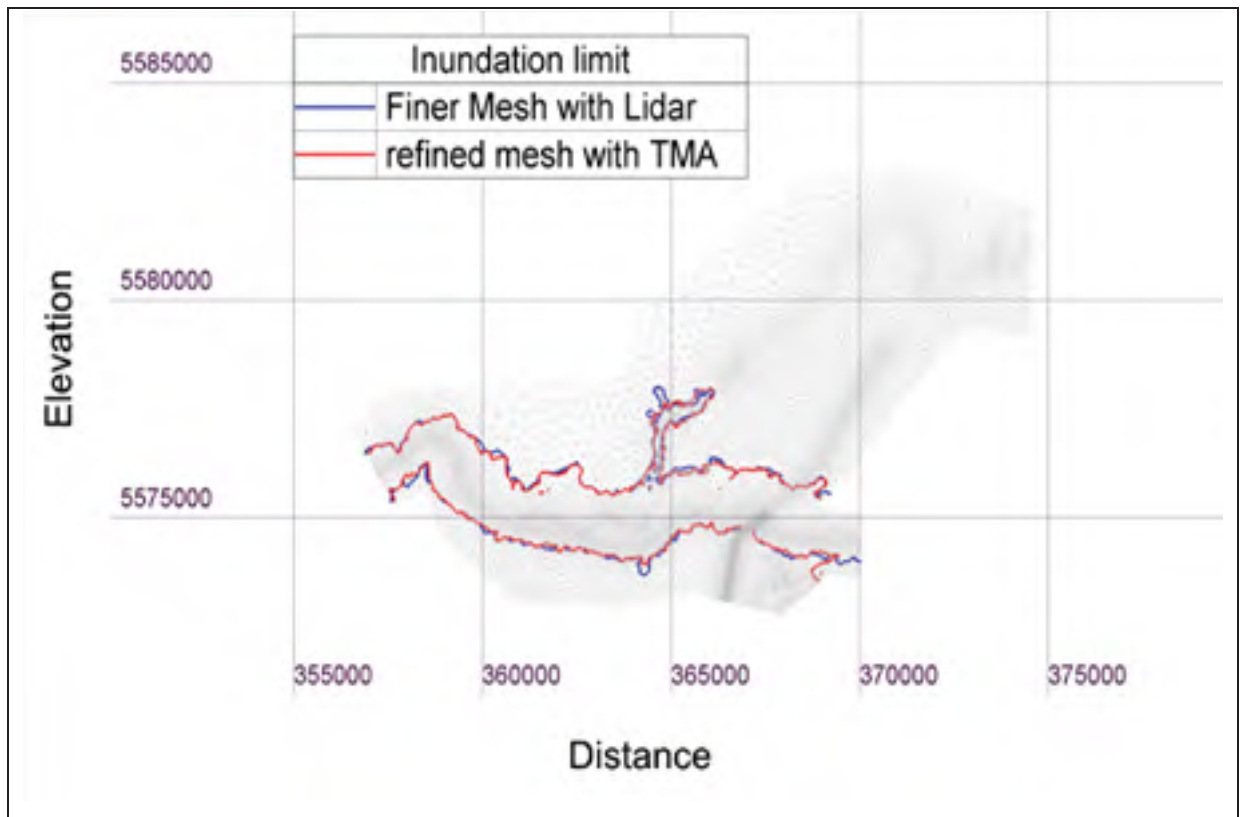


Figure 3-41 Comparing the inundation limit for the mesh refined with TMA and the finer mesh with LIDAR.

Figure 3.42 and 3.43 shows the changes in maximum water surface along the Puyjalon River in the initial mesh and in the refined mesh with TMA.

Considering the modified friction factor along the shoreline increases the accuracy of simulations and reduces the number of refinements.

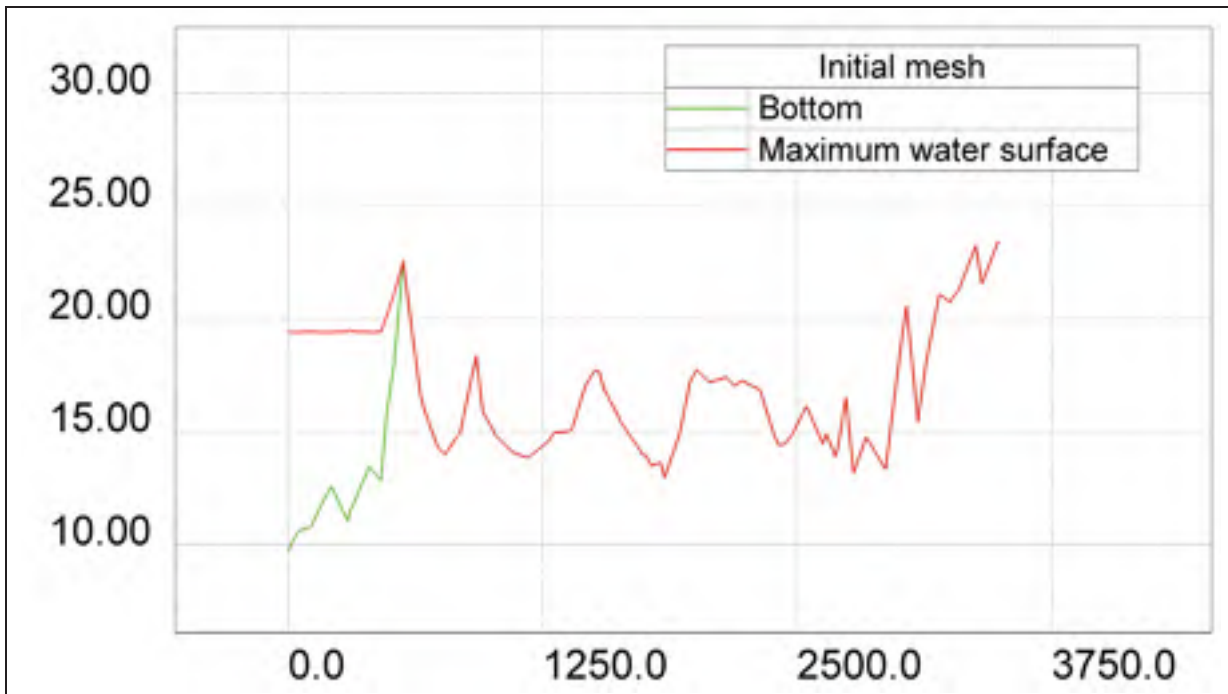


Figure 3-42 The bottom and maximum free surface along the Puyjalon branch before the refinement.

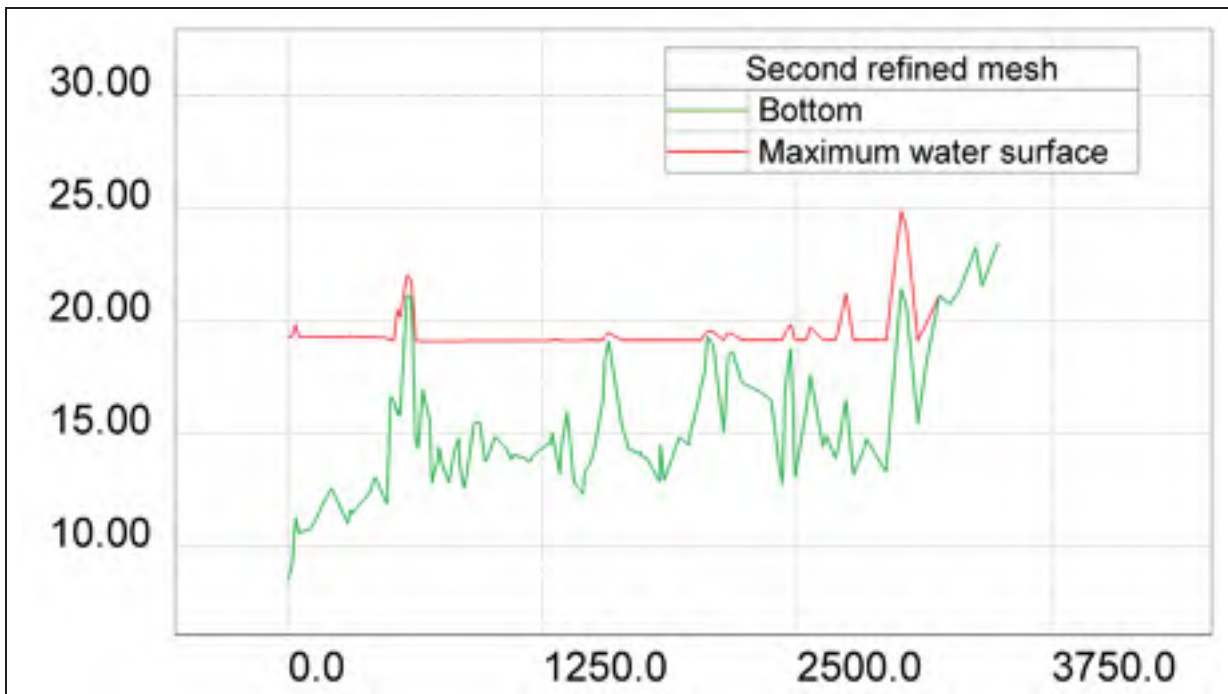


Figure 3-43 The bottom and maximum free surface along the Puyjalon branch after mesh refinement with TMA.

3.3 Conclusion

In this section TMA was used for a dambreak analysis of the Romaine-Puyjaon River. For this case, the boundary conditions were a dambreak hydrograph discharge at the inlet and a water surface validation at the outlet. After post-processing the simulation results, TMA was applied using LIDAR data to modify the slope of the dam in the way of the water flow. For this case, in addition to modifying the slopes near the shoreline by doing the adaptive meshing methodology based on topometric data, the friction coefficients for these parts were also calculated. It was expected that the modified friction coefficients would help solving the free surface equations and reduce the number of refinements. It was concluded that another method to help momentum equation analysing over the wet dry elements would be modifying friction coefficient properly to damp the wave motion and avoid creation of standing waves, at the end of the streams which is the main reason of having numerical problem in dry bed, due to the high change in gradient of velocity (sudden change in slope of velocity characteristic curves).

CONCLUSION

One of the main problems in the numerical modeling and drawing of floods map produced by dam breaks is the uncertainty in the geometry representation of a model. This uncertainty might look like a big bump or a weir in the flow path in the numerical model that could act like a vertical wall in the way of the water, especially over a dry bed or on tidal flat areas. During the numerical analysis for flood mapping, engineers have faced a variety of uncertainties in their simulations. One of the uncertainties in creating the numerical model is the uncertainty about bathymetric data captured from LIDAR data or topographic data along with. In such a case, the generated numerical model using only a limited portion of LIDAR data that might cause inaccuracy in the numerical analysis.

In this thesis, to reduce this uncertainty in bathymetry, we looked at to do mesh refinement techniques namely TOPO MESH ADAPTOR (TMA) along the shoreline, where the situation is more sensitive. By, having finer meshes in these areas can improve numerical analysis. To our knowledge, we propose for the first time an adaptive meshing methodology based on real topometric data to improve the precision of flood maps in particular for secondary valleys, in the cases where rich topographic data of the LIDAR type is available for a complex bathymetry, such as that of the Eastmain River and Romain-Puyjalon River. A code called TMA was written in Python to implement this method.

We propose an adaptive meshing methodology to perform mesh refinements where discontinuity arises from projecting new nodes on the topometric data. The adaptive meshing methodology is applied on fluid flows to get more accurate results for the numerical simulation of water motion. Mesh refinement method can be classified into the Eulerian approach, the Lagrangian approach, and the Eulerian-Lagrangian approach (whether it is based on water depth, velocity or both). These adaptive methodologies, for dam breaks analysis are based on four main steps:

- 1) Applying the numerical approach on the initial mesh;

- 2) Post-processing approaches based on a water depth algorithm to determine which elements are wet, dry, or partially wet-dry;
- 3) Applying the mesh adaptive methodology on the wet-dry elements;
- 4) Interpolating the results of the previous mesh on the new mesh and comparing them with the new mesh, and then selecting the elements that need to be refined in the next mesh refinement step.

As stated before we developed a code in Python named TOPO MESH ADAPTOR (TMA) to do the mesh refinement on the basis of the adaptive mesh methodology. After the mesh refinement using TMA around these points, it was discovered that observed weirs were like small dikes submerged after the mesh refinement, and that further zones could be affected by the water, or vice versa. The TMA processing is summarized below:

- TMA uses linear 2D interpolation to calculate the water depth at every point inside each element on the basis of the nodal values of the elements. This linear interpolation helps to reduce the local uncertainty of bathymetry along the main flow direction. In addition, the effect of bottom elevation is mainly sensitive at the upstream end of the streams where the water depth is very low, but the velocity could be high. Thus, the mesh refinement technique focuses in these areas.
- TMA uses two points from the LIDAR data retrieving the minimum and maximum values of the elevation inside each element to find the maximum real slope in the cut-elements. Most previous efforts controlled the slope in the wet-dry elements in order to avoid having a high velocity or a negative depth. Within this new approach, the results are more trustworthy as it can apply where the maximum water level is more than its neighbor bottom elevation in dry bed by considering appropriate weight factor.
- TMA considers a higher weight factor around the V shape channels to find the freeboard of channels and obtain a more accurate shape of the channels. Better shape of the narrow channels helps in producing standing waves and increase the water level so that the water can pass through the high elevated bumps or dikes.

TMA is applied to two cases, the Eastmain River and the Romaine-Puyjalon River, and the results are presented. Mesh refinements with TMA where we have a belt of dikes or a channel with complex cross section in dry bed, yield more accurate result compare with coarse meshes. In this approach, we tried to reach a final solution by first, post-processing the results from a coarse model and drawing the inundation limit to determine where the mesh should be refined then mesh refinement only is performed locally along the inundation limit, which is the most sensitive part of the domain. For the second case, modifying and calibrating the friction coefficient along the shoreline could improve the analysis, and should be considered for future efforts.

FUTURE WORK

There are various adaptations, tests and analyses left for the future. In this work we have focused on solving the uncertainty of bathymetry for two cases. However, there are more uncertainties of input data and more mathematical models in different areas that we suggest be studied in future. They are as follows:

- Define additional factors for doing mesh refinement to enhance the mesh refinements and develop a Lagrangian approach for doing the mesh refinement and testing it on more cases (adaptive meshing methodology in existence of bubble velocity).
- Improve the mathematical model by considering other factors, such as mesh size or order of polynomial degree, in order to improve the simulation on cut-elements.
- Study the effect of uncertainty in changes in bathymetric data during the simulation (e.g. sediment transfer);
- Study the effect of uncertainty of bathymetry in a 3D model if there is a velocity component in the z direction.

APPENDIX I

USER GUIDE OF TOPO MESH ADAPTOR

Adaptive mesh refinement methodology for shallow water wave propagation

BY

Mahdi Moslemi

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR A MASTER'S DEGREE
WITH THESIS IN MECHANICAL ENGINEERING

Montreal, Québec
H3C 1K3

© Copyright 2019 reserved by Mahdi Moslemi

© Copyright reserved

It is forbidden to reproduce, save or share the content of this document either in whole or in parts. The reader who wishes to print or save this document on any media must first get the permission of the author.

TOPO-MESH-ADAPTOR

Adaptive mesh refinement methodology for Shallow Water Wave Propagation

(By python)

USER GUIDE

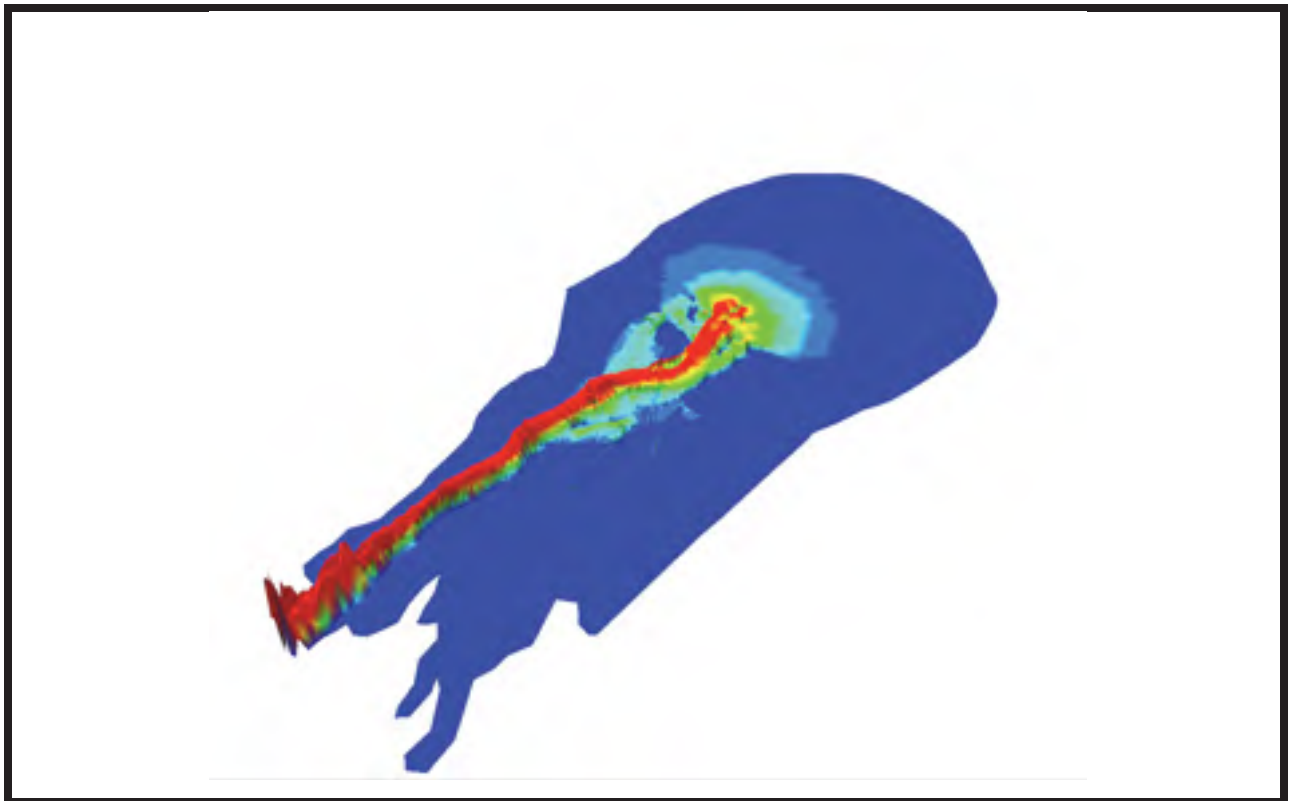
Mahdi Moslemi

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

UNIVERSITÉ DU QUÉBEC

Montreal, Québec

H3C 1K3



Contents:

1	Introduction.....	102
2	Instruction on download and using python3 for running code.....	103
3	Mesh refinement algorithms	104
3.1	Summary of using First algorithm	104
3.2	First Algorithm	105
3.2.2	Generating the input file for running the code in serial.....	106
3.2.3	Generating the input file for running the code in parallel.....	107
3.3	Second algorithm	109
3.3.1	Summary of Using Second algorithm.....	109
3.3.2	Interpolating the result of previous mesh on the new mesh.....	110
3.3.3	Generating input files for second algorithm	112
3.3.4	Parallel mode	113
3.4	Generating the mesh file and running TELEMAC after each refinement...	115
4	Results.....	117
5	Summary.....	119
6	Appendix A.....	120

Automatic mesh refinement methodology for Shallow water wave propagation

1) Introduction

Flood maps are the final product of the dam failure studies required by dam safety regulations. The flood's limit which represents the maximum amplitude of the flood wave's envelope, is generally the result of a dam break scenario simulated by a hydraulic numerical model. However, the numerical model uses only a limited portion of the available topometric data data to build the terrain model (a 2D mesh plus topometric elevation at nodes). Particularly in the cases where the topo-metric data has been recorded by LIDAR, this data is estimated in several million of points. But, the hydraulic numerical models rarely exceed hundreds of thousands of nodes due to the computer constraints and time associated with the operation of these models. The production of the final flood map requires consistency between projected levels and elevations for all points on the map. This verification may be tedious if the area becomes large; also there are several small secondary valleys of tributary streams which could not be represented by the original hydraulic numerical model. The aim of this work is to propose an automatic strategy of re-meshing which uses the envelope of the maximum dimensions reached with the original model coupled with the available LIDAR data to produce an improved mesh that will succeed in accurately capturing the wet/dry fronts and the overflows in secondary valleys and the area where we have changes in the fluid regime over dry bed. The algorithm is based on few basic steps such as:

- (i) finding the elements cut by the envelope of the wet/dry interfaces;
- (ii) projecting the topo-metric points onto the cut elements;
- (iii) if these points are very close to the interface or if they are found in a valley or if they are more elevated than the corresponding cut elements, then these points will be added to the previous nodes which are included in a subsequent triangulation step that produce a new mesh;

- (iv) re-run the simulation on the new mesh.
- (v) Continue using the above algorithm till it does not produce new nodes.

This algorithm has been implemented and validated in the study of a dam break flow over a complex river bathymetry.

Key words: crude mesh, fine mesh, interpolation, flood mapping, automatic mesh refinement

2) Instruction for download and using python3 for running code

Python is a free programming language which has a lots of powerful libraries and is less complicated than other programming languages. In comparison with the c++ language, code snippet are often 5-10 times shorter. One can argue that this advantage make python a popular programming language to interact with open source programming like TELEMAC, QGIS and etc.

We used python3 for programming the code. You can install python3.5 from <https://www.python.org/downloads/> . It is also recommended to have a good editor for python commands, so you can modify easily the code whenever you want. For this case I used Pycharm, but you can also use IDLE (the original editor of python).

After installing python, sometime it is necessary to use new libraries. For installing the libraries of python3.5 or higher version do as follow:

- For MAC OS (or LINUX) users, open Terminal (Console) and simply type: pip install “name of library” like “pip install math”
- For Windows users, open the “cmd”(command line),then go to the directory of script in python folder by writing: “cd C:\Python35\Scripts” in cmd , and then write pip install “name of library”

For running the python codes one should change the current directory to the folder where the code file is located by using cmd for windows (or Terminal for mac) i.e. “cd C:\Desktop\Code_Folder” then you can write:

“ python file.py ”

The libraries which we used for the code are as follow: (If you didn't have any of them you can install it following the steps previously described)

- Numpy
- Numpy.linalg
- Pandas
- Math
- Random
- CSV
- tkinter
- pathlib
- mpi4py (this library is required only for running the codes in parallel and you need to have already installed MPI executable on your system like Open MPI, MPI1 or MPI2)

Another way to run the python codes is to use the “Run” tab in IDLE, and then click on Run Module, for running the python files.

Mesh refinements should be performed on the elements where there is change of fluid regime from supercritical to subcritical. This will happen where there are hydraulic jumps or walls in wet-dry elements. For this purpose, we produce two algorithms for mesh refinements.

3) Mesh Refinements Algorithm

3.1. Summary of Using First Algorithm

1. Go to the directory “TOPO_MESH_ADAPTOR”;
2. Create the directory “INPUT_FILES”;
3. Generate the “HAUTEUR D’EAU(MAX).t3s” file from the results of TELEMAC (HAUTEUR D’EAU (max)). Put all the necessary files: geometry mesh file, “maximum water depth” file, and the topo-metric file(s) “i.e. *.grd or *.xyz” inside this folder.
4. Generate the “parameters.txt” file and put it in to the “INPUT_FILES” folder (a description of this file will be detailed).

5. Use the command window and locate into the “TOPO_MESH_ADAPTOR” directory and run the python code “TMA_serial.py” by typing:

```
python MTMA_Serial.py
```

Another way to run the python file by IDLE: open the python file with “Edit with IDLE”, then click on “run” tab and select “Run Module”.

6. If you’d like to run the code in parallel, first you need to divide your LIDAR data to the number of the processors which will be employed for the parallel mode (you get topo-metric files as *.grd).
7. Generate the “parameters.txt” based on the new topo-metric files you have and put all the necessary files in the “INPUT_FILES” directory.
8. Use the following command window to run the code in parallel with np (example: np=6) processors:

```
mpiexec -n 6 python TMA_FA_Parallel.py
```

or

```
mpirun -n 6 python TMA_FA_Parallel.py
```

3.2. First Algorithm (detailed example)

“MOSLMEI-TOPO-MESH-ADAPTOR” has two algorithms for automatic mesh refinement. In the first algorithm we do the mesh refinement for all of the cut-elements (for more details please refer to the methodology in my thesis).

This refinement will be done for the elements in the dry zones.

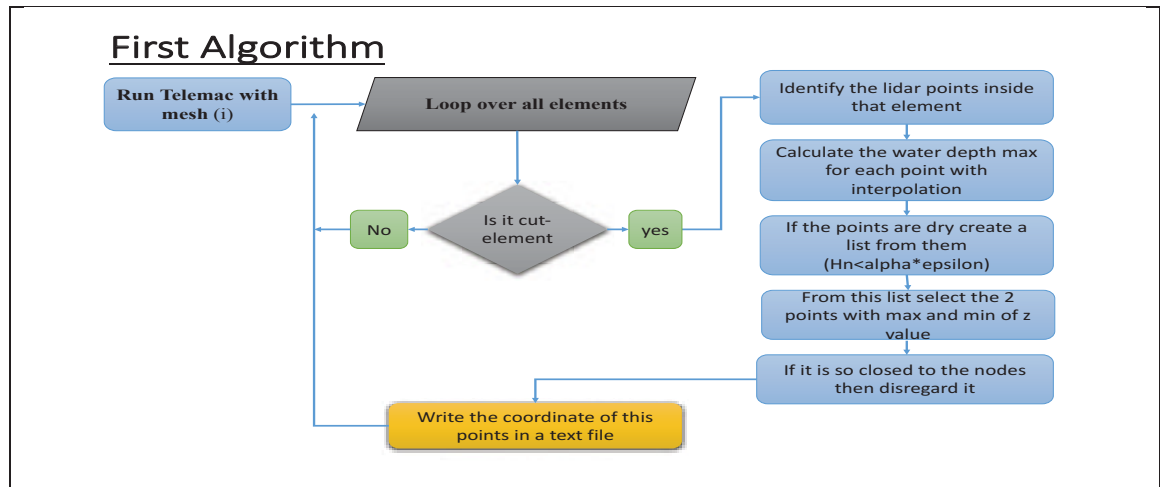


Figure a. First algorithm for the mesh refinement.

For using the first algorithm we need to create some input files.

3.2.1 Generating the input files for running the code in serial

Before running the code, you have to create a folder in the source folder and name it “INPUT_FILES”. Inside the INPUT_FILES put the following files:

- topo-metric data (i.e. *.xyz, *.grd)
- maximum water depth for initial mesh computed with Blue Kenue, from the results of TELEMAC, (HAUTER D’EAU(max).t3s) and
- the geometry file for the initial mesh (FOND_S1_initial.t3s).
- parameters.txt file.

The “parameters.txt” file must be created. You can see below an example of the parameters.txt file in serial.

In the 5th line and 8th line, name of output files for “new added points”, and “all_points_1.xyz” should be written. These names can be defined by the user. In the following example, after running the algorithm, the new mesh will be generated using Blue Kenue and the list of all points in “all_points_1.xyz”.

The word “first” at the line 7 means we want to use “First Algorithm”

```

parameters.txt - Edited
HAUTEUR D'EAU(Max).t3s
33d012000201_utm_17.grd
0.05
2
refinement_1_
FOND_S1_initial.t3s
first
all_points_1.xyz

```

← Maximum water depth file
 ← Topo metric data
 ← Epsilon (ϵ)
 ← α (α)
 ← Output file for new points
 ← Geometry
 ← Final result

Figure b. parameter file for running the first algorithm.

Using the command window, access into the directory “TOPO_MESH_ADAPTOR” and run “TMA_Serial.py” as follow:

python MTMA_Serial.py

3.2.2 Generating the input file for running the code in parallel (First Algorithm)

In the parallel approach we divide LIDAR data into several smaller parts and employ each processor to run on one of the LIDAR files. You can see an example of parameter file for running the code in parallel with 4 processors as follow:

```

parameters.txt - Edited
HAUTEUR D'EAU(Max).t3s
33d012000201_utm_17.grd 33d012000202_utm_17.grd 33d022000202_utm_17.grd 33d072000102_utm_17.grd
0.05
2
refinement_1_
FOND_S1_initial.t3s
first
all_points_1.xyz

```

Figure c. Symbolic parameter file for running first algorithm in parallel.

Two codes can be used to partition LIDAR data into tiles. The code “divided_rectangular.py” divides a rectangular LIDAR domain into smaller rectangles. This code may be used to find the region where we want to apply the adaptive mesh refinement methodology (AMR). You may find that for even small regions, there are a lot of points. In this case you can run the “divided_rectangular.py” again to divide your LIDAR data into more partitions, or use “divided_line_by_line.py” which divides the LIDAR file based on the number of points that exist in that LIDAR file.

After dividing LIDAR data in to the smaller parts like “LIDAR1”, “LIDAR2”, ..., “LIDAR6”, again the “INPUT_FILES” folder should be created and all the LIDAR data files, the geometry file and the maximum water depths should be placed inside it.

Then, it is possible to run this algorithm in the parallel mode. For example, if we like to employ six different LIDAR files, we should work with six processors. In this case we should use “Terminal” in Mac OS or “Command line” in Windows to change the directory to the location of the code. As the name of python file in the first algorithm for running the code in parallel is “TMA_FA_Parallel.py”, we should write in “cmd”:

```
mpiexec -np 6 python MTMA_FA_Parallel.py
```

While the code is running, it makes some log files which give you more details about the running’s progress for each processor.

```

rank=0 processor=0          processor=0.8200
rank=1 processor=1          processor=0.8201
rank=2 processor=2          processor=0.8202
rank=3 processor=3          processor=0.8203
rank=4 processor=4          processor=0.8204
rank=5 processor=5          processor=0.8205
rank=0 processor=0          processor=0.8206
rank=1 processor=1          processor=0.8207
rank=2 processor=2          processor=0.8208
rank=3 processor=3          processor=0.8209
rank=4 processor=4          processor=0.8210
rank=5 processor=5          processor=0.8211
rank=0 processor=0          processor=0.8212
rank=1 processor=1          processor=0.8213
rank=2 processor=2          processor=0.8214
rank=3 processor=3          processor=0.8215
rank=4 processor=4          processor=0.8216
rank=5 processor=5          processor=0.8217
rank=0 processor=0          processor=0.8218
rank=1 processor=1          processor=0.8219
rank=2 processor=2          processor=0.8220
rank=3 processor=3          processor=0.8221
rank=4 processor=4          processor=0.8222
rank=5 processor=5          processor=0.8223
rank=0 processor=0          processor=0.8224
rank=1 processor=1          processor=0.8225
rank=2 processor=2          processor=0.8226
rank=3 processor=3          processor=0.8227
rank=4 processor=4          processor=0.8228
rank=5 processor=5          processor=0.8229
rank=0 processor=0          processor=0.8230
rank=1 processor=1          processor=0.8231
rank=2 processor=2          processor=0.8232
rank=3 processor=3          processor=0.8233
rank=4 processor=4          processor=0.8234
rank=5 processor=5          processor=0.8235

```

Figure d. Example of the log file generated by the first algorithm. In this file we have some information about the cut-elements.

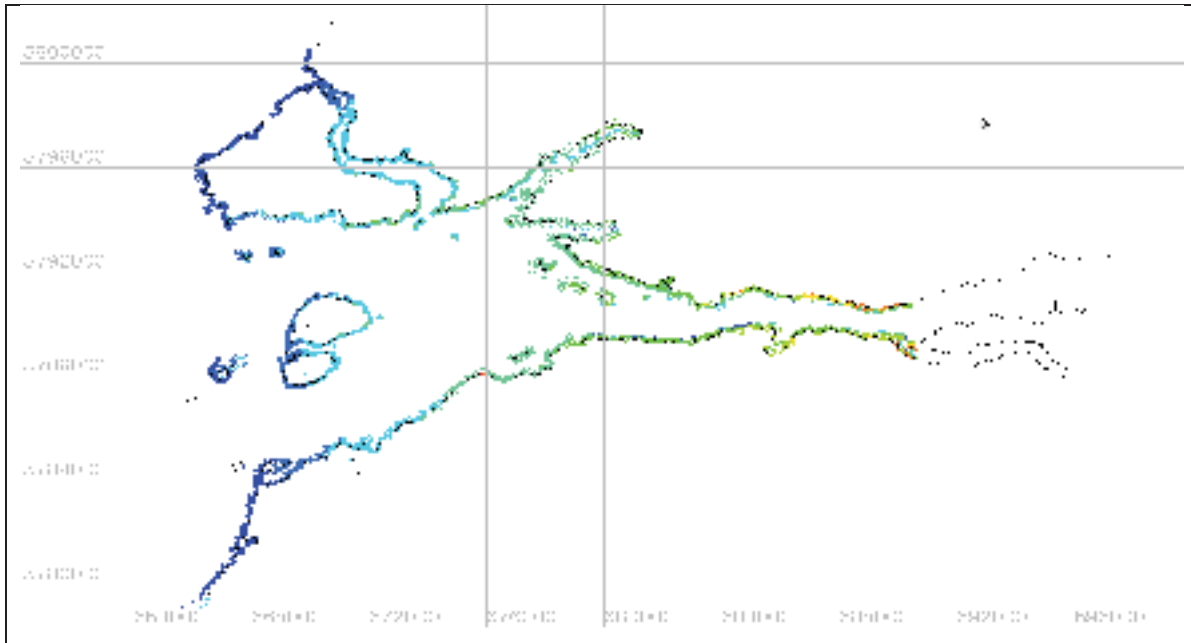


Figure e. The location of the new points added by running the algorithm to the previous mesh.

3.3. Second Algorithm

3.3.1 Summary of Using Second algorithm

1. Go to the directory “TOPO_MESH_ADAPTOR”
2. Create a directory “INPUT_FILES”
3. Create the water depth difference file with Blue Kenue (a description will follow);
4. Generate the “parameters.txt” file;
5. Put all the necessary files such as geometry (FOND_initial.t3s), water depth different file (error_1_2.t3s), topo-metric files, “parameters.txt” file, and “maximum water depth” file in INPUT_FILES;
6. Run the python code “TMA_Serial.py”;
7. If you like to run the code in parallel, first you need to divide your LIDAR data into the (np= number of processors);
 - Generate the “parameters.txt” file appropriate for running the code in parallel by writing the name of all topo-metric data files in “parameters.txt” file;
 - Use the following command to run the code in parallel with 6 processors:

mpiexec - n 6 python TMA_SA_Version1_Parallel.py

Or

mpirun - n 6 python TMA_SA_Version1_Parallel.py

in the second algorithm we try to do mesh refinement for the elements with significant differences in the water depths between two meshes.

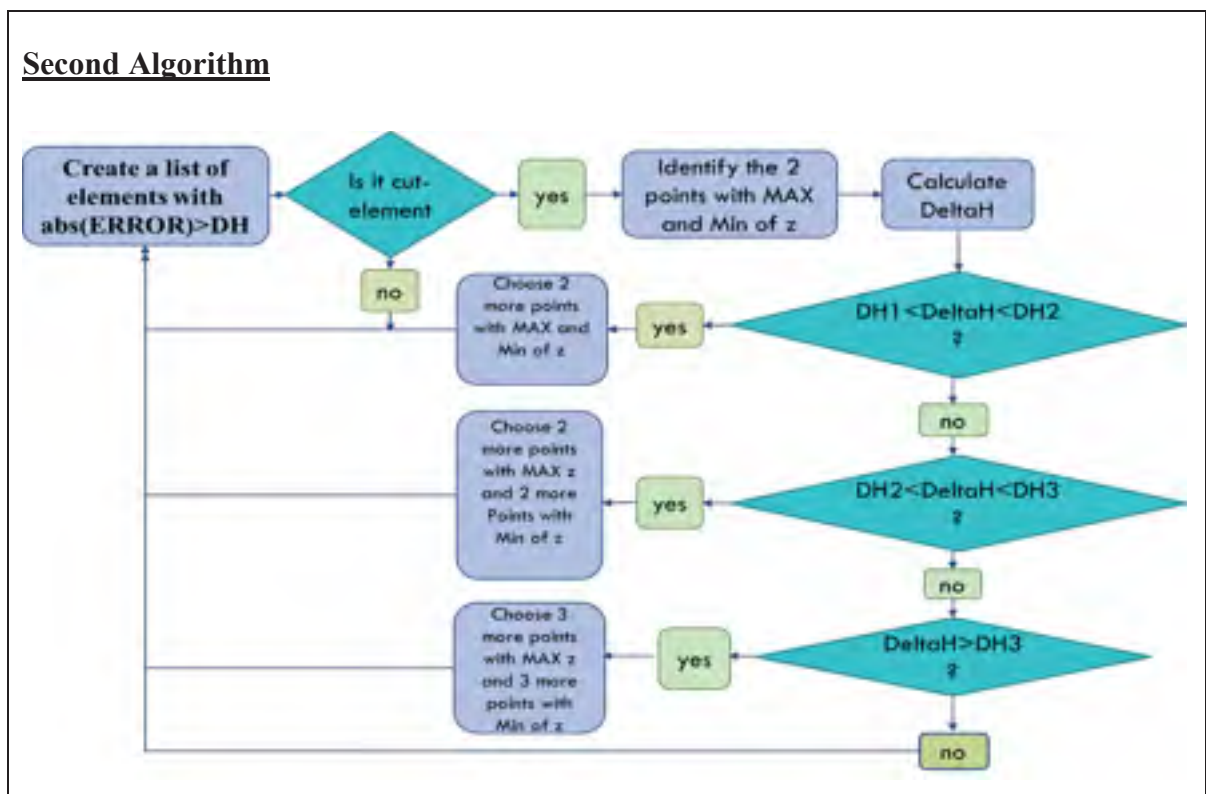


Figure f. Second mesh refinement algorithm.

In the second algorithm it is necessary to generate a mesh name “error.t3s” or mesh refinement indicator file. This file generated when interpolating the result of previous mesh (maximum water depth initial mesh) on the new mesh with the help of Blue Kenue.

3.3.2 Interpolating the result of previous mesh on the new mesh

With Blue Kenue we can generate a mesh with nodal values of maximum water depth:

- 1) Open the result of TELEMAC in Blue Kenue;
- 2) Choose the “water depth” and then we click on Tools>Extract Surface>Temporal Maximum to generate the “water depth (max)” mesh file.

The “water depth (max)” file is a mesh file with nodal values of maximum water depth over the simulation for different time steps.

With the help of this file we can figure out the inundation lines and the flood map. We can also see the effect of the wave simulation on the dry bed.

The “water depth (max)” file is a mesh file with the format of “*.t3s” that has the nodal values and the mesh connectivities while we need just the nodes for processing the interpolation. We can open “water depth (max)” file (HAUTER D’EAU(MAX).t3s) as a text file and collect the nodes manually or use the following python code to have the nodes automatically. The name of this python code is “CollectNodes.py”. With this code you can browse your “water depth (max)” file and click on run to have the nodes of your mesh.

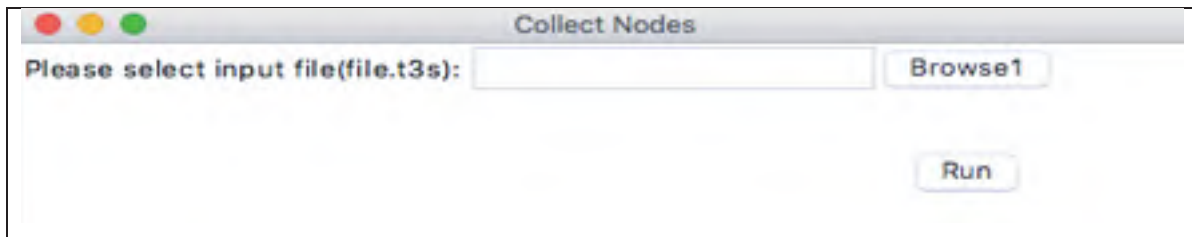


Figure g. Using CollectNodes.py for collecting the nodes for interpolation process.

After collecting the nodes in a text file, we name it “WDMAX_POINTS.xyz” and we do as follow:

- 1) In Blue Kenue we generate a 2D interpolation with WDMAX_POINTS.xyz;
- 2) Apply this 2D interpolation on the new mesh and name it “HAUTER D’EAU(MAX)_IP_1_2”;

- 3) Go to Tools>Calculator and generate a file with nodal value $abs("HAUTER D'EAU(max)_2" - "HAUTER D'EAU(max)_IP_1_2")$ and name it "error_1_2.t3s".

Sometimes you might see empty lines in the files generated by Blue Kenue. It is a bug. So, it is important to always check for any empty line in the "error_1_2.t3s" or "HAUTEUR D'EAU(max)_2.t3s" or "FOND_S_1.t3s" before running the algorithms.

Here you can see the location of the points added to the previous mesh by the second algorithm.

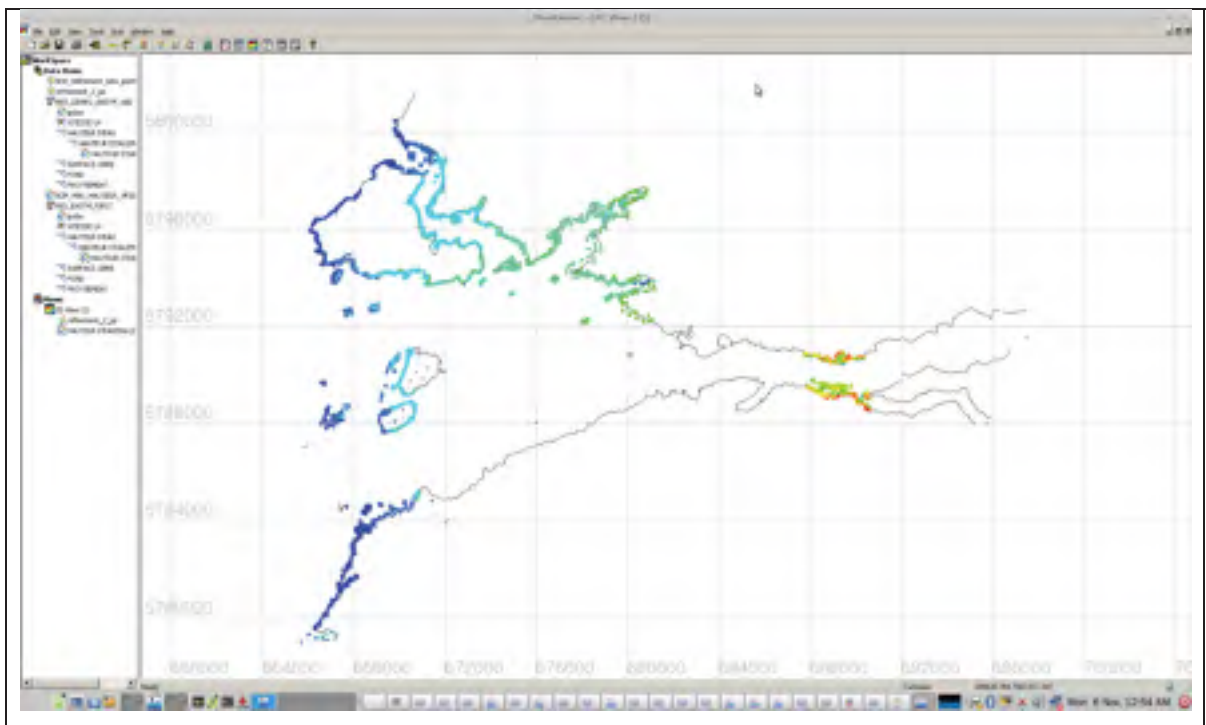
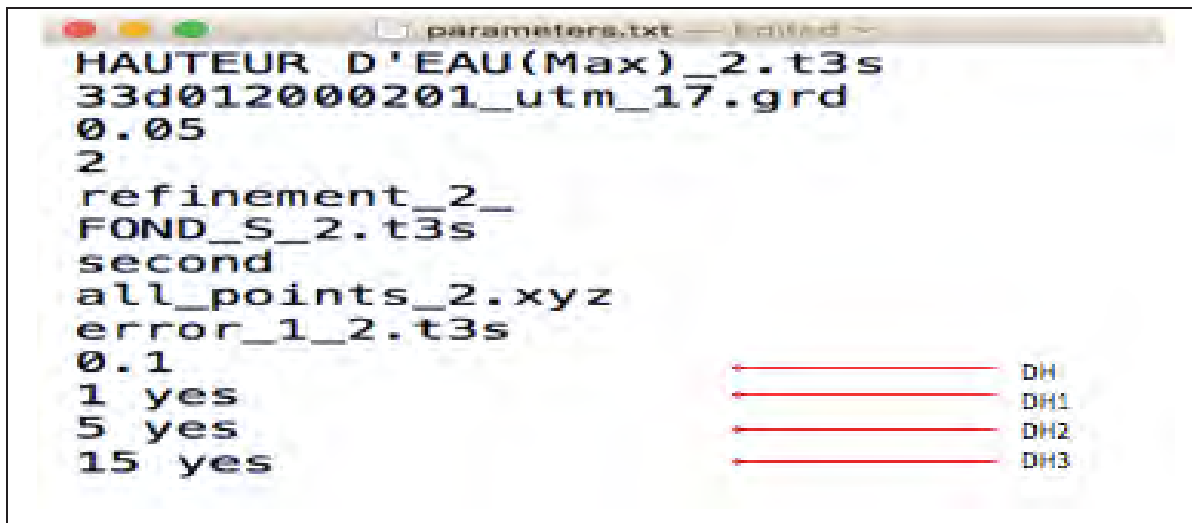


Figure h. Location new pointed added by the second algorithm to previous mesh.

3.3.3 Generating input file for second algorithm

After finishing the following steps, one should prepare a text file named "parameters.txt". More details are described in the methodology chapter of my thesis. We define some of the necessary parameters and variables for running the algorithm in this text file.

Again, here the 5th line and the 8th line are the names of output files and can be modified with the user. In the following example, the next mesh will be generated using the list of all points “second_refinement_points.xyz”. Here you can see a sample “parameters.txt” file:



```

HAUTEUR D'EAU(Max)_2.t3s
33d012000201_utm_17.grd
0.05
2
refinement_2_
FOND_S_2.t3s
second
all_points_2.xyz
error_1_2.t3s
0.1
1 yes
5 yes
15 yes
  
```

Figure i. Example of parameter text file for running the second algorithm.

As it is shown in the figure i, the word “yes” in the last three lines of parameter.txt file means the user like to choose multiple points. If the user was satisfied with the results without choosing multiple points, he should write “No” in front of the last 3 line.

Using the appropriate weight factor ‘ α ’ can help reaching the results sooner. For choosing weight factor please refer to my thesis. As a first guess, we use $\alpha=2$.

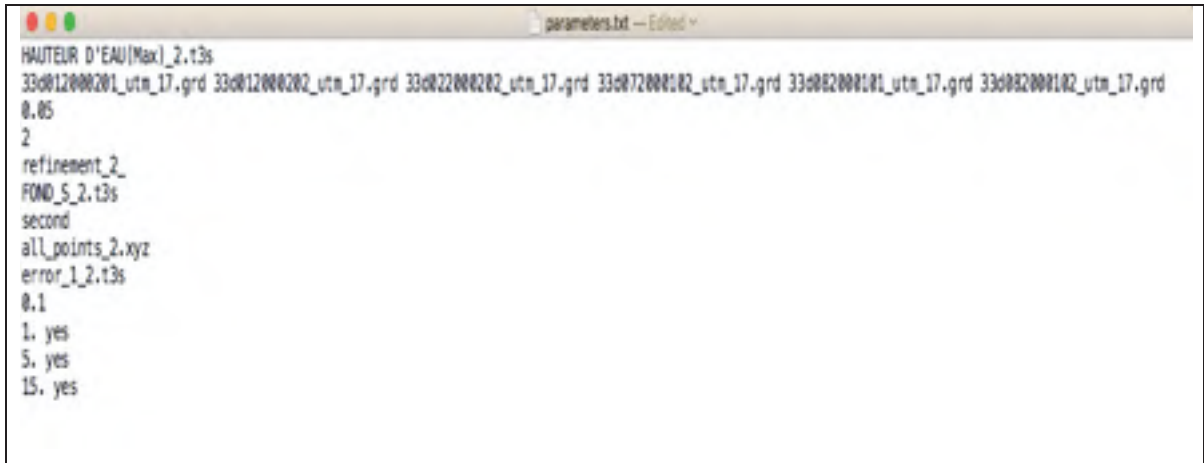
After Generating this parameters.txt file we should run the “TMA_Serial.py”.

3.3.4 Parallel mode

After generating the file “error_1_2.t3s”, one might see the errors exist just in a specific domain. So instead of applying all the LIDAR data, it is better to use just the topo-metric data of that region.

So, based on the results of the second algorithm, we can understand very well, which part of the domain needs finer meshes. In this case we can just apply the LIDAR data which is related

to those regions. As we explained before, two separated codes for dividing our LIDAR data can be used. One of them divides the domain based on the number of available points in that file, and the other one divides the domain in some smaller rectangular domains. These python codes are named “divided_rectangular.py” and “divided_line_by_line.py”. The user guide for these codes has been prepared in appendix A.



```

HAUTEUR D'EAU(Max)_2.t3s
33d012000201_utm_17.grd 33d012000202_utm_17.grd 33d022000202_utm_17.grd 33d072000102_utm_17.grd 33d082000101_utm_17.grd 33d082000102_utm_17.grd
0.05
2
refinement_2
FOND_5_2.t3s
second
all_points_2.xyz
error_1_2.t3s
0.1
1. yes
5. yes
15. yes

```

Figure j. Example of parameter text file for running second algorithm in parallel with 6 processor.

For the second algorithm, we should write the following command in Terminal or “cmd”
mpirun -np 6 python TMA_SA_Version1_Parallel.py

Here you can find the example of log files generated after running the second algorithm with each processor. It is clear, that if you run the code with one processor, it will act like running the code in serial.

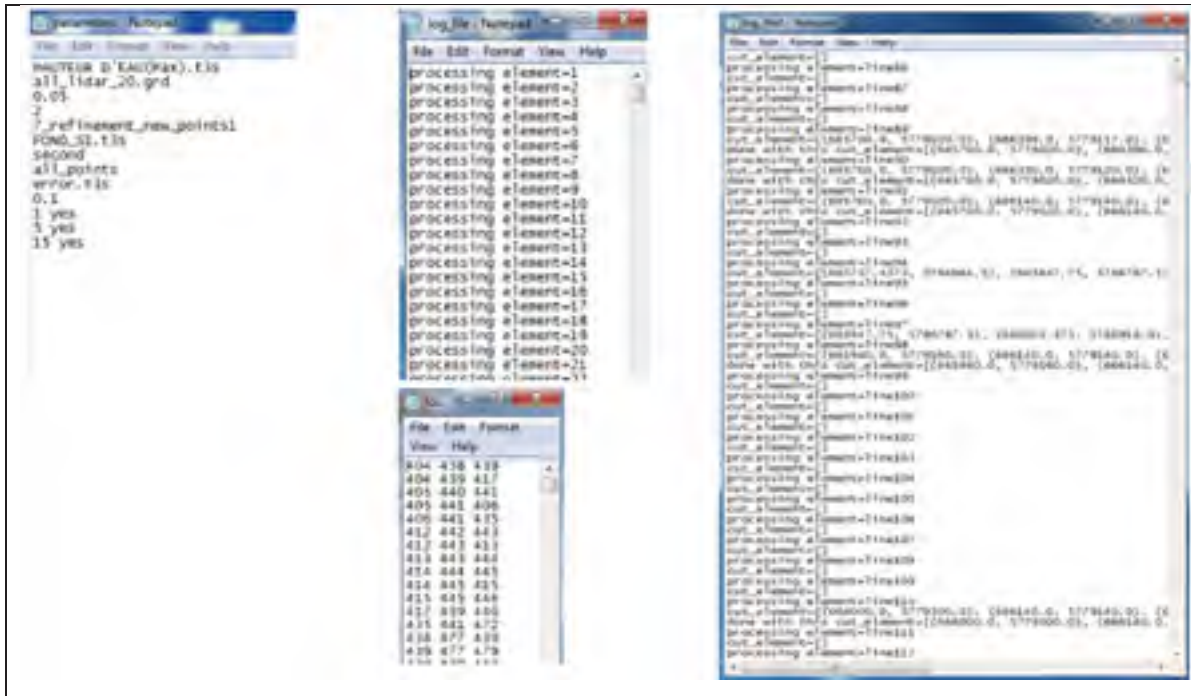


Figure k. Log Files generated with the second algorithm.

It is important to remember, in order to avoid overwriting the results, the “OUTPUT_FILES” folder should always be empty before running the “TOPO_MESH_ADAPTOR”.

3.4 Generating the mesh file and running TELEMAC after each refinement

After each refinement a new mesh is generated using Blue Kenue as follows:

- generate triangular mesh with the file which contains ‘all points’, and the outline of the domain i.e. “BND_EASTM.i2s”;
- create a SELAFIN file;
- impose the boundary condition.

Now that we have all the necessary files, we generate a new CASE file (CAS_EASTM.TXT) with the text format for the TELEMAC by declaring the necessary properties and the files related to the new meshes (SELAFIN file, Boundary Condition file, ...). For running TELEMAC, again we go to the directory of CASE file and use the following command:

telemac2d.py CAS_EASTM.TXT

```

CAS_EASTM.TXT — Edited
TITRE = 'TELEMAC 2D ; RIVIERE EASTMAIN; TOPOGRAPHIE LIDAR'
/
PROCESSEURS PARALLELES : 24
/
/
FICHIER DES CONDITIONS AUX LIMITES : CL_GEO_EASTM.TXT
FICHIER DE GEOMETRIE : GED_EASTM.SLF
FICHIER DES FRONTIERES LIQUIDES : EASTM_DEBIT.TXT
FICHIER FORTRAN : EASTM_COINI.FOR
FICHIER DES RESULTATS : RES_EASTM.SLF
/
/-----/
/----- POSTPROCESSING -----/
/-----/
/
SUITE DE CALCUL = NON
VARIABLES POUR LES SORTIES GRAPHIQUES = U,V,H,S,B,W
DUREE DU CALCUL : 86400.
PAS DE TEMPS VARIABLE : OUI
NOMBRE DE COURANT SOUHAITE : 1.0
/NOMBRE DE PAS DE TEMPS = 28800 PAS DE TEMPS = 3.
PERIODE POUR LES SORTIES GRAPHIQUES = 300
PERIODE DE SORTIE LISTING = 300
INFORMATIONS SUR LE SOLVEUR : OUI BILAN DE MASSE : OUI
/-----/
/----- CONDITIONS INITIALES -----/
/-----/
CONDITIONS INITIALES : 'PARTICULIERES'
/FICHIER DU CALCUL PRECEDENT : './COINI_EASTM.SLF'
/-----/
/----- CONDITIONS AUX LIMITES -----/
/-----/
DEBITS IMPOSES : 0.0; 150. COTES IMPOSEES : 1.15; 0.0
/-----/
/----- PARAMETRAGE NUMERIQUE -----/
/-----/
PRECONDITIONNEMENT = 2
BANCS DECOUVRANTS = VRAI
FORME DE LA CONVECTION = 1;5
OPTION DE SUPG = 0;0
MAXIMUM D'ITERATIONS POUR LE SOLVEUR = 200
SOLVEUR = 7 OPTION DU SOLVEUR = 3 PRECISION DU SOLVEUR = 0.0001
STOCKAGE DES MATRICES : 3 PRODUIT MATRICE-VECTEUR : 1
IMPLICITATION POUR LA HAUTEUR : 0.6 IMPLICITATION POUR LA VITESSE = 0.6
MASS-LUMPING SUR H = 1.
CLIPPING DE H = NON LOI DE FROTTEMENT SUR LE FOND = 3
COEFFICIENT DE FROTTEMENT : 30.
MODELE DE TURBULENCE = 1 COEFFICIENT DE DIFFUSION DES VITESSES = 1.
EQUATIONS : 'SAINT-VENANT EF'
OPTION DE TRAITEMENT DES BANCS DECOUVRANTS : 1
TRAITEMENT DU SYSTEME LINEAIRE : 1
DISCRETISATIONS EN ESPACE : 11;11

/- DISCRETISATIONS EN ESPACE : 12 ; 11
/- FORME DE LA CONVECTION : 6;5 MODELE DE TURBULENCE = 1
/- OPTION DE SUPG : 1;2
/- LOI DE FROTTEMENT SUR LE FOND = 3 COEFFICIENT DE FROTTEMENT = 40.
/- PROPAGATION = OUI COEFFICIENT DE DIFFUSION DES VITESSES = 0.005
/- IMPLICITATION POUR LA HAUTEUR = 0.6 IMPLICITATION POUR LA VITESSE = 0.6
/- SOLVEUR : 1 PRECONDITIONNEMENT : 2 ORDRE DU TIR INITIAL POUR H : 1
/- PRECISION DU SOLVEUR = 1.E-5
/- OPTION DU SOLVEUR : 1

&FIN

```

Figure h. Example of CAS_EASTM.TXT.

4) Results

Here you can see the change in a cross section and maximum water depth of “Riviere a lê pêche” in the following figures.

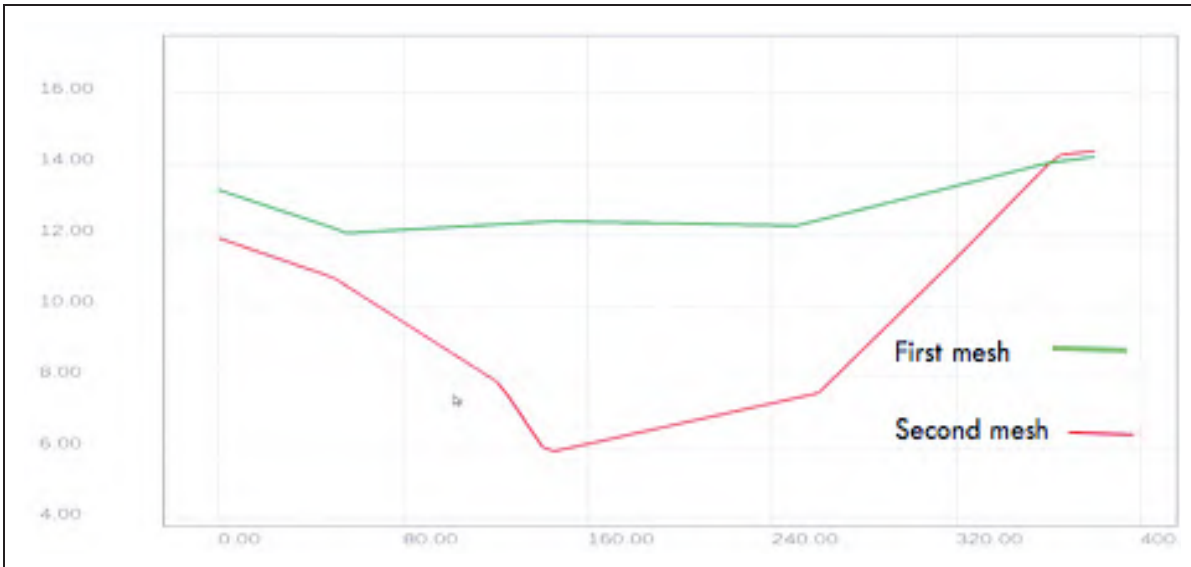


Figure 1. Change of cross section for mesh generated by TMA_SA_Version1_Parallel.py.

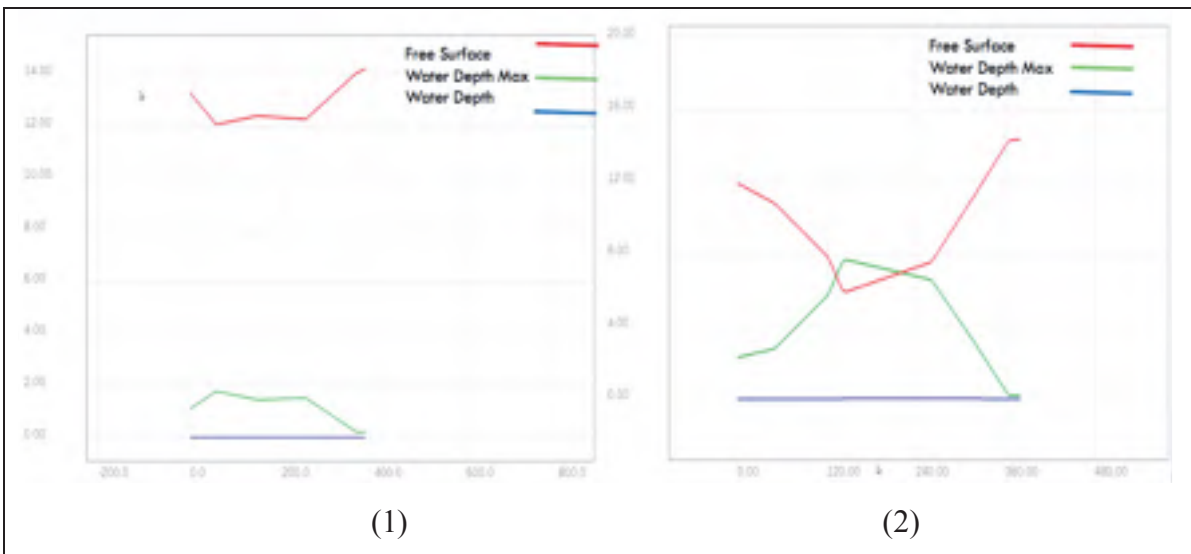


Figure m. Comparing the change of maximum water depth in 1) First refinement and 2) Second refinement.

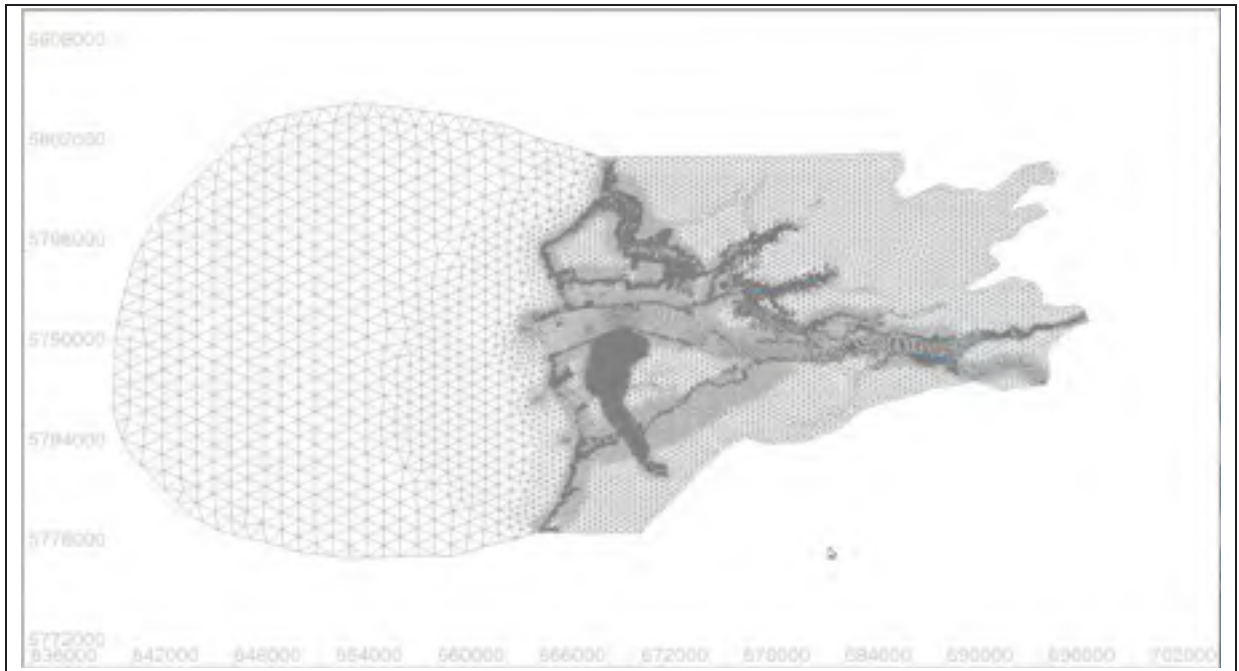


Figure n. Illustration of error_16_17.t3s. This figure gives us some information about where the refinement should be performed. So, in some case we may also do the refinements manually.

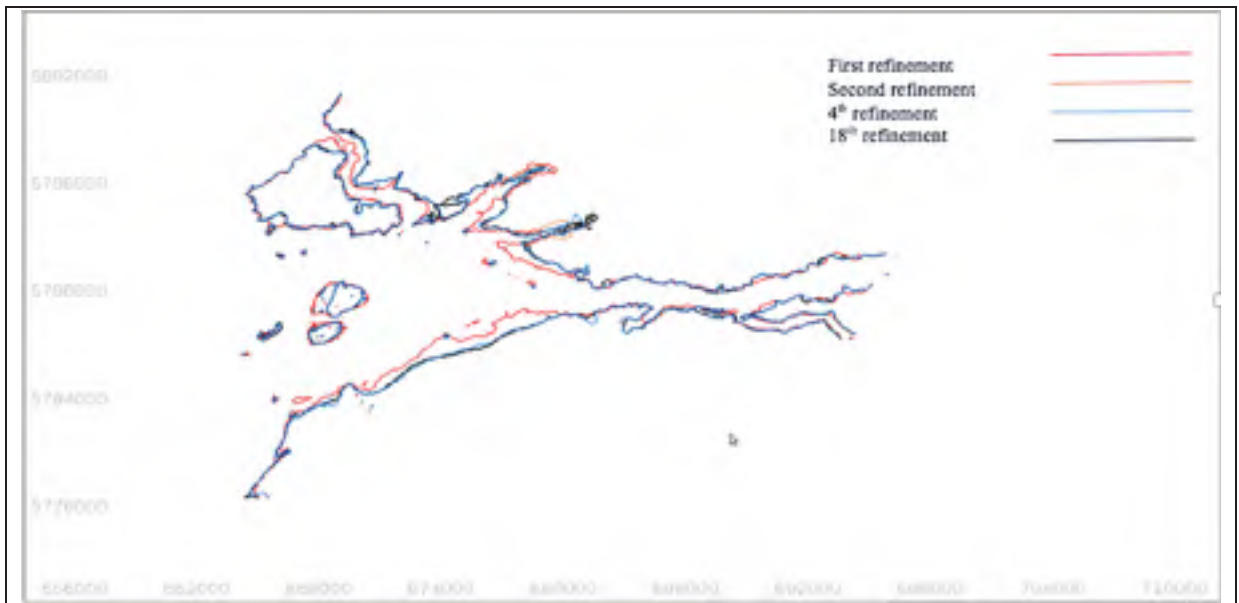


Figure o. mesh refinement occurs across the inundation line.

5) Summary

An automatic mesh refinement with TMA which has been done along the shoreline improves the simulation results over the inundation limit. Modifying the friction coefficient along the shoreline can also be considered to analyze more accurately the water depth value in a river cross section.

Appendix A

The following python code has been employed in TMA:

« **Cut_elements.py** »:

has been written to find the cut-elements in dam breaks simulation

« **Interpolation.py** »:

has been written to do the triangular interpolation for calculating the water depth for points inside each element.

« **Inside_polygon.py** »:

Is used to identify the points inside each element;

« **sum_points.py** »:

which has been used to collect all the new points found by the algorithms with the previous mesh nodes'

« **TMA_Serial.py** »:

is the command file for using both algorithms in serial mode;

« **TMA_FA_Parallel.py** »:

is the code for running the first algorithm in parallel;

« **TMA_SA_Version1_Parallel.py** »:

is the code for running the second algorithm in parallel;

« **TMA_SA_RandomPoints_Parallel.py** »:

This code snippet can be used for running the algorithm by selecting multiple random point inside each element. We can use this code same as “TMA_SA_Version1_Parallel.py”. You can find more details about this algorithm in the following figure.

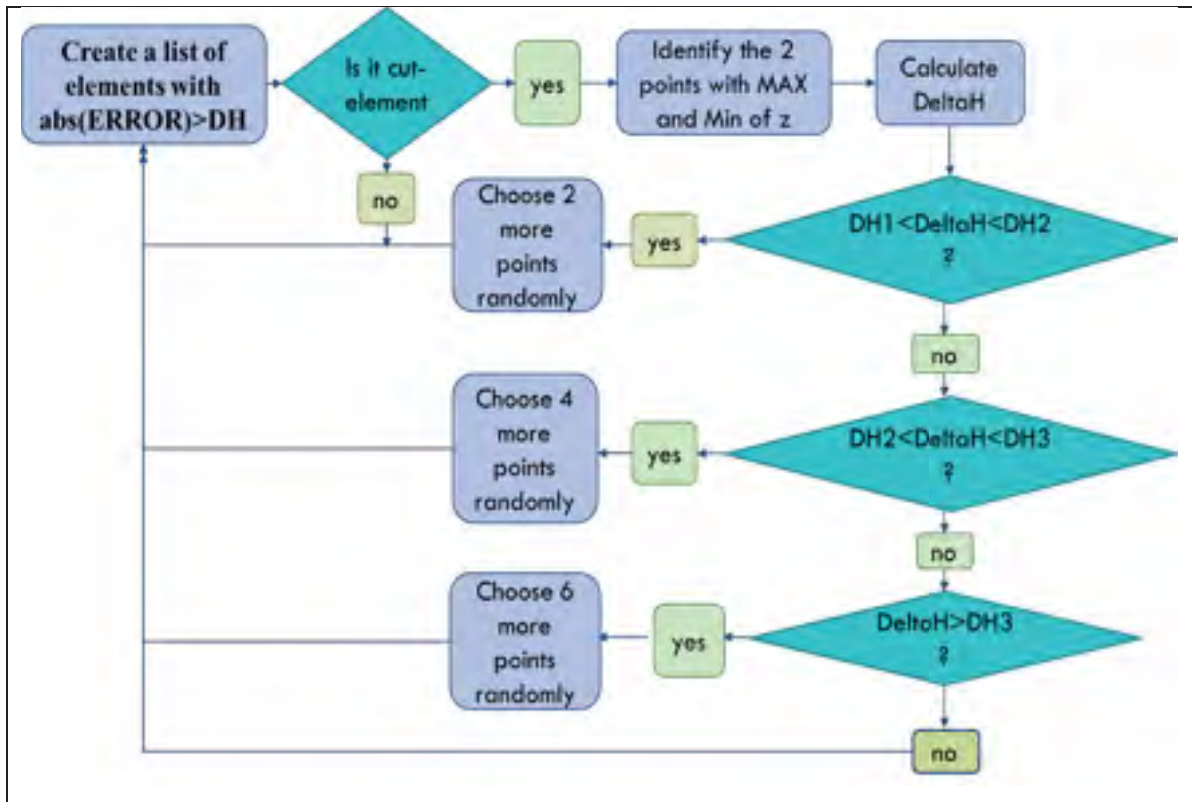


Figure p. Automatic mesh refinement algorithm for choosing nodes randomly.

« TMA_4.py »:

With this code like before, we select the two dry points inside cut-elements with minimum and maximum value of elevation (z). In addition, we tried to find the points in the middle of edges of each element. However, TELEMAC couldn't use the files generated with these nodes. The elevation for the points in the middle of the edges, considered to zero. So, it is clear that we must interpolate our topo-metric data again on the meshes generated from TMA_4.py.

Here we are going to explain some of the utilities codes which can help in preprocessing of the algorithm.

«**CollectNodes.py**»:

This code can be used to collect the nodes of any mesh generated by Blue Kenue;

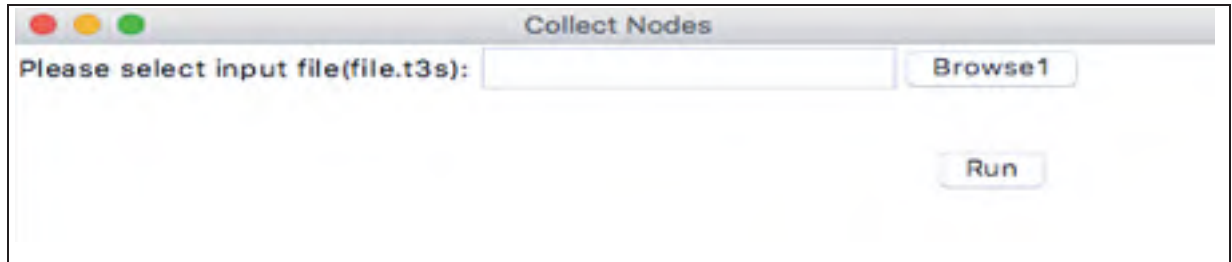


Figure q. Illustration of CollectNodes.py.

You just need to brows the *.t3s file and click on “run” to have the nodes of your mesh. The result will be in the directory of “CollectNodes.py”

«**Merge_mesh.py**»:

This file will be use to add or subtract the nodes of one mesh file generated by Blue Kenue (*.t3s) to the other one.

If you want to add the nodes of two different mesh files you should first select the directory you like to save the result by bottom “Browse2”, and then select the desire mesh files one by one, with the bottom “Browse1”, then click on “run” bottom. If you like to subtract the nodes of a mesh file from the other one, again you should select your original file with “browse2” and then select the files which you want to reduce its nodes from the original file one by one, by “browse3” and click on the “subtract” bottom.

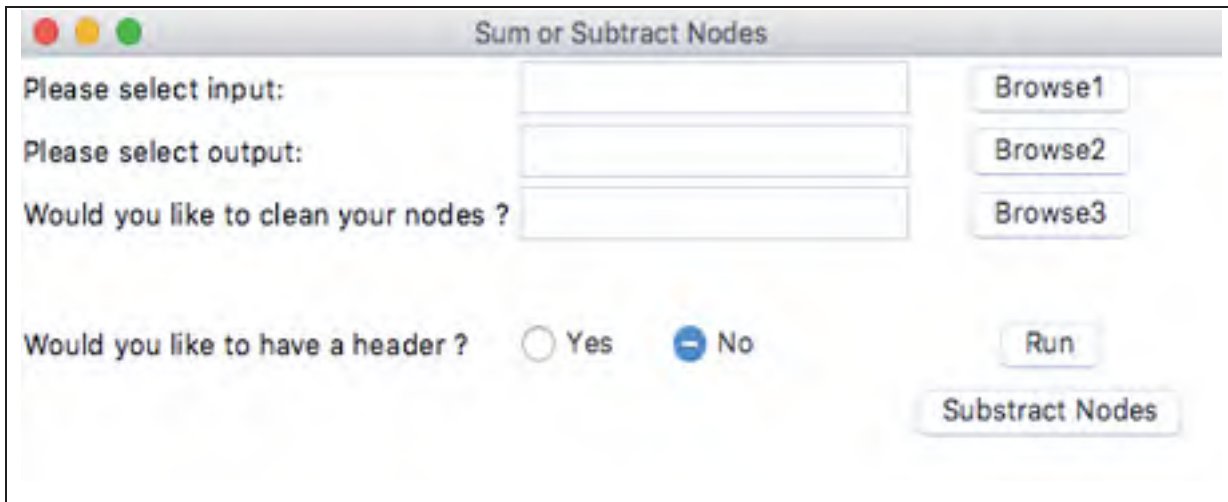


Figure r. Illustration of Merge_mesh.py.

«divided_rectangular.py»:

This file is for partitioning topo-metric files. For using this code, you need to generate a text file in a source directory of this code and name it “divided_rectangular.txt”. In the first line of this file, you should write the address of the topo-metric file which you like to partition it, the second line is the number of partitions in x direction, and the third line is the number of partitions in y direction. In the follow you can see an example of generating “divided_rectangular.txt” for partitioning a topo-metric file in to 20 partitions.

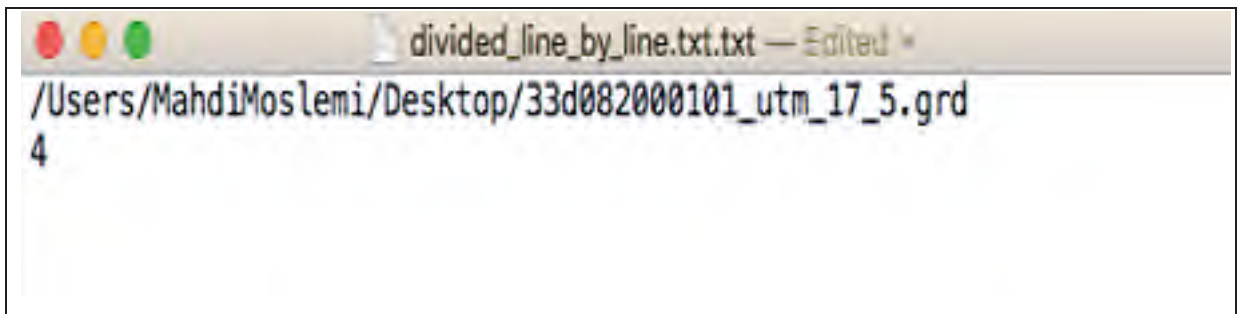


Figure s. an example of divided_rectangulat.txt.

«divided_line_by_line.py»:

This code is for cutting a big topo-metric file into smaller ones. For using this code, you create a text file in the source directory of this code and name it “divided_line_by_line.txt”. In the first line of this text file you write the complete address of a topo-metric file which you want

divided, and in the second line you specify the number of files created by dividing the original file; for example, if you like to divide your files in to 4 smaller files, you generate the “divided_line_by_line.py” as follow:

A screenshot of a text editor window titled "divided_line_by_line.txt.txt — Edited". The window contains two lines of text: the first line is the file path "/Users/MahdiMoslemi/Desktop/33d082000101_utm_17_5.grd" and the second line is the number "4".

```
divided_line_by_line.txt.txt — Edited *  
/Users/MahdiMoslemi/Desktop/33d082000101_utm_17_5.grd  
4
```

Figure t. an example of divided_line_by_line.txt.

REFERENCES

- 2017 Quebec floods. (2017, 2017, December 29). Wikipedia, The Free Encyclopedia. Retrieved from https://en.wikipedia.org/w/index.php?title=2017_Quebec_floods&oldid=817612468
- Askes, H., & Sluys, L. J. (2000). Remeshing strategies for adaptive ALE analysis of strain localisation. *European Journal of Mechanics-A/Solids*, 19(3), 447-467.
- Barbosa, H. J., & Hughes, T. J. (1991). The finite element method with Lagrange multipliers on the boundary: circumventing the Babuška-Brezzi condition. *Computer Methods in Applied Mechanics and Engineering*, 85(1), 109-128.
- Barrenechea, G. R., & Chouly, F. (2012). A local projection stabilized method for fictitious domains. *Applied Mathematics Letters*, 25(12), 2071-2076.
- Belytschko, T., Liu, W. K., Moran, B., & Elkhodary, K. (2013). *Nonlinear finite elements for continua and structures*. John Wiley & Sons.
- Berger, M. J., & Colella, P. (1989). Local adaptive mesh refinement for shock hydrodynamics. *Journal of computational Physics*, 82(1), 64-84.
- Berger, M. J., & Oliger, J. (1984). Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of computational Physics*, 53(3), 484-512.
- Blain, C. A., & Massey, T. C. (2005). Application of a coupled discontinuous–continuous Galerkin finite element shallow water model to coastal ocean dynamics. *Ocean Modelling*, 10(3), 283-315.
- Bunya, S., Kubatko, E. J., Westerink, J. J., & Dawson, C. (2009). A wetting and drying treatment for the Runge–Kutta discontinuous Galerkin solution to the shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, 198(17), 1548-1562.
- Burman, E., & Hansbo, P. (2010). Fictitious domain finite element methods using cut elements: I. A stabilized Lagrange multiplier method. *Computer Methods in Applied Mechanics and Engineering*, 199(41-44), 2680-2686.

- Canada, N. R. (2010, 2010-12-02). Flooding events in Canada: Quebec. *Canada*. Retrieved from <https://www.canada.ca/en/environment-climate-change/services/water-overview/quantity/floods/events-quebec.html>
- Codina, R., & Baiges, J. (2009). Approximate imposition of boundary conditions in immersed boundary methods. *International journal for numerical methods in engineering*, *80*(11), 1379-1405.
- Codina, R., Houzeaux, G., Coppola-Owen, H., & Baiges, J. (2009). The fixed-mesh ALE approach for the numerical approximation of flows in moving domains. *Journal of computational Physics*, *228*(5), 1591-1611.
- Dai, M., & Schmidt, D. P. (2005). Adaptive tetrahedral meshing in free-surface flow. *Journal of computational Physics*, *208*(1), 228-252.
- Díez, P., & Huerta, A. (1999). A unified approach to remeshing strategies for finite element h-adaptivity. *Computer Methods in Applied Mechanics and Engineering*, *176*(1-4), 215-229.
- Donea, J., Huerta, A., Ponthot, J.-P., & Rodriguez-Ferran, A. (2004). Encyclopedia of Computational Mechanics Vol. 1: Fundamentals., Chapter 14: Arbitrary Lagrangian-Eulerian Methods. In: Wiley & Sons.
- Edom, E. (2008). *Numerical calculation of the dam-break Riemann problem with a detailed method and comparison with a simplified method*. Diploma Thesis of Leibniz Universität Hannover,
- Egelja, A., Schäfer, M., & Durst, F. (1998). An adaptive grid Eulerian method for the computation of free surface flows. *International Journal of Computational Fluid Dynamics*, *10*(3), 213-224.
- Fondelli, T., Andreini, A., & Facchini, B. (2015). Numerical simulation of dam-break problem using an adaptive meshing approach. *Energy Procedia*, *82*, 309-315.
- Gardin, E. (2017). Blue Kenue™: Software tool for hydraulic modellers. *National Research Council Canada*. Retrieved from https://www.nrc-cnrc.gc.ca/eng/solutions/advisory/blue_kenue_index.html
- Gardin, E. (2017-04-26). Blue Kenue™: Software tool for hydraulic modellers. *NRC-CNRC*.

- George, D. (2013). Modeling Hazardous, Free-Surface Geophysical Flows with Depth-Averaged Hyperbolic Systems and Adaptive Numerical Methods. In *Computational Challenges in the Geosciences* (pp. 25-48): Springer.
- George, K., & Thomas, H. (2014). Simulation of groundwater flow based on adaptive mesh refinement.
- Glowinski, R., Du, Q., Hintermüller, M., & Suli, E. (2017). *Handbook of Numerical Methods for Hyperbolic Problems: Applied and Modern Issues* (Vol. 18): Elsevier.
- Glowinski, R., Pan, T. W., Hesla, T. I., Joseph, D. D., & Périaux, J. (1999). A distributed Lagrange multiplier/fictitious domain method for flows around moving rigid bodies: application to particulate flow. *International Journal for Numerical Methods in Fluids*, 30(8), 1043-1066.
- Hansbo, P., & Larson, M. G. (2002). Discontinuous Galerkin methods for incompressible and nearly incompressible elasticity by Nitsche's method. *Computer Methods in Applied Mechanics and Engineering*, 191(17-18), 1895-1908.
- Hirt, C. W., & Nichols, B. D. (1981). Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of computational Physics*, 39(1), 201-225.
- Hoppe, C., & Krömker, S. (2009). Adaptive meshing and detail-reduction of 3D-point clouds from laser scans. *International archives of photogrammetry, remote sensing and spatial information sciences*, 38, 5.
- Hossaini, N. (2003). *Simulation des problèmes de bris de barrage par des solveurs de Riemann*. École de technologie supérieure,
- Jahandar Malekabadi, M., & Kalateh, F. (2017). Gradually varied flow profile based on adaptive pattern. *ISH Journal of Hydraulic Engineering*, 1-9.
- Jakeman, J. (2006). On numerical solutions of the shallow water wave equations. *B. Sc. Hons. thesis, Australian National University*.
- Kesserwani, G., & Liang, Q. (2012). Locally limited and fully conserved RKDG2 shallow water solutions with wetting and drying. *Journal of scientific computing*, 50(1), 120-144.

- Lai, M.-C., & Peskin, C. S. (2000). An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of computational Physics*, 160(2), 705-719.
- Lai, W., & Khan, A. A. (2014). Discontinuous Galerkin Method for 1D Shallow Water Flows in Natural Rivers. *Engineering Applications of Computational Fluid Mechanics*, 6(1), 74-86. doi:10.1080/19942060.2012.11015404
- Landry, J., Soulaïmani, A., Luke, E., & Ali, A. B. H. (2016). Robust moving mesh algorithms for hybrid stretched meshes: Application to moving boundaries problems. *Journal of computational Physics*, 326, 691-721.
- Legat, V., & Oden, J. T. (1995). An adaptive hp-finite element method for incompressible free surface flows of generalized Newtonian fluids. In *Theoretical, Experimental, and Numerical Contributions to the Mechanics of Fluids and Solids* (pp. 643-678): Springer.
- Litrico, X., & Fromion, V. (2009). *Modeling and control of hydrosystems*: Springer Science & Business Media.
- Mapfrappe. (2012). hydroelectric-dams-and-reservoirs. *Compare areas on Google Maps*. Retrieved from <http://mapfrappe.blogspot.ca/2012/06/hydroelectric-dams-and-reservoirs.html>
- Marqués, J. M. F. Introduction to the Finite Volumes Method. Application to the Shallow Water Equations.
- Mestreau, E., Löhner, R., & Aita, S. (1993). *TGV tunnel entry simulations using a finite element code with automatic remeshing*. Paper presented at the 31st Aerospace Sciences Meeting.
- Mittal, R., & Iaccarino, G. (2005). Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37, 239-261.
- Munson, B. R., Young, D. F., Okiishi, T. H., & Huebsch, W. W. (1990). Fundamentals. of Fluid. *Mechanics*.
- Nguyen, V. P., Rabczuk, T., Bordas, S., & DufLOT, M. (2008). Meshless methods: a review and computer implementation aspects. *Mathematics and Computers in Simulation*, 79(3), 763-813.

- Nitsche, J. (1971). *Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind*. Paper presented at the Abhandlungen aus dem mathematischen Seminar der Universität Hamburg.
- Oñate, E., Idelsohn, S. R., Del Pin, F., & Aubry, R. (2004). The particle finite element method—an overview. *International Journal of Computational Methods*, 1(02), 267-307.
- Peskin, C. S. (1972). Flow patterns around heart valves: a numerical method. *Journal of computational Physics*, 10(2), 252-271.
- Rath, S., & Pasche, E. (2004). Hydrodynamic Floodplain Modeling based on high-resolution LiDAR measurements. In *Hydroinformatics: (In 2 Volumes, with CD-ROM)* (pp. 486-493): World Scientific.
- Schott, B., & Wall, W. (2014). A new face-oriented stabilized XFEM approach for 2D and 3D incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 276, 233-265.
- Soulaïmani, A., Fortin, M., Dhatt, G., & Ouellet, Y. (1991). Finite element simulation of two- and three-dimensional free surface flows. *Computer Methods in Applied Mechanics and Engineering*, 86(3), 265-296.
- Soulaimani, A., & Saad, Y. (1998). An arbitrary Lagrangian-Eulerian finite element method for solving three-dimensional free surface flows. *Computer Methods in Applied Mechanics and Engineering*, 162(1-4), 79-106.
- Steininger, A. (2014). *Dam overtopping and flood routing with the TREX watershed model*. Colorado State University.
- Tchamen, G., & Kahawita, R. (1998). Modelling wetting and drying effects over complex topography. *Hydrological Processes*, 12(8), 1151-1182.
- Tennessee, U. o. Water waves. *Elements of Physics*. Retrieved from http://labman.phys.utk.edu/phys221core/modules/m12/Water_waves.html.
- Tsubaki, R., & Fujita, I. (2010). Unstructured grid generation using LiDAR data for urban flood inundation modelling. *Hydrological Processes: An International Journal*, 24(11), 1404-1420.

- Wikipedia contributors. (2018, 3 Feb. 2018). Dam failure. *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/Dam_failure
- Zagonjoli, M. (2007). *Dam break modelling, risk assessment and uncertainty analysis for flood mitigation* (Doctoral dissertation, UNESCO-IHE, Institute for Water Education).
- Zokagoa, J.-M., & Soulaïmani, A. (2010). Modeling of wetting–drying transitions in free surface flows over complex topographies. *Computer Methods in Applied Mechanics and Engineering*, 199(33-36), 2281-2304. doi:10.1016/j.cma.2010.03.023