Comparison of LISFLOOD-FP and Delft3D FM for large scale flood inundation modeling

30 November 2016

CSE, Applied Mathematics, EEMCS

J.G.P.M.M. Peterse 1390791

Technische Universiteit Delft

Acknowledgements

During this thesis many people supported me. This page is dedicated to them. For starters, I would like to thank my supervisors Michalis Vousdoukas, Kees Vuik and Martin Verlaan. Michalis, thank you for allowing me the possibility to do a large part of my thesis at JRC, I had a great time and an interesting look inside the JRC. Kees, thank you for your relentless support, both during times when it went well and times when it could have gone better. And Martin, thank you for the endless feedback on the final versions of my report, the long discussions, the always optimistic attitude. The positive atmosphere you create helped me many times.

Like any author of any book I have ever read, I will also address some words to my mother. This work would have been a lot more impossible without both her mental and financial support over the years. I hope you know how much appreciated this help is. The same holds for my two brothers Stephan and Remy, well except for the financial support.

Many of my friends supported me during this not always easy process as well. Some by talking about the thesis, most by making it possible to forget it completely every now and then. Both methods of help are appreciated equally. I would especially like to thank Ines, Tijn, Dorien, Claire, Eline, MC, Esther, Ludo, Tamara, Peter, Maria, Bob and Joran (there is no order of importance since the list is randomized).

And of course Nadja, for both your mental and textual support I am grateful.

I was not only supported by people. Three institutes helped me to achieve the final part of my master program. Without JRC, Deltares and the Technical University of Delft this project would not have been possible. I would like to thank all three parties for the opportunity they gave me.

Contents

List of variables

t	Time [s]
x	Horizontal direction from west to east [m]
y	Horizontal direction from south to north [m]
z	Vertical direction [m]
ζ	Water level [m], relatively to the reference plane
h	Water depth (total fluid column height) [m]
b	Bed level elevation [m]
$\boldsymbol{u}$	Velocity vector [m/s]
u	x-Direction of the velocity vector [m/s]
v	y-Direction of the velocity vector [m/s]
w	z-Direction of the velocity vector [m/s]
g	Grativy [m/s^2]
ρ	Density [kg/m^3]
ν	Viscosity [kg/(ms)]
$\boldsymbol{\tau}$	Bottom friction

Chapter 1

Introduction

Living in coastal areas brings many opportunities. Since the beginning of history people have been building settlements near rivers, estuaries and coastlines. Water is a source of life: It enables agricultural facilities, can be employed to transport resources and human beings and is needed for humans to drink. However building settlements near oceans or large rivers also has downsides. One of those downsides is that the water sometimes reaches locations where it can cause severe damage, like flooding cities, industrial areas and farmland. In recent years, many of such floods have occurred. Some of them cost us dearly, and are paid with loss of capital or even with loss of life. For the year 2014, the overall losses from hydrological events (flood, mass movement) are estimated at $29 bn and around 5,000 lives by Munich RE[1].

One of those hydrological events are floods from the sea in coastal areas. Records of flooding of coastal areas goes back to at least 26 December, 838, when "a large area of the northwest of the Netherlands was hit by a storm tide". According to the annals of Bishop Prudentius of Troyes[2] there were 2,437 victims. A flood in 1530 (also in the Netherlands), known as St. Felix's Flood[3], had a reported death toll of over 100,000. A more recent example is the North Sea flood of 1953, which occurred in the night of 31 January and morning of 1 February. During this flood, the Netherlands, Belgium and the United Kingdom where affected. The reported death toll[4] for this flood was 2,551, with most of the deaths (1,836) occurring in the Netherlands. This event triggered major studies on the strengthening of coastal protection in the Netherlands. For example, after this event the Dutch committee "Deltacommissie (1953)" was established on 18 February of 1953. Although all of the above floods were severe, they are of a different scale compared to the flooding occurred around the Indian Ocean in 2004, on 26 December[5]. This flood had a reported death toll of 200,000+ people. These few examples of coastal flooding clearly show the danger and devastation that they cause for human lives.

In recent years, much research has been done to prevent this type of disaster. In order to do that, it is necessary to identify the threats and assess possibilities to eliminate them. One technique to identify the threats is to use computer models that simulate coastal flooding events. Those computer models give insight in what happens during a flood and can be used both for improving the coastal protection and to provide information on escape routes and safe locations in case a flood occurs. These models have been continuously improved throughout the last years. Using new techniques and computing power made it possible to model larger areas (for example the whole European continent) with higher resolution. During the last decades many computer programs have been developed in order to simulate this type of floods. Recent software packages such as 3Di (Stelling, 2012), LISFLOOD-FP (LFP, see Section 4.1 or (Bates, 2013)) or Delft3D FM (DFM, see Section 4.2 or (Kernkamp and Dam, 2011)) are becoming more capable of simulating flooding events. If it is possible to predict areas that will flood during certain events, measures can be taken to prevent the flood, for example by adapting the coastal landscape. Alternatively, people and properties can be evacuated in time by using escape routes and safe locations that were determined by those simulations.

[1] http://www.munichre.com/en/reinsurance/business/non-life/natcatservice/annual-statistics/index.html
[2] http://www.deltawerken.com/De-eerste-vloeden/227.html
[3] https://en.wikipedia.org/wiki/List_of_deadliest_floods
[4] https://en.wikipedia.org/wiki/North_Sea_flood_of_1953
[5] http://news.bbc.co.uk/2/hi/asia-pacific/4204385.stm

In Europe, one of the organizations tasked with the protection of the European citizens against this kind of threats is the Joint Research Centre (JRC[6]) in Ispra. One of their current projects is to generate accurate flood maps of Europe. This information can then be used to reduce the risk of flooding on locations where it matters most, which is a good starting point in order to help the European citizens to be more safe. Those flood maps are based on hydrological modeling, which is conducted by JRC using the software package LFP.

The topic of coastal flooding simulation is a vast one with many challenging aspects. When looking at large scale flooding simulation on a pan-European scale, this is even more so. One of the largest challenges is the amount of computational resources necessary to simulate floods on such scales. In order to restrict the calculations, researchers are constantly searching for ways to reduce the computational demands. This research will compare the simulation package currently used by the JRC with the more complex simulation package Delft3D FM developed by Deltares[7]. We first assess the differences between the two strategies and then we investigate how computational time can be saved. This research assesses the different approaches on flooding simulation and compare their computational speed, flood outcome and limitations on a representative coastal area. The scope of the project is to assess what actions can be undertaken to reduce the computational demands of the currently employed flooding simulations, without significantly affecting the results.

This study will focus on the differences between LFP and DFM, and on the effect of including and excluding the momentum advection term in both LFP and DFM models, applied to a coastal flooding simulation representative for the European coastline. The reason for this is that currently, JRC is running LFP without the momentum advection term, and it is important to analyze the consequences of neglecting this term. In addition, in DFM, the momentum advection term is solved explicitly, which can limit the maximum allowed time step. This can have a negative effect on the computational demands. If it can be shown that the momentum advection term is not of significant influence for coastal flooding simulations in Europe, and furthermore LFP and DFM give comparable results, then using DFM without momentum advection might reduce the necessary computational resources. However, if the momentum advection term is influencing the results considerably, then it is clear that the currently run LFP model is neglecting a relevant term. In summary, the goals of this research are the following:

- Compare LFP and DFM software packages in order to assess the difference in flood inundation extent and computational speed.
- Investigate the effect of the momentum advection term for LFP and DFM.
- Investigate other possibilities for speed up in both packages without affecting the flooded extent significantly.

1.1 Research questions

The introduction and focus of the study leads to the following research questions

1. How do the hydrological modeling packages LISFLOOD-FP (LFP) and Delft3D FM (DFM) compare in terms of

   Flooded extent?

   Computational time?

2. What is the effect of the momentum advection term on the

   Flooded extent?

   Computational time?

[6]The JRC is the in-house science service for the European Commission. Its main task is to provide scientific background for policy makers. For more information see https://ec.europa.eu/jrc/

[7]http://www.deltares.nl

3. What other settings can be tweaked in DFM in order to reduce the computational demand without affection the flooded extent?

In order to address the research questions, different models are run and compared. An overview of this can be seen in Figure 1.1. Whether or not the model outcomes resemble the real world is not in the scope. In order to compare the influences of the differences in the models, first in both LFP and DFM a model is created with the goal to have the two software packages running as similar as possible. Both software packages use settings without advection and run with a small time step of five seconds. Those simulations will be called the base simulations. Then for each of the packages different runs are conducted, to see the effect of this compared to the base simulations. Variations will be made in the chosen maximum time step, including additional terms of the differential equations, allowing different CFL conditions numbers in both LFP and DFM. DFM has more options to adjust than LFP. Various flux limiters and bed level types (see Section 4.2) are tested. The influence of including turbulence closure will also be addressed in DFM. This option is not available in LFP, which tacitly assumes that turbulence is negligible. The validity of this assumption will be tested. The different model runs are then assessed on

- Flooded extent
- (Maximum) water velocities
- Computational time

If those outcomes are comparable with the base simulation, the new model is said to be an alternative setup for the current setting. To help answering the question of whether or not modeling advection

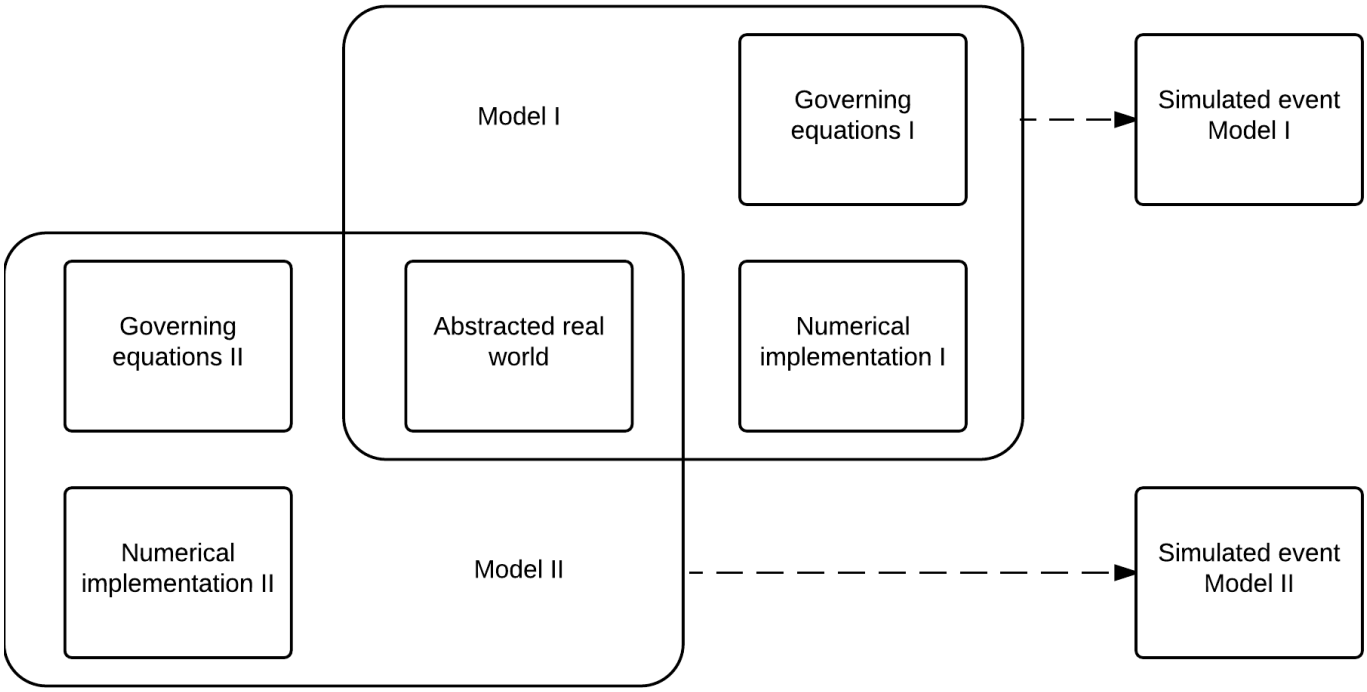


Figure 1.1: Schematic of proposed strategy to isolate the differences between models. The differences between the models have been reduced to the governing equations and the numerical implementation of thos equations.t

makes a difference in the current flooding simulations, a number of extra experiments will be conducted. The experiments that are run represent a one-dimensional flow with an obstruction. At this obstruction, the effects of modeling advection will be investigated. The effects will be measured in terms of water level and velocity across the jump.

Chapter 2

The shallow water equations

The shallow water equations (SWE) form the basis of this research. They are partial differential equations that are derived from the full Navier-Stokes equations (NS), and are very often used to solve shallow water flow. This chapter will give an overview of the shallow water equations, explain the terms of the equations, and list some frequently used simplifications. These simplifications are the cornerstone of this research, because we will investigate the effect of neglecting one of the terms in the SWE. We begin by giving a short introduction as to how the Navier-Stokes equations are derived and what assumptions are made in the process. Then, we continue with explaining how the SWE are derived.

2.1 The Navier-Stokes equations

The Navier-Stokes equations are one of the main pillars for simulations of fluid flow, they are build up with continuity equations. This means that the sum of the change of some quantity ϕ in a control volume Ω is equal to the amount gained/lost through the boundary $\partial\Omega$ of the control volume plus what is created/consumed by sources and sinks in this control volume. This principle is called "Reynolds transport theorem". For the Navier-Stokes equations, the quantities that are conserved are mass (m) and momentum (mu). Another possible quantity often used to conserve is energy. In general, we can express a continuity equation as

$$\frac{d}{dt}\int_{\Omega}\phi dV = -\int_{\partial\Omega}\phi\mathbf{u}\cdot\mathbf{n}dA + \int_{\Omega}sdV, \tag{2.1}$$

where the first term on the left describes the change of the quantity ϕ, the first term on the right is what is lost/gained through the boundary and the term on the right is the amount created/consumed in the so called control volume.
By using the divergence theorem

$$\int_{\partial\Omega}\phi\mathbf{u}\cdot\mathbf{n}dA = \int_{\Omega}\nabla\cdot(\phi\boldsymbol{u})\,dV, \tag{2.2}$$

and Leibniz's rule

$$\frac{d}{dt}\int_{\Omega}\phi dV = \int_{\Omega}\frac{\partial\phi}{\partial t}dV, \tag{2.3}$$

we get

$$\int_{\Omega}\left(\frac{\partial\phi}{\partial t} + \nabla\cdot(\phi\mathbf{u}) - s\right)dV = 0. \tag{2.4}$$

Since this has to hold for an arbitrary control volume V, this means

$$\frac{\partial\phi}{\partial t} + \nabla\cdot(\phi\mathbf{u}) = s, \tag{2.5}$$

2.1.1 Momentum equations

If we define the quantity ϕ of Equation 2.5 to be the so called mass flux, which is the product of mass density ρ and flow velocity $\mathbf{u}$, so $\phi = \rho\mathbf{u}$, we end up with the momentum equations, which consists of three equations in the three velocity directions.

$$\frac{\partial}{\partial t}(\rho\mathbf{u}) + \nabla \cdot (\rho\mathbf{u}\mathbf{u}) + \mathbf{s} = 0. \tag{2.6}$$

In this setting, $\mathbf{uu}$ is a dyad, which is a special case of a tensor product, and looks like

$$\mathbf{uu} = \begin{pmatrix} uu & uv & uw \\ vu & vv & vw \\ wu & wv & ww \end{pmatrix}, \tag{2.7}$$

$\mathbf{s}$ represents the amount created/destroyed, and consists of

$$\boldsymbol{s} = +\rho\boldsymbol{b} + \nabla \cdot \boldsymbol{T}, \tag{2.8}$$

where $\mathbf{b}$ represents the body forces, and $\mathbf{T}$ the stress tensor.

2.1.2 Mass equation

The same approach can be used for mass. We define the quantity ϕ from Equation 2.5 as $\phi := \rho$, where ρ is the mass density. Furthermore, we assume that there is no mass created/consumed in a control volume, so $s = 0$. Then we get the so called mass continuity equation (which is also often refered to as the continuity equation);

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\mathbf{u}) = 0. \tag{2.9}$$

Combining Equations 2.6 and 2.9 results in the equations:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho\mathbf{u}) & = 0, \\ \dfrac{\partial}{\partial t}(\rho\mathbf{u}) + \nabla \cdot (\rho\mathbf{u}\mathbf{u}) - \rho\boldsymbol{b} - \nabla \cdot \boldsymbol{T} & = 0. \end{cases} \tag{2.10}$$

This form of the Navier-Stokes equations is capable of describing turbulent flow. However to solve this turbulence, the scale on which the equations have to be solved is not desirable. In order to account for this, the turbulence part can be taken out of the equations and solved separately. By replacing

$$u = \bar{u} + u', \tag{2.11}$$

where $\bar{u}$ is some slowly varying mean value and u' is the turbulent part, we can rewrite the equations in 2.10. If we take averages and drop the bar, by adding a term that accounts for the turbulence

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho\mathbf{u}) & = 0, \\ \dfrac{\partial}{\partial t}(\rho\mathbf{u}) + \nabla \cdot (\rho\mathbf{u}\mathbf{u}) - \rho\boldsymbol{b} - \nabla \cdot \boldsymbol{T} - \nabla \cdot \boldsymbol{T}_{turb} & = 0. \end{cases} \tag{2.12}$$

The turbulence term $\boldsymbol{T}_{turb}$ is neglected in this research and it will not be investigated in more depth. The interested reader is referred to the literature.

Now we make the following assumptions

- The fluid is incompressible, this mean ρ is not dependent on p.
- The fluid is Newtonian, so we can write $\boldsymbol{T} = -p\boldsymbol{I} + \bar{\boldsymbol{T}}$, where $\bar{\boldsymbol{T}}$ is the viscous stress tensor defined by $\bar{\boldsymbol{T}}_{ij} = \mu\left(\dfrac{\partial u_i}{\partial x_j} + \dfrac{\partial u_j}{\partial x_i}\right)$. μ is the shear viscosity of the fluid.

- Gravity is the only body force (so we neglect Coriolis forces), resulting in $\rho\boldsymbol{b} = \rho\boldsymbol{g}$.
- For realistic variations of temperature and salinity, only small changes in density occur. This means that the variation of density can be ignored everywhere except for the gravity term $\rho\boldsymbol{g}$. This approach is called the Boussinesq approximation.

The resulting Reynolds averaged Navier-Stokes equations are

$$\begin{cases} \nabla \cdot \boldsymbol{u} & = 0, \\ \rho \dfrac{\partial}{\partial t}(\boldsymbol{u}) + \rho \nabla \cdot (\boldsymbol{u}\mathbf{u}) & = -\nabla p + \rho \boldsymbol{g} + \nabla \cdot \bar{\boldsymbol{T}}. \end{cases} \tag{2.13}$$

The equations can also be written in less compact notation as

$$\begin{cases} \dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y} + \dfrac{\partial w}{\partial z} & = 0, \\ \rho \left[\dfrac{\partial u}{\partial t} + \dfrac{\partial uu}{\partial x} + \dfrac{\partial uv}{\partial y} + \dfrac{\partial uw}{\partial z} \right] & = -\dfrac{\partial p}{\partial x} + \dfrac{\partial \tau_{xx}}{\partial x} + \dfrac{\partial \tau_{xy}}{\partial y} + \dfrac{\partial \tau_{xz}}{\partial z}, \\ \rho \left[\dfrac{\partial v}{\partial t} + \dfrac{\partial vu}{\partial x} + \dfrac{\partial vv}{\partial y} + \dfrac{\partial vw}{\partial z} \right] & = -\dfrac{\partial p}{\partial y} + \dfrac{\partial \tau_{yx}}{\partial x} + \dfrac{\partial \tau_{yy}}{\partial y} + \dfrac{\partial \tau_{yz}}{\partial z}, \\ \rho \left[\dfrac{\partial w}{\partial t} + \dfrac{\partial wu}{\partial x} + \dfrac{\partial wv}{\partial y} + \dfrac{\partial ww}{\partial z} \right] & = -\dfrac{\partial p}{\partial z} + \dfrac{\partial \tau_{zx}}{\partial x} + \dfrac{\partial \tau_{zy}}{\partial y} + \dfrac{\partial \tau_{zz}}{\partial z} - \rho g. \end{cases} \tag{2.14}$$

2.2 Derivation of the shallow water equations

In this section, we derive the shallow water equations from the Navier-Stokes equations. We begin by listing the necessary boundary condition assumptions.

- No slip condition at the bottom $z = b$, so

$$u_b = v_b = 0 \text{ (which implies } w_b = 0) \tag{2.15}$$

- No normal flow at the bottom, so

$$u_b \frac{\partial b}{\partial x} + v_b \frac{\partial b}{\partial y} - w_b = 0 \tag{2.16}$$

- The bottom shear stress in the x-direction is defined as

$$\tau_{bx} = -\tau_{xx} \frac{\partial b}{\partial x} - \tau_{xy} \frac{\partial b}{\partial y} + \tau_{xz}, \tag{2.17}$$

  with τ_{bx} being the bottom friction. The y-direction is defined similarly.

- No relative normal flow at the surface $z = \zeta$,

$$\frac{\partial \zeta}{\partial t} + u \frac{\partial \zeta}{\partial x} + v \frac{\partial \zeta}{\partial y} - w_b = 0. \tag{2.18}$$

- p $= p_a$ at $z = \zeta$
- The surface shear stress is defined as

$$\tau_{\zeta x} = -\tau_{xx} \frac{\partial \zeta}{\partial x} - \tau_{xy} \frac{\partial \zeta}{\partial y} + \tau_{xz}, \tag{2.19}$$

  and similar for the y-direction.

Now, we consider the last line in Equation 2.14. Using the assumption of small vertical velocities, we can ignore all terms except the pressure gradient and the constant effect of gravity, which results in

$$\frac{\partial p}{\partial z} = -\rho g \rightarrow p = -\rho g z + p_0. \tag{2.20}$$

with the assumption that ρ is independent of z. At the water surface $z = \zeta$ the pressure is $p = p_a$. This results in

$$p_a = -\rho g \zeta + p_0 \rightarrow p_0 = \rho g \zeta + p_a, \tag{2.21}$$

which then gives the hydrostatic pressure distribution

$$p = \rho g \left(\zeta - z\right) + p_a. \tag{2.22}$$

Now we can write

$$\begin{cases} \dfrac{\partial p}{\partial x} = \rho g \dfrac{\partial \zeta}{\partial x} + g\left(\zeta - z\right)\dfrac{\partial \rho}{\partial x} + \dfrac{\partial p_a}{\partial x}, \\ \dfrac{\partial p}{\partial y} = \rho g \dfrac{\partial \zeta}{\partial y} + g\left(\zeta - z\right)\dfrac{\partial \rho}{\partial y} + \dfrac{\partial p_a}{\partial y} \end{cases}. \tag{2.23}$$

This means Equation 2.14 reduces to

$$\begin{cases} \dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y} + \dfrac{\partial w}{\partial z} & = 0, \\ \dfrac{\partial \rho u}{\partial t} + \dfrac{\partial \rho u u}{\partial x} + \dfrac{\partial \rho u v}{\partial y} + \dfrac{\partial \rho u w}{\partial z} & = -\rho g \dfrac{\partial \zeta}{\partial x} + \dfrac{\partial \tau_{xx}}{\partial x} + \dfrac{\partial \tau_{xy}}{\partial y} + \dfrac{\partial \tau_{xw}}{\partial z}, \\ \dfrac{\partial \rho v}{\partial t} + \dfrac{\partial \rho v u}{\partial x} + \dfrac{\partial \rho v v}{\partial y} + \dfrac{\partial \rho v w}{\partial z} & = -\rho g \dfrac{\partial \zeta}{\partial y} + \dfrac{\partial \tau_{yx}}{\partial x} + \dfrac{\partial \tau_{yy}}{\partial y} + \dfrac{\partial \tau_{yw}}{\partial z}, \\ \dfrac{\partial p}{\partial z} & = -\rho g \end{cases} \tag{2.24}$$

In order to reduce the 3D equations to 2D, we integrate the mass equation over the whole z-direction, resulting in

$$\begin{aligned} 0 &= \int_b^\zeta \nabla \cdot \boldsymbol{u} dz \\ &= \int_b^\zeta \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} dz. \end{aligned} \tag{2.25}$$

Applying Leibniz rule (only shown here for $\int_b^\zeta \frac{\partial u}{\partial x} dz$)

$$\int_b^\zeta \frac{\partial u}{\partial x} dz = \frac{\partial}{\partial x}\left(\int_b^\zeta u dz\right) - u_\zeta \frac{\partial \zeta}{\partial x} + u_b \frac{\partial b}{\partial x}, \tag{2.26}$$

to Equation 2.25, we get

$$\begin{aligned} 0 &= \int_b^\zeta \nabla \cdot \boldsymbol{u} dz \\ &= \frac{\partial}{\partial x}\left(\int_b^\zeta u dz\right) - u_\zeta \frac{\partial \zeta}{\partial x} + u_b \frac{\partial b}{\partial x} \\ &+ \frac{\partial}{\partial y}\left(\int_b^\zeta v dz\right) - v_\zeta \frac{\partial \zeta}{\partial y} + u_b \frac{\partial b}{\partial y} + w_\zeta - w_b. \end{aligned} \tag{2.27}$$

If we now insert the boundary conditions shown in Equations 2.16 and 2.18, we can rewrite Equation 2.27 as

$$\begin{aligned} 0 &= \int_b^\zeta \nabla \cdot \boldsymbol{u} dz \\ &= \frac{\partial h\hat{u}}{\partial x} + \frac{\partial h\hat{v}}{\partial y} + \frac{\partial \zeta}{\partial t} - w_\zeta + w_\zeta + w_b - w_b \\ &= \frac{\partial h\hat{u}}{\partial x} + \frac{\partial h\hat{v}}{\partial y} + \frac{\partial \zeta}{\partial t}, \end{aligned} \tag{2.28}$$

where $\hat{u}$ and $\hat{v}$ are defined as

$$\hat{u} = \frac{1}{h}\int_b^\zeta u dz, \qquad \hat{v} = \frac{1}{h}\int_{-b}^\zeta v dz. \tag{2.29}$$

The next step is to integrate the LHS of the momentum Equation 2.24 in the x-direction over the depth. This is done for each term separately. Note that ρ is divided out, since it is assumed constant.

$$\begin{aligned} \int_b^\zeta \frac{\partial u}{\partial t} dz &= \frac{\partial}{\partial t}\int_b^\zeta u dz - u_\zeta \frac{\partial \zeta}{\partial t} + u_b \frac{\partial b}{\partial t}, \\ \int_b^\zeta \frac{\partial u^2}{\partial x} dz &= \frac{\partial}{\partial x}\int_b^\zeta u^2 dz - u_\zeta^2 \frac{\partial \zeta}{\partial x} + u_b^2 \frac{\partial b}{\partial x}, \\ \int_b^\zeta \frac{\partial uv}{\partial y} dz &= \frac{\partial}{\partial y}\int_b^\zeta uv dz - u_\zeta v_\zeta \frac{\partial \zeta}{\partial y} + u_b v_b \frac{\partial b}{\partial y}, \\ \int_b^\zeta \frac{\partial uw}{\partial z} dz &= u_\zeta w_\zeta - u_b w_b \end{aligned} \tag{2.30}$$

Putting it all together yields

$$\begin{aligned} &\int_b^\zeta \frac{\partial u}{\partial t} + \frac{\partial uu}{\partial x} + \frac{\partial uv}{\partial y} + \frac{\partial uw}{\partial z} dz \\ &= \frac{\partial}{\partial t}\int_b^\zeta u dz + \frac{\partial}{\partial x}\int_b^\zeta u^2 dz + \frac{\partial}{\partial y}\int_b^\zeta uv dz \\ &+ u_\zeta \underbrace{\left[-\frac{\partial \zeta}{\partial t} - u_\zeta \frac{\partial \zeta}{\partial x} - v_\zeta \frac{\partial \zeta}{\partial y} + w_\zeta \right]}_{=0, \text{ see Eq 2.18}} \\ &+ u_b \left[\underbrace{\frac{\partial b}{\partial t}}_{=0} + \underbrace{u_{-b}\frac{\partial b}{\partial x} + v_{-b}\frac{\partial b}{\partial y} - w_b}_{=0, \text{ see Eq 2.16}} \right] \\ &= \frac{\partial}{\partial t}\int_b^\zeta u dz + \frac{\partial}{\partial x}\int_b^\zeta u^2 dz + \frac{\partial}{\partial y}\int_b^\zeta uv dz. \end{aligned} \tag{2.31}$$

Just as in the continuity equations, we use depth average values. In order to do this, we consider $\int_b^\zeta (u-\hat{u})(v-\hat{v})\, dz$, where we use the definitions introduced in Equation 2.29.

$$\begin{aligned} &\int_b^\zeta (u-\hat{u})(v-\hat{v})\, dz \\ &= \int_b^\zeta uv dz - \int_b^\zeta u\hat{v} dz - \int_b^\zeta \hat{u}v dz + \int_b^\zeta \hat{u}\hat{v} dz \\ &= \int_b^\zeta uv dz - \hat{v}\int_b^\zeta u dz - \hat{u}\int_b^\zeta v dz + \int_b^\zeta \hat{u}\hat{v} dz \\ &= \int_b^\zeta uv dz - \hat{u}\hat{v}h, \end{aligned} \tag{2.32}$$

which can be rewritten as

$$\int_b^\zeta uvdz = \hat{u}\hat{v}h + \int_b^\zeta (u-\hat{u})(v-\hat{v})\,dz. \tag{2.33}$$

The second term on the RHS is difficult to estimate, since it requires knowledge about the distribution of velocity in the vertical direction, which is not solved for in the current set of equations. Alternatively, it is often modeled as eddy viscosity, or ignored, which is done in this research. This means

$$\begin{aligned}&\int_b^\zeta \frac{\partial u}{\partial t} + \frac{\partial uu}{\partial x} + \frac{\partial uv}{\partial y} + \frac{\partial uw}{\partial z}dz \\ \approx &\frac{\partial \hat{u}h}{\partial t} + \frac{\partial \hat{u}^2 h}{\partial x} + \frac{\partial \hat{u}\hat{v}h}{\partial y}.\end{aligned} \tag{2.34}$$

Now we take the RHS of the same momentum equation and integrate over the depth

$$\begin{aligned}&\int_b^\zeta \frac{\partial\left(\tau_{xx} - \rho g\zeta\right)}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z}dz \\ &= -\rho g\frac{\partial \zeta}{\partial x}\int_b^\zeta dz + \frac{\partial}{\partial x}\int_b^\zeta \tau_{xx}dz + \frac{\partial}{\partial y}\int_b^\zeta \tau_{xy}dz \\ &\left[-\tau_{xx}|_\zeta \frac{\partial \zeta}{\partial x} - \tau_{xy}|_\zeta \frac{\partial \zeta}{\partial y} + \tau_{xz}|_\zeta\right] \\ &+ \left[\tau_{xx}|_b \frac{\partial b}{\partial x} + \tau_{zy}|_b \frac{\partial b}{\partial y} - \tau_{xz}|_b\right] \\ &= -\rho g h\frac{\partial \zeta}{\partial x} + \frac{\partial}{\partial x}\int_b^\zeta \tau_{xx}dz + \frac{\partial}{\partial y}\int_b^\zeta \tau_{xy}dz + \tau_{\zeta x} - \tau_{bx}.\end{aligned} \tag{2.35}$$

We define

$$F_x = \frac{\partial}{\partial x}\int_b^\zeta \tau_{xx}dz + \frac{\partial}{\partial y}\int_b^\zeta \tau_{xy}dz, \tag{2.36}$$

$$F_{sx} = \frac{1}{\rho}\left[F_x + \tau_{\zeta x} - \tau_{bx}\right]. \tag{2.37}$$

Combining the above and dropping the hat results in the Shallow Water Equations

$$\begin{cases} \dfrac{\partial \zeta}{\partial t} + \dfrac{\partial hu}{\partial x} + \dfrac{\partial hv}{\partial y} &= 0 \\ \dfrac{\partial hu}{\partial t} + \dfrac{\partial hu^2}{\partial x} + gh\dfrac{\partial \zeta}{\partial x} + \dfrac{\partial huv}{\partial y} &= F_{sx} \\ \dfrac{\partial hv}{\partial t} + \dfrac{\partial hv^2}{\partial y} + gh\dfrac{\partial \zeta}{\partial y} + \dfrac{\partial huv}{\partial x} &= F_{sy} \end{cases} \tag{2.38}$$

2.3 Shallow Water Equations

In the previous section the SWE have been derived.In this section we will focus on the different terms that make up the SWE, and present some well known forms in which certain terms are neglected. In order to do this, we first introduce the notation $q_x = hu$ and $q_y = hv$, consider the bottom level constant in time (i.e. $\dfrac{\partial h}{\partial t} = \dfrac{\partial \zeta}{\partial t}$) and give names to the terms in the differential equations

$$SWE = \begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial q_x}{\partial x} + \dfrac{\partial q_y}{\partial y} = 0, \\ \underbrace{\dfrac{\partial q_x}{\partial t}}_{\text{local acceleration}} + \underbrace{\dfrac{\partial uq_x}{\partial x} + \dfrac{\partial vq_x}{\partial y}}_{\text{advection}} + \underbrace{gh\dfrac{\partial \zeta}{\partial x}}_{\text{water slope}} = \underbrace{F_{sx}}_{\text{friction slope}}, \\ \underbrace{\dfrac{\partial q_y}{\partial t}}_{\text{local acceleration}} + \underbrace{\dfrac{\partial vq_y}{\partial y} + \dfrac{\partial uq_y}{\partial x}}_{\text{advection}} + \underbrace{gh\dfrac{\partial \zeta}{\partial y}}_{\text{water slope}} = \underbrace{F_{sy}}_{\text{friction slope}}. \end{cases} \tag{2.39}$$

There are several options to solve the above equations numerically. However it not always necessary to solve the complete equations. In certain applications, it is possible to neglect terms which are small. By neglecting different terms, the equations change characteristics. Below, three popular approximations are described. They all solve the complete continuity equation, but neglect various terms in the momentum equations.

Local Inertial Equations

By neglecting the advection terms, we arrive at the Local Inertial Equations (LIE). Those equations form an important role in this research, since they will be compared to the full SWE

$$(LIE): \begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial q_x}{\partial x} + \dfrac{\partial q_y}{\partial y} & = 0, \\ \underbrace{\dfrac{\partial q_x}{\partial t}}_{\text{local acceleration}} + \underbrace{gh\dfrac{\partial \zeta}{\partial x}}_{\text{water slope}} & = \underbrace{F_{sx}}_{\text{friction slope}}, \\ \underbrace{\dfrac{\partial q_y}{\partial t}}_{\text{local acceleration}} + \underbrace{gh\dfrac{\partial \zeta}{\partial y}}_{\text{water slope}} & = \underbrace{F_{sy}}_{\text{friction slope}}. \end{cases} \tag{2.40}$$

Diffusive Wave Equations

A further simplification is achieved when the local acceleration term is also neglected. This simplification does not take into account the local rate of change of the momentum. The result is

$$(DWE): \begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial q_x}{\partial x} + \dfrac{\partial q_y}{\partial y} & = 0, \\ \underbrace{gh\dfrac{\partial \zeta}{\partial x}}_{\text{water slope}} & = \underbrace{F_{sx}}_{\text{friction slope}}, \\ \underbrace{gh\dfrac{\partial \zeta}{\partial y}}_{\text{water slope}} & = \underbrace{F_{sy}}_{\text{friction slope}}. \end{cases} \tag{2.41}$$

Kinematic Wave Equations

If we now also ignore the pressure gradient, we end up with the Kinematic Wave Equations (KWE).

$$(KWE): \begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial q_x}{\partial x} + \dfrac{\partial q_y}{\partial y} & = 0, \\ \underbrace{gh\dfrac{\partial b}{\partial x}}_{\text{bed gradient}} & = \underbrace{F_{sx}}_{\text{friction slope}}, \\ \underbrace{gh\dfrac{\partial b}{\partial y}}_{\text{bed gradient}} & = \underbrace{F_{sy}}_{\text{friction slope}}. \end{cases} \tag{2.42}$$

For a list of papers that address the DWE or KWE, see (Almeida and Bates, 2013).

Chapter 3

Literature

This chapter will describe some key papers. The papers that are selected show an (but not exhaustive) overview of the currently available techniques and their capabilities. This chapter contains a number of papers concerning LISFLOOD-FP (LFP, see Section 4.1 or (Bates, 2013)). It shows the evolution of the software package and the capabilities. For a more detailed description of the software package itself, the reader is referred to Section 4.1. The LFP software package was first released in 2000, where it solved the kinematic wave equations shown in Section 2.3. Later the options to solve the diffusive wave, inertial wave and shallow water equations were added. The first six papers describe the evolution of this software package and shows test cases where the package is evaluated. The papers are chosen to show the current developments for LFP, but also to show the effectiveness of this code compared to other methods. The first paper described one of the first large scale flood simulations, executed using LFP. This package is still in use, and is the package LFP described in this thesis. Since its introduction in 2000, a lot of development has taken place, but the fundamentals are still the same. The fifth paper compares three different implementations of the LFP software package with industrial code. This paper is closely related to the research carried out in this thesis. The papers after the first six papers describe different flooding simulation packages or techniques and are included to show the current state of the art for flood inundation simulation. Before we continue with the papers themselves, different methods to compare flooded areas are discussed.

3.1 Definitions for flood inundation comparison

Different definitions to compare flooded areas will be used in the papers and also throughout the report. Four definitions will mainly be used. Consider F_{m1} and F_{m2} as respectively the flooded areas of two methods, where the methods can be simulations or observations. Then the four criteria are defined as follows:

$$\text{Hit ratio H1} = 100\frac{F_{m1} \cap F_{m2}}{F_{m2}}, \tag{3.1}$$

this represents how many cells flooded in F_{m2} are also flooded in F_{m1}. Note that this measure is not symmetric. We will also use $H2$ which is defined as

$$\text{Hit ratio H2} = 100\frac{F_{m2} \cap F_{m1}}{F_{m1}}. \tag{3.2}$$

$$\text{False alarm ratio F} = 100\frac{F_{m1}/F_{m2}}{F_{m2}}, \tag{3.3}$$

where $F_{m1}/F_{m2} := \{x \in F_{m1} \cup F_{m2} : x \in F_{m1}, x \notin F_{m2}\}$. This measures the amount of cells flooded in F_{m1} that are not flooded in F_{m2} relatively to the amount of cells flooded in F_{m2}. Note that this measure is not symmetric.

$$\text{Critical success ratio C} = 100\frac{F_{m1} \cap F_{m2}}{F_{m1} \cup F_{m2}}, \tag{3.4}$$

This measure can be viewed as a combination of the two above measures. It account both for how well the flooded areas overlap as for how much mismatch there is between the two flooded areas compared.

It divides the area that is flooded in both methods by the combined flooded area of the two methods. This measure is symmetric.

$$\text{Correctly Predicted ratio CP} = \text{number of cells classified correctly as either wet or dry as a percentage of the total area.} \tag{3.5}$$

3.2 Paper: A simple raster-based model for flood inundation simulation (Bates and Roo, 2000)

Authors: Paul Bates and Ad de Roo.
Year: 2000

The paper compares three different inundation simulation techniques applied to a 35 km reach of the River Meuse in The Netherlands. The techniques used are:

1. A planar approximation to the free surface.
2. A relatively coarse resolution (50-250 m) two-dimensional finite element scheme on an unstructured grid.
3. A simple raster-based model for flood inundations simulation (their technique, LFP, flow-limited solver).

According to the authors of the paper, technique 3. outperforms the other two techniques (69.5%, 63.8% and 81.9% of inundated and non-inundates areas correctly predicted (see Equation 3.5) for respectively technique 1., 2. and 3.).
Until 2000 the most popular approaches to modeling fluvial hydraulics at the scale of 5-50 km have been one-dimensional finite difference solutions of the full Saint-Venant equations (which are the one-dimensional version of the Shallow Water equations described in Chapter 2). An example of such a program is MIKE11 (DHI, 2003). Those one-dimensional solutions solve the problem for cross sections. One of the main problems with this is that the cross sections are defined manually. This does not only require considerable skill, but also prevents an automatic process. The raster-based model introduced in this paper (Bates and Roo, 2000) only requires a DEM as a base for the program to run. The model consists of two parts. The first part is the one-dimensional representation of the channel and the second part deals with the floodplains. For the one-dimensional part a simplified version of the saint-Venant equations is used:

$$\begin{aligned} \frac{\partial Q}{\partial x} + \frac{\partial A}{\partial t} + q = 0 \qquad & \text{(Continuity)}, \\ S_0 = S_f \qquad & \text{(Momentum)}. \end{aligned} \tag{3.6}$$

Where S_0 is the channel bed slope and S_f is the friction slope here approximated as the water surface slope. For explanation of the variables, see Table 3.1. This means that the local acceleration, convective acceleration (also known as (momentum) advection) and pressure terms in the momentum equation are eliminated. This is the same as stating that the friction and gravity forces balance. They use an explicit finite difference procedure based on a backward difference scheme that is described in (Chow et al., 1988). One of the limitations of this simplified momentum equations is that only down gradient characteristics of the hydraulics are considered and hence backward effects are ignored (i.e., water can only flow from high water level to low water level), and there is the possibility of shock waves in areas of flow convergence.

Once a certain bankful depth is exceeded in a channel cell, this water may be be routed into adjacent floodplain areas. Note that a floodplain cell is only defined by its dimensions, elevation and a user-defined friction coefficient.

V	The cell volume
Q_{up}:	flow rate from the upstream adjacent cell
Q_{down}:	flow rate from the downstream adjacent cell
Q_{left}:	flow rate from the left adjacent cell
Q_{right}:	flow rate from the right adjacent cell
A_{ij}:	The cross sectional area at the interface of the two cells
R_{ij}	The hydraulic radius at the interface of the two cells.
S_{ij}	The water surface slope between the two cells
n	The Manning friction coefficient

Table 3.1: Definitions of variables

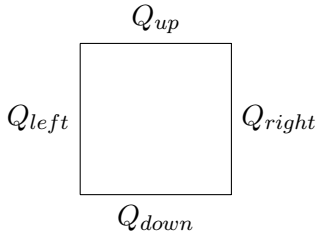


Figure 3.1

The distribution of water in the floodplain has been kept as simple as possible, each cell is treated as a storage volume for which a continuity equation is solved (see Figure 3.1):

$$\frac{dV}{dt} = Q_{up} + Q_{down} + Q_{left} + Q_{right}, \tag{3.7}$$

where the flow rates are calculated using the Manning equation (see Table 3.1 for the definitions of the variables)

$$Q_{ij} = \frac{A_{ij} R_{ij}^{2/3} S_{ij}^{1/2}}{n}. \tag{3.8}$$

So the floodplain flow is approximated as a two-dimensional diffusion wave. In order to prevent more water leaving a cell than it contains, the flow rates are weighted according to

$$c = \frac{V_t}{(Q_{up} + Q_{down} + Q_{left} + Q_{right})\,\Delta t}, \tag{3.9}$$

where V_t is the volume of water in the cell at time t and Δt the time step. It has to be noted that in the floodplain there is no restriction in the flow direction (which is restricted in the channel).

The only boundary condition applied to the model is an upstream inflow hydrograph. No boundary condition is applied at the downstream end of the model and water is able to leave freely with the flow rate calculation based on the local water slope between the penultimate and final cells. Initial conditions are derived from measured water values.

The model was tested on the section of the Maas river between Borgharen in the Netherlands (upstream) and Maaseik in Belgium (downstream), which has a length of 35 km.

As a reference, the "true" air photo derived inundation was used to compare the model results with the planar surface method and the 2D FEM model. The two ways used to measure the inundation simulation compared to the "real" air photo derived inundation are;
a) the Critical success value C (Equation 3.4);
b) the Correctly Predicted value CP (Equation 3.5)

Using the above definitions, the result shown in Table 3.2 was obtained. It can be seen that the steady state version of the LFP model outperforms all the other models tested here.

Simulation	Critical hit (%)	Correctly Predicted (%)
25 m 35 km DEM, planar surface	64.0	69.5
25 m 35 km DEM, steady state	77.3	**81.9**
50 m 35 km DEM, planar surface	63.8	69.4
50 m 35 km DEM, steady state	**78.3**	81.6
50 m 35 km DEM, dynamic	77.1	80.6
100 m 35 km DEM, planar surface	63.3	69.2
100 m 35 km DEM, steady state	69.2	70.2
100 m 35 km DEM, dynamic	69.6	70.9
2D FE model (element sizes in the range 50-250m)	48.1	56.2

Table 3.2: Results obtained for the different models in (Bates and Roo, 2000). Planar surface is model 1, 2D FE is model 2 and steady state and dynamic are model 3. Steady state and dynamic are model runs with a steady inflow hydrograph and dynamic inflow hydrograph, respectively.

3.3 Paper: A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling (Bates et al., 2010)

Authors: Paul Bates, Matthew Horritt and Timothy Fewtrell.
Year: 2010

This paper describes the acceleration method (inertial formulation of the shallow water equations) used in LFP. For the differential equations the reader is referred to Section 2.3 and for their numerical implementation to Section 4.1.2. The method is compared to the diffusive model (see Section 2.3 or 4.1.2), which is also implemented in LFP((Hunter et al., 2005)), and analytical solutions. For a horizontal beach, the inertial model performs slightly better than the diffusive model, however slightly worse for a planar beach. The Root Mean Square Error (RMSE) is however always less than the typical vertical error in the available DEM. When comparing the two methods on a more realistic flooding simulation, the Root Mean Square Difference (RMSD) is in the order of 1 cm, which is relatively small. The inertial model is significantly faster, with a maximum speed up of a factor 1120. They also conclude that the inertial model might show instabilities when the manning friction coefficient tends to zero. (This issue is solved, see (Almeida et al., 2012) or Section 3.4)

3.4 Paper: Improving the stability of a simple formulation of the shallow water equations for 2-D flood modeling (Almeida et al., 2012)

Authors: Gustavo De Almeida, Paul Bates, Jim Freer and Maxime Souvignet.
Year: 2012

The inertial formulation implemented in LFP as described in Section 3.3 has been reported to show instable behavior with low Manning friction coefficients. This paper tests two numerical schemes to improve on the stability of this inertial formulation. Applying a Taylor expansion to the implementation of Section 3.3 shows that all second order spatial derivatives were canceled out, resulting in no diffusion of the error terms. Which they claim is the cause for the instabilities. This paper describes and tests two proposed solutions, the so called q-upwind numerical scheme and q-centered numerical scheme. Both schemes add numerical diffusion to the momentum equations. The methods differ in how the time derivative of the flux q is calculated. In Section 3.3, this is done by

$$\left.\frac{\partial q}{\partial t}\right|_{i-1/2} \simeq \frac{q_{i-1/2}^{n+1} - q_{i-1/2}^{n}}{\Delta t}, \tag{3.10}$$

and in the q-upwind scheme by

$$\left.\frac{\partial q}{\partial t}\right|_{i-1/2} \simeq \frac{q_{i-1/2}^{n+1} - \left[\theta q_{i-1/2}^{n} + (1-\theta)\, q_{upw}^{n}\right]}{\Delta t}, \tag{3.11}$$

where θ is the weighting factor and q_{upw}^{n} takes the value of q^n at the upstream cell interface. The q-centered scheme uses the following differential

$$\left.\frac{\partial q}{\partial t}\right|_{i-1/2} \simeq \frac{q_{i-1/2}^{n+1} - \left[\theta q_{i-1/2}^{n} + \frac{(1-\theta)}{2}\left(q_{i-3/2}^{n} + q_{i+1/2}^{n}\right)\right]}{\Delta t}. \tag{3.12}$$

The q-upwind method shows, when applying a Taylor expansion, an unwanted first order derivative error term which scales the same as the CFL condition, and thus does not vanish with grid refinement. The q-centered method does not show this term and scales properly with grid refinements.

The schemes are compared on three different test cases. The first and second are respectively the third and fourth test cases described in Section 3.5 and the third test case is a simulation of urban inundation. The area is 4.6 x 2.5 km and is represented by a DEM at 2 m resolution. The simulations for Test case 3 were run with $\theta = 0.9$ and $n = 0.025$ or $n = 0.035$. For the simulations with a Manning coefficient of 0.035, the differences between the q-centered and the original inertial formulation where less than 5 cm almost everywhere. However when the friction coefficient is reduced to 0.025, the locations where the difference is more than 5 cm increased substantially. Also, some regions of the domain show a checkerboard water level for the original method. These oscillations are not observed for the q-centered approach. It is noted that Manning coefficients of the order 0.01 to 0.015 occur commonly in urban areas and thus the lower values of friction coefficients are important for flood inundation simulation. Since the oscillation can also cause negative water heights, and the program resets those heights to 0 m, there will be a significant mass error in the original simulation. It is also noted that checking the mass error is a good indicator to see if those model instabilities have occurred.

3.5 Paper: Applicability of the local inertial approximation of the shallow water equations to flood modeling (Almeida and Bates, 2013)

Authors: Gustavo De Almeida and Paul Bates
Year: 2013

The objective of this paper is to analyze how the assumption of neglecting the momentum advection term in LFP affects the accuracy of the model. The inertial formulation is compared with analytical solutions to the full SWE. The equations used can be found in Equation 4.4. The numerical implementation of the inertial formulation in LFP is described in Section 4.1.2. The paper is divided into two sections, the first is concerned with steady flow and the second with unsteady flow.

3.5.1 Steady flow

Steady flows are simulated and compared against analytical solutions of the full SWE. The equation describing steady, gradually varied flow derived from the inertial model yields

$$\frac{dh}{dx} = S_0 - S_f. \tag{3.13}$$

and for the full SWE

$$\frac{dh}{dx} = \frac{S_0 - S_f}{1 - Fr^2}, \tag{3.14}$$

where S_0 and S_f are respectively the bed and friction slopes $gh\frac{\partial b}{\partial x}$ and $\frac{gn^2\|\boldsymbol{q}\|q_x}{h^{7/3}}$ and $Fr = u/\sqrt{gh}$. So, for low Fr numbers ($<< 1$), the depth gradient given by both equations are roughly the same. And for numbers below unity, called subcritical flow, the flows show similar behavior however the gradient of the full SWE will always be larger. For near critical flow (i.e. transcritical flow), the full SWE will lead to a near vertical depth profile whereas this effect is not the case for the inertial approximation. As soon as the flow gets supercritical, the flow changes characteristics with a change in sign for the water slope. So for supercritical flow, the SWE predicts water surface profiles approaching the normal depth whilst the inertial model predicts water surface profiles going away from the normal depth. This shows the unsuitability of the inertial model for supercritical flows.

For the steady state flow, two test cases are used. Both based on the solution of an inverse problem: given the water surface profile, flow discharge and channel characteristics, the bed profile is obtained by integrating the channel slope, which is derived from the steady flow form of the SWE.

Test case 1: Near critical steady subcritical flow in a rectangular channel

A 1 km long and 10 m wide channel with a Manning coefficient of 0.03 and discharge of 20 m^3/s is simulated, with $0.54 < Fr < 0.99$. The near critical conditions (i.e. Fr close to unity) occur near both boundaries. The analytically derived channel slope is

$$S_0(x) = \left[1 - \frac{4}{gh(x)^3}\frac{dh(x)}{dx} + \frac{0.0036}{h(x)^{10/3}}\right], \tag{3.15}$$

which is integrated to obtain the channel profile $b(x)$. The solution to the SWE is

$$h(x) = \left(\frac{4}{g}\right)^{1/3}\left\{1 + \frac{1}{2}\exp\left[-16\left(\frac{x}{L} - \frac{1}{2}\right)^{1/2}\right]\right\}. \tag{3.16}$$

This analytical solution was used as the initial condition for the inertial model and ran until a new steady state was reached. The results show a very good agreement between the two methods, even when Fr close to one near the boundaries. It is observed that higher bed gradients results in higher deviations from the analytical solutions. The same experiment has been repeated for $0.35 < Fr < 0.54$ by increasing the depth profile by a constant. The results show that the error also increases with increasing depth gradients. However, the error is reduced considerably compared to the tests with higher Fr number.

Test case 2: Sinusoidal bed elevation in a rectangular channel

The second experiment is similar to the first, but now the bed level follows a sinusoidal shape with about five periods. The equation for the channel slope is the same as in Test case 1 (except that h is different here of course), but the water level is

$$h(x) = \frac{9}{8} + \frac{1}{4}\sin\left(\frac{\pi x}{500}\right). \tag{3.17}$$

Increasing the water depth gradients again results in increased departure from the analytical solution, just as in Test case 1.

3.5.2 Unsteady flow

In the unsteady flow problem, the eigenvalues of the Jacobian matrix are

$$\lambda_{IN}^{1,2} = \pm\sqrt{gh} \tag{3.18}$$

for the inertial model and

$$\lambda_{SWE}^{1,2} = u \pm \sqrt{gh} \tag{3.19}$$

for the full SWE. For the local Riemann problem this results in that the inertial model admits only solutions with one shock traveling in one direction and an opposite rarefraction wave, where the SWE also admits to more complex solutions. This leads to the fact that the inertial model is again unable to simulate supercritical flow. In the unsteady flow case, the error made by the inertial model increases with an increased Fr number, just as in the steady flow cases. This time it is reflected in the propagation speed of the waves, which is slower with the inertial model.

Test case 3: Nonbreaking wave propagation over a horizontal plane

This test case simulates the propagation of a flood wave over a horizontal plane that is initially dry. The analytical solution to the water depth is

$$h(x,t) = \left[-\frac{7}{3}\left(n^2u^2\left[x-ut\right]\right)\right]^{3/7}, \tag{3.20}$$

which will yield the upstream boundary by setting $x = 0$. n^2u^3 was held constant, but n and u were varied. The error in the position of the wave front for the different values of n and u can be seen in Table 3.3

Manning n	0.005	0.01	0.02	0.03
Velocity u (m/s)	0.95	0.60	0.38	0.29
Relative error Test case 3 (%)	15.4	6.2	2.1	1.0
Relative error Test case 4 (%)	9.9	5.2	2.4	1.2

Table 3.3: Error of inertial model compared to the analytical solution of the shallow water equations for Test case 3 and Test case 4.

Test case 4: Nonbreaking wave propagation over an adverse slope

Test case 4 is the same as Test case 3, except that now there is a non horizontal plane. For this test case, no analytical solution is known and therefore the solutions to the SWE were obtained with a fourth order Runga-Kutta method. The bed elevation increases linearly from 0 to 4 m over a distance of 4 km. The results can be seen in Table 3.3.

Discussion and Conclusion

They conclude that the inertial model "is capable of providing highly accurate solutions to the shallow water system, particularly in the lower range of subcritical flows (e.g., $Fr < 0.5$). This covers a wide range of floodplain and lowland river flows of practical interest. For $0.5 < Fr < 1$, the inertial models accuracy is influenced by Fr and flow spatial gradient. If a high accuracy for Fr close to unity and steep depth changes is required, the inertial model does not suffice. For Fr near unity, the predicted speed of a propagating wave by the inertial model is lower than the speed obtained by the full SWE.

3.6 Paper: How much physical complexity is needed to model flood inundation? (Neal et al., 2012)

Authors: Jeffrey Neal, Ignacio Villanueva, Nigel Wright, Thomas Willis, Timothy Fewtrell and Paul Bates
Year: 2012

In this paper, three two-dimensional explicit hydrological models have been benchmarked against the following industrial software packages:

- JFLOW (diffusive)

- Flowroute (diffusive)
- IW2D (Shallow water)
- ISIS2D (Shallow water)
- SOBEK (Shallow water)
- TUFLOW (Shallow water)

The three models tested against the above list can be broadly defined as simulating diffusive (LFP, adaptive solver, see Section 4.1.2), inertial (LFP, acceleration solver, see Section 4.1.2) or shallow water waves (LFP, Roe solver, see Section 4.1.2) models. The models are compared for different test cases. The general conclusion is: "For flows that vary gradually in time, differences in simulated velocities and depths due to physical complexity were within 10% of the simulations from a range of industry models". So, for this type of flows it seems unnecessary to solve the complete shallow water equations and only solving for the diffusive or inertial formulation is sufficient. This is not the case for supercritical flow, where significant differences are measured. Furthermore, it is observed that the diffusive model required much longer simulation times.

Test cases

The test cases are chosen from a list of ten test cases from the Environment Agency for England and Wales. The four chosen in the paper can shortly be described as

1. Test 3: Momentum conservation over a bump. Water enters the domain on the west and flows down a slope. At the end of the slope is a bump. Some of the water should overtop the bump. As the authors expected, the diffusive model did not simulated the desired overtopping effect, while the other two models did. Furthermore, the inertial model required an increased friction coefficient for stability. Even with this higher friction, the inertial model simulated a final depth within the range simulated by industry code.

2. Test 4: Rate of flood propagation over extended floodplains. A 1000 by 2000 m floodplain with a 20 m wide inflow hydrograph at the west side of the floodplain is simulated. The inertial model showed a greater flood extent in the diagonal direction when compared to the full SWE model, and less in the horizontal direction. However, the overall differences are small relative to typical vertical errors in survey data. Overall, it can be observed that the models solving similar differential equations result in very similar outcomes, while changing the solved differential equation seems to have more impact (this may seem obvious, but the results of my thesis (see Chapter 7) shows the opposite effect for the Xynthia test case). Overall, the depths where within 10% of each other, and inundation time at one of the observation points were within 3 minutes after 60 minutes of simulation.

3. Test 5 (50 m): Dam failure in a valley. The three different LFP models were compared for flow depth, velocity and arrival time in a dam failure setup. It was observed that the full SWE model was affected by instabilities of the flood edge. The peak velocities of all the LFP models where always within the range of the industry codes, where peak water levels were up to 10% lower for some observation points. The codes not solving the SWE showed a quicker rate of rise in the water surface elevation, which resulted in greater peak velocities.

4. Test 6b: Dam break. This test evaluates the ability to simulate hydraulic jumps and wake zones behind buildings. This test case has a dominant supercritical flow. The inertial LFP model showed a volume error of 30%, due to negative cell water depths (which means that this water depth is reset to zero and thus adding water to the system). The SWE LFP model showed similar results to other SWE industrial code. The authors concluded that for these conditions a shock capturing shallow water model is required, and the diffusive and inertial LFP models "should be avoided in situations where hydraulic jumps are expected to affect flood wave propagation".

The inertial LFP model results "indicate that this model may be suitable for similar test cases where flows are subcritical and friction is greater than $n = 0.03$." Another observation is that in the test cases, every time the inertial LFP "model was unable to emulate the full shallow water models depths and velocities to within $\sim 10\%$, its mass balance error increased by many orders of magnitude". Finally, executing Test 5 on a grid of 10 m (instead of 50 m) showed that the sensitivity to this grid size change is similar to the sensitivity of the model choice. Their final conclusion is: "We show that for gradually varied flow, full shallow water models may be unnecessarily complex, and simpler, cheaper schemes, such as the inertial wave formulation in LISFLOOD-ACC, can perform just as well, both in terms of velocity and depths. Moreover we show that subtle modelling decisions can often have more effect on results than selecting a more physically complex model."

3.7 Paper: Hyper-resolution mapping of regional storm surge and tide flooding: comparison of static and dynamic model (Ramirez et al., 2016)

Authors: Jorge Ramirez, Michal Lichter, Tom Coulthard and Chris Skinner.
Year: 2016

A dynamic reduced-complexity model (DRC model) and a static model are compared at three test sites for a storm tide flooding. This DRC model has cells of < 100 m and is applied at regional scales (100-200 km of coastline, 50-100 km inland). The DRC model is based on CAESAR-Lisflood and solves the inertial equations described in (Bates et al., 2010) (it solves the Shallow Water Equations ignoring the advection term). The method is compared to a statical approach on three locations: The central Bay of Biscal during the Xynthia storm; the north-eastern US coast near New York City during hurricane Sandy; the southern coast of Myanmar during cyclone Nargis. The models have a grid based on Space Shuttle Radar Topography Mission (SRTM) DEM at 90 m resolution. A 2.5 km seaward shelf near the coastline is represented by a -5 m bathymetry. Both methods are assessed against measurements of the flooded extend and High Water Marks (HWM). The DRC model does not have to be run on a supercomputer and can be applied to any coastal area, since the input data used is from open source sources and the topology data is available for all locations on earth.

3.7.1 Models

The RCD model developed in this study is largely based on CAESAR-Lisflood, which is a open-source, freely available model. The CAESAR-Lisflood solves the simplified shallow water equations described in (Bates et al., 2010). The model can run regional landscapes of 15,000 km^2 represented by DEMs with a resolution of < 90 m (so $> 1{,}000{,}000$ cells) on a normal laptop in the scope of minutes to hours. The static model is a hydraulically connected model.

3.7.2 Model input

The RCD model requires three sources of input:

1. A DEM representing the land and the near shore bathymetry
2. Land cover to assign roughness values
3. The duration and height of the storm tide

For the DEM, SRTM data at the resolution of 90 m is used. This DEM is in EGM96 datum, and all other datasets are converted to this datum. Since SRTM data does not only represent the bare earth, but also vegetation, they investigated the study areas but no changes where made in the DEM (it is deemed unnecessary to do so). The bathymetry is represented by a -5 m elevation stretching over a 2.5 km wide seaward shelf. The claim is made that adding GEBCO bathymetry data results in very small differences in maximum flood extents and water depths, and is therefor omitted.

Test site	Dynamic		Static	
	HWM flooded (%)	Vertical error (RSME, m)	HWM flooded (%)	Vertical error (RSME, m)
France	62	0.81	65	0.85
USA	32	0.94	32	0.86
Myanmar	40	2.01	52	2.97

Table 3.4: Results obtained in (Ramirez et al., 2016).

Test site	Observed	Dynamic				Static			
	Flood area (km^2)	Flood area (km^2)	Hit (%)	False (%)	Under (%)	Flood area (km^2)	Hit (%)	False (%)	Under (%)
France	444	611	79	59	21	1327	95	204	5
USA	553	328	51	8	49	371	57	10	42
Myanmar	4219	4139	65	33	35	8096	92	99	8

Table 3.5: Results obtained in (Ramirez et al., 2016).

Roughness values are often calibrated to match observed results, however this calibration data is not likely to be available in an operational context. Therefore the model uses typical roughness values for land cover classes and utilizes the GlobCover 2009 land cover maps (Defourny et al., 2011). Rivers where included in the model by using the SRTM water body dataset and assigning roughness coefficients for water ($n = 0.02$) within river channel locations.
The storm tide is included in the model by raising and lowering storm tide water levels near the coastline. The water levels were obtained from databases, where the REFMAR[1] database was used for the water levels during Xynthia, and NOAA[2] for the Sandy storm. For Nargis, due to lack of tide stations, a simulation was run in order to create this data. Using this input, a time series of 62 hour was constructed, with an interval of 10 minutes for Xynthia and Nargis, and 6 minutes for Sandy. Furthermore, waves are not explicitly reproduced in the model, although they may be included in the observed water levels.

3.7.3 Model output and validation data

Both the static and DRC model produce a map of maximum flood water heights (DEM elevation + water depth) for the entire DEM, and a delineation of the maximum flood extent. As for the validation data, high water marks (HWM) and the delineated area were available.

3.7.4 Results

In Tables 3.4 and 3.5, the results are summarized. For the definitions of Hit and False the reader is referred to Equations 3.2 and 3.3. The "Under" percentage in the table is simply 100% - Hit. The water heights at HWM locations are in general overestimated by both models. The performance of both models was good for France and USA (median vertical error < 0.5 m), however less successful for Myanmar. A general result is that the static model is less conservative, having a higher value of correctly (Hit) flooded area however also resulting in overestimation (False) of up to 204% (versus 59% for the DRC model).

3.7.5 Discussion

For France and USA, the water levels produced by the dynamic model are within 1 m of the observed values, which is comparable to the performance of dynamic models of greater physical complexity (Forbes et al., 2014). Furthermore, the underestimation of the flood extent in the USA could be due

[1] www.refmar.shom.fr
[2] http://tidesandcurrents.noaa.gov/

to that only one geographic location is used for storm tide water levels, even though storm tide heights can vary along the coast. The flood extent of the DRC is comparable with the flood extent created by a dynamic coupled model of surge, tide and wave flooding, as done in (Bertin et al., 2014), which is described in Section 3.10.

3.7.6 Conclusions

A successful implementation of the DRC has been created that is able to run on the spatial resolution of 90 m and regional scale. The model is applicable at any location in the world and can be run with limited resources. As input, only freely available data of global extent has been used. DRC models offer a more conservative alternative to existing models while remaining easily implementable. It is suggested that more research should be done regarding the role of DEM resolution on the results.

3.8 Paper: Assessment of static flood modeling techniques: application to contrasting marshes flooded during Xynthia (western France) (Breilh et al., 2013)

Authors: Jean Breilh, Eric Chaumillon, Xavier Bertin and M. Gravelle
Year: 2013

An assessment of static flood modeling techniques is presented. The techniques are applied to an area in western France and compared with post-storm delineation of flooded areas. The storm for which this is done is the Xynthia storm, which also acts as a case for the current thesis report. The first static technique they apply is called SM1 (Static Model 1): A static flood modeling method that uses the maximum sea level recorded during the storm at La Pallice tide gauge, the second technique (called SM2, Static Model 2) incorporates a spatial varying maximum sea level rise extracted from a model. A third method, called SO (Surge Overflow), computes the water volume discharge over the dikes based on time series of modeled water levels. SM1 and SM2 consider a cell flooded if the elevation is below the maximum sea level and only if they are connected to an adjacent cell that is flooded or connected to open water. The cells used are raster based, and a cell is adjacent to another cell if the cell boundary or corner touches another cell, see the 8 point rule in Figure 3.2. Using this simple static techniques, they find that marshes close to the ocean are modeled more accurately than marshes further away from the ocean. The marshes that are further away than 10 km all have a rather poor F-value[3], as can be seen in Figure 3.3. This is not a surprise, since static methods will decrease to be of use when the water has to travel further over land to reach a marsh. For SM1 and SM2, the results obtained can be viewed in Table 3.6.

For two of the marshes the surge overflow method has been applied. In both the marshes floods where heavily overestimated by the static methods. For this semi-dynamic model the weir discharge

[3]An F-value is defined here as F = A/(A+B+C), where A correctly predicted area, B flooded in model, not flooded in observation and C flooded by observation, not flooded in simulation)

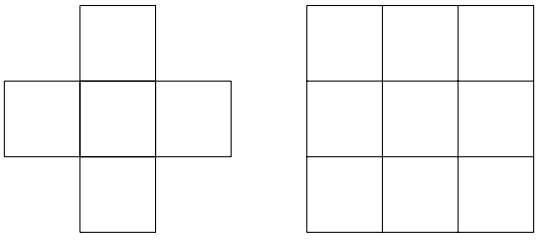


Figure 3.2: Left: 4 point rule of connection. Right: 8 point rule of connection. The cell in the middle is connected to either 4 or 8 cells.

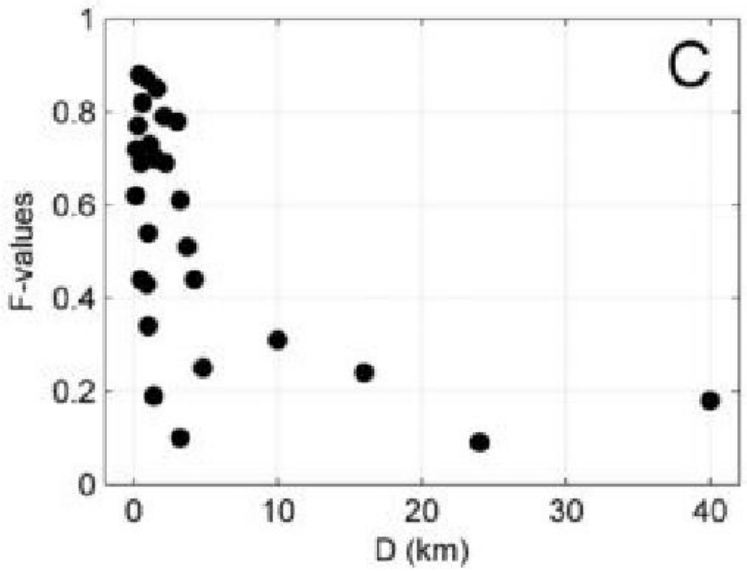


Figure 3.3: F-values found in (Breilh et al., 2013) using SM2 for the 27 marshes regarding distance between coastline and landward boundary of the marsh.

Marsh classes	Mean F-value using method SM1	Mean F-value using method SM2
all marshes	0.51	0.54
small marshes	0.55	0.58
large marshes	0.41	0.42
very large marsh	0.17	0.16

Table 3.6: Fit measurement of flooded extent in (Breilh et al., 2013)

equation of Kindsvater and Carter (1957) has been used;

$$Q = \mu L\,(2g)^{1/2}\,h^{3/2}), \tag{3.21}$$

where h is the water depth over the dike in meters, calculated by subtracting the dike crest height to time series of modeled sea level at the closest computational node. Using this method on two marshes increased the F-value from 0.24 and 0.17 for SM2 to 0.41 and 0.39 for SO respectively.

The static modeling techniques applied here are really easy to implement, but as can be seen comes at the cost of low F-value levels. But maybe, it can be used to give a first indication of what (large) areas may or may not flood at all, and be used as a first estimate to indicate where the focus of further modeling should lie.

3.9 Paper: Quadtree flood simulations with sub-grid digital elevation models (Stelling, 2012)

Author: Guus Stelling
Year: 2012

This paper describes the numerical software packages 3Di and applies it to two examples. One of the examples is a flooding event on the Dutch coastline. This paper is included in order to show different software packages than the one described in Software (Chapter 4). First, the key aspects of the software package will be described and then the flooding event will be presented.

3.9.1 The 3Di software package

3Di is a software package that employs the finite volume staggered grid method for the shallow water equations. It uses Cartesian grids, but in combination with a quadtree implementation to allow for vayring cell sizes. The software package is designed to readily implement an available DEM (which might have a very high resolution) with the model, and generate the grid based on this DEM. Afterwards, the DEM is used to determine the roughness values of the cells. This approach leads to a flexible grid with the resolution available where necessary and coarse cells where possible. The interesting thing is that the size of the DEM does not have a bad influence on the simulation speed (after once being indexed). This can be achieved by pre-computing the necessary tables. Using this technique, DEMs with tens of millions of pixels can be analyzed.

The program solves the 2D shallow water equations given below

$$\begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial uh}{\partial x} + \dfrac{\partial vh}{\partial y} & = 0, \text{(Continuity)} \\ \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} + v\dfrac{\partial u}{\partial y} + g\dfrac{\partial \zeta}{\partial x} + \dfrac{c_f}{h}u\|u\| & = 0, \text{(x-momentum)} \\ \dfrac{\partial v}{\partial t} + u\dfrac{\partial v}{\partial x} + v\dfrac{\partial v}{\partial y} + g\dfrac{\partial \zeta}{\partial y} + \dfrac{c_f}{h}v\|u\| & = 0, \text{ (y-momentum)} \end{cases} \tag{3.22}$$

where we have $h(x, y, t) = \zeta(x, y, t) - b(x, y)$ denoting the water depth. The mass and momentum equations are solved on a staggered grid, where the mass is solved in each cell and the momentum is solved over the edges of the cells. This is done in such a way that if there is a bigger cell next to a smaller cell, the bigger cells uses only the quarter of the cell that shares a border with the smaller cell and the smaller cell uses half of the cell that shares the border with the bigger cell, as can be seen in Figure (3.4) and Figure (3.5). This will be explained in more detail in Section (3.9.1).

Grid

In order to be able to talk about a grid in 3Di, first a short introduction to some notation will be given. 3Di is based on rectangular cells of different sizes. And has a DEM that supports those cells. In order to be able to have cells of different sizes 3Di applies a method called quadtree method.

Starting with a DEM consisting of pixels with size $\delta x \times \delta x$. Based on this DEM a course grid is generated, where cells can have size $\Delta x_l = \Gamma \delta x 2^l$, where $\Gamma \geq 2, L \geq l \geq 0$. So the smallest available cell contains four DEM pixels. In this notation, l refers to the so called level of a cell (so, the smallest cells available are level 0). First, the grid is divided into cells of the highest level, which will be denoted by L. Now let us zoom in on one of those cells. This cell can be divided into four smaller cells, having $l = L - 1$. Each of those cells can be divided into four smaller cells again. In this way it is possible to create a grid with varying cell sizes. This dividing of cells cannot always occur but has to follow one simple rule. Two cells adjacent to each other may not have more than one difference in level.

3Di uses a staggered grid approach to solve the differential equations. The continuity equation is solved for each cell, and the momentum equations are solved over the edge of two adjacent cells. In order to do this, the domain where to solve has to be defined. If two adjacent cells have the same size, this is done in the usual setting. For example, see Figure (3.4), the domain used to solve the momentum in x-direction will be the quarters B,D,E,G. If two cells next to each other do not have equal size, Figure (3.5) illustrates the domain used for the momentum equation in the x-direction.

Subgrid in 3Di

As stated in Section (3.9.1), 3Di uses a DEM as the input for bathymetry data. The power of 3Di is that it uses the high resolution DEM data to calculate the cross section at the cell faces and is also used to compute the volume of water in a cell. For example, the cross section in the y-direction is

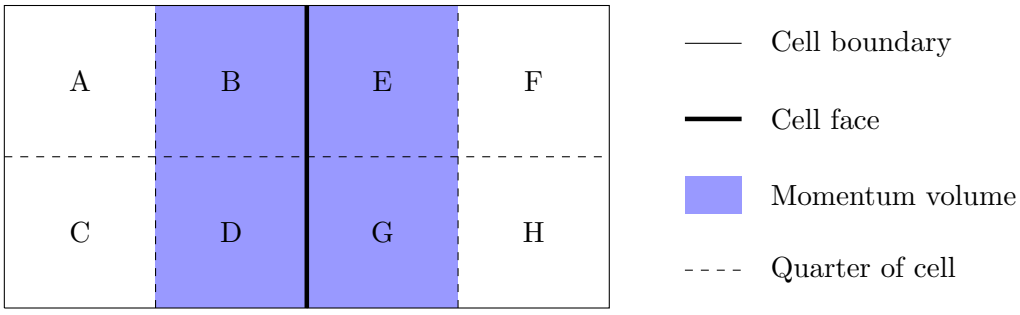


Figure 3.4: Two cells divided in four parts. The blue part is used for the momentum equation

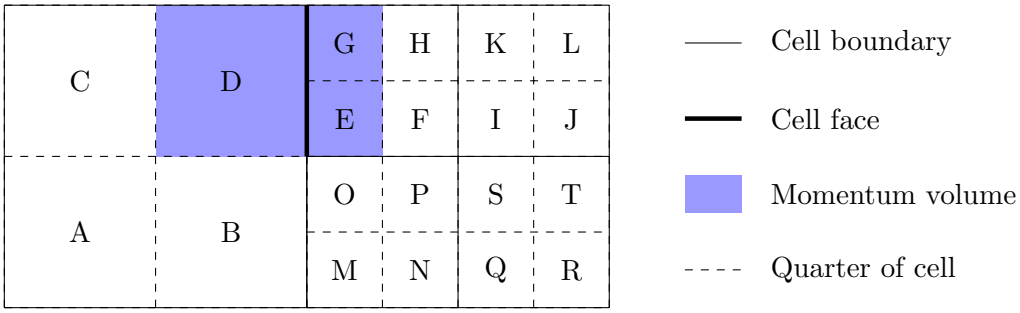


Figure 3.5: Five cells, blue part is used for the momentum equation over the thick edge.

defined as:

$$A^{y}_{l,m,n+1/2}(t) = \delta x \sum_{j=j_0}^{j=j_1} max\left(0, \zeta^{*}_{l,m,n+1/2} - b_{i+1/2,j}\right), \tag{3.23}$$

where $\zeta^{*}_{l,m,n+1/2}$ is calculated by a first order upwind method and $b_{i+1/2,j} = \max(b_{i,j}, b_{i+1,j})$. For the exact definition and explanation, the interested reader is referred to the paper since it's not the goal to specify this here. The goal is to indicate how the pixels are used to incorporate a high resolution DEM in a lower resolution grid. A graphical representation of the calculation of $A^{y}_{l,m,n+1/2}(t)$ is given in Figure 3.7.

3.9.2 Flooding case

As an example, a polder area in the Netherlands (near Petten) has been simulated. An area containing 1 386 x 886 = 1 227 996 pixels was chosen, which three different coarse grids: grid 1, $\Gamma = 4, l_{\max} = 4$; grid 2, $\Gamma = 2, l_{\max} = 1$; grid 3, $\Gamma = 2, l_{\max} = 3$. They have respectively 4 880, 10 736 and 13 014 cells, which is a factor 100 to 250 less than the number of pixels. Simulation of three days is done in less than five minutes on a simple laptop. The main difference in flooding area between the grids is in the inundation time. However, the final inundated area is very similar.

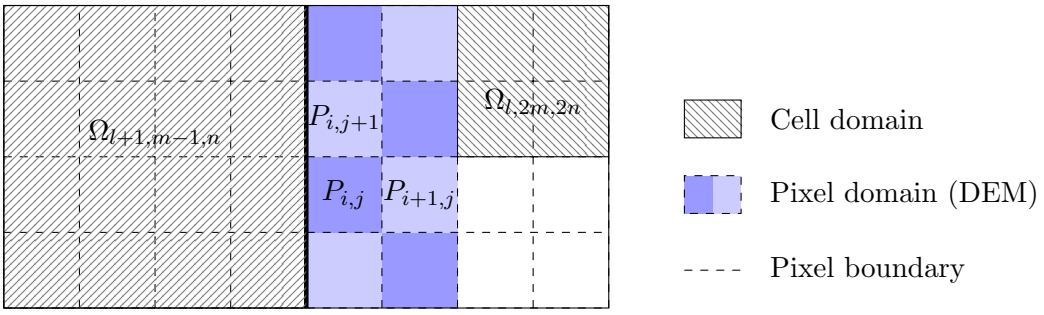


Figure 3.6: Explanation of domain notation of 3Di.

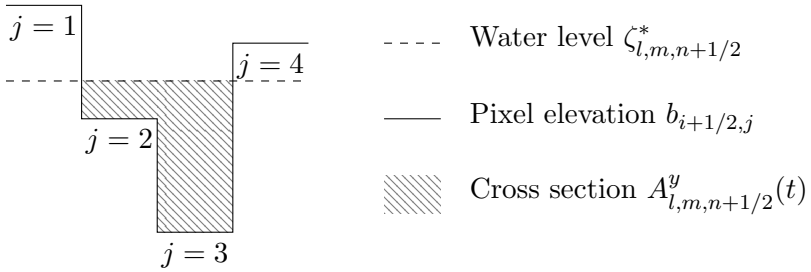


Figure 3.7: cross section calculation in 3Di.

3.10 Paper: A modeling-based analysis of the flooding associated with Xynthia, central Bay of Biscay (Bertin et al., 2014)

Authors: Xavier Bertin, Kai Li, Aron Roland, Yinglong Zhang, Jean Breilh and Eric Chaumillon.
Year: 2014

A high resolution, 2D fully coupled modeling system hindcast of the flooding associated with the tropical storm Xynthia is presented in this paper. This summary will follow the same outline as their paper.

3.10.1 The studied area and storm

The studied area is located in the central Bay of Biscay in western France and is similar to the area depicted in Section 6.2. It consists of two big islands and several embayments and estuaries. Furthermore, 50% of the area less than 10 km inland is below the highest astronomical tide, rendering this area the most vulnerable in France. The storm Xynthia hit the bay in the night of the 27th and 28th of February in 2010.

3.10.2 The storm surge and flooding modeling system

The numerical model SELFE is employed, which realizes the coupling in 2D and 3D between a circulation model and the spectral wave model WWM-II. Both of the models have the same unstructured grid and domain decomposition. SELFE solves the 2D shallow water equations using a semi-implicit Galerkin finite elements method. The area is is discretized using 888 248 nodes and 1 765 765 elements which resolution ranging from 5 m to 30 km in order to capture dikes and dunes. The model was driven by hourly fields of atmospheric pressure and 10 m wind speeds and by astronomical tides both in the domain and at the open boundary. The model is run on 144 cores on a 512 core SGI ALTIX ICE cluster. About 60 hours of wall clock time was used to simulate 8 days. This is a factor three times faster than real time. This setup is not feasible for pan-European scales.

3.10.3 Modeling results

The waves were predicted within reasonable error, but some showed a delayed shift in wave direction of 3 hours when compared to observations. The water levels from three tidal gauges and the model results show a Root Mean Square Discrepancy (RMSD) ranging from 0.15 to 0.25 m with a negative bias in the order of centimeters. The flooding predictions were compared to observations using Equation 3.4. This value was computed for seven main areas and ranged from 0.5 to 0.79.

3.10.4 Discussion

Comparison of the model data showed that small marshes located along the coastline were not flooded at all in model while being flooded in the observations. Those marshes all have barriers higher than the still water levels computed with the model. which ignored wave runup and infragravity waves.

This paper questions the one-way nesting approach very often used (also in this thesis). This approach assumes that the boundary conditions at the coastline or close to it can be determined by simulation of the coastal sea conditions without accounting for the flood. Then, use the acquired results as a boundary condition for the flood simulation. It is stated that: "The comparison between our baseline simulation and a simulation where the flooding is disabled by increasing the dike height reveals differences in maximum water levels locally reaching 1.0 m. This result is of key importance for coastal management strategies and also questions classical engineering approaches relying on one-way nesting." This results in that a local flood inundation model may have boundary conditions with too high water levels. Also, raising the protection level locally might have negative effects on the safety of the surrounding areas due to higher water levels and thus that the management of protection should be coordinated on a larger scale.

3.11 Paper: A methodology for flood susceptibility and vulnerability analysis in complex flood scenarios (Dottori et al., 2016)

Authors: Franceso Dottori, Mario Martina and Rui Figueiredo
Year: 2016

This paper proposes "a methodology for large-scale analysis of flood susceptibility and vulnerability. The methodology is based on a mathematical index, which considers local topography and basic information about the flood scenario to reproduce flooding processes." The mathematical index is based on the one dimensional uniform flow equations for large rectangular channels

$$\frac{d\zeta}{dx} = -\frac{u^2 n^2}{h^{4/3}}, \tag{3.24}$$

where $d\zeta/dx$ is the water surface slope, u the flow velocity, h the water depth and n the Manning coefficient. Based on this equation, a flood intensity index is defined as

$$I_{w_i} = \zeta_{\text{source}} - z_i - \lambda d, \tag{3.25}$$

where z_i is the bed elevation of cell i, d is the distance between the cells and λ is based on

$$\lambda = \frac{u^2 n^2}{h^{4/3}}, \tag{3.26}$$

but in practice is a calibrated value. It is considered constant over the whole domain. The index is determined on an iterative basis and influences the neighboring cells their index value. It is tested on two areas in Northern Italy. Both of them are river floods, but the method could also be applied to coastal scenarios.

3.11.1 Test cases and results

The first test site is located 30 km above Bologna, Italy. A breach in the river Reno caused a flooding of the test site in 1990 and resulted in a total flooded area of about 5.6 km^2. There is a complex drainage network in the area.

The second test site has a total extent of around 200 km^2, located North West of the city Modena, Italy. The area is a "lowland agricultural area with with several small towns, village and sparsely distributed farmland." It was hit by a flood on January 2014 and had a flooded area of about $90km^2$. This area also has a complex drainage network.

The parameter λ is set to one value for the whole area and calibrated to reproduce both the flooded extent and point measurements, and hence not based on Equation 3.26. The results could be improved by allowing locally varied values for λ, but this has not been done in order to prevent over-parameterization. The RMSE with respect to observed depths is 51 cm, which is not negligible however close to the results of a CA2D model obtained earlier by the same researchers.

3.12 Conclusion of the papers

The papers based on LFP all state generally the same: If the flow is sub critical and slowly varying, then the inertial equations are a good approximation of the full SWE. This has been tested for many test cases. However not for a typical flooding simulation with grid cells in the range of 90 m SRTM data. Furthermore, hydrologically connected models show large overestimations of the flooded extent, however are cheap. During this thesis, models with cells of size of 90 m are compared for different hydrological packages to investigate the differences between packages and the importance of including the advection term (so the question is whether the inertial formulation is a good approximation of the full SWE in this setting).

In the above literature there is a lack of comparison of flood inundation based on larger grid cells in a two-dimensional area. This study will address this lack and compare two software packages with larger grid cells. The two packages chosen are LISFLOOD-FP and Delft3D FM. They are chosen because LISFLOOD-FP is currently used at JRC and Delft3D FM is developed by Deltares. Also, the two packages differ greatly in ease of use and available option. Therefore, those two software packages are an interesting choice for a basis of comparison.

Chapter 4

Simulation packages used

This chapter will address the two simulation packages used, being LISFLOOD-FP (LFP) and Delft3D FM (DFM). Some general structure of the packages will be explained. Features that are of key importance for later sections will be emphasized.

4.1 LISFLOOD-FP

The first model that will be used is LISFLOOD-FP (LFP). LFP has been created for non experts and has been shown to give good results (Bates et al., 2005). Currently, JRC uses LFP for their European scale flooding simulations. This section gives an overview of the LFP simulation package, for a more in depth account of LFP the reader is referred to the manual (Bates, 2013).

4.1.1 Grid

A key feature in the LFP package is the simplicity of the grid. The grid is based on square cells, all with an uniform size. This brings the advantage of being able to generate grids fast (for example by directly inserting a DEM as grid), and using simple computational techniques to solve the flow of water. Figure 4.1 shows where the variables are defined in LFP.

4.1.2 Differential equations

The full shallow water equations described below form the basis for the LFP software package are (Almeida and Bates, 2013)

$$\begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial q_x}{\partial x} + \dfrac{\partial q_y}{\partial y} = 0, \\ \underbrace{\dfrac{\partial q_x}{\partial t}}_{\text{local acceleration}} + \underbrace{\dfrac{\partial u q_x}{\partial x} + \dfrac{\partial v q_x}{\partial y}}_{\text{advection}} + \underbrace{gh\dfrac{\partial \zeta}{\partial x}}_{\text{water slope}} + \underbrace{\dfrac{gn^2\|\boldsymbol{q}\|q_x}{h^{7/3}}}_{\text{friction slope}} = 0, \\ \underbrace{\dfrac{\partial q_y}{\partial t}}_{\text{local acceleration}} + \underbrace{\dfrac{\partial v q_y}{\partial y} + \dfrac{\partial u q_y}{\partial x}}_{\text{advection}} + \underbrace{gh\dfrac{\partial \zeta}{\partial y}}_{\text{water slope}} + \underbrace{\dfrac{gn^2\|\boldsymbol{q}\|q_y}{h^{7/3}}}_{\text{friction slope}} = 0. \end{cases} \tag{4.1}$$

For the definitions of the variables, the reader is referred to the list of variables at the beginning of the report. LFP solves the equations on a staggered grid. The continuity equation is solved at the cell centers while the momentum equations are solved over the cell edges. The mass conservation is implemented in the LFP model as (Neal et al., 2012)

$$h_{i,j}^{t+\Delta t} = h_{i,j}^{t} + \Delta t \frac{q_{i-1/2,j}^{t+\Delta t} - q_{i+1/2,j}^{t+\Delta t} + q_{i,j-1/2}^{t+\Delta t} - q_{i,j+1/2}^{t+\Delta t}}{\Delta x^2}. \tag{4.2}$$

For the momentum conservation equation, LFP offers multiple options. An overview can be found in Table 4.1. During this thesis, two of the solver types available in LFP will be used. Those are referred

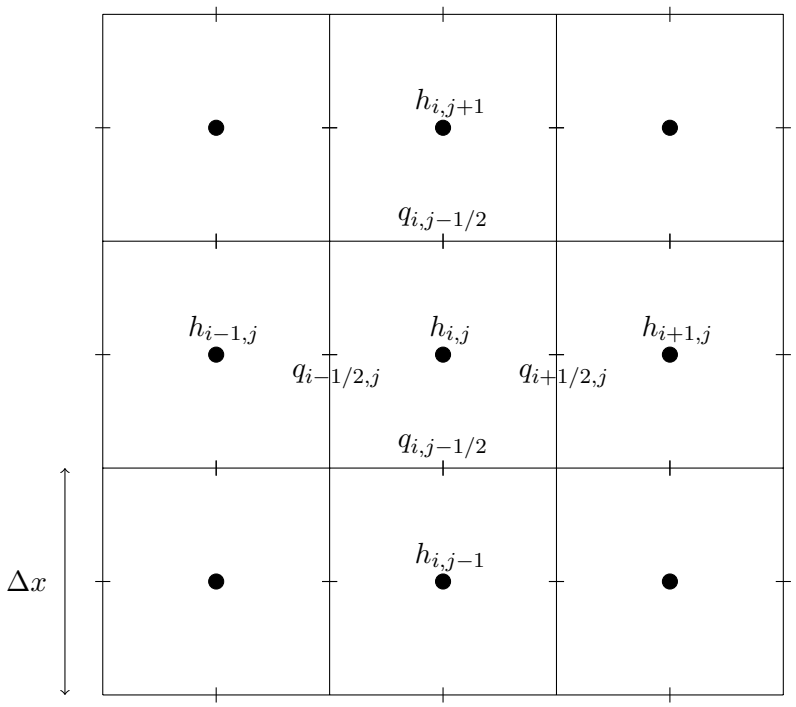


Figure 4.1: Grid definition in LISFLOOD-FP. The variables h, ζ and b are in the cell center (only h shown here) q and u are on the cell edges (only q shown here).

to as the acceleration and Roe solver, and they are discussed in more detail below. LFP describes the friction slope in Equation 2.39 as

$$F_{sx} = -\frac{gn^2\|\boldsymbol{q}\|q_x}{h^{7/3}}, \qquad F_{sy} = -\frac{gn^2\|\boldsymbol{q}\|q_y}{h^{7/3}} \tag{4.3}$$

Adaptive solver

The adaptive solver in LFP solves the parts of the equations known as the diffusion wave approximation (see Section 2.3), which means that the local acceleration and advection term are neglected in the momentum equations (see Section 2.3). It has been shown that the maximum stable time step for the adaptive solver scales with the grid size according to $(1/\Delta x)^4$(Bates et al., 2010). This renders the computational times unfeasible for our study. Therefore, this solver has not been used for the thesis and the details of the solver are omitted in this report.

Acceleration solver

The acceleration method takes all terms into account except for the advection term. This simplification is called the local inertial equations (see Section 2.3). This means that the equations solved in this solver are (Almeida et al., 2012)

$$\begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial q_x}{\partial x} + \dfrac{\partial q_y}{\partial y} = 0, \\ \underbrace{\dfrac{\partial q_x}{\partial t}}_{\text{local acceleration}} + \underbrace{gh\dfrac{\partial \zeta}{\partial x}}_{\text{water slope}} + \underbrace{\dfrac{gn^2\|\boldsymbol{q}\|q_x}{h^{7/3}}}_{\text{friction slope}} = 0, \\ \underbrace{\dfrac{\partial q_y}{\partial t}}_{\text{local acceleration}} + \underbrace{gh\dfrac{\partial \zeta}{\partial y}}_{\text{water slope}} + \underbrace{\dfrac{gn^2\|\boldsymbol{q}\|q_y}{h^{7/3}}}_{\text{friction slope}} = 0. \end{cases} \tag{4.4}$$

The momentum equations are solved separately for the x and y directions. Flows between cells are calculated as (Almeida et al., 2012) (x-direction given here only)

$$q_{i-1/2}^{t+\Delta t} = \frac{\left[\theta q_{i-1/2}^{t} + \frac{(1-\theta)}{2}\left(q_{i-3/2}^{t} + q_{i+1/2}^{t}\right)\right] - g h_f^t \frac{\Delta t}{\Delta x}\left(\zeta_i^t - \zeta_{i-1}^t\right)}{1 + g\Delta t n^2 \|\boldsymbol{q}_{i-1/2}^t\| / h^{7/3}}, \tag{4.5}$$

where $\zeta = h + b$ is the water surface elevation, h_f^t is the depth at the interface between the cells calculated by $max(\zeta_i^t, \zeta_{i-1}^t) - max(b_i, b_{i-1})$. q is the flow per unit width. θ is a factor that determines the amount of artificial numerical diffusion. The default value for θ is 1, which means that there is no artificial diffusion added to the scheme. In this case the equation reduces to

$$q_{i-1/2}^{t+\Delta t} = \frac{q_{i-1/2}^{t} - g h_f^t \frac{\Delta t}{\Delta x}\left(\zeta_i^t - \zeta_{i-1}^t\right)}{1 + g\Delta t n^2 \|\boldsymbol{q}_{i-1/2}^t\| / h^{7/3}}, \tag{4.6}$$

The time stepping is adaptive in the sense that it is limited by

$$\Delta t_{max} = \alpha \frac{\Delta x}{\sqrt{g h^t}}, \tag{4.7}$$

where α ranging between 0.2 and 0.7 gives a stable result for most floodplain flow situations. Using this formulation for the momentum term, the time step scales with $1/\Delta x$. It should be noted that although the difference method uses an implicit formulation, Equation 4.6 is completely explicit in the sense that the old and new time step variables are separated by the equal sign, allowing for a simple implementation. Whether or not this implementation is faster than the implicit implementation of DFM is in the scope of this project.

Roe solver

The exact implementation of the Roe solver has not been found. In the papers referenced, only a brief glance is given on the Roe method. (Neal et al., 2012) describes the Roe method as follows:

"LISFLOOD-Roe is the two-dimensional shallow water model from Villanueava and Wright ((Villanueva and Wright, 2006)); thus, it calculates the flow according to the complete Saint Venant formulation. The method is based on the Godunov approach and uses an approximate Riemann solver by Roe ((Roe, 1981)). The explicit discretisation is first order in space on a raster grid. It solves the full shallow water equations with a shock capturing scheme. LISFLOOD-Roe uses a point-wise friction based on the Mannings equation, while the domain boundary/internal boundary (wall) uses the ghost cell approach. The stability of this approach is approximated by the CFL condition for shallow water models, which is shown in Table II. As the complete model formulation is quite lengthy and relatively well known, it is not reproduced here."

The time step limiter used for Roe's method that is referred to in the above text, when is referred to Table II, is

$$\Delta t_{max} = \alpha \frac{\Delta x}{|u| + \sqrt{g h_t}}, \tag{4.8}$$

In the one dimensional case, the variable u_i^{n+1} is calculated as

$$u_i^{n+1} = u_i^n \frac{\Delta t}{\Delta x}\left[\delta f_{i-1/2}^{+} + \delta f_{i+1/2}^{+}\right] + \Delta t\left[\delta h_{i-1/2}^{+} + \delta h_{i+1/2}^{-}\right], \tag{4.9}$$

where f and h are defined in (Burguete and Garca-Navarro, 2001). The interested reader is encouraged to read the papers and investigate the implementation of Roe's method.

Solver	Dimensions	Terms included	Terms neglected	Time step	Further Tech Det
Routing	1D on 2D grid	User specified velocity and bed slope direction only	All	Adaptive	(Sampson et al., 2012)
Flow-limited	1D on 2D grid	Friction and water slopes	Local and convective acceleration	Fixed	(Bates and Roo, 2000)
Adaptive	1D on 2D grid	As above	As above	Adaptive	(Hunter et al., 2005)
Acceleration	1D on 2D grid, friction terms in 2D	Friction and water slopes, local acceleration	Convective acceleration	Adaptive	(Almeida et al., 2012), (Bates et al., 2010)
Roe	2D	All terms	None	Adaptive	(Neal et al., 2012), (Villanueva and Wright, 2006)

Table 4.1: List of terms that can be used in the LISFLOOD-FP solver

4.1.3 Combining rivers and floodplains

Furthermore, the program is able to combine one dimensional rivers and two dimensional floodplains, where it is possible for a floodplain and a river to be located in the same grid cell. The river and the floodplain are solved separately and when the river overflows there is a mass coupling between the two systems, so the momentum is not coupled here. Also, the program is able to account for rainfall (spatially uniform value), infiltration (spatially uniform value), and evaporation. And finally, the program is able to include weirs in the model.

4.1.4 Advantages

LISFLOOD-FP is a very simple program to setup and offers different simulation techniques easily. It is capable of running the ocean part of the model, the weirs between the ocean and the land, the floodplain and sub-channels in the floodplain. So it could be used to simulate the whole domain that we are interested in. Another advantage is the fact that the software is shareware. The model can solve simplifications of the SWE, which leads to computational efficient variants to a more advanced hydrodynamical model and has been reported giving quite reasonable results.

4.1.5 Disadvantages

The biggest disadvantage is that there is no possibility to have a flexible cell size. This increases the number of necessary cells needed to obtain a similar accuracy to programs that are able to have different cell sizes. Hence, the needed computational time to acquire the necessary accuracy can become significantly larger using this program than some other programs that are capable of having a non-uniform grid. Furthermore, the fact that only the water height is taken into account in the weir overflow formula might be an issue for storm surges. Also, the program requires a DEM that is on the same grid as the raster used in the model, this means that the DEM should be preprocessed for different rasters. Also, the program lacks some error warnings, such as that there will be no error if some control parameter is not recognized, but this control parameter will simple be ignored. So if you make a typo in one of the control parameters, which can happen easily, this control parameter is ignored rather than the program gives you feedback on this unrecognized control parameter. This can lead to serious problems with interpreting results and or error handling. At last, the program does not include the Coriolis force. This implies that the boundary condition needs to be close to the coast.

4.2 Delft3D FM

This section will describe the Delft3D FM suite. DFM is a hydraulic software package developed by Deltares[1] that solves the SWE and is widely applicable. DFM can be applied to problems on many scales and is for example applied for dam break simulations and global storm surge models. Emphasis will be on the differential equations implemented, and the different techniques to implement the bottom into DFM.

[1] http://www.deltares.nl

4.2.1 Grid

We now give a short introduction on the notation. The FM in Delft3d FM stands for Flexible Mesh, thus allowing for non uniform cells. A computational cell in Delft3D FM is defined by three to six netnodes, which are the corners of the cell. Those cells are connected by netlinks. A flownode is defined as the cell circumcentre, and a flowlink is defined as the line segment between two flownodes, see Figure 4.2.

The grid in Delft3D FM is a non uniform grid and can be build up from rectilinear grids, curvilinear grids and unstructured grids composed of triangles, quads, pentagons and hexagons. Those unstructured cells can be used to connect two parts of a grid that have a different resolution, as can be seen in Figure 4.3. It can also be used to create more general shapes near a boundary for example. The combination of those networks have the advantage that they are more suited for arbitrary shapes, but this comes at the cost of extra computational time and memory usage. It is also possible to couple 1D networks to 2D networks.

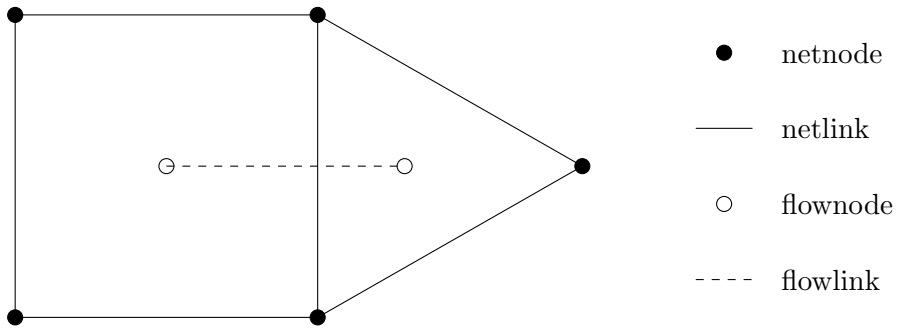


Figure 4.2: Domain definition in Delft3D FM

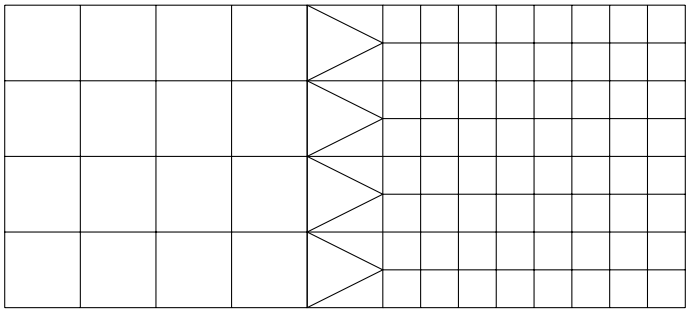


Figure 4.3: Example of how two different rectangular grids can be connected by employing triangles in Delft3D FM.

4.2.2 Partial differential equations

The main partial differential equations solved by DFM are (see the technical reference manual of Delft3D FM[2], hereafter referred to as TRM)

$$\frac{\partial h}{\partial t} + \nabla \cdot (h\mathbf{u}) = 0, \tag{4.10}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{h}\left(\underbrace{\nabla \cdot (h\mathbf{u}\mathbf{u})}_{\text{Advection}} - \underbrace{\mathbf{u}\nabla \cdot (h\mathbf{u})}_{\text{Storage term}}\right) = -g\nabla\zeta + \underbrace{\frac{1}{h}\nabla \cdot \left(\nu h\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)\right)}_{\text{Diffusion}} + \frac{\boldsymbol{\tau}}{h\rho}, \tag{4.11}$$

[2]https://content.oss.deltares.nl/delft3d/manuals/D-Flow_FM_Technical_Reference_Manual.pdf

where $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)^T$, ζ is the water level, h the water depth, $\mathbf{u}$ the velocity vector, g the gravitational acceleration, ν the viscosity, ρ the water mass density and $\boldsymbol{\tau}$ is the bottom friction. Equation 4.11 is derived from the momentum equation

$$\frac{\partial h\mathbf{u}}{\partial t} + \nabla \cdot (h\mathbf{u}\mathbf{u}) = -gh\nabla\zeta + \nabla \cdot \left(\nu h \left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)\right) + \frac{\boldsymbol{\tau}}{\rho}, \tag{4.12}$$

by expanding $\frac{\partial h\mathbf{u}}{\partial t} = h\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\frac{\partial h}{\partial t}$, using Equation 4.10 and dividing by h.

4.2.3 Spatial discretization of the continuity equation

These partial differential equations are spatially discretized, as will be discussed in this part of the report. The discussion will be concise and only state the results, for a full review on the matter the reader is referred to the TRM. The continuity equation is discretized as

$$\frac{dV_k}{dt} = -\sum_{j \in \mathcal{J}(k)} A_{u_j} s_{j,k}, \tag{4.13}$$

where $\mathcal{J}(k)$ is the set of vertical faces that bound cell k and $s_{j,k}$ accounts for the orientation of face j with respect to cell k V_k is the volume of the water column at cell k, computed using algorithm 20 of the technical reference manual. A_{u_j} approximates the flow area of face j and h_{u_j} is the water depth of face j, both will be described below.

Bed level types used in DFM

Delft3D FM can use six different definitions to implement the bathymetry into the model. Two of those six are described here since they will be used later on. The first is "bed level type = 1" (blt1), and the second is "bed level type = 3 (blt3)". The key difference is that for blt1 the bed geometry is user defined by specifying the cell-centered values, whereas blt3 has the bed geometry defined at the net nodes, see Figure 4.5. The difference in implementation means that the face-based water depth h_{u_j} and the wet cross-sectional area A_{u_j} are calculated differently. Let us look at the face z_1z_2 in Figure 4.5, and assume the water flows from ζ_1 to ζ_2. for blt1, DFM defines

$$h_{u_j} = \zeta_1 - \max\{bl1, bl2\}, \tag{4.14}$$

$$A_{u_j} = \omega_{u_j} h_{u_j}, \tag{4.15}$$

where ω_{u_j} is the width of the face. But on the other hand, for blt3, DFM defines

$$h_{u_j} = \zeta_1 - \min\{z_1, z_2\}, \tag{4.16}$$

and A_{u_j} is defined as the area of Figure 4.4.

In order to make a fair comparison between LFP and DFM, blt = 1 will be used in DFM. The default way of implementing the DEM in DFM is to apply a bottom level at the corners of a cell (blt3), but since LFP applies the bottom level at the cell center, this is also done in DFM. This allows us to keep the grid the same in both methods (and does not require interpolation of the DEM). In this way, the input data is equal in all methods. Since DFM is designed to have the bottom level applied at the cell corners, a second DFM model is evaluated to study the effect of using the default option (and thus letting DFM interpolate the DEM in order to obtain values at the cell corners). The results of this can be seen in Section 7.4.2.

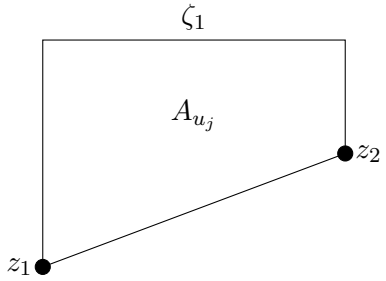


Figure 4.4: Wet cross-sectional area A_{u_j} for blt3

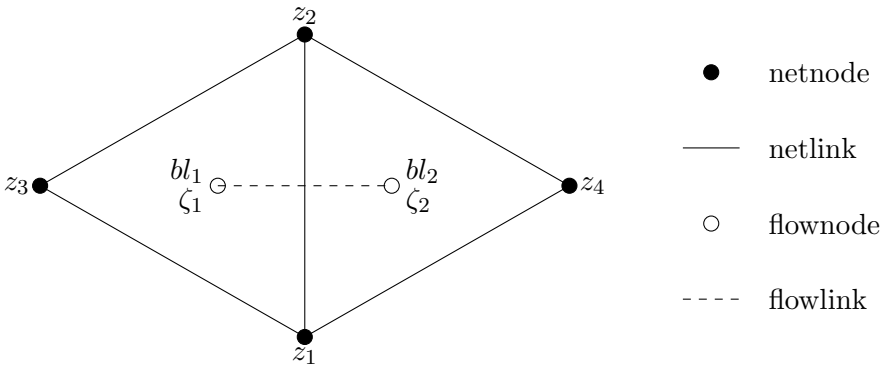


Figure 4.5: Different bottom level types

4.2.4 Spatial discretization of the momentum equation

The momentum equation (Equation 4.11) is discretized at the faces and in face-normal direction. The water level gradient term projected in the face-normal direction is discretized as

$$g\nabla\zeta|_j \bullet \boldsymbol{n}_j \approx \frac{g}{\delta x_j}\left(\zeta R(j) - \zeta L(j)\right), \tag{4.17}$$

and the bed friction term is discretized as

$$\left.\frac{\boldsymbol{\tau}}{h\rho}\right|_j \bullet \boldsymbol{n}_j \approx -\frac{g\|\boldsymbol{u}_j\|}{C^2\hat{h}_j}u_j. \tag{4.18}$$

$\hat{h}_j$ acts as an hydraulic radius, the definition can be found in the TRM, and depends on the conveyance type. The advection is discretized as

$$\left[\frac{1}{h}\left(\nabla \bullet (h\boldsymbol{u}\boldsymbol{u}) - \boldsymbol{u}\nabla \bullet (h\boldsymbol{u})\right)\right]_j \bullet \boldsymbol{n}_j \approx \mathcal{A}_{\mathrm{i}j}u_j + \mathcal{A}_{\mathrm{e}j}, \tag{4.19}$$

where the terms $\mathcal{A}_{\mathrm{i}j}$ (implicit part) and $\mathcal{A}_{\mathrm{e}j}$ (explicit part) play an important role and are called advi and adve respectively. The diffusion term is defined as

$$\left[\frac{1}{h}\nabla \bullet \left(\nu h\left(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T\right)\right)\right]_j \bullet \boldsymbol{n}_j = \mathcal{D}_j. \tag{4.20}$$

The above equations result in the following spatial discretization of Equation 4.11

$$\frac{\mathrm{d}u_j}{\mathrm{d}t} = -\frac{g}{\Delta x_j}\left(\zeta R(j) - \zeta L(j)\right) - \mathcal{A}_{\mathrm{i}j}u_j - \mathcal{A}_{\mathrm{e}j} - \mathcal{D}_j - \frac{g\|\boldsymbol{u}_j\|}{C^2\hat{h}_j}u_j. \tag{4.21}$$

4.2.5 Momentum advection term in Delft3D FM

DFM offer many options to implement the momentum advection terms and the general basis for this implementation will be outlined here. Since the focus of this thesis will be mainly on two implementation types, those will be further described below. The general notation used for the advection terms is

$$\begin{aligned}\mathcal{A}_{\mathrm{e}j} =& A_{L_j} \sum_{l\in\mathcal{J}^*(L(j))} q_l^* s_{l,L(j)} \boldsymbol{u}_{ul} \bullet \boldsymbol{n}_j - q_l^{**} s_{l,L(j)} \left(1-\theta_{l,L(j)}\right) u_{L_j}^* + \\ & A_{R_j} \sum_{l\in\mathcal{J}^*(R(j))} q_l^* s_{l,R(j)} \boldsymbol{u}_{ul} \bullet \boldsymbol{n}_j - q_l^{**} s_{l,R(j)} \left(1-\theta_{l,R(j)}\right) u_{L_j}^*,\end{aligned} \tag{4.22}$$

and

$$\begin{aligned}\mathcal{A}_{\mathrm{i}j} =& - A_{L_j} \sum_{l\in\mathcal{J}^*(L(j))} q_l^{**} s_{l,L(j)} \theta_{l,L(j)} \\ & - A_{R_j} \sum_{l\in\mathcal{J}^*(R(j))} q_l^{**} s_{l,R(j)} \theta_{l,R(j)}.\end{aligned} \tag{4.23}$$

The terms $\mathcal{J}^*, q_l^*, q_l^{**}, \theta_{l,L(j)}, \theta_{l,R(j)}, u_{L_j}^*, u_{R_j}^*, A_{L_j}, A_{R_j}$ depend on the chosen advection scheme. For the full list of variables, the reader is referred to the manual.

No advection

When switching off advection (setting the option Advection type = 0 in the parameter mdu file), we get both $\mathcal{A}_{\mathrm{e}j} = \mathcal{A}_{\mathrm{i}j} = 0$. This means that the spatial discretization of the momentum equation is reduced to

$$\frac{\mathrm{d}u_j}{\mathrm{d}t} = -\frac{g}{\Delta x_j}\left(\zeta R(j) - \zeta L(j)\right) - \frac{g\|\boldsymbol{u}_j\|}{C^2\hat{h}_j} u_j. \tag{4.24}$$

This corresponds more or less with the equations solved using the acceleration method of LFP (see Section 4.1.2), with key difference that DFM solves for the primitive variable u whereas LFP solves for the conservative variable $q = u * h * \Delta y$. In this sense, the formulation used in DFM without advection is not momentum conserving.

Advection

When using the default option for advection (Advection type = 33 in the parameter mdu file), we get a more complicated formulation for $\mathcal{A}_{\mathrm{e}j}$ and $\mathcal{A}_{\mathrm{i}j}$. The unknowns listed below Equation 4.23 are

$$\begin{aligned}\mathcal{J}^* &= \mathcal{J} \\ q_l^* &= q_{al} \\ q_l^{**} &= q_{al} \\ \theta_{l,L(j)} &= 0 \\ \theta_{l,R(j)} &= 0 \\ u_{L_j}^* &= u_j \\ u_{R_j}^* &= u_j \\ A_{L_j} &= \frac{\alpha_j}{\alpha_j V_{L(j)} + (1-\alpha_j) V_{R(j)}} \\ A_{R_j} &= \frac{(1-\alpha_j)}{\alpha_j V_{L(j)} + (1-\alpha_j) V_{R(j)}},\end{aligned} \tag{4.25}$$

where $\mathcal{J}$ is the set of faces, q_{al} is the upwind flux, u_j is the face normal velocity at face j, α_j is the non-dimensional distance from the left cell center to the face j and $V_{L(j)}$ and $V_{R(j)}$ are the volumes of the cell left and right of the face j. For the exact algorithm that calculate the variables, the reader is referred to the TRM.

Diffusion term

The diffusion is discretized as

$$\left[\nabla \bullet \left(\nu h \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right)\right)\right]_{\Gamma_j} \bullet \boldsymbol{n}_j = \left(\alpha_j \boldsymbol{d}_{L(j)} + (1 - \alpha_j)\right) \bullet \boldsymbol{n}_j, \tag{4.26}$$

where $\boldsymbol{d}_{L(j)}$ and $\boldsymbol{d}_{R(j)}$ are respectively the cell-centered diffusion terms, and α_j is again the non-dimensional distance from the left cell center to the face. The viscosity coefficient ν can be computed with different techniques, the interested reader is referred to the manual. No further attention will be dedicated to this since diffusion is mostly switched off during the research.

Chapter 5

One dimensional test cases

In order to get an understanding of the differences in software packages and setups, two simple one-dimensional test cases are performed. The main goal is to see if characteristics can be found for which the different simulation packages and settings behave similar or not. Both test cases are derived from the Xynthia model described in Chapter 6. and represent a one-dimensional slice of this model.

5.1 Test case I

5.1.1 Model setup

Test case I is a channel with cells of 90 m with obstacle. Water levels are supplied at both sides. It is a simplified one dimensional section of the model used in Chapter 6. This simple test case consists of 41 cells of 90 x 90 m, where the first 20 cells and the last 16 cells have a bed level elevation (b) of 0 m. The five cells in the middle have a bed level elevation of y meters, which represents an obstacle:

$$b = \begin{cases} 0 \text{ m}, & 0 \text{ m} < x < 1800 \text{ m} \\ y \text{ m}, & 1800 \text{ m} \leq x \leq 2250 \text{ m} \\ 0 \text{ m}, & 2250 \text{ m} < x < 3690 \text{ m} \end{cases}, \tag{5.1}$$

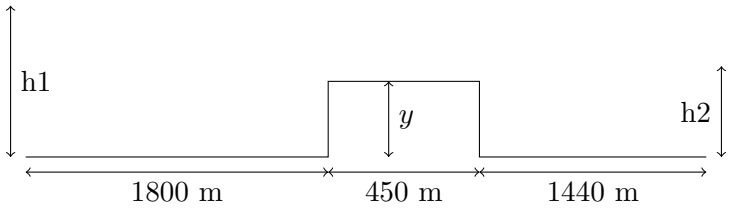


Figure 5.1: Graphical representation of Equation 5.1

where y takes the values 2, 3, 4 and 4.5 m for different model runs. At the boundaries (left and right side) of the channel, a water level is applied. They are represented by h1 and h2 in Figure 5.1. They are time dependent and their values can be seen in Figure 5.2. Normally, the right boundary would be represented by an water slope. LFP for example, uses the water slope that is found in the cells before the last cells and applies the same slope at the boundary. In this experiment, the water levels are supplied externally at both boundaries to focus on the effect the chosen method has around the obstacle.

The main goal of this analysis is to see the difference between the LFP and DFM models, and the effect of advection. Four different model setups are run and compared: LFP without advection; LFP with advection; DFM without advection; DFM with advection. The steady state solution of the four runs are compared, and for some key locations along the channel the water level is plotted against time to see how the different model setups evolve in time. Although the goal of this thesis is to reduce computational time, the time will not be reported in this analysis. The reason for this is that they

are too small to be measured accurately with the knowledge and tools available to the author. Thus, the focus here will be on the model results themselves. The time step used is 5 seconds for all the models.

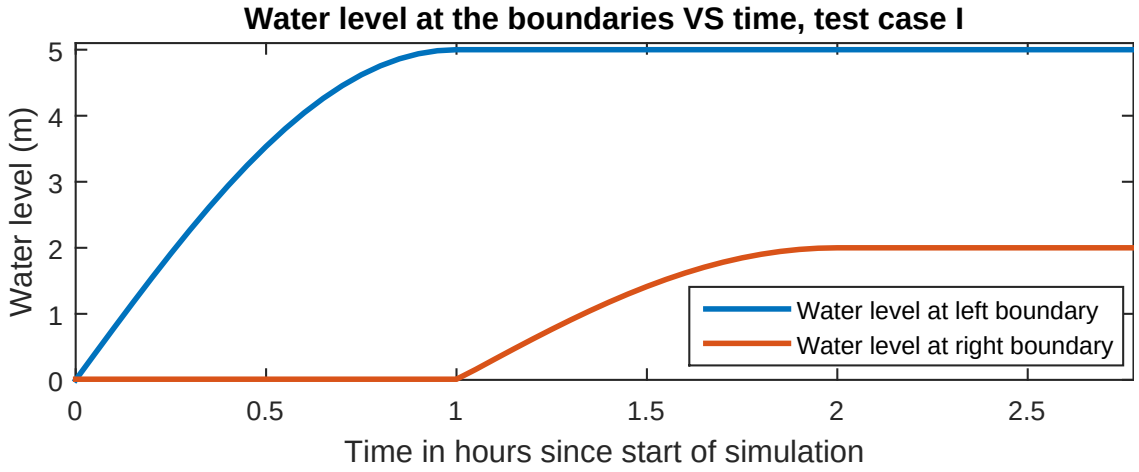


Figure 5.2: Boundary conditions for test case I.

5.1.2 Results

For each of the obstacle heights, the steady state results for the different model setups can be found in Figure 5.3. The LFP with advection model setup results in very different behavior compared to the other three model setups and shows a different characteristic across the jump than the other three. This occurs for all of the investigated obstacle heights. The three other models show deviations of up to the order of decimeters, but only across the jump, or in the first cells after the jump.

In the lower part of the figures, the water of the DFM no adv model is subtracted from the water levels of the LFP no adv, LFP with adv and DFM no adv. When comparing the DFM models, it can be seen that for all obstacles, small deviations occur across the jump. The largest deviations are found for the obstacle of three meters, but for all cases there is an increased water level in the last cell on the jump for the DFM no adv models. When comparing both the LFP and DFM models without advection, it can be seen for the obstacle of 2 and 3 meter that the LFP model has a more conservative water level after the obstacle. Overall, when ignoring the LFP with adv model, the resulting water levels a little after the obstacles deviate less than 10 cm between the models. It is interesting to see that for the obstacle of 3 meter, the water levels of LFP no adv and DFM with adv (after the jump) are closer to each other than when comparing the two DFM models.

In Table 5.2, the Root Mean Square Difference (RMSD) between the different models can be seen for the obstacle of 3 meters height. Note that in this test case, the deviation caused by switching to another simulation package is of the same order as in or excluding advection in the DFM simulation package, both being in the order of centimeters, where the total water level is in the order of meters. This deviation is smaller than the typical vertical error ($\sim$ 0.1 m) in high resolution terrain data (Bates et al., 2010). The deviations with the LFP with adv model and the others are $\sim$ 0.2 m.

The total flux of water for the stationary situation can be seen in Table 5.3. Again, it is observed that the LFP with advection model shows very different behavior compared to the other models, in this case when looking at the higher obstacles. For the obstacle of 4.5 m, the flux is about 5.5 times as large as the fluxes given by the other model setups. For the other models, it can be seen that for the cases without advection, LFP is more conservative in terms of steady state volume flux than its DFM counterpart. For DFM, excluding the advection term leads to higher fluxes than when including this

term. It is again notices that switching between two software packages has roughly the same impact as in or excluding advection in DFM.

The Froude number

$$Fr = u/\sqrt{gh}, \tag{5.2}$$

has been calculated for the DFM with adv simulations. For the respective obstacles of 2, 3, 4 and 4.5 meters, the maximum Froude number is 0.52, 1.15, 2.61 and 5.72. For the obstacle of 4.5 meters, the Froude number is plotted for each cell center in Figure 5.4. The other obstacle heights show a similar graph for the Froude number. The Froude number is an important indicator here since the inertial models, i.e. the models without advection, will simulate a slope that is different from the full SWE model, see Section 3.5 for details. As soon as the Froude number reaches unity or is larger than unity, the water slopes will behave differently. It is expected that the differences between the model with and without advection will start to deviate when the Froude number reaches unity (Almeida and Bates, 2013). Comparing the RMSD of DFM no adv and DFM with adv for the four different obstacle heights with the maximum Froude number can be seen in Table 5.1. No apparent relation is seen by the maximum Froude number and the RMSD. The simulation with the highest Froude number for the advection model has the smallest RMSD with the model without advection.

	Maximum Froude number	RMSD
2.0 meter	0.52	0.0283
3.0 meter	1.15	0.0342
4.0 meter	2.61	0.0094
4.5 meter	5.72	0.0019

Table 5.1: Maximum Froude number and RMSD of DFM no adv and DFM with adv for the four different obstacle heights.

The above steady state only tells part of the story. For the Xynthia floods, which has water levels rising in the order of hours, it is also important to investigate the evolution with respect to time between the simulation approaches. For different observation points, located at 855, 1755, 1845, 2205, 2295 and 2655 meter (cells 10, 20, 21, 25, 26 and 30 respectively), the time evolving water levels have been plotted in Figure 5.5a. This has been done for the obstacle of 3 meter height. At location p1, it is interesting to see that there is a clear difference between the LFP and DFM model, but the effect of advection can not be observed so clearly here. Overall the time dependencies of the models show similar behavior, even for the LFP with advection model. Wetting occurs roughly at the same time and the curve is similar. This is to be expected for such a simple test case.

	LFP no adv	LFP with adv	DFM no adv	DFM with adv
LFP no adv		0.2149	0.0406	0.0448
LFP with adv			0.2235	0.2201
DFM no adv				0.0342
DFM with adv				

Table 5.2: RMSD (meters) for the water levels obtained for the four different model setups with coastal protection of 3 meter for test case I.

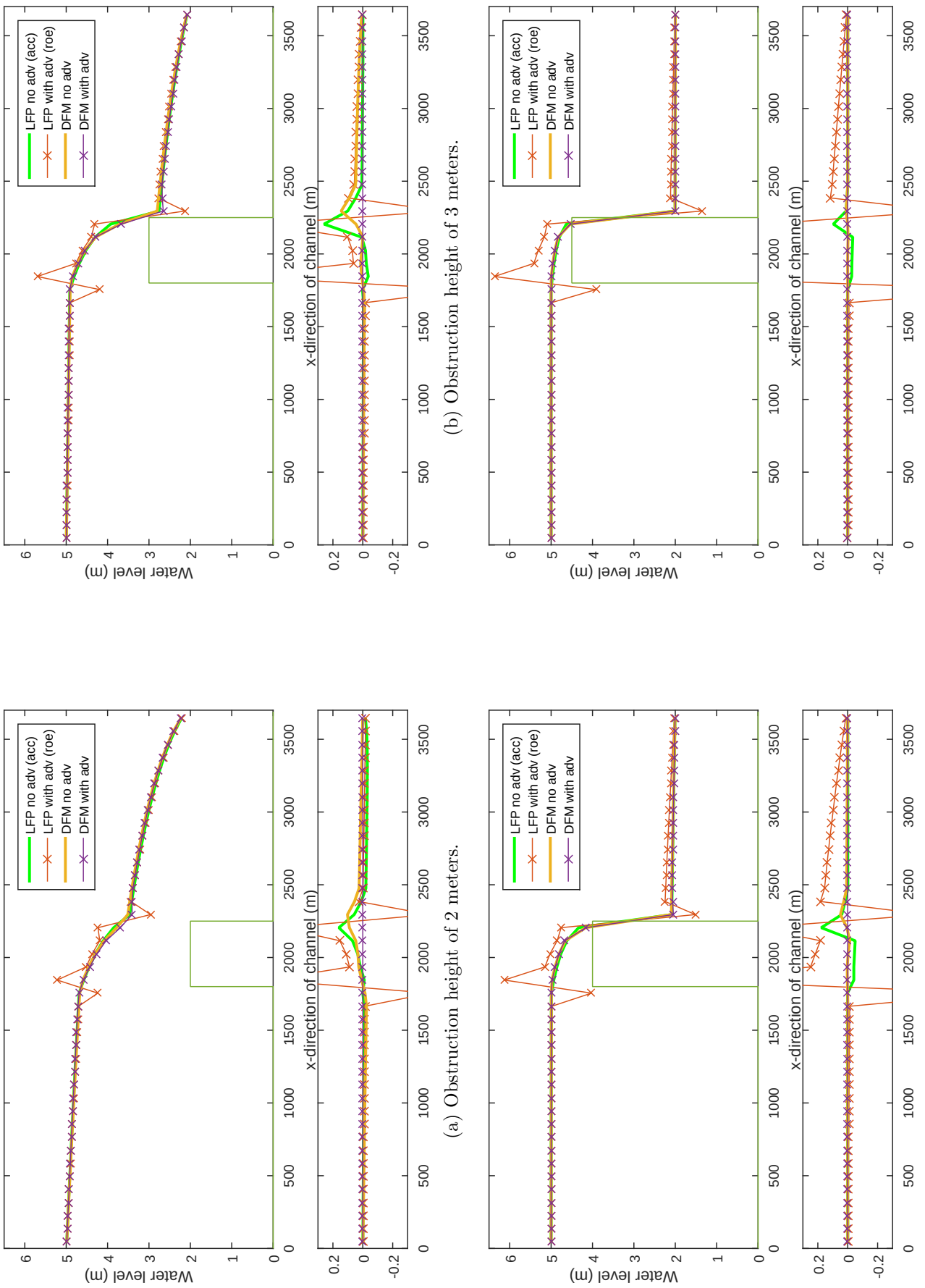
5.1.3 Conclusion and discussion

It can be concluded that all models except the LFP with advection model behave rather similar in terms of water level and flux. The deviation introduced by including or excluding advection in the DFM model is of the same order as the deviation between the LFP and DFM models without advection. The same holds for the volume flux.

	LFP no adv	LFP with adv	DFM no adv	DFM with adv
2.0 meter	329.7	332.3	336.5	322.6
3.0 meter	180.5	193.4	187.7	175.8
4.0 meter	46.0	87.6	48.3	46.1
4.5 meter	10.5	56.8	11.1	10.9

Table 5.3: Steady state volume flux (m^3/s) for different model setups for test case I.

The boundary condition on the right side of the model is not perfect. Although the model shows what happens across the jump, the water levels on the right side of the model are forced in the direction of the 2 meter height boundary condition. This has been acknowledged and is addressed in Test case II described in the next section.



(a) Obstruction height of 2 meters.

(b) Obstruction height of 3 meters.

(c) Obstruction height of 4 meters.

(d) Obstruction height of 4.5 meters.

Figure 5.3: The lower figures represents the difference between a model and the model DFM with advection.

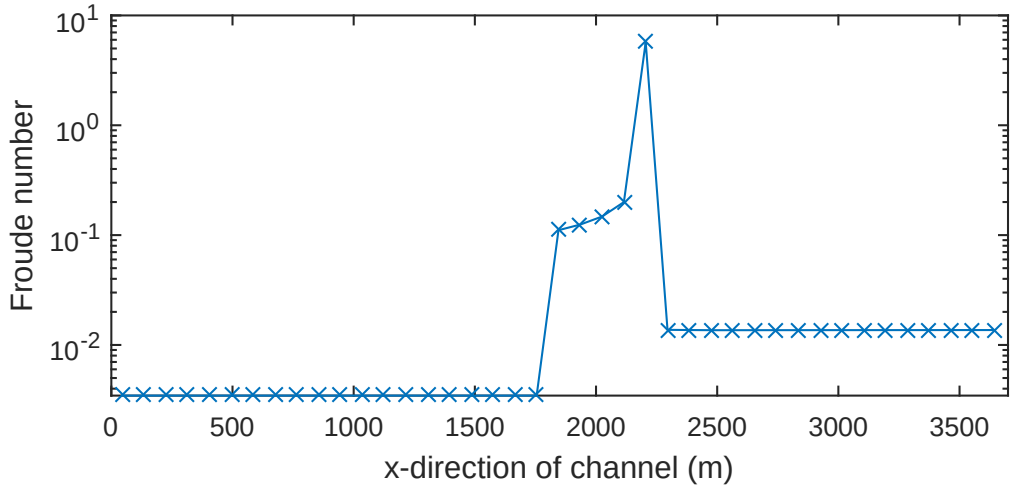
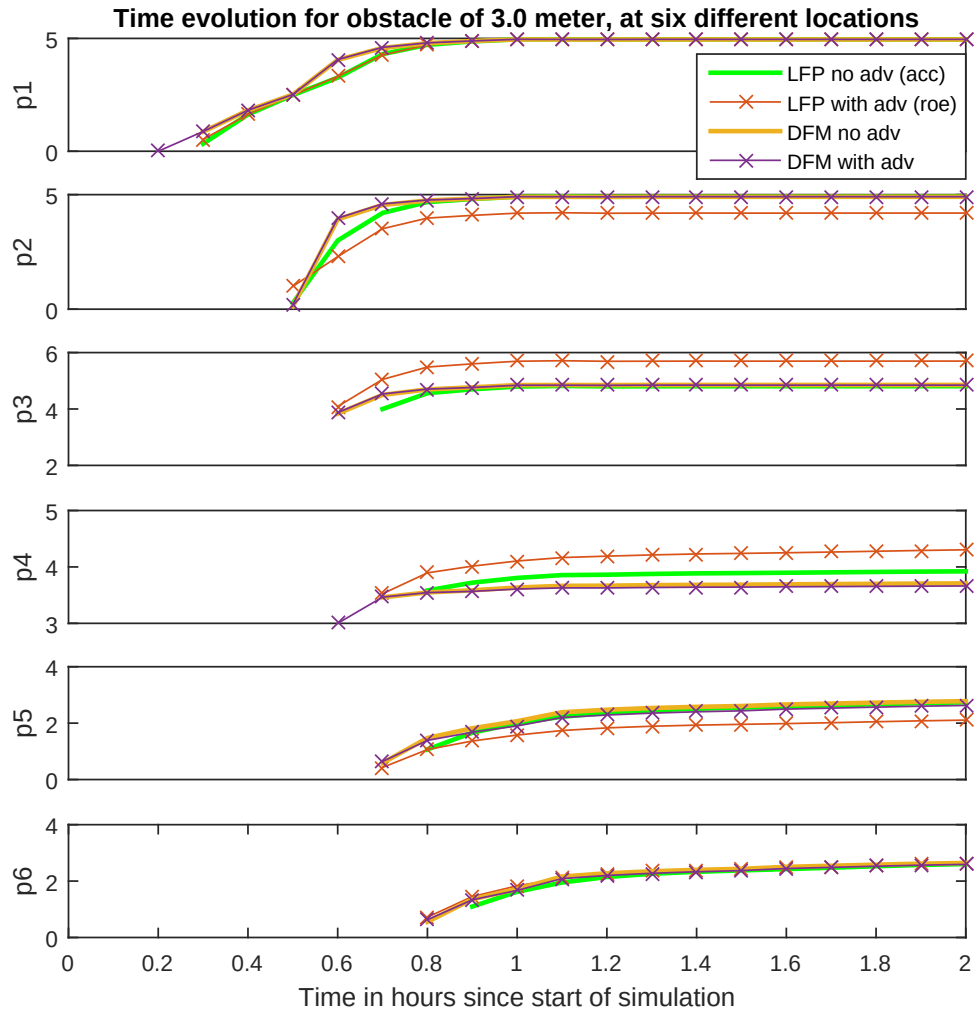
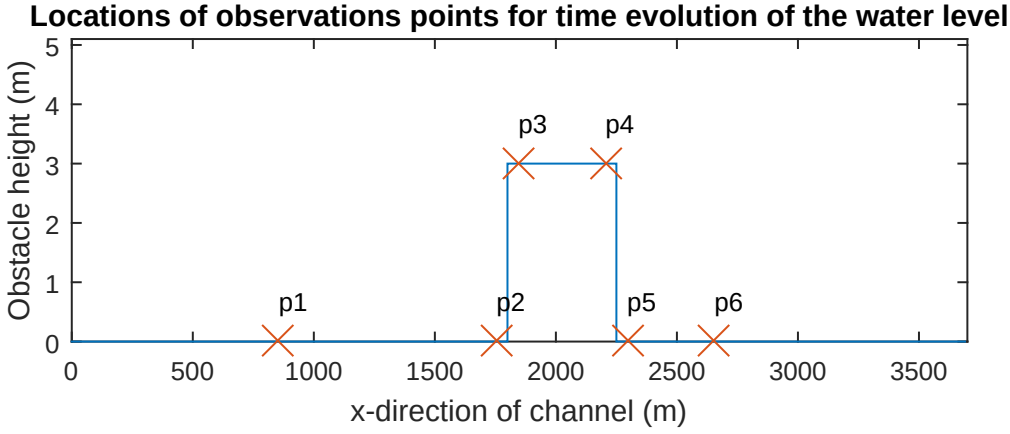


Figure 5.4: Froude number for the DFM with adv model with obstacle height of 4.5 m, Test case I.



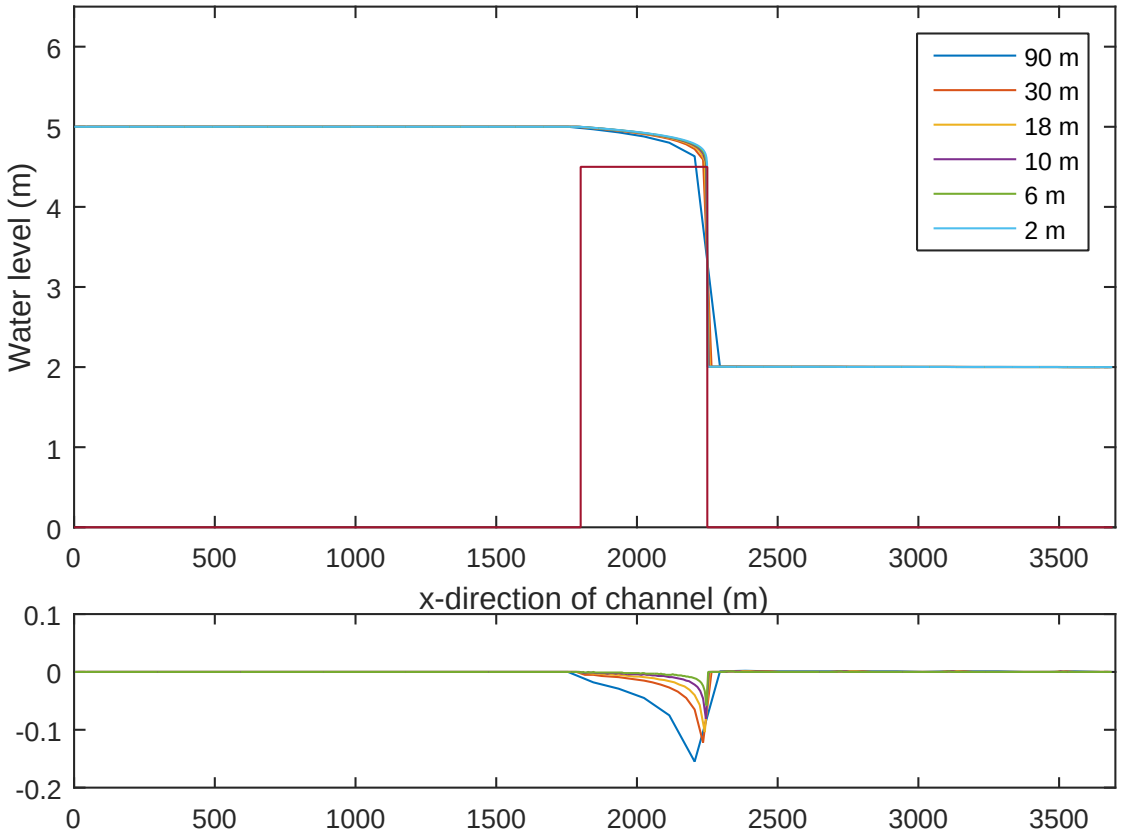
(a) Time evolution of water level for six different points for test case I. On the y-axis, the water level is represented in meters. The location of the observations points are visualized in Figure 5.5b.



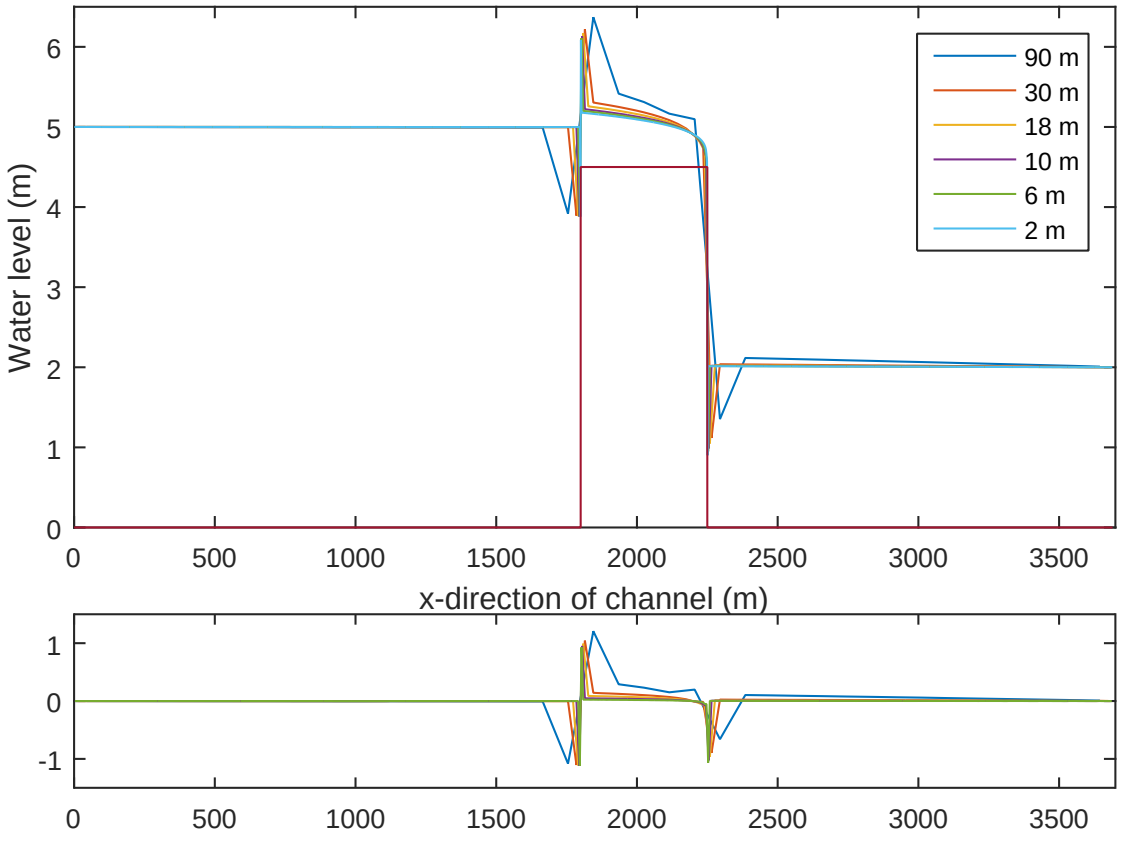
(b) Locations of the observations points used in Figure 5.5a.

5.2 Dependency on cell size

In order to analyze the large differences between the LFP with adv model and the other models, grid refinements have been performed for both the LFP models. Cell sizes of 2, 6, 10, 18 and 30 meters have been used. The water levels for the LFP models can be seen in Figure 5.6. In the lower figure the water levels are subtracted by the water levels occurred in the 2 m model. It can be seen that the LFP no adv is only locally dependent on the grid refinements. However this effect does not propagate throughout the model. For the LFP with adv model, it can be seen that the effects are also mainly local, however when comparing the 90 m model with the 2 m model, the deviations in water level remains visible until almost the end of the model. When analyzing the peaks occurring around the jumps it is observed that this peaks always occur in the first cell center before and after a jump.



(a) LFP without advection.



(b) LFP with advection.

Figure 5.6: Different grid sizes for Test Case I for the LFP model. (a) LFP without advection, (b) LFP with advection. Note the difference in units on the y-axis

5.3 Test case II

5.3.1 Model setup

As a second experiment, a similar setup as for test case I is used. The difference is that now there is a floodplain of $125 * 90$ m $= 11250$ m behind the obstacle, and there is a closed boundary at the end of the floodplain. The length of the floodplain has been chosen in such a manner that the water overtopping the obstacle does not reach the end of the floodplain before the water level in front of the obstacle is lower than the obstacle again. The boundary condition on the west has been slightly altered. Where in the previous test case the water level stayed at 5 m, this would not give the desired result for this experiment. A boundary of a sine function has been chosen, which can be seen in Figure 5.7. This boundary condition is similar in amplitude and time scale as the one used for the pilot study described in Chapter 7. This experiment is conducted in order to see the effect of the chosen method on the amount of overtopped water. The model has been run for different obstacle heights than the previous test case.

5.3.2 Results

The average water level behind the obstacle is plotted for the simulations in Figure 5.9, which represents the total overtopped volume. In Table 5.4, the water levels with respect to the water levels of DFM with adv are shown. It is interesting to observe that the LFP no adv and DFM with adv show very similar results, where DFM no adv is a little less conservative. LFP with adv again shows behavior which does not correspond to the other three models. Furthermore, LFP no adv is always within 1% of DFM with adv in the current settings, while DFM no adv and DFM with adv show a difference of 4 to 6%. It is interesting to note that LFP no adv and DFM with adv are generally more alike than DFM no adv and DFM with adv.
In Figure 5.8, the propagation of the waterfront can be observed. This shows very similar results for all the models except for LFP with adv model. LFP with adv models shows faster propagation speed than the other models for the current test case.

	LFP no adv	LFP with adv	DFM no adv	DFM with adv
2.5 meter	1.00	0.96	1.04	1
3.0 meter	1.01	1.20	1.06	1
3.5 meter	1.01	1.65	1.06	1
4.0 meter	0.99	2.95	1.04	1

Table 5.4: Water level after the obstacle at the end of the simulation, scaled to 1 for DFM with adv.

5.3.3 Conclusion and discussion

For Test case II, three of the four models behave very similar in terms of water level, propagation speed and total volume that flowed over the obstacle. The fourth model, LFP with adv, shows very different behavior. The higher the obstacle, the larger the differences with LFP with adv and the other models become. In general, for these problems the three other models behave similar.

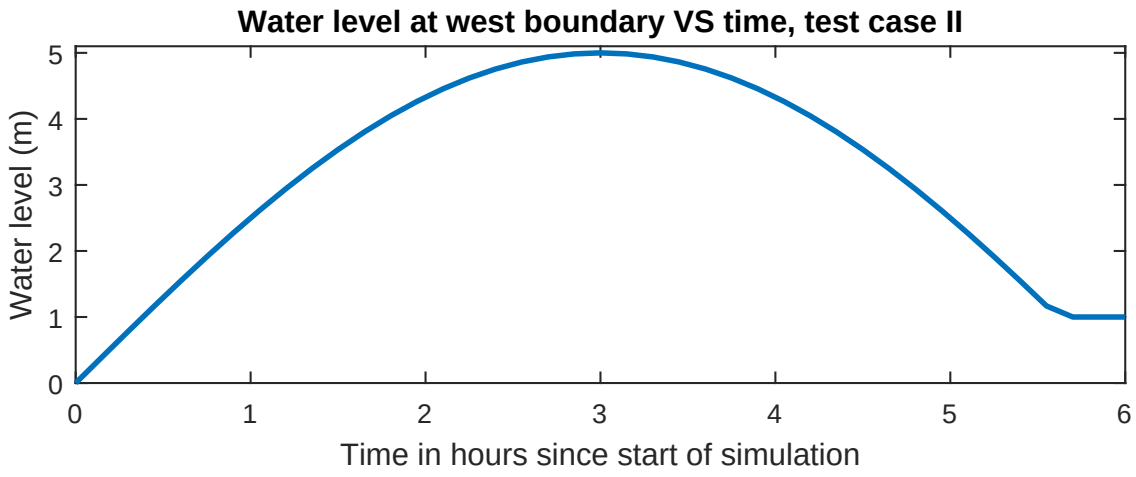


Figure 5.7: Boundary conditions for test case II.

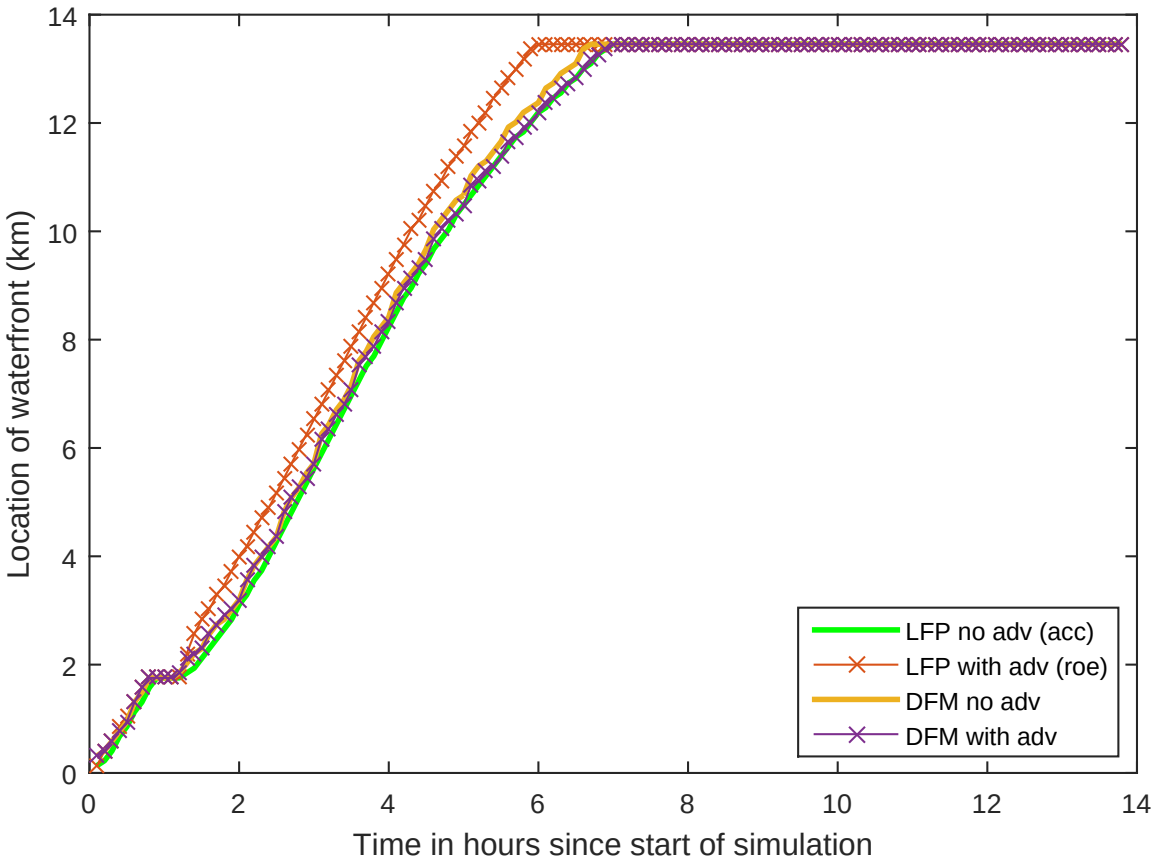
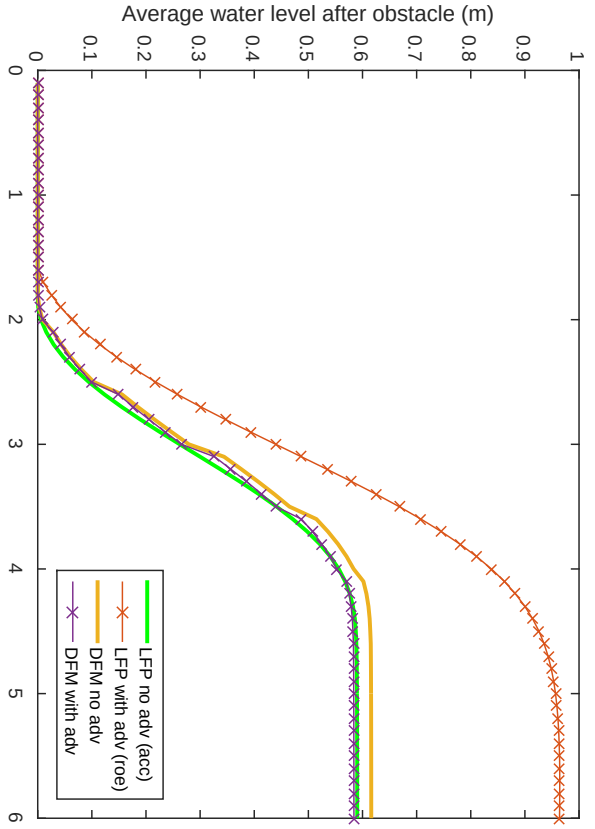
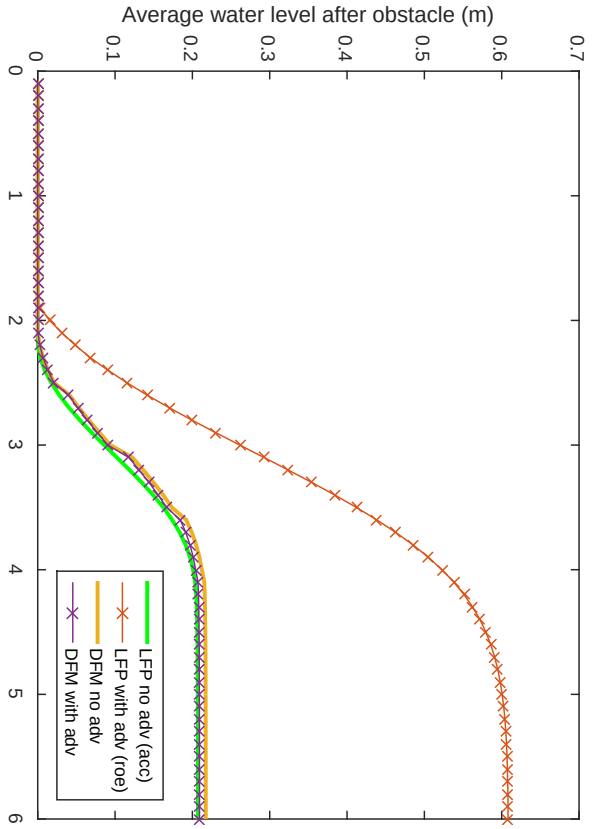


Figure 5.8: The propagation of the waterfront in the different model setups for test case II. Obstacle height is 3.0 meter.



(a) Obstruction height of 2.5 meters.



(b) Obstruction height of 3 meters.

(c) Obstruction height of 3.5 meters.

(d) Obstruction height of 4 meters.

Figure 5.9: Average water level after the obstacle at the end of the simulation for Test case II.

Chapter 6

Two dimensional real world models

A model based on an internally produced 90 m SRTM dataset by JRC has been build. The JRC 90 m STRM dataset is a slightly altered version of the 90 m SRTM dataset. The altercations have been made in order to implement the coastal protection in the DEM. This model has been set up in order to compare the results obtained with the currently used methods at JRC with a DFM model. The data used for this model is based on the Xynthia storm that occurred in Western France in 2010, and is described by (Bertin et al., 2014) and (Breilh et al., 2013) in more detail. Data used as input for the models are: An hydrograph based on measurements from the La Pallice tide gauge, a DEM based on SRTM adapted to include coastal protection and manning friction coefficients based on the MARINE land cover map. Since the exact procedure on how this input data is generated is not known by the author, the results in this section will be mostly restricted to the comparison between the methods. Any comparisons with the observed values are only indicative. The chapter starts by showing the input data and then shows how this data is implemented in the models. The actual model runs and results are described in Chapter 7.

6.1 Input data and model setup

6.1.1 The domain of the model

The domain is an area in Western France in the Bay of Biscay. The area is depicted on a map in Figure 6.1. It is roughly 90 x 70 km in size. The corners of the domain can be seen in Table 6.1. Not the whole domain is simulated. Only the land and a seaward shelf of 2.5 km is simulated. This shelf has a bed level of -1 m. The simulated area can be seen in Figure 6.2.

6.1.2 DEM

The DEM used as an input for this model can be seen in Figure 6.2. The white area on the left is masked (LFP) or removed from the grid (DFM). This white area is not the same as the wet area defined in the section land definition. The DEM is an adapted version of the 90 m STRM dataset. The adaptions that are made are done in order to incorporate coastal protection in the DEM, which is usually missing in data of this resolution. The method employed to incorporate coastal protection is unclear to the author, however is not of importance to the test case.

6.1.3 Manning friction coefficient

The manning coefficients are generated using the MARINE land cover map in combination with known Manning friction values corresponding to this land cover type. A map can be seen in Figure 6.3.

6.1.4 The grids used

The grids of the models will be addressed here. The goal is to have the grids as similar as possible in the two software packages. A representation of the grid used in LFP can be seen in Figure 6.2, where the white area is masked (and thus not simulated). The grid used in DFM can be seen in Figure 6.4.

Figure 6.1: The study area used in Chapter 7

In LFP, the domain is implemented by using the DEM directly. In order to use the same input data for the DFM bottom level model, this data has been transformed into a .xyb file (a file used in DFM to allocate depth values corresponding to an x- and y-coordinate), where the data has been shifted by half the cell size to account for the fact that LFP uses the lower left corner as a reference point for a cell and uses this value for the whole cell. So this actually means that the value used as input for DFM should be shifted half a cell up and to the right.

6.1.5 Boundary condition

As a boundary condition, a time varying Hydrograph is applied to the meandering boundary at the left. The north, east and south boundary have zero flux imposed. The hydrograph can be seen in Figure 6.5. The hydrograph is the same as used in (Vousdoukas et al., 2016). The location of where the boundary is implemented can be seen in Figure 6.6 for both LFP and DFM. Note the small difference near $(x, y) = (3.46, 2.64) \times 10^6$.

6.1.6 DFM with default bed level type

The third model that has been set up is done with the default settings for the bed level type (bed level type = 3) in DFM. Before going in detail on how this model is implemented, a remark is made. In order to compare LFP and DFM as well as possible, the design choice was made to keep the cells equal, and furthermore to define the bed level in the center of the cell for DFM, since this is also how the bed level is defined in LFP. This brings a complication to implementing the original bed level type in DFM, since the DEM is only giving information at the cell centers. But now, with the bed level defined at the corners, we face the problem depicted in Figure 6.7. This problem can be solved by shifting the whole model by 45 meters to the northwest, but this would results in issues comparing the flooded extent because the cells will represent a different area. A way to circumvent this would be to divide each cell in 4 smaller cells, and compare the flooded extent based on those smaller cells. However, this would bring forward other issues. It has been chosen for this study to

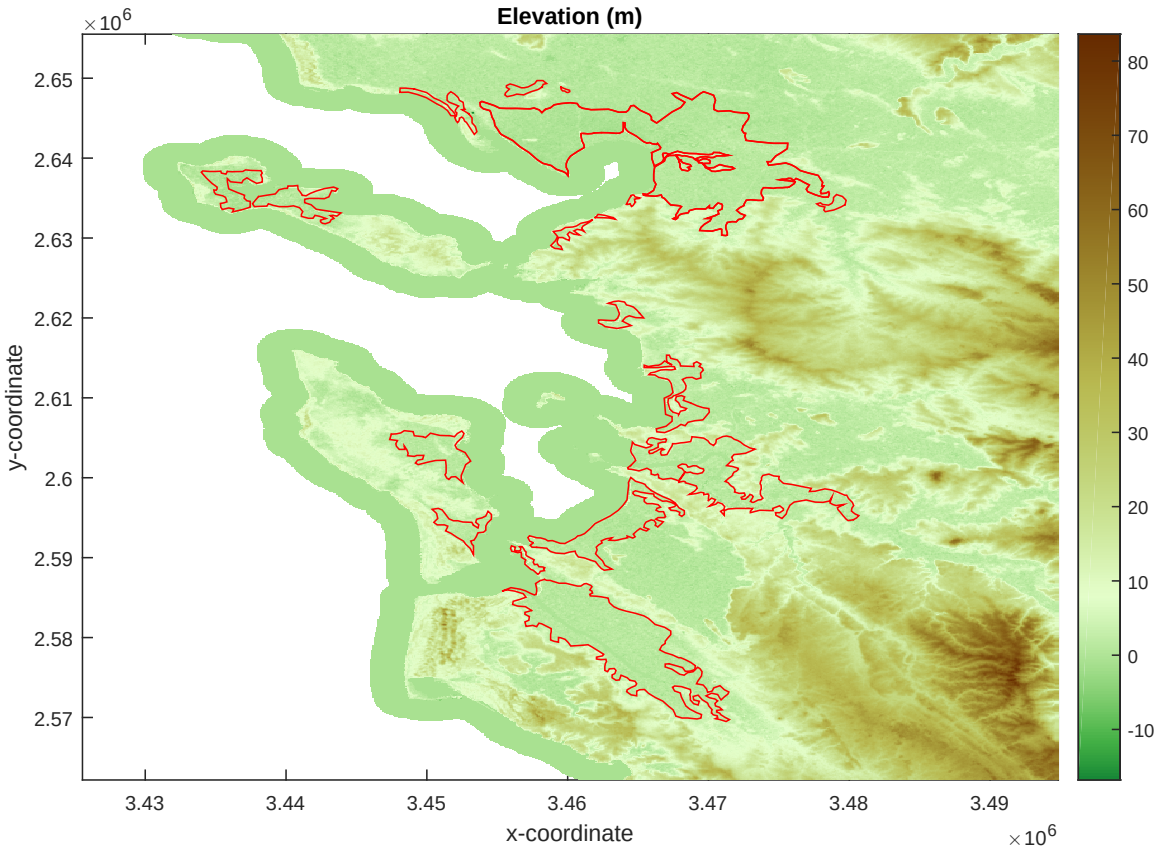


Figure 6.2: 90 m SRTM data set used as input DEM for models. The red lines are the observed marshes.

accept the smoothing of the DEM. It is expected that this smoothing will result in a larger flooded area. The effect of this smoothing should be investigated further.

6.1.7 Key differences of implementation of the input data between LFP and DFM

In spite of the effort to equal both models as much as possible, there are some differences. This section is an effort to list those differences as accurately as possible.
In LFP, the boundary condition is applied in a ghost cell next to the boundary cells, whereas in DFM the boundary condition is applied as a polyline bordering the cells.
Looking closely at the location around $(x, y) = (3461500, 2635800)$, you can see there are a few cells that are represented in DFM which have been masked in LFP. This is due to the way the grid has been made, and is considered not to have a significant effect on the outcome of the simulations. This difference in grid also extends to how the boundary is implemented, as can be seen in Figure 6.6.

6.2 Projection used

In this simulations, the projection used is the European Terrestrial Reference System 1989 (ETRS 1989). The study area has the lower left corner located at $(x, y) = (3425542, 2562178)$. The study area consists of cells with dimension 90 x 90 m, and has 771 and 1038 cells in respectively the x- and y-direction. The corners of the area can be converted to WSG84 coordinates, and the result is seen in Table 6.1.

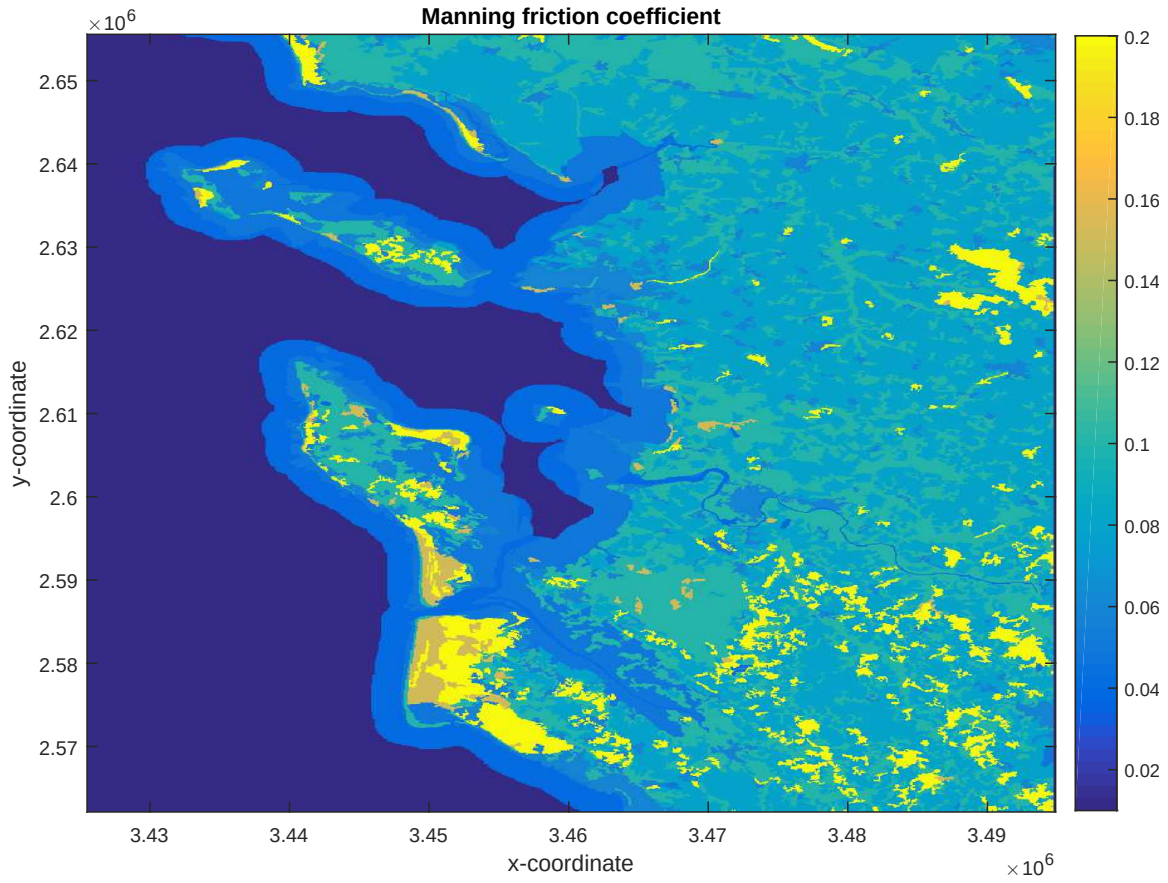


Figure 6.3: Manning friction coefficient map used for the 90 m SRTM simulations.

Corner	x-coord	y-coord	lat	long
lower left	3425542	2562178	45.5482	-1.5021
lower right	3494932	2562178	45.6408	-0.6217
upper right	3494932	2655598	46.4724	-0.7878
upper left	3425542	2655598	46.3781	-1.6815

Table 6.1: Corner coordinates projected to WSG84.

6.3 Land definition

In order to compare flood maps, it is necessary to have a definition of what is supposed to be dry. An area is only considered flooded if it has been wet during an event, but was marked as an area that should remain dry. The areas marked in this way are defined as land, so land is defined in this research as an area that is dry under normal circumstances. The definition is taken from JRC, and is used only to compare methods employed by me to methods employed by JRC. It is unknown to the author how this data is generated. The area defined as land can be found in Figure 6.8. As you can see, the observed flooded areas are also plotted in the same figure. Using this definition of land results in a problem, due to the fact that the marshes only cover the land (as defined in (Breilh et al., 2013)), and there is a different definition of land. This results in areas that will be considered as a false hit, while in reality it is not. This is another reason why the observations are only considered as indicative.

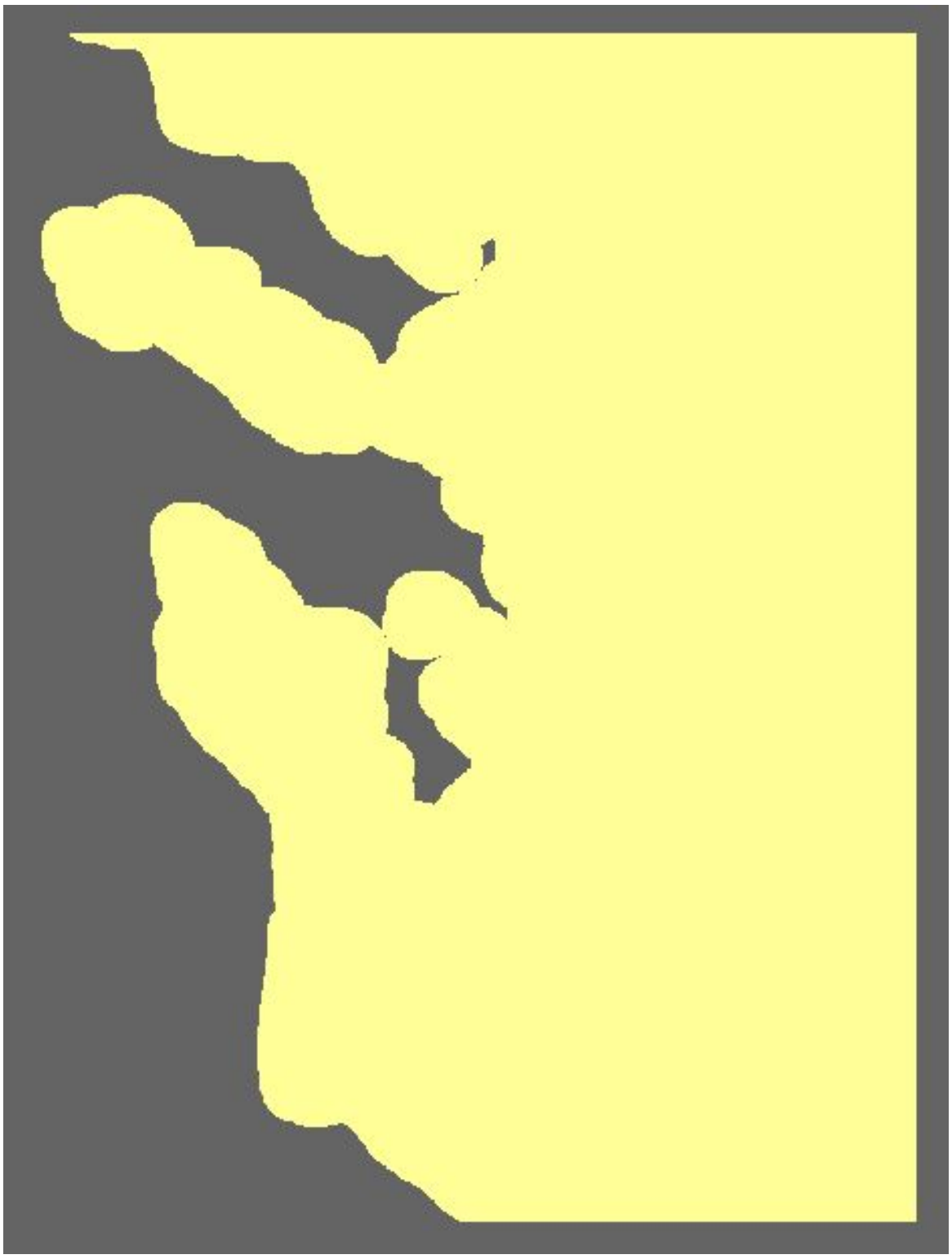


Figure 6.4: The domain used in DFM

6.4 Observed flooded areas (Marshes)

In Figure 6.2, the observed marshes are plotted over the DEM. The marshes are acquired based on (Breilh et al., 2013), by manually copying the data shown in the paper. by comparing Figure 6.8 with Figure 1 from (Breilh et al., 2013), it can be seen that they used a different definition of land (the black line, I assumed).

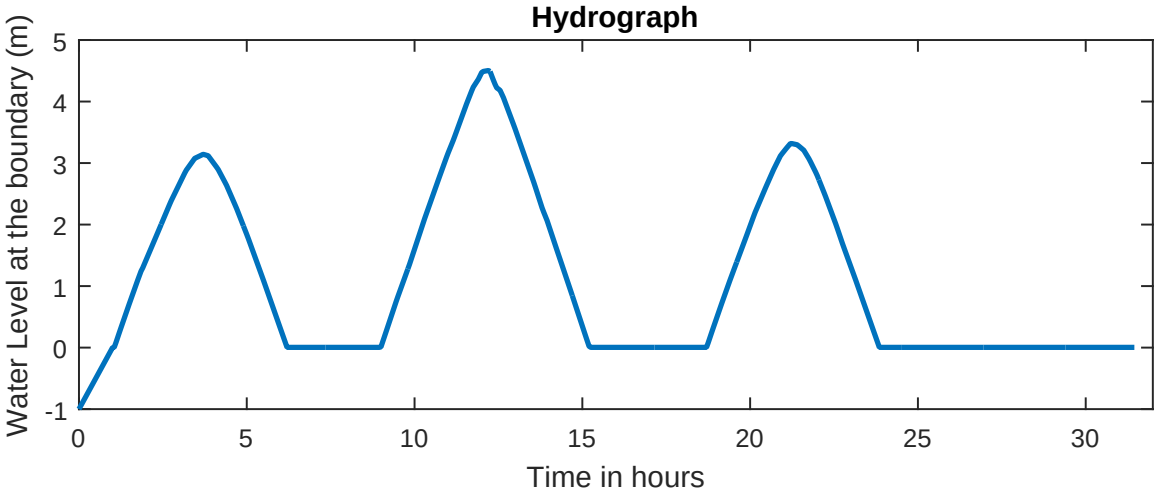


Figure 6.5: Hydrograph applied to the boundary of the models

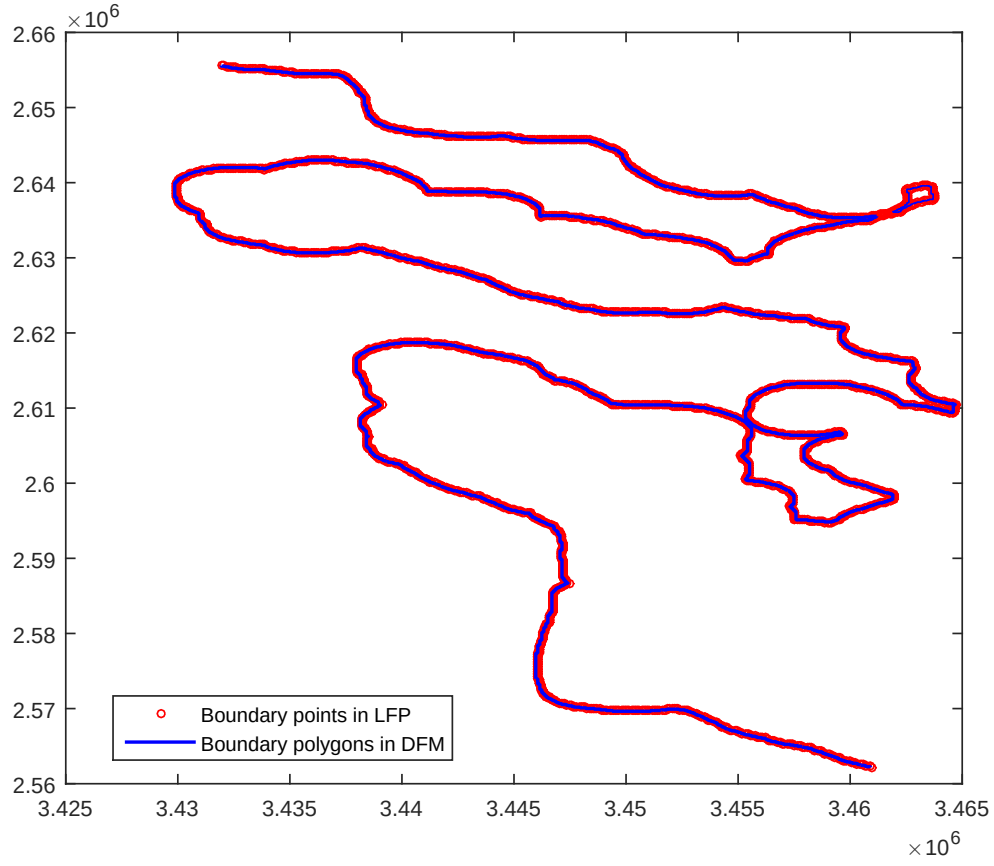


Figure 6.6: Location of the boundary in the LFP and DFM models.

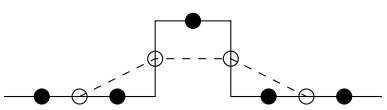


Figure 6.7: Smoothing occurring when changing to bed level type = 3

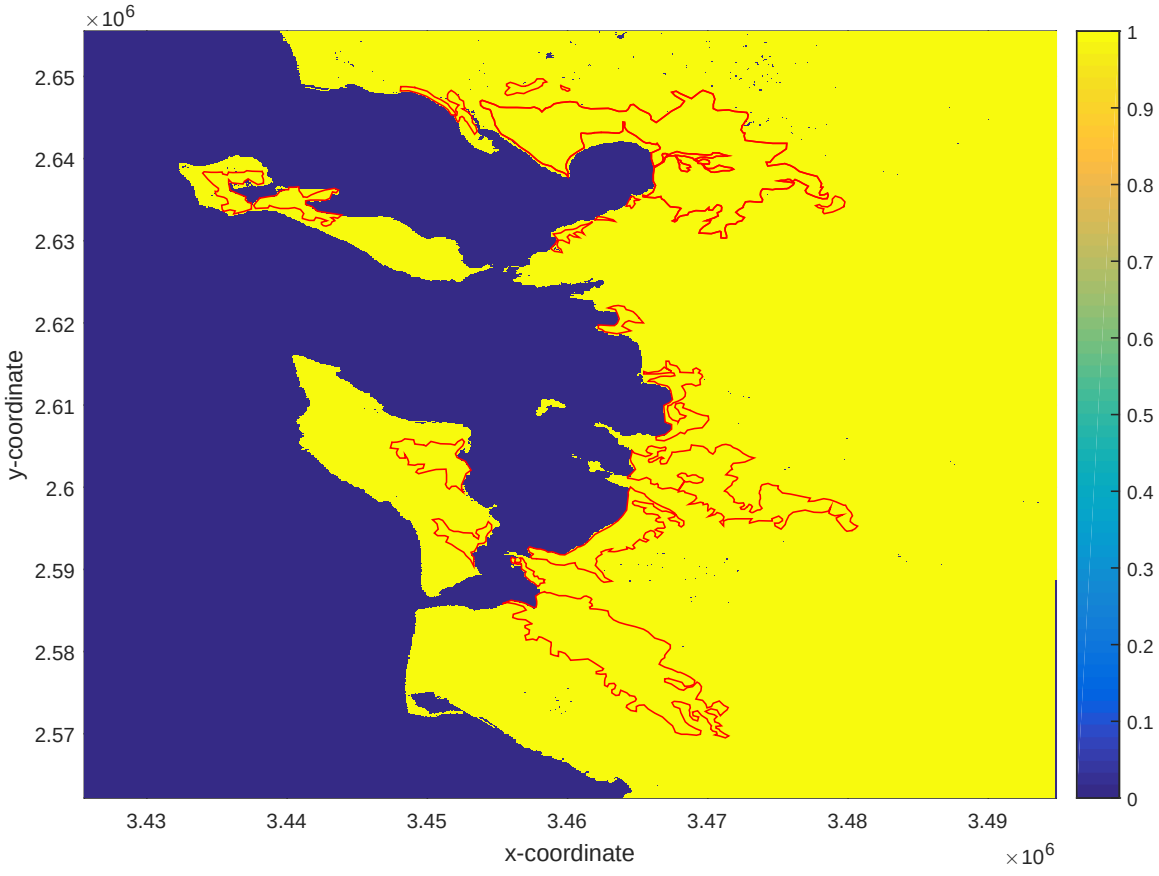


Figure 6.8: Land as defined by JRC. Yellow is land and blue is sea. The red areas are the observed flooded marshes.

6.5 Implementation in LFP

The main input file is shown in Table 6.2. The DEM has been implemented as Table 6.3. The first lines of the Hydrograph can be seen in Table 6.4 and the location is defined as in Table 6.5.

DEMfile	dem.asc
resroot	res
dirroot	results
sim_time	113220
initial_tstep	5
massint	113220
saveint	113220
manningfile	Manning.asc
bcifile	lfp.bci
bdyfile	lfp.bdy
drycheckoff	
hazard	
#Roe	
acceleration	

Table 6.2: Main input file LFP models

ncols	771			
nrows	1038			
xllcorner	3425542			
yllcorner	2562178			
cellsize	90			
NODATA_value	-9999			
-9999	-9999	-9999	-9999	-9999
-9999	-9999	-9999	-9999	-9999
-9999	-9999	-9999	-9999	-9999
-9999	-9999	-9999	-9999	-9999

Table 6.3: Input file of the DEM for LFP

# inflow hydrograph	
lfp	
630	hours
-1	0
-0.95	0.05
-0.9	0.1
-0.85	0.15
-0.8	0.2
-0.75	0.25

Table 6.4: Input file of the Hydrograph for LFP

6.6 Implementation in DFM

The main file in Delft3D FM is too long to show here. The important settings are discussed in Chapter 7. The main file redirects to other files, which are shown here. The file shown in Table 6.7 is the external force file, which defines what quantities are taken into account and where the input files for those can be found. As can be seen, there are two water level bounds and a friction coefficient file. The boundary files consists of two files, a polyline that represents the location and a time varying value file. This is similar to the two files used for the boundary in LFP. The first few lines of the files can be seen in Tables 6.8 and 6.9. This is not how the DEM is normally implemented. Normally the

P	3429907	2639983	HVAR	lfp
P	3429907	2639893	HVAR	lfp
P	3429907	2639803	HVAR	lfp
P	3429907	2639713	HVAR	lfp
P	3429907	2639623	HVAR	lfp

Table 6.5: Input file for the location of the boundary in LFP

ncols	771			
nrows	1038			
xllcorner	3425542			
yllcorner	2562178			
cellsize	90			
NODATA_value	-9999			
0.01	0.01	0.01	0.01	0.01
0.01	0.01	0.01	0.01	0.01
0.01	0.01	0.01	0.01	0.01
0.01	0.01	0.01	0.01	0.01

Table 6.6: Input file of the Manning friction coefficient for LFP

values are stored at the net nodes, however the bed level is not defined at the net nodes when using bed level type = 1. The bed level is implemented by supplying a separate file containing list of x and y coordinates and bed level values. The first lines of this file can be seen in Table 6.10. The Manning friction coefficient is a similar file, the first few lines are shown in Table 6.11.

QUANTITY	=	waterlevelbnd
FILENAME	=	bdrcond1.pli
FILETYPE	=	9
METHOD	=	3
OPERAND	=	O
QUANTITY	=	waterlevelbnd
FILENAME	=	bdrcond2.pli
FILETYPE	=	9
METHOD	=	3
OPERAND	=	O
QUANTITY	=	frictioncoefficient
FILENAME	=	manning.xyz
FILETYPE	=	7
METHOD	=	6
OPERAND	=	O

Table 6.7: External force file for DFM

0	-1.00000000
3	-0.95000000
6	-0.90000000
9	-0.85000000
12	-0.80000000
15	-0.75000000
18	-0.70000000
21	-0.65000000

Table 6.8: Input file of the Hydrograph for DFM

L1	
2212 2	
3431932	2655598
3431932	2655508
3432112	2655508
3432112	2655418
3432292	2655418

Table 6.9: Input file for the location of the boundary in DFM

3425587	2562223	-1.00
3425587	2562313	-1.00
3425587	2562403	-1.00
3425587	2562493	-1.00
3425587	2562583	-1.00
3425587	2562673	-1.00

Table 6.10: Input file of the DEM for DFM

3425587	2562223	0.01
3425587	2562313	0.01
3425587	2562403	0.01
3425587	2562493	0.01
3425587	2562583	0.01
3425587	2562673	0.01

Table 6.11: Input file of the Manning friction coefficient for DFM

Chapter 7

Results

This chapter will report on the obtained results. For all simulations, a HP Elitebook from 2015 is used. The software versions are 1.1.191.47915 for Delft3D FM and 5.8.9 for LISFLOOD-FP. First the two software packages are run without advection and compared to each other in terms of flooded extent and computational time. Then the same is compared for the model runs with advection. After this, different model setups for LFP are compared to each other, and different model runs for DFM are compared to each other. The notation for the different models is as follows: The name of a model starts either with LFP, DFM or DFMbl. LFP stands for a model run with LISFLOOD-FP, DFM stands for a model run with Delft3D FM that used bed level type = 3, and DFMbl stands for a model run with Delft3D FM with bed level type = 1. Then, the time step is indicated in seconds. After this, it is indicated whether or not the momentum advection term is included, by the terms A (advection included) and NoA (no advection included). At the end, the CFL number multiplied by ten is given. As an example, "LFP5A7" means: The model is a LISFLOOD-FP model with a maximum time step of 5 seconds, advection is included in the model (so the Roe's method is used), and the CFL number is 0.7. Extensions to the names indicate extra settings. It will always be explained what those extra settings are. In this chapter, a cell is called inundated or flooded if at some point during the simulation the water level was 10 cm or higher.

7.1 LFP and DFMbl without advection

First, a LFP and DFMbl model are run. Both models are run without advection, use a time step of 5 seconds and have no diffusive or turbulence model. We start by describing their setups and reporting the results.

7.1.1 LFP5NoA7 model

In order to give an impression of the results obtained by the LFP models, a detailed account of the outcome of the LFP5NoA7 model is given here. This model uses similar settings to the settings used at JRC for large scale flood inundation simulation and used in (Vousdoukas et al., 2016). The results for maximum water levels as well as maximum water velocitiese are shown in Figure 7.1.

The maximum water level that occurred during simulation LFP5NoA7 can be observed in Figure 7.1a. The maximum is 4.64 m, which is roughly 10 cm larger than the highest water level supplied at the boundary (4.50 m) and therefore seems feasible.

LFP supports the output of the maximum water velocity in the x- and y-direction (Vx and Vy). Those velocities are given at the cell boundaries, since this is where the velocities are calculated for the simulations. Those maximums can be seen in Figure 7.1c and 7.1d. In the figures, the water velocities are shown up to a speed of 1.5 m/s. Values higher than this are truncated to 1.5 m/s for visualization motives. For the interest of speeding up the simulation, those points are definitely important since they are likely to limit an increase in time step. The number of cell boundaries having a maximum speed higher than 1.5 m/s are 280 and 144 for the x- and y-direction, respectively, on a

total of 800298 cells. Furthermore, the maximum velocity is 4.5 and 3.4 m/s, respectively. In Figure 7.1b, the maximum water velocities computed by

$$Vc_{i,j} = \left(\left[\max \left(V_{i-1/2,j}, V_{i+1/2,j} \right) \right]^2 + \left[\max \left(V_{i,j-1/2}, V_{i,j+1/2} \right) \right]^2 \right)^{0.5} \tag{7.1}$$

can be observed. This figure is also truncated at 1.5 m/s. The maximum velocity is 5.3 m/s and the number of cells having a speed over 1.5 m/s is 976.

7.1.2 DFMbl5NoA7 model

A similar setup has been run in Delft3D FM. The water level can be found in 7.2a and velocities in x- and y-direction can be found in Figures 7.2b and 7.2c. The maximum water level that occurred during the simulation is 4.62 m, which is almost the same as in LFP5NoA7 (4.64 m). The maximum velocities at cell the boundaries in x- and y-direction are 7.7 and 7.6 m/s. The Vc value is not shown here because this value is defined differently in DFM. The Vx and Vy values are also truncated in the figures. The number of cells having a speed over 1.5 m/s are 412 and 285, respectively.

7.1.3 LFP5NoA7 and DFMbl5NoA7 models compared

Figure 7.3a shows a comparison of the inundations obtained in LFP5NoA7 and DFMbl5NoA7. The LFP model is more conservative than the DFMbl model. The H1-value of 91.7 means that the inundated area of the second model (DFMbl5NoA7) is flooded for 91.7% in the first model (LFP5NoA7), and the H2-value of 99.8% means that the inundated area of the first model is flooded for 99.8% in the second model. This means that almost everything that is flooded in the LFP model is also flooded in the DFMbl model, but only 91.7% of the area flooded in DFMbl is also flooded in LFP. The C-value is a symmetric measure of the inundated areas. It divides the intersection by the union of the inundated areas, and in this case is 91.5%. This indicates that they compare relatively well in terms of flooded extent. The exact definitions can be found in Section 3.1. The computational time was 24 minutes and 39 seconds for LFP5NoA7 versus 63 minutes and 51 seconds for the DFM5NoA7 model.

7.2 LFP and DFMbl with advection

Now, we report the results of model runs that include advection terms. For LFP, including advection is done by switching to the Roe solver, see Section 4.1. For DFM, this is done by setting the advection type to 33, see Section 4.2 for more details.

7.2.1 LFP5A7 model

Including advection by changing the numerical solver (see Section 4.1) from acceleration to Roe has a huge influence on the outcome of the simulation. Figure 7.4 shows the maximum water levels and water velocities. The flooded extent obtained by both models is compared in Figure 7.3c. It can be observed that including advection leads to a severe increase in flooded extent. However, it is worth noting that the flooded extent obtained with the Roe solver does not look unrealistic, and that only after analyzing other quantities such as water velocities or water depths one realizes that the results are an artifact of an unstable model.
The maximum water velocities obtained during the LFP5A7 simulation are 13421, 21613 and 21613 m/s for Vx, Vy, and Vc. Those values seem very unlikely. To further investigate the model's validity, some model verifications are conducted. Starting with an analyzes of the expected maximum time step used by the program. LFP with Roe's method uses the time step limiter described in Equation 4.8. The smallest time step used by the model during this simulation is 4.49 s. Using the limiter and picking the maximum velocity obtained (and setting $h = 0$, since setting a higher water level will result in an even smaller Δt_{max}), results in the upper limit for Δt_{max}

$$\Delta t_{max} = \alpha \frac{\Delta x}{|v| + \sqrt{g h_t}} = 0.7 \frac{90m}{21613m/s + \sqrt{(9.81 * 0m^2/s^2)}} = 0.0042s. \tag{7.2}$$

This means that the smallest time step used during the simulation should be 0.0042 s or smaller. However in our simulation it is 4.49 s. Furthermore, along the shoreline, many locations have a water level of over 12 m. Note that the maximum water level supplied at the boundary is about 4.5 meters. So even if a wave fully reflected at the boundary and did not show any diffusive behavior, the maximum water level should only be 10 m ($(4.5+1)*2-1=10$), where 1 is added because the bed level is -1 m on the shelf. This points in the direction of either an input error in the model, or in a bug in the LFP software package. Since the reason that this instabilities occur is unknown to us, the Roe method will not be used in the remaining part of the study. It should be noted that changing the "depththresh" setting from the default of 0.001 to 0.1 has a large influence on the maximum velocity. When the depththresh is set to 0.1, the maximum velocities are in the order of 200 m/s. Such velocities are still not feasible, but more reasonable than velocities in the order of ten thousands.

7.2.2 DFMbl5A7 model and comparison with DFMbl5NoA7

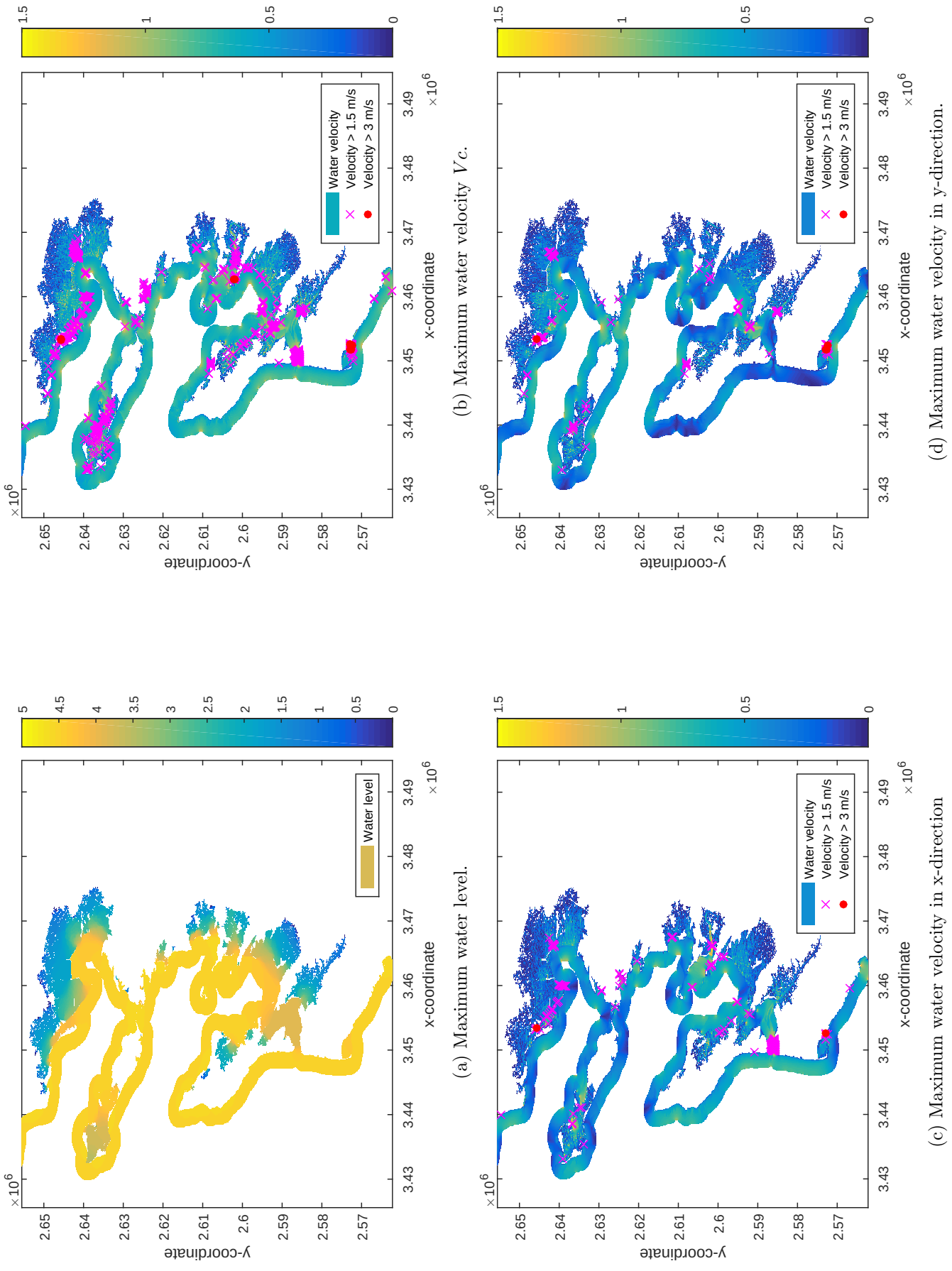
A DFM model is also run with the advection momentum term included, this is done by setting the advection type to 33 in the MDU file, see Section 4.2 for more details. The water levels and velocities for this model can be found in Figure 7.5. During this simulation, the maximum water level is 4.62 m and the maximum velocities in the x- and y-direction are 4.1 and 4.4 m/s. The comparison of the inundated area simulated by the DFMbl5A7 and DFMbl5NoA7 model can be seen in Figure 7.3d. It is interesting to see that the effect of including advection in the simulation on the inundated area is smaller than the effect of the chosen software package. The DFMbl models with and without advection show a C-value of 98.5 %, which shows that the results are almost identical. The computational times for the two models are also identical, being 64 and 63 minutes for the model without and with advection. The reason for the similar computational times is that in both simulations the maximum allowed time step of 5 seconds is used for all time steps and not limited by the CFL number. In later sections, the maximum allowed time step will be increased and the CFL condition for the model without advection will be relaxed (which is possible in the case without advection, since the solver is completely implicit).

7.3 Comparison between different LFP models

Different model setups have also been run for LFP. Since the software package is explicit for all solver types, the CFL has to remain smaller than one. This means that the parameter varied is only the time step. Table 7.1 shows the computational time, C-value with LFP5NoA7 and velocities in x- and y-direction. In Figure 7.6 a representative flood inundation comparison can be found.

Model name	runtime (minutes)	nr of time steps	average ts (seconds)	C with LFP5NoA7	max Vx (m/s)	max Vy (m/s)
No advection						
LFP1NoA7	100.07	113220	1	98.53	4.5	3.4
LFP3NoA7	31.23	37740	3	99.30	4.5	3.4
LFP5NoA7	24.65	23697	4.8		4.5	3.4
LFP10NoA7	18.62	22640	5.0	99.79	4.5	6.3
LFP20NoA7	20.95	22398	5.1	99.78	4.5	7.0
LFP40NoA7	19.60	22364	5.1	99.78	4.5	6.8
Advection						
LFP5A7	43.28	23049	4.9	73.08	13421*	21613*
LFP10A7	39.90	21681	5.2	73.02	13415*	21603*
LFP20A7	38.37	21517	5.3	73.03	13416*	21606*
LFP40A7	36.02	21507	5.3	73.03	13420*	21608*

Table 7.1: LFP results



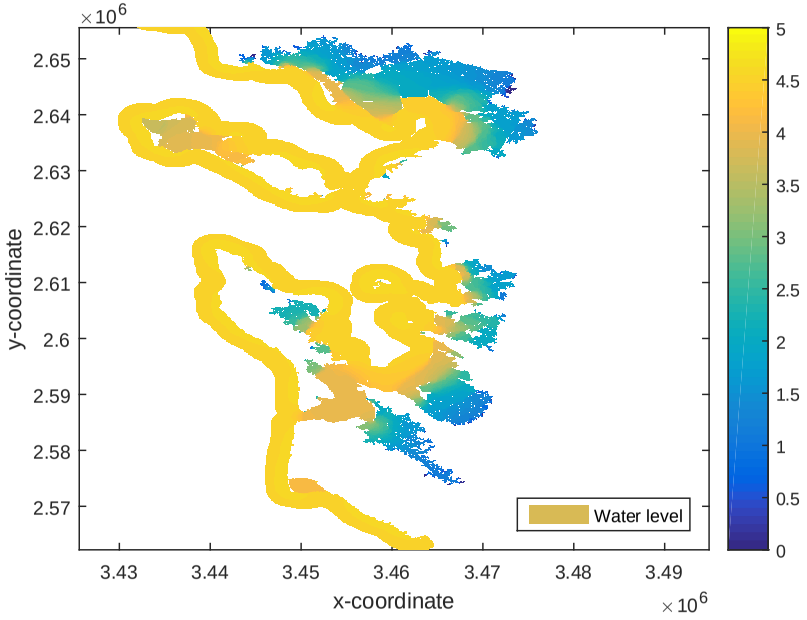
(a) Maximum water level.

(b) Maximum water velocity Vc.

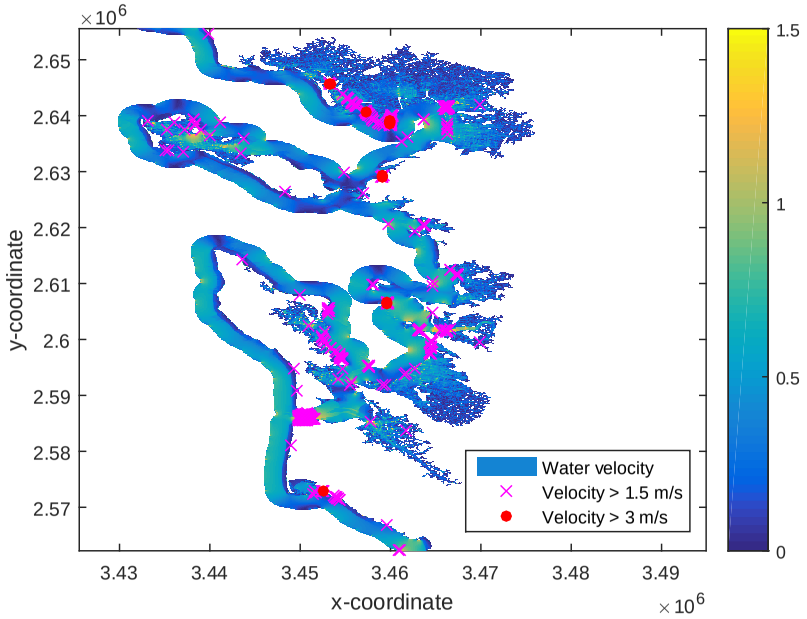
(c) Maximum water velocity in x-direction

(d) Maximum water velocity in y-direction.

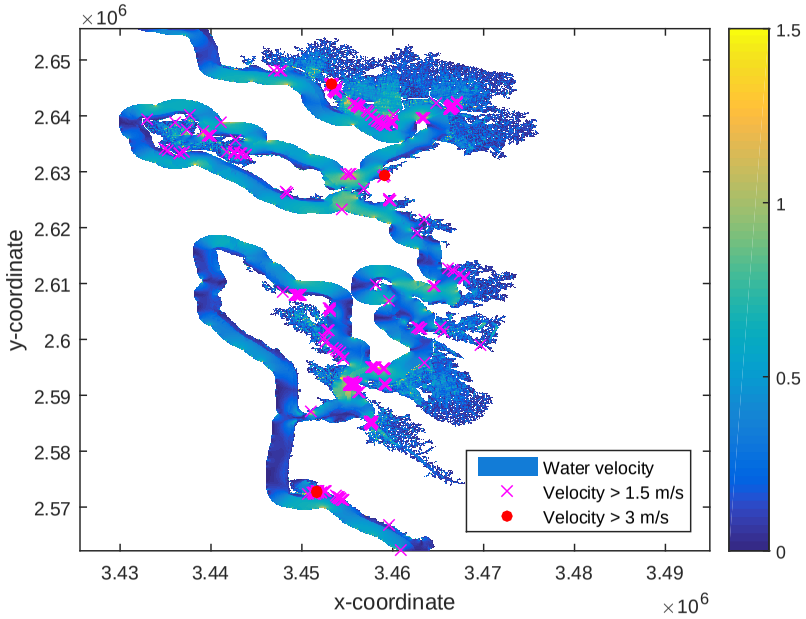
Figure 7.1: Maximum water level and maximum water velocities for LFP5NoA7 model.



(a) Maximum water level (m) for DFMbl5NoA7.

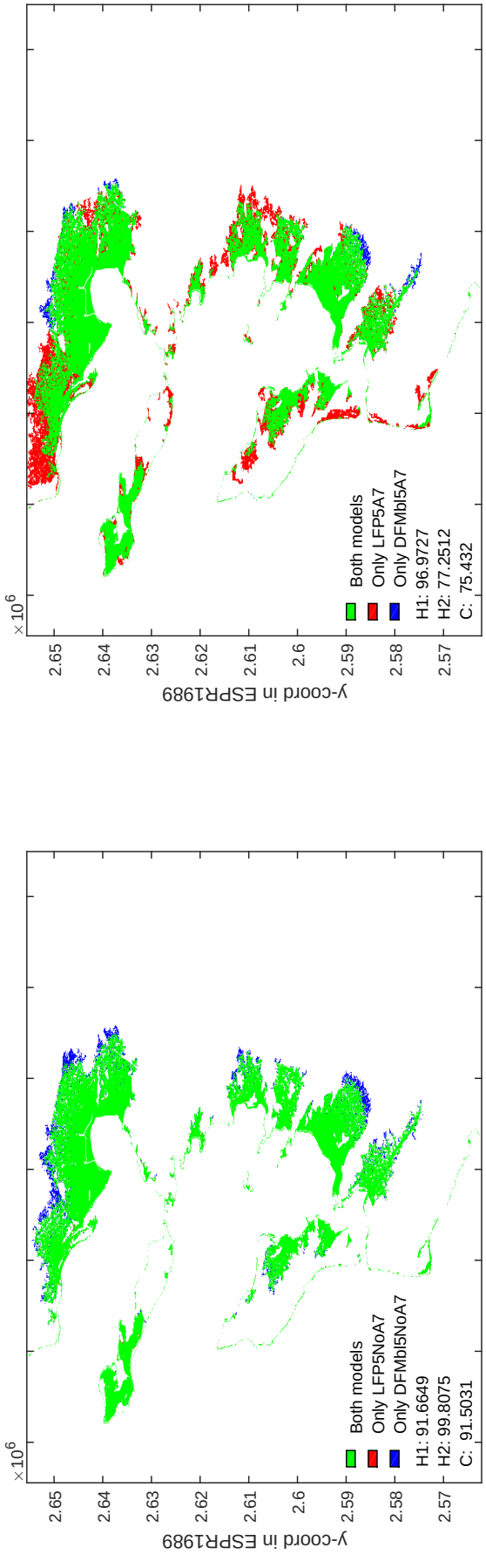


(b) Maximum water velocity in x-direction for DFMbl5NoA7



(c) Maximum water velocity in x-direction for DFMbl5NoA7

Figure 7.2: Maximum water level and velocity for DFMbl5NoA7 model.



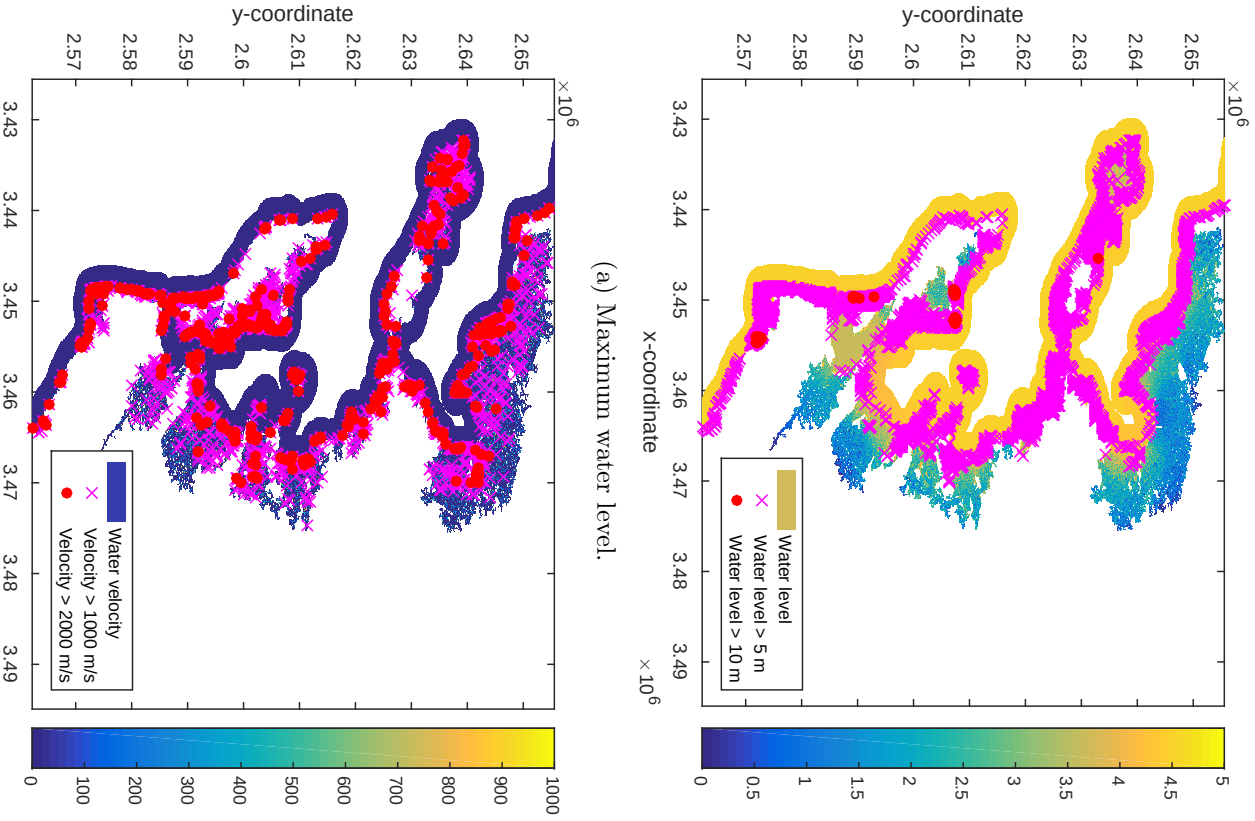
(a) LFP5NoA7 and DFMbl5NoA7 models compared, i.e., both the LFP and DFMbl model without advection.

(b) LFP5A7 and DFMbl5A7 models compared, i.e., both the LFP and DFMbl model with advection.

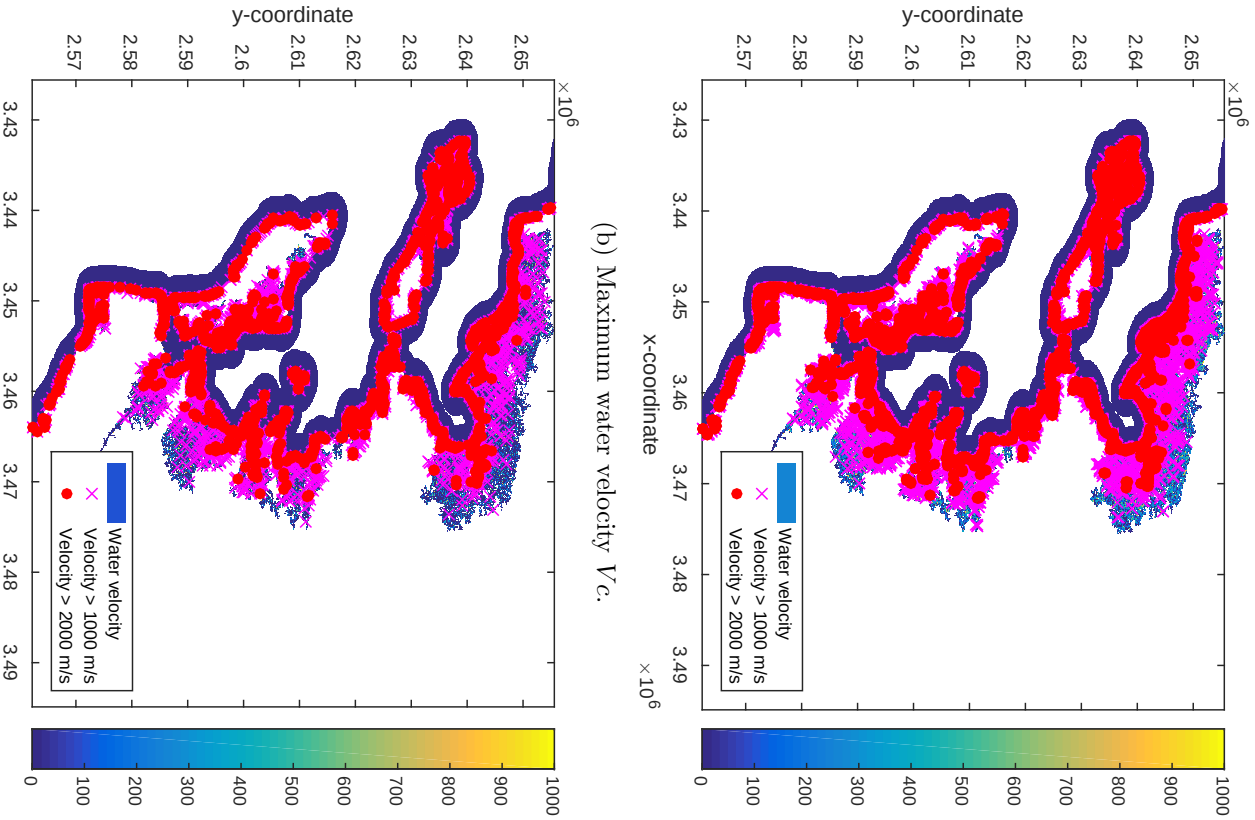
(c) LFP5NoA7 and LFP5A7 models compared. Shows the influence of advection in the LFP models.

(d) DFMbl5NoA7 and DFMbl5A7 models compared. Shows the influence of advection in the DFMbl models.

Figure 7.3: Inundation results compared for the four base simulations LFP5NoA7, DFMbl5NoA7, LFP5A7 and DFMbl5A7.



(a) Maximum water level.

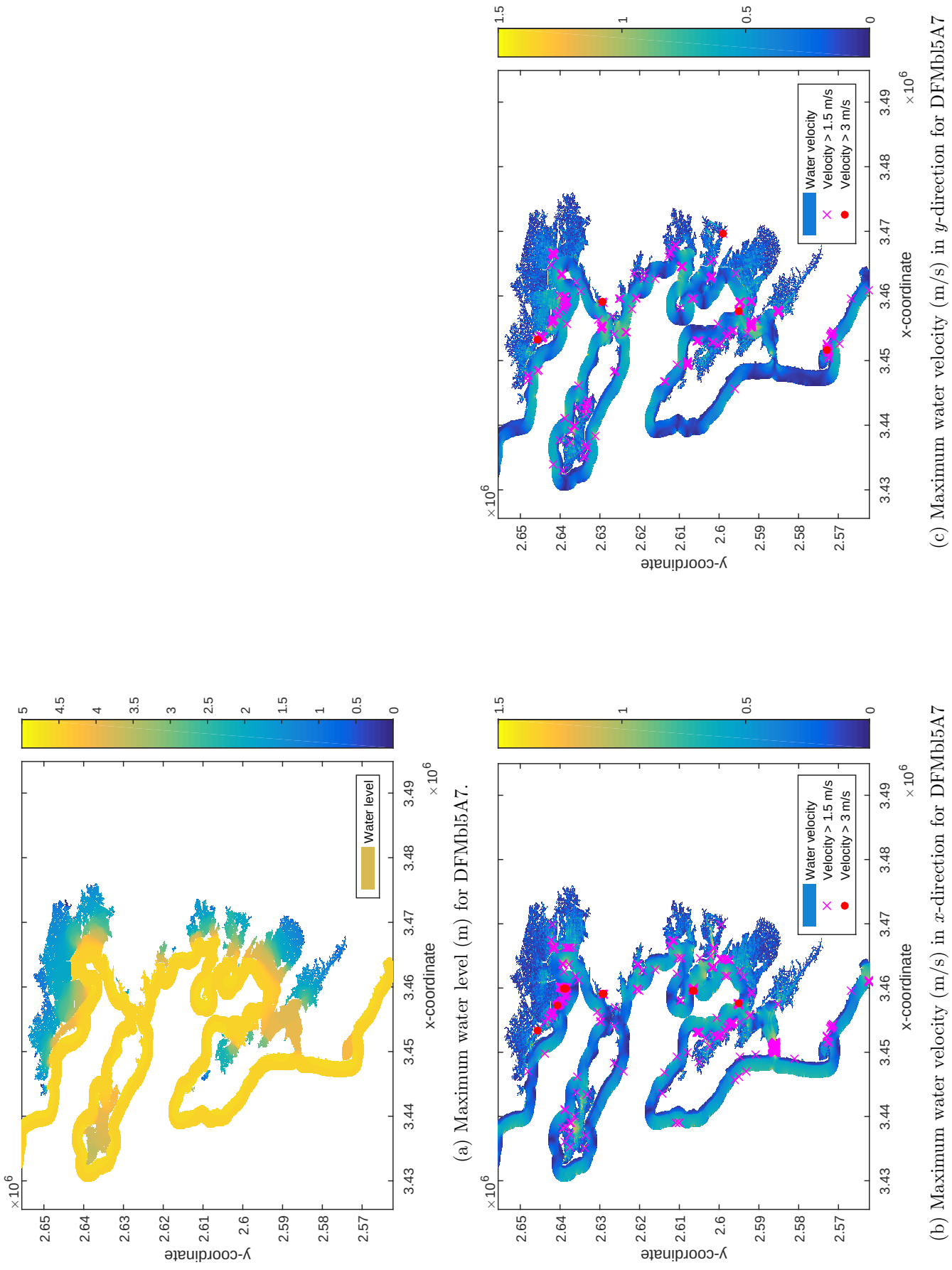


(b) Maximum water velocity Vc.

(c) Maximum water velocity in x-direction

(d) Maximum water velocity in y-direction.

Figure 7.4: Maximum water level and maximum water velocities for LFP5A7 model.



(a) Maximum water level (m) for DFMbl5A7.

(b) Maximum water velocity (m/s) in x-direction for DFMbl5A7

(c) Maximum water velocity (m/s) in y-direction for DFMbl5A7

Figure 7.5: Maximum water level and velocity for DFMbl5A7 model.

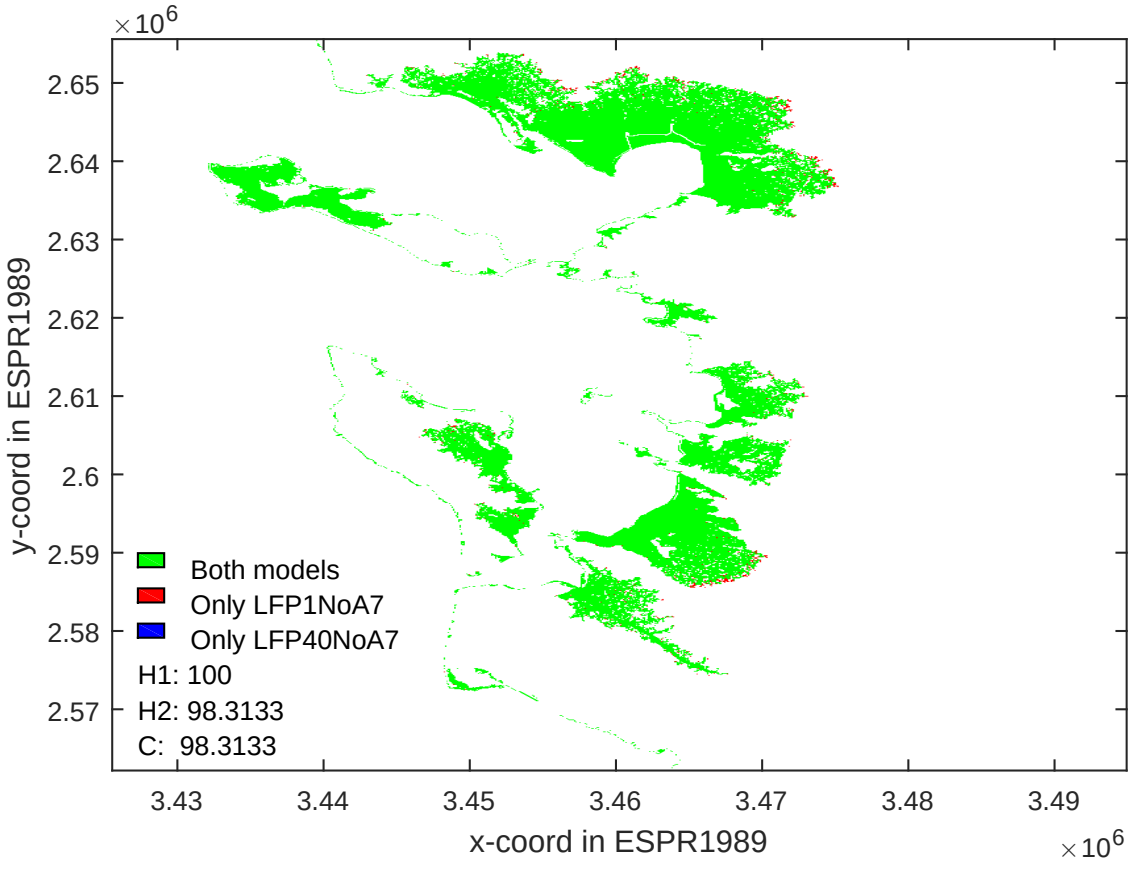


Figure 7.6: Flooded extent simulated by the LFP1NoA7 and LFP40NoA7 models compared. As can be seen, the simulations results in very similar flooded extents.

7.4 Comparison between different DFMbl models

In order to investigate the possibilities for speed up within the DFMbl models, some model parameters are varied. The two key parameters that are varied are the maximum allowed time step and the CFL number. Note that a necessary condition for stability with the DFM advection models is that the CFL number should be smaller than unity. For the models without advection, this restriction is not necessary. For those models the limit on the CFL number is imposed by the speed of the matrix solver compared to the size of the time step. First, the models without advection are discussed, and later the models with advection.

The time step is increased from 5 to 180 s, with intermediate steps. Furthermore, the CFL condition is relaxed from 0.7 to 2.8. The results obtained for some of the simulations are shown in Figure 7.7. Table 7.2 shows the computational time, C-value with DFMbl5NoA7 and velocities in the x- and y-direction. It can be seen that the difference in inundated area between the base simulation DFMbl5NoA7 and the others is not significant. Looking at the values in this table, it seems that DFMbl60NoA7 and DFMbl80NoA7 are the same, and DFMbl60NoA14 and DFMbl80NoA14 are also the same. After verification of output velocities, this hypothesis is verified. This indicates that all time steps are 60 or smaller, and thus the CFL number is always the limiting factor. DFMbl5NoA7 and DFMbl5NoA28 are also the same, but those simulations are limited by the maximum allowed time step instead of the CFL number. The C-value, which is a good indicator for comparability of the inundated area, is 98 or higher (out of 100) for all of the simulations shown in the Table, and 99 for all the models without advection. This means that changing the CFL number and time step does not have a significant influence on the inundated area. And also that in or excluding the advection term results in similar inundated ares. Since the computational time is greatly reduced for some of the models compared to the DFMbl5NoA7 model, it is favorable to increase the CFL number and time step in the current simulations. So far the computational time has been reduced from 64 to 11 minutes. It is interesting to see that even if the inundated area remains rather similar, the same can not be said for the maximum velocities. Two advection simulations with a CFL number over 1 are also run (DFMbl80A14 and DFMbl80A28). In contrast with the expectations of the author, the simulations are stable and give feasible results. It is interesting to see that those simulations show very similar results with the other advection models, both in terms of inundated area and maximum velocity.

7.4.1 Most limiting cells for DFMbl80NoA28

There are a few cells that limit the time step frequently. As can be seen in Figure 7.8, the simulation required a total time steps of 2364, and 1733 of those time steps were limited by the CFL condition. 45% of the limited time steps are caused by just five cells. It could be interesting to investigate the possibilities to locally change the bathymetry of those cells and check the effect on the flooded extent and the simulation time required. It is possible that changing the bathymetry locally has a positive influence on computational time, without changing the flooded outcome of the simulations too much.

7.4.2 DFM with Bed level type = 3

The bed level type has been changed to 3. This is done since this setting is the default value for DFM, and the effect of changing this parameter should be stated. As expected, using bed level type = 3 in stead of type = 1 gives a different result (see Section 6.1.6). The comparison of flooded extent with DFMbl90NoA70 can be found in Figure 7.9. It can be observed that the flooded extent changes significantly when changing the bottom level type. The model with the default bottom level is less conservative. This could be explained by the smoothing shown in Figure 6.7. In order to assess the actual effect of the bottom type, different simulations could be run. The simulation time for the DFM model with the default bed level type is shorter than with the cell centered bed level type of the DFMbl models, being respectively 8 and 10 minutes. This can be seen in Table 7.2.

Model name	runtime* (minutes)	nr of time steps	average ts (seconds)	fraction solve/steps	C with DFMbl5NoA7	max Vx (m/s)	max Vy (m/s)
No advection, see Section 7.1.2							
DFMbl5NoA7[1]	63.84	22652	5.0	0.31		7.7	7.6
DFMbl5NoA28[1]	61.40	22652	5.0	0.31	100.00	7.7	7.6
DFMbl10NoA7	32.88	11499	9.8	0.35	99.96	4.2	4.6
DFMbl10NoA14	32.80	11341	10.0	0.34	99.95	4.2	4.6
DFMbl10NoA28	32.75	11337	10.0	0.35	99.95	4.2	4.6
DFMbl20NoA7	23.80	8039	14.1	0.37	99.92	4.2	4.9
DFMbl20NoA14	18.53	6024	18.8	0.38	99.87	4.3	4.9
DFMbl20NoA28	18.18	5689	19.9	0.41	99.86	5.7	4.9
DFMbl40NoA7	22.65	7350	14.4	0.38	99.92	4.2	5.0
DFMbl40NoA14	14.81	4123	27.5	0.45	99.81	6.4	6.4
DFMbl40NoA28	14.31	3251	34.8	0.53	99.73	4.6	6.3
DFMbl60NoA7[2]	22.55	7317	15.5	0.38	99.91	4.2	5.0
DFMbl60NoA14[3]	14.74	4016	28.2	0.45	99.81	6.2	4.6
DFMbl60NoA28	11.77	2393	47.3	0.56	99.60	8.3	4.6
DFMbl80NoA7[2]	22.58	7317	15.5	0.38	99.91	4.2	5.0
DFMbl80NoA14[3]	14.53	4016	28.2	0.45	99.81	6.2	4.6
DFMbl80NoA28	11.22	2364	47.9	0.56	99.62	4.2	4.6
DFMbl80NoA35	11.14	2187	51.8	0.56	99.54	4.4	4.5
DFMbl90NoA35	10.99	2003	56.5	2003	99.47	4.4	4.5
DFMbl90NoA70	**10.00**	1640	69.0	0.60	99.27	4.7	4.8
Advection, see Section 7.2.2							
DFMbl5A7	63.32	22652	5.0	0.31	98.51	4.1	4.5
DFMbl10A7	34.46	11457	9.9	0.34	98.48	4.1	4.5
DFMbl20A7	23.70	7512	15.1	0.37	98.45	4.1	4.5
DFMbl40A7	22.79	7321	15.5	0.37	98.45	4.1	4.5
DFMbl60A7	23.14	7295	15.5	0.37	98.45	4.1	4.5
DFMbl80A7	24.85	7295	15.5	0.38	98.45	4.1	4.5
DFMbl80A14	15.18	4012	28.2	0.44	98.35	4.9	4.5
DFMbl80A28	**11.32**	2342	48.3	0.55	98.15	4.4	5.2
Default Bed level type and no advection, see Section 7.4.2							
DFM90NoA70	8.47	1421	79.7	0.63	80.71	7.0	8.3
DFM180NoA70	7.02	1115	101.5	0.63	80.92	4.4	14.2
Default settings for DFM, see Section 7.5.							
DFMbl5A7default	83.75	22652	5.0	0.24	98.76	4.1	4.5

Table 7.2: Performance indicators for different model setups. Exponents at the end of the model name indicates which model setups result in exactly the same run. This indicates that either the CFL or the maximum allowed time step is always limiting the time step. *: Runtime is defined as the value of "time steps + plots" in the .dia file.

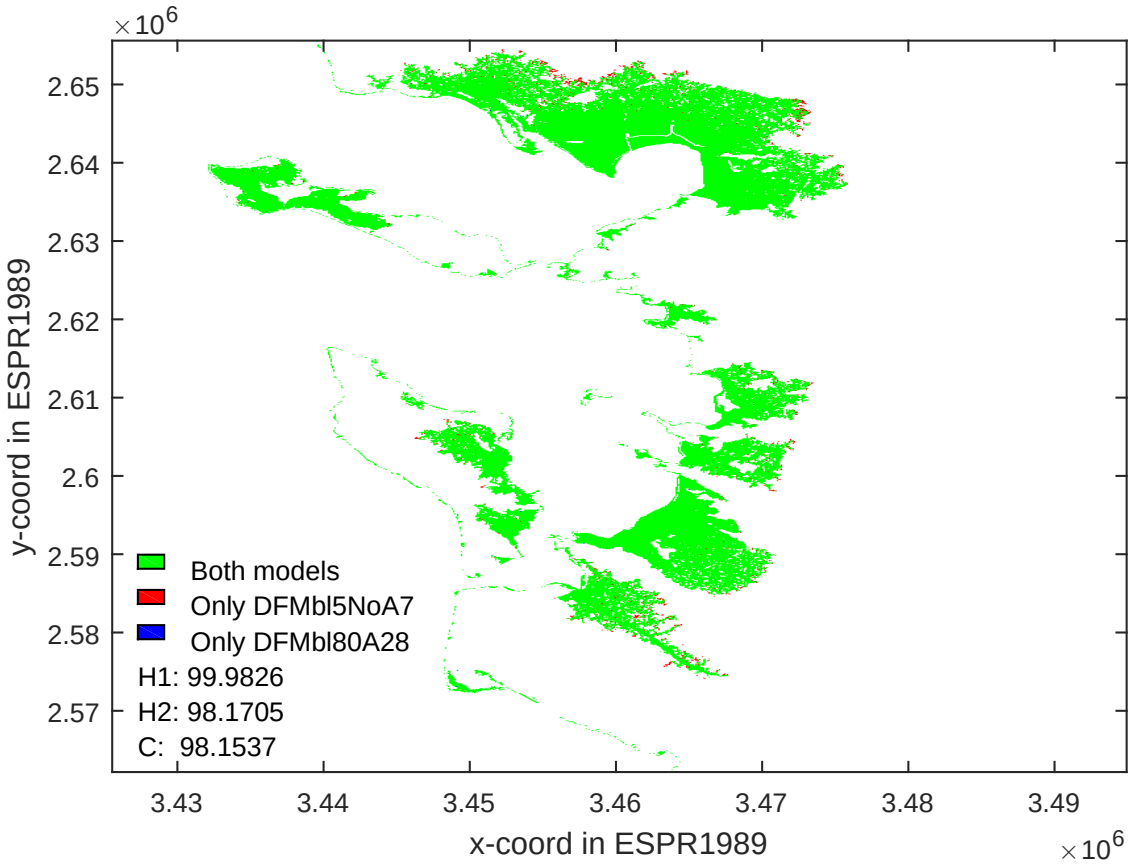


Figure 7.7: Flooded extent simulated by the DFMbl5NoA7 and DFMbl80A28 models compared. As can be seen, the simulations results in very similar flooded extents.

7.5 DFM with default settings, including turbulence etc.

A default Delft3D FM model has been run. In order to do this, a default parameter file (mdu file) is created. The following settings have been adapted in order to create a comparable simulation

1. Software version used: 1.1.145.40991, Jul 17 2015[1]
2. WaterLevIni = -20
3. BedlevType = 1
4. DtUser = 180
5. Dtmax = 5
6. TStop = 113220
7. Epshu = 1.d-3
8. Output settings have been changed.
9. Conveyance2D = 0.

The difference between flood maps simulated by DFMbl5A7 and DFMbl5A7default can be seen in Figure 7.10. As can be seen, the resulting inundated extent is similar to the model where advection and turbulence modeling are switched off. The computational time required is 84 minutes, which is 21 minutes more than what was required for DFMbl5A7. Furthermore, the maximum velocities obtained

[1]The version used for this thesis produces an error.

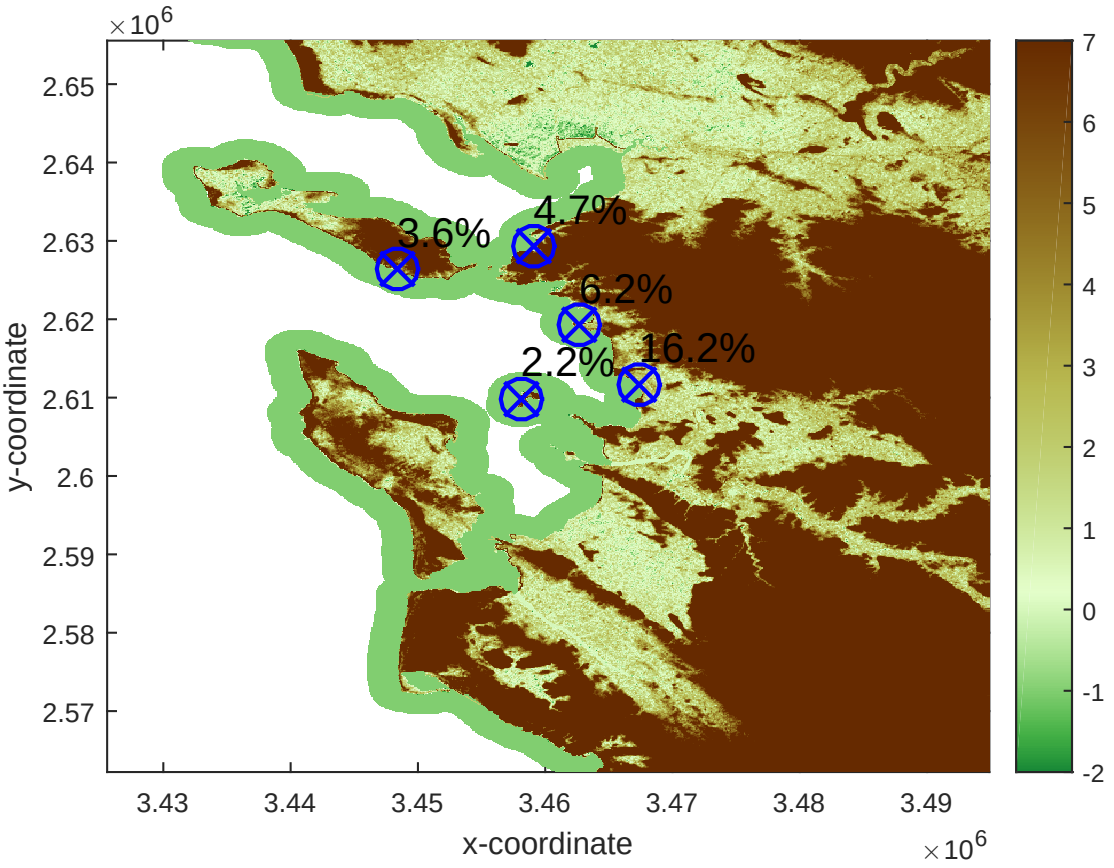


Figure 7.8: Five most limiting cells for DFMbl80NoA28 model. The values show how often the cell is time step limiting over the complete simulation. Total number of time steps: 2364. Total number of time steps limited by the CFL number: 1733. The five cells shown limit 778 time steps, and are responsible for 45% of the limited time steps.

are 4.1 and 4.5 for the x and y direction, which is similar to the DFMbl5A7 as well. In conclusion, the flooded extent for both models is very comparable, and the DFMbl5A7 model is less computationally demanding.

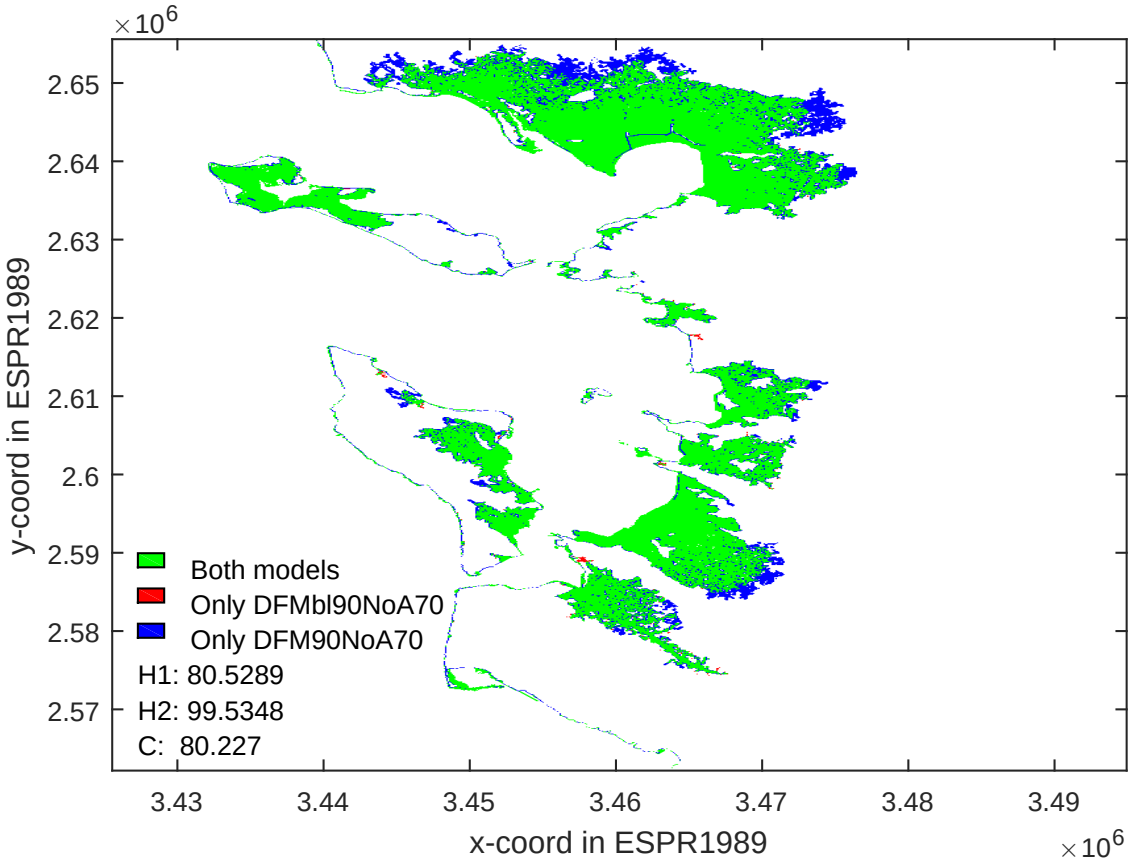


Figure 7.9: Flooded extent simulated by the DFMbl90NoA70 and DFM90NoA70 models compared.

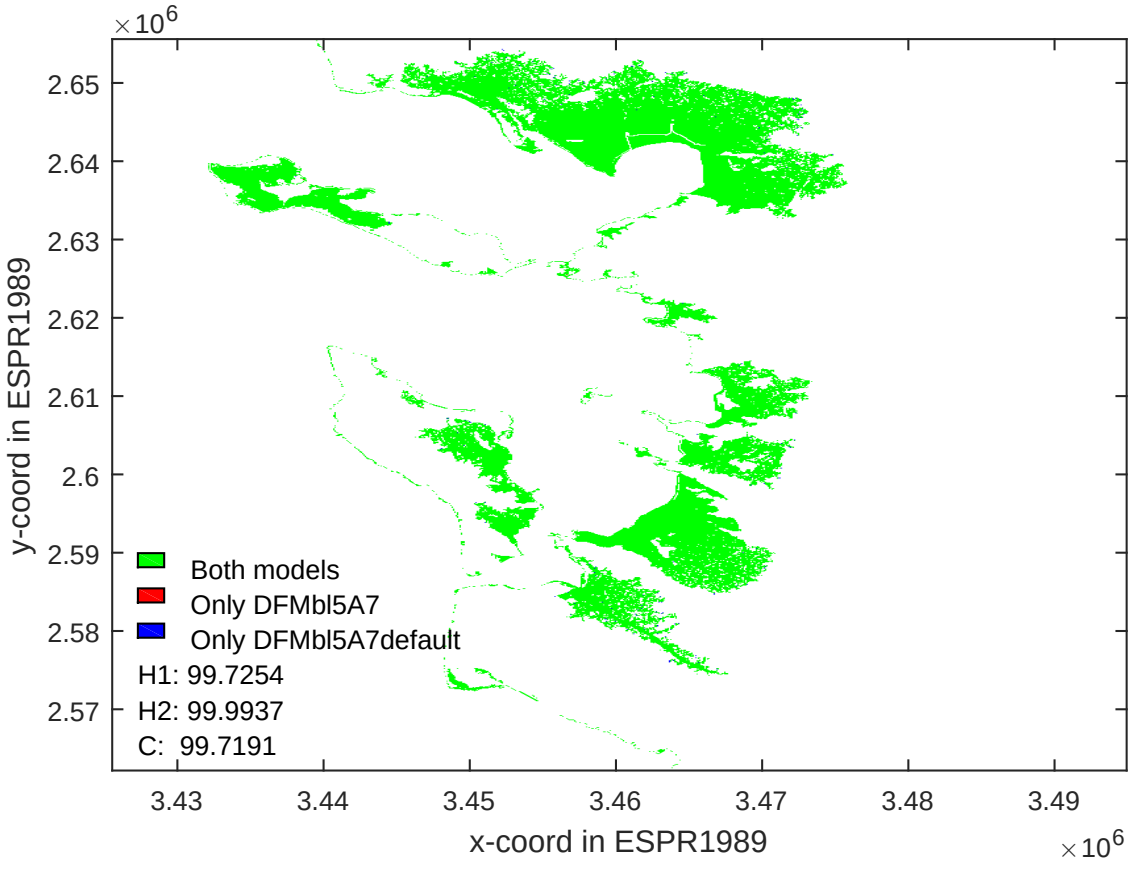


Figure 7.10: Flooded extent simulated by the DFMbl5A7 and DFMbl5A7default models compared. The default model includes advection and turbulence modeling.

Chapter 8

Conclusions

The answer to the research question will be given in this section, concluding the research. Two programs capable of flood inundation simulation have been run and compared, both with and without the momentum advection term of the shallow water equations.

The first question was

1. How do the hydrological modeling packages LISFLOOD-FP (LFP) and Delft3D FM (DFM) compare in terms of

   Flooded extent?

   Computational time?

For the one dimensional test cases, three out of four models compare well in terms of water level, inundation arrival time and flux over the obstacle. The fourth model, LFP with adv, shows a different behavior than the other three models. For the higher obstacles, LFP with advection simulates a much larger volume flux and a higher water level after the obstacle than the other models. It also predicts different water levels at the jump locations.
The 90 m SRTM models LFP and DFM without advection compare well on the inundated area. Typically, C-ratios of around 92% are found between the methods. For LFP and DFM with advection however, the methods do not compare well. As Figure 7.3b shows, a typical C-ratio is 75%. Furthermore, LFP with advection is less conservative than DFM with advection. The reason for this is that the LFP model shows instabilities. Water velocities of 21613 m/s arise, and the time step limiter has been shown not to work as indicated. As for computational time, the LFP model without advection has a minimum computational time of 19 minutes, whereas the DFMbl model can run in 10 minutes. When allowing for different bed level types this can be further reduced, at least to 7 minutes. It was seen that the effect of chaning the bed level type does have a large influence on the inundated area than in- or excluding advection in DFM, having a C-ratio of 80%. The effect of the bed level type and the overall implementation of the topography is something that should be investigated in more depth.

The second question was

2 What is the effect of the momentum advection term on the

   Flooded extent?

   Computational time?

This question is answered for both simulation packages separately. For LFP, it has been shown that the results obtained by including the momentum advection term are not comparable to the model without the advection term in terms of flooded extent, see Figure 7.3c. The Critical hit ratio is 73%. More importantly, the Roe model (LFP with advection) has maximum water velocities of 21613 m/s, which seem to reflect an unstable simulation. The computational time required for the advection model is about 2.5 times as large as for the model without advection. For DFMbl, including advection

does not influence the inundated area much, typical C-values are in the order of 98%. The required computational time however is greatly increased when including advection. Especially when allowing larger maximum allowed time steps, and allowing for a larger CFL number for the models without advection while keeping the CFL number fixed at 0.7.

The final research question was

3 What other settings can be tweaked in DFM in order to reduce the computational demand without affection the flooded extent?

In both LFP and DFM the maximum allowed time step has been varied and in DFM also the CFL condition. For LFP, the computational time ranges between 100 minutes and 19 minutes if the time step is increased from 1 to 10 seconds. The flooded extent and maximum occurred velocities are not influenced significantly for the case without advection. For the DFMbl models, the computational time varies between 64 and 10 minutes when increasing the maximum allowed time step from 5 to 90 seconds and using a CFL condition of 7 in stead of 0.7.

When comparing DFMbl5A7 with DFMbl5A7default, the inundated extent is similar. However, DFMbl5A7default needs 21 minutes extra time. This answers the questions whether complex techniques as turbulence modeling should be included with a negative. If the grid is refined, this question will have to be addressed again.

Overall, it can be concluded that most of the studied cases show very similar results. Exceptions to this, however, are that including advection in LFP results in unstable behavior and that varying the bed level type in DFM does have a significant influence. In- or excluding advection in DFM seems to have almost no influence on the inundated area. Varying the allowed time step and CFL number is equally not influential when considering the inundated area. To a lesser degree, the same can be said for changing the software package. In addition, allowing for a larger time step generally leads to a reduction of computational time.

We were able to run a simulation of a coastal segment in just 10 minutes. The European continent consists of about 11000 of such segments (Vousdoukas et al., 2016). When computing the complete coast of Europe, the laptop used (HP Elitebook from 2015) would require about 75 days. On a single laptop this is impractical. However, on a cluster computer similar to the one Deltares has, this would easily be reduced to a feasible number of days by employing multiple processors. In order to assess the computational time needed for a single simulation on such a cluster, the models should be tested there.

In order to improve the speed of European size simulation, preliminary studies can be conducted to restrict the number of segments to be simulated. They can be excluded based on the topography. For example, an area like a fjord will not need complex models. The land inward depth of a segment can be restricted by conducting a static hydrologically connected model. Areas that are clearly not reachable from the sea do not have to be included in the model domain. This also includes local high lands. Applying such preliminary and cheap methods could potentially decrease the amount of segments.

Bibliography

Gustavo De Almeida and Paul Bates. Applicability of the local inertial approximation of the shallow water equations to flood modeling. *Water Resources Research*, 49:4833–4844, 2013.

Gustavo De Almeida, Paul Bates, Jim Freer, and Maxime Souvignet. Improving the stability of a simple formulation of the shallow water equations for 2-D flood modeling. *Water Resources Research*, 48:1–14, 2012.

Paul Bates. LISFLOOD-FP user manual, 2013. URL `http://www.bristol.ac.uk/geography/research/hydrology/models/lisflood/`.

Paul Bates and Ad De Roo. A simple raster-based model for flood inundation simulation. *Journal of Hydrology*, 236:54–77, 2000.

Paul Bates, Richard Dawson, Jim Hall, Matthew Horritt, Robert Nicholls, Jon Wicks, and Mohammed Hassan. Simplified two-dimensional numerical modelling of coastal flooding and example applications. *Coastal Engineering*, 52:793–810, 2005.

Paul Bates, Matthew Horritt, and Timothy Fewtrell. A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *Journal of Hydrology*, 387:33–45, 2010.

Xavier Bertin, Kai Li, Aron Roland, Yinglong Zhang, Jean-Francois Breilh, and Eric Chaumillon. A modeling-based analysis of the flooding associated with Xynthia, central Bay of Biscay. *Coastal Engineering*, 94:80–89, 2014.

Jean-Francois Breilh, Eric Chaumillon, Xavier Bertin, and M Gravelle. Assessment of static flood modeling techniques: Application to contrasting marshes flooded during Xynthia (western France). *Natural Hazards and Earth System Sciences*, 13:1595–1612, 2013.

Javier Burguete and Pilar Garca-Navarro. Efficient construction of high-resolution TVD conservative schemes for equations with source terms : application to shallow water flows. *International Journal for Numerical Methods in Fluids*, 37:209–248, 2001.

Ven Chow, David Maidment, and Larry Mays. *Book: Applied hydrology.* 1988.

Pierre Defourny, Eric Van Bogaert, Vasileios Kalogirou, and Jose Ramos Perez. GLOBCOVER 2009 Products Description and Validation Report. Technical report, 2011.

DHI. Mike 11 user manual, 2003. URL `https://dhigroup.com`.

Francesco Dottori, Mario Lloyd Virgilio Martina, and Rui Figueiredo. A methodology for flood susceptibility and vulnerability analysis in complex flood scenarios. *Journal of Flood Risk Management*, 2016.

Cristina Forbes, Jamie Rhome, Craig Mattocks, and Arthur Taylor. Predicting the Storm Surge Threat of Hurricane Sandy with the National Weather Service SLOSH Model. *Journal of Marine Science and Engineering*, 2:437–476, 2014.

Neil Hunter, Matthew Horritt, Paul Bates, Matthew Wilson, and Micha Werner. An adaptive time step solution for raster-based storage cell modelling of floodplain inundation. *Advances In Water Resources*, 28:975–991, 2005.

Herman Kernkamp and Arthur Van Dam. Efficient scheme for the shallow water equations on unstructured grids with application to the Continental Shelf. *Ocean Dynamics*, 61:1175–1188, 2011.

Jeffrey Neal, Ignacio Villanueva, Nigel Wright, Thomas Willis, Timothy Fewtrell, and Paul Bates. How much physical complexity is needed to model flood inundation? *Hydrological Processes*, 26: 2264–2282, 2012.

Jorge Ramirez, Michal Lichter, Tom Coulthard, and Chris Skinner. Hyper-resolution mapping of regional storm surge and tide flooding: comparison of static and dynamic models. *Natural Hazards*, 82:571–590, 2016.

Philip Roe. Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. *Journal of Computational Physics*, 43:357–372, 1981.

Christopher Sampson, Timothy Fewtrell, Alastair Duncan, Kashif Shaad, Matthew Horritt, and Paul Bates. Use of terrestrial laser scanning data to drive decimetric resolution urban inundation models. *Advances In Water Resources*, 41:1–17, 2012.

Guus Stelling. Quadtree flood simulations with sub-grid digital elevation models. *Water Management*, 165:567–580, 2012.

Ignacio Villanueva and Nigel Wright. Linking Riemann and storage cell models for flood prediction. *Water Management*, 159:27–33, 2006.

Michalis Vousdoukas, Evangelos Voukouvalas, Lorenzo Mentaschi, Francesco Dottori, Alessio Giardino, Dimitrios Bouziotas, Alessandra Bianchi, Peter Salamon, and Luc Feyen. Developments in large-scale coastal flood hazard mapping. *Natural Hazards and Earth System Science*, 16:1841–1853, 2016.