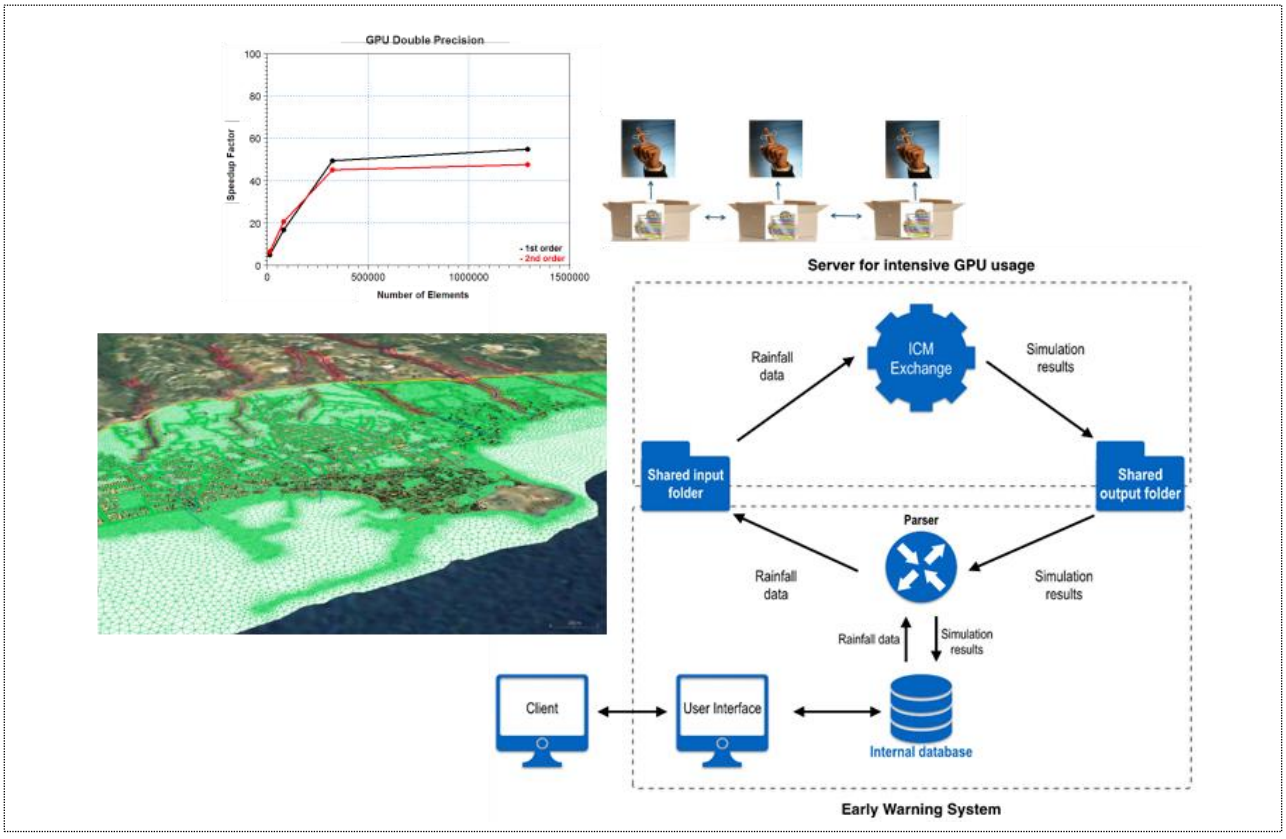
D4.2 Report

Guidelines on Achieving Faster Simulations for Early Warning



This project has received funding from the European Union´s Seventh Framework Programme for Research, Technological Development and Demonstration under Grant Agreement N° 603663 for the research project PEARL (Preparing for Extreme And Rare events in coastal regions).

D4.2 Report

Guidelines on Achieving Faster Simulations for Early Warning

Authors: *DHI, TUHH*

Contributors: *Hyds*
NTUA (Makropoulos, C. and Lykou, A.)

Dissemination level (select one):
PU = Public
PP = Restricted to other programme participants (including the Commission Services).
CO = Confidential, only for members of the consortium (including the Commission Services).

Document Information

Project Number	***603663***	Acronym	**PEARL**
Full Title	Preparing for Extreme and Rare events in coastal regions		
Project URL	http://www.pearl-fp7.eu/		
Document URL			
EU Project Officer	Eleni Manoli		

Deliverable	Number	D.4.2	Title	Guidelines on Achieving Faster Simulations for Early Warning
Work Package	Number	WP4	Title	Flood forecasting and early warning systems for coastal regions

Date of Delivery	Contractual	MM.DD.YYYY	Actual	MM.DD.YYYY
Status	version 1.0		final X	
Nature	prototype □ report X dissemination □			
Dissemination level	public X consortium			

Abstract (for dissemination, 100 words)	This guideline document presented and discussed options for achieving faster model simulations in the context of flood early warning. These options involve different aspects of the modelling process, from model conceptualisation and construction, to computational hardware capabilities. The various techniques were organised in two categories in the discussion: 1. Techniques for minimising computational load • Model (concept) simplification • Model (construction) optimisation 2. Techniques for maximising computational power • CPU parallelisation • Hydbrid/GPU computing
Keywords	Flooding, fast simulations, CPU parallelisation, GPU computing

Version Log				
Issue Date	Rev. No.	Author	Change	Approved by

Disclaimer

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under Grant agreement n° 603663 for the research project PEARL (Preparing for Extreme And Rare events in coastal regions).

Summary

This report presents guidelines for achieving faster simulations in urban coastal flood modelling. It focuses on two approaches:
(1) Minimising the computational load for models used in early warning
(2) Maximising computational power.

After a general introduction of concepts in Chapter 1, the proposed approaches for improving simulation times are discussed in Chapter 2. Various techniques related to minimising computation loads (1) are presented in Chapter 2.1, and Chapter 2.2 lists techniques for maximising computation power for model simulations (2). Example application of the various techniques in the project case areas of Hamburg (Germany), Greve (Denmark), Marbella (Spain) and Rethymno (Greece), are then presented in Chapter 3. Finally, the report ends with Chapter 4 containing a summary and conclusions about the proposed guidelines for achieving faster simulations for early warning.

Contents

List of figures

List of tables

Abbreviations/Acronyms

CPU	Central Processing Unit
GPU	Graphics Processing Unit
MPI	Message Passing Interface
OpenMP	Open Multi-Processing

API	Application Programming Interface
1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
NWP	Numerical Weather Prediction
CET	Central European Time
FLOPS	FLoating-point Operations Per Second

1 Introduction

Complex model chains, particularly integrated models considering the complete spectrum of climate-ocean-coastal-overland flows, are notoriously computationally-heavy, leading to their limited use in early warning applications. An objective in the PEARL Project is to develop techniques for fast flood simulations for early warning in areas with combined risks from coastal, fluvial and pluvial flooding. Here fast means sufficiently fast models as are fit for the purpose.

These guidelines present speed-up techniques that have been identified, implemented, and evaluated in the project focusing on two approaches: (1) minimising the computational load for models used in early warning (Chapter 2.1), and (2) maximising computational power (Chapter 2.2). Minimising the computational load for integrated models involves optimising model efficiency through solutions such as:

- optimizing grid resolutions,
- sub-grid scale characteristics,
- grid types,
- application of simplified modelling techniques.

Maximising computational power, on the other hand, involves optimal use of technological advances, including use of:

- parallelisation,
- super computers,
- cloud computing,
- GPU (Graphical Processing Unit) parallelisation.

Example applications of these techniques in several case studies are presented in Chapter 3.

Computational models are used in real-time forecasting or offline scenario analysis for flood early warning systems. Offline analysis involves prior modelling with various conditions to obtain a catalogue of flooding that may be used as a reference for future flood estimates. Real-time forecasting, on the other hand, involves the use of current and short-term future predicted driver and system conditions to determine if, when and where flooding would occur.

Cartesian grid-based hydrodynamic models simulate urban flooding with high resolution and accuracy, and the state-of-the-art method of 1D-2D coupling of network and Cartesian hydrodynamic models has been proven to simulate urban coastal flooding with high resolution and reliability (Schmitt et al., 2004; Sto. Domingo et al., 2010; Hénonin et al., 2013; Leitão et al., 2013). However, their long computational times limit their viability in early warning applications. These integrated models are computationally expensive and often cannot be run fast enough to predict urban floods in real time (Yu & Lane, 2006; Chen et al., 2012). Simulation times with 2D inundation models can be excessive (the order of days) at grid resolutions considering flows around buildings and other surface structures (Liu and Pender, 2010). Consequently, recent research efforts have focused on improving the efficiency of these models and speeding up simulations.

The efficiency of 2D models is a major challenge to flood modellers (Chen et al, 2012). This issue is especially relevant in risk analyses involving a large number of simulations, and especially in flood warning applications, which require timely and accurate estimation of potential flooding in real-time. This efficiency depends on the chosen time steps, size of the study area, the efficiency of numerical algorithms and the use of multi-processing, among others. This has prompted

unceasing efforts towards developing other methods for flood analysis, or new techniques for speeding-up 2D model simulations. As summarized by Chen et al. (2012), methodologies that have been developed to improve the performance of modelling include grid coarsening (Yu & Lane, 2006), reduced complexity models (Liu et al., 2013); simplified governing equations (Bates et al., 2011); parallelization (Neal et al., 2010; Smith et al., 2014); unstructured mesh (Wang et al., 2010) and adaptive grid based methods (Liang & Wang, 2011). Some of the latest investigations speed up simulations, whilst preserving the flood prediction accuracy, through the application of flexible meshes and computation parallelization techniques. Open Multi-Processing (Open MP), Message Passing Interface (MPI), and Graphical Processing Unit (GPU) parallelisation techniques have been developed and implemented to solve efficiency problems with 2D flood models (Hankin et al., 2008; Neal et al., 2010; Kalyanapu et al., 2011; Ghimire et al., 2013; Smith et al., 2014).

2 Techniques for Faster Model Simulations

In this report, techniques for improving model efficiency are categorized according to options for: a) minimising computational load, and (b) maximising computational power. The first set of options relates to modification of 'soft' components, such as model setup or software code (i.e. equations), while the second option involves the use of new developments related to hardware. In addition, the discussions focus on speed-up techniques for fully hydrodynamic flood modelling methods. The ultimate goal is to attain a flood modelling solution appropriate for warning applications, which needs to be, not only time-efficient, but also accurate with its flood estimates.

Figure 1 shows a typical work process for systematically developing a suitably fast-running flood model. The type of model to use, which influences expected computation times, is initially determined based on factors such as dominant processes to be simulated, data availability, and main purpose of the modelling. Then, as the model domain directly impacts the number of computational points and computation times, it needs optimisation based on the main area of interest, scale levels and information required from the modelling. Once the model extents have been set, additional techniques may be used to further optimise the number of computation points in the domain, such as changing the computational grid type. These first few phases in the work process relate to the configuration of the flood model. After which, further options for model speed-up then involve maximising the use of available hardware/computing power in the modelling. Repetition of the process is performed if speed-up targets remain unachieved (Figure 1).

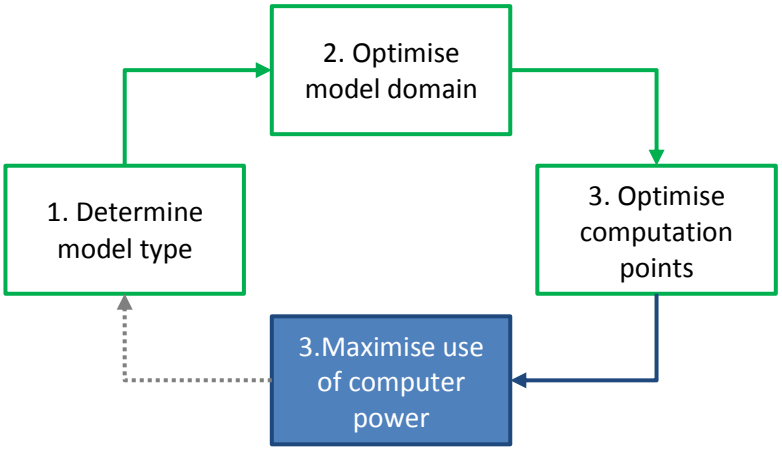


Figure 1 Illustration of example work process for investigating model speed-up options.

Techniques for achieving faster model simulations are presented in the succeeding sections following the work process shown in Figure 1.

2.1 Minimising Computational Load

Minimising computational loads in hydrodynamic flood modelling may be achieved through modification of the computational grid or model domain, as well as simplification of the model and the governing equations used in the calculations.

2.1.1 Model simplification

The use of simplified models is a way of achieving fast simulations in inundation modelling for early warning. This technique involves the use of simplified governing equations in hydrodynamic

modelling, or using a totally different type of less complex (i.e. non-physics-based) model (Chen at al., 2012). Different modelling techniques are available for flood analysis, ranging from simple terrain analysis, to 2D surface flow modelling, to advanced modelling systems integrating 1D network and 2D terrain models, or even 3D ocean models (Figure 2). The model type is selected based on detail and accuracy levels needed. The model must consider the dominant processes involved, and have input requirements corresponding to available data (Parkinson and Mark, 2005). It may not be necessary to choose an advanced, physically based model over a simpler model, which could afford savings with respect to computation times.

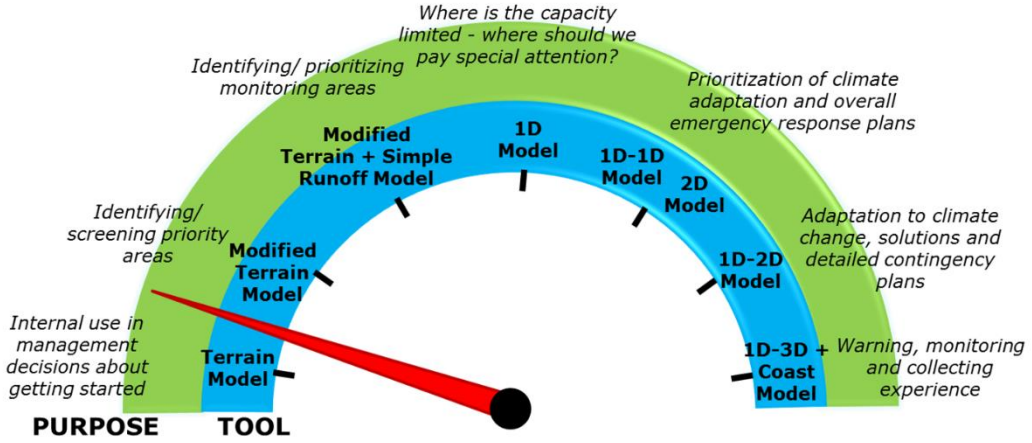


Figure 2 Illustration of how various flood modelling tools of varying complexity are used for different purposes. Source: (DANVA, 2007).

Despite the availability of a wide range of computationally efficient simplified inundation modelling methods, warning systems are subject to expectations regarding forecasting technique and accuracy, together with forecast lead-time and service delivery. Forecasts upon which warnings are based should be as accurate as possible (Parkinson & Mark, 2005), and hydrodynamic models that provide high resolution depth and discharge information for flood predictions are considered advantageous in flood risk assessment and forecasting (Price & Vojinovic, 2008). Thus, in these guidelines, speed-up techniques maintaining the level of accuracy as obtained with physically based modelling are favoured over simplified modelling in flood early warning applications.

2.1.1.1 Uncoupled 2D hydrodynamic models

Coupled 1D-2D models are considered the state-of-the-art in urban flood modelling. These models are able to simulate flows in underground sewers and over the land surface, as well as the flow exchanges between the two systems (right in Figure 3). However, this type of integrated model involves multiple systems that entail additional computational points compared to when only one of the coupled systems is used. However, 2D surface flow models (left in Figure 3) are also viable options for modelling flooding. They simulate flows and water levels based on computations solving the Saint-Venant equations in two dimensions (i.e. Shallow Water Equations (SWEs)). Flood modelling with this approach assumes that flooding primarily occurs according to runoff, accumulation and flow mechanisms and processes overland.

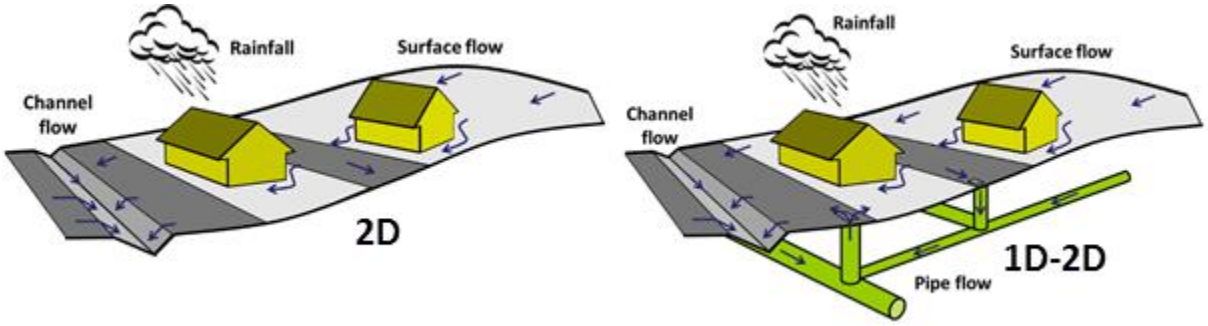


Figure 3 Illustrations comparing 2D (left) and coupled 1D-2D (right) flood models.

For 2D models, rainfall may be applied directly on the surface, and sea level data defined along open boundaries as are relevant in coastal flooding applications. However, underground sewers are unrepresented, and if important, approximation techniques must be applied to consider their impact in the modelling, such as through rainfall pre-processing, and the like. Hence, the use of 2D models for flood analysis is mainly relevant for areas where:

- Sewer network data is lacking and unavailable
- Sewer network does not exist
- Sewer network does not have a significant impact on the flood process being analysed

Part of the analysis performed in the Greve Case Study (see Chapter 3.1) was investigating the appropriate type of urban coastal flood model to use, wherein the sensitivity of model results efficiency were analysed when using 2D or 1D-2D modelling methods. Using the same sea level boundary conditions, results obtained from the flood model with and without the 1D sewer network are shown in Figure 4.

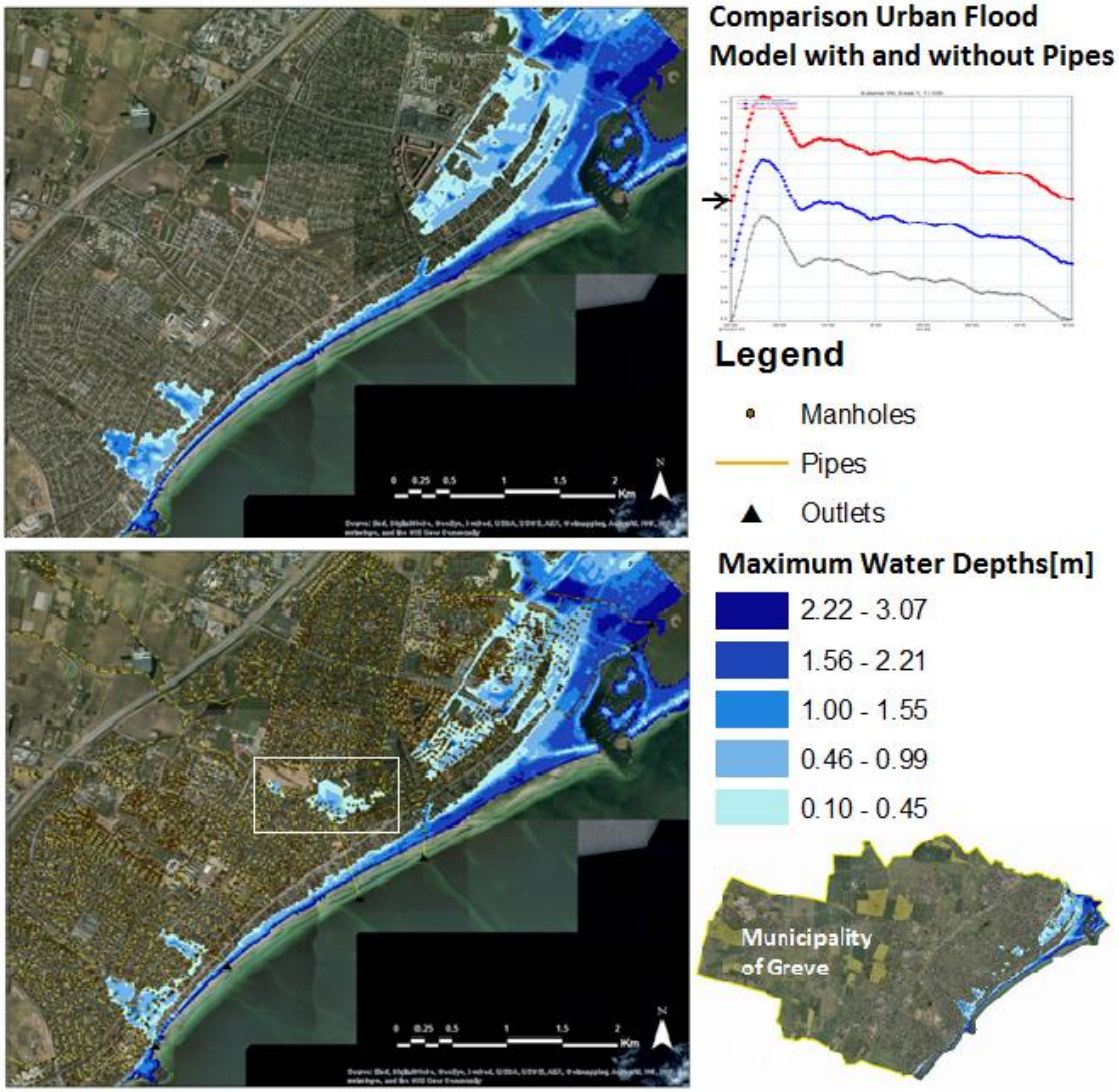


Figure 4 Comparison of computed urban coastal flooding in Greve using a 2D model without the 1D sewer network (upper), and a coupled 1D-2D model with the sewer network (bottom) for a 100-year extreme sea level event under climate change conditions. (Source: Berbel Roman, 2014)

Results showed more flooding computed with the 1D-2D model compared with the 2D model, amounting to around a 10% difference in flood extents between the two. It was apparent that water from the sea could enter the drainage network and cause surcharges in low-lying areas further inland. In terms of computational speed, simulation times with the 1D-2D model largely depend on 2D model computations; therefore, the difference in simulation times between the 2D and 1D-2D models for Greve were relatively not substantial, amounting to around a 20% reduction going from a 1D-2D to a 2D model (Table 1). Thus, in the case of Greve, the sensitivity analysis showed that a 1D-2D model was more prudent to use for flood analysis, as the relatively flat terrain, and presence of a dense sewer network that is well-connected to the sea necessitates consideration of the 1D drainage network in the flood model, and that the simulation time savings afforded by removing the 1D model were not substantial in comparison to the loss in model accuracy.

Table 1 Sensitivity of simulation time with flood model type (i.e. 2D versus 1D-2D) in Greve. (Source: Berbel Roman, 2014)

Model	Simulation Time [min]	Simulation Time [h]
2D Flexible Mesh	576	9.6
1D-2D Flexible Mesh	733	12.2

2.1.1.2 Pseudo 2D models

2D flood models, nevertheless, still have high computational resource requirements for solving the SWEs, so efforts have been made to improve modelling performance, while maintaining accuracy, by reducing the complexity of these governing equations. This is, for example, achieved through reducing the terms of the equations (Hunter et al., 2007). The JFLOW model (Bradbrook et al., 2004), Urban Inundation Model (UIM) (Chen et al., 2007), and the diffusive version of LISFLOOD-FP (Bates and De Roo, 2000) solve the 2D diffusive wave equations that ignore inertial terms.

These 'Pseudo 2D models' implement simplified versions of the shallow water equations in calculating inundation. These types of simplified models are computationally efficient as less calculation terms are handled. Moreover, they may be implemented using numerically implicit formulations affording high Courant numbers and possibilities for using larger computational time steps. However, they use a flood spreading approach not simulating velocity patterns and giving a highly simplified description of reality. Thus, they should be used with caution, and are generally appropriate for simple preliminary flood screening applications to help identify potential flood risk areas for further detailed analysis and monitoring.

The Flood Screening Tool (FST) by DHI (2014), also employs the diffusive wave approximation, wherein convective acceleration, viscous, and external forcing terms are disregarded from the governing equations. Computational performance and simulation results with FST have been compared with those from a full 2D hydrodynamic model (i.e. MIKE 21), for overland flow simulations in an area in Denmark. The comparative evaluations were based on flood extents and water volumes on the surface at various stages during the storm event, flood depths at selected locations in the model, and overall water balance with respect to rainfall volume, system volume and boundary flows. From the comparisons, it was found that model speed-up factor with the simplified FST solution, compared to MIKE 21, was around 1.5. Excellent mass balance was achieved with FST, while the 2D model generated water amounting to around 6.6% of the total rainfall volume. Simulated flood extents (for water depths > 5 cm) during the times of well-developed flood (i.e. after the peak of the rain), differed by only a small margin (around 2%), although in the earlier phases, the relative differences were more significant, indicating differences in flood movement dynamics (Figure 5) (DHI, 2016).

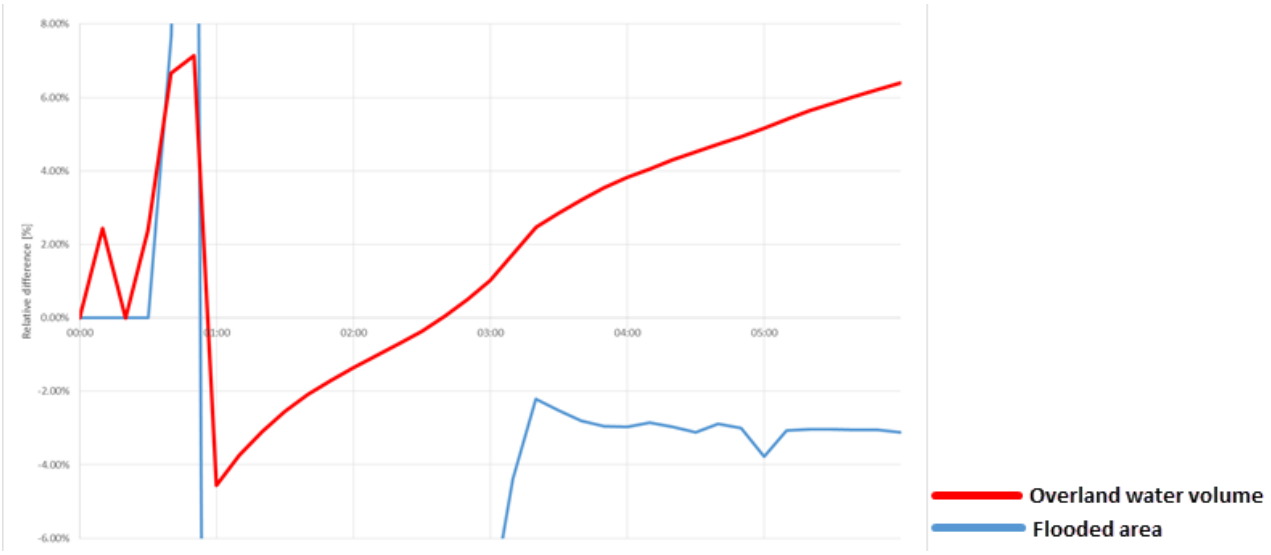


Figure 5 Relative deviation (%) between the simulated flooded areas (blue) and overland water volume (red) for the two models (MIKE 21 Flow Model vs. MIKE 21 FST). (Source: DHI, 2016)

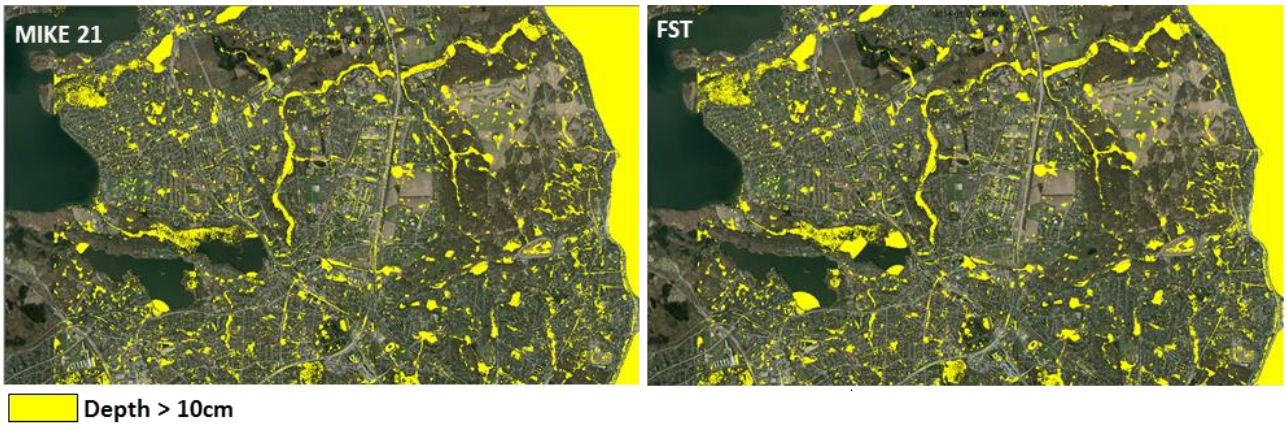


Figure 6 Comparison of computed flood areas (Depth >10cm) from the 2D MIKE 21 (left) and FST (right) models at the end of the simulation (06:00 hrs). (Source: DHI, 2016)

Another example of simplified models are the so called Rapid Flood Inundation Models (RFIMs) based on a flood cell storage algorithm, such as those presented in Liu and Pender (2010), Lhomme et al. (2008), and Krupka et al. (2007). Liu and Pender (2010) presented an RFIM that predicts water depths and extents using a volume spreading algorithm that also accounts for inflow and surface characteristics, which offers an improvement to that of Krupka et al. (2007). The RFIM by Krupka et al. (2007) involves identification of flood storage cells in the model domain based on a digital elevation model (DEM), and then flood estimation through distribution of a specified volume of water across the storage cells. Improvements to the RFIM were also proposed by Lhomme et al. (2007), incorporating additional physical process, such as multiple spilling and friction, in the spreading algorithm.

2.1.1.3 Terrain analysis

Terrain Analysis is another flood estimation technique, wherein inundation depths are computed solely based on topographical information from the Digital Terrain Model (DTM) of the area (Figure

7). This method, using raster calculator tools, is very quick to implement. In coastal flooding applications, it calculates inundation in all areas lying below the projected maximum sea level, even without topographical or underground drainage connections to the sea. To deal with this issue, the method may be modified, such that inundation is computed only in areas that have hydraulic connections to the source (e.g. the sea for coastal flooding), and no flooding is computed beyond areas of high elevation.

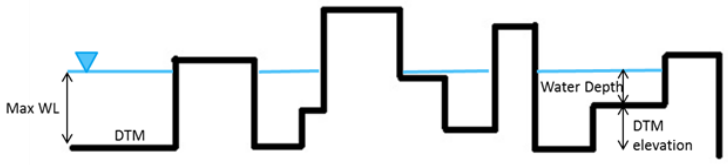


Figure 7 Profile view illustration of Terrain Analysis method estimating inundation, wherein water depths are computed by subtracting terrain elevations from a maximum water level value (Water Depth = Max WL – DTM elevation).

In addition, using DTMs for analysing inland flooding, inundation may be considered to occur in and around local depressions on the terrain, as in so-called 'Depression Maps' /'Blue Spot Maps' in DANVA (2007). Identification of these depressions and their hydraulic connectivity is performed using GIS tools, and used to estimate inland flood extents.

In the Greve Case Study (see Chapter 3.1), Terrain Analysis was tested and evaluated against the use of 2D hydrodynamic models for analysing urban coastal flooding in the process of identifying the most appropriate, fast (and accurate) model to use for the flood forecasting system. An extreme climate change scenario 100-year sea level event condition was used as input in the three different approaches tested: (1) Terrain Analysis, (2) Cartesian grid 2D model, and (3) Flexible mesh 2D model (see Chapter 3.1.1). To assess their performance in terms of accuracy and speed, results were compared with respect to flood extent, ease of model set up, and computational time (Table 2).

Table 2 Summary of comparison results between Terrain Analysis and 2D hydrodynamic modelling approaches for urban coastal flood analysis in the Greve Case Study. (Source: Roman, 2014)

	Terrain Analysis	Cartesian Grid 2D	Flexible Mesh 2D
Inundated area [km^2] (Model area = 19 km^2)	5.14	3.72	3.75
Ease of model setup	Fast	Easy	Complex
Computation time [h] (Event duration = 42.5 h)	N/A (Very low)	~35.8	~9.6

Figure 8 Maximum flood maps in the Greve Case Study obtained using Terrain Analysis (left) and 2D Cartesian grid modelling (right) using a 100-year climate change scenario extreme sea level event as input.

The results illustrate overestimation of flooding with Terrain Analysis compared to hydrodynamic modelling. Maximum flood extents amounted to 5 km^2 with Terrain Analysis, while the 2D models both calculated around 4 km^2. This is because all areas below a certain level were considered flooded with Terrain Analysis, even areas without connections to the sea. The 2D surface flow models provide a more accurate description of flooding, calculating flows and velocities over the surface using physics-based equations. In the study, the use of a flexible mesh computational grid revealed potential model speed-up of up to 4 times compared to use of Cartesian grid. However, whether the model speed-up was due to the use of the flexible mesh, or because the number of elements with the flexible mesh is lower than with the Cartesian grid, needs to be further clarified.

Nevertheless, the tests showed that the main advantage of the Terrain Analysis method over 2D hydrodynamic flood modelling is that it is easy to setup and implement, and provides a quick overview of the problem, showing areas where water can potentially be stored in depressions on the terrain. However, the uncertainty of this method is significant since it ignores physical processes, such as friction losses and flow propagation over the terrain. Terrain Analysis is most suitable for coastal areas with large slopes, where only areas near the coast are most likely to be inundated; otherwise, flood extents are easily overestimated. Hence, Terrain Analysis is suited for problem identification, whereas detailed analysis involves a hydrodynamic model.

2.1.2 *Selecting the model domain*

When the model type has been determined, optimising the domain is then investigated as an effort to further achieve fast simulations with the flood forecast model (Figure 1). In general, model extents vary proportionally with the number of (active) computational points, and accordingly, computation times, in hydrodynamic flood modelling.

Therefore, potential model speed-up may be achieved by careful delineation of the model domain considering the following:

- **Dominant processes**. Modelling coastal flooding could involve considering not only inland coastal areas, but also near-shore sea areas in the model domain. In addition, significant hydrologic contributions from upstream rural areas to coastal flooding may necessitate inclusion of upstream catchments in the model.
- **Interest area**. The model domain may be defined as only the area on which the main stakeholders of the flood forecasting site are interested. For example, hydrologic delineations seldom follow municipal or City boundaries, but the model domain may be constrained to an

area smaller than that defined by hydrological delineation if they go beyond the municipal or City lines for a flood forecasting system owned/maintained by a certain municipality or City.

- **Area of influence**. The model domain may be limited to areas where boundary forcing impact the flood process of interest. For example, when modelling coastal flooding due to extreme sea levels with a 2D model, extents may be limited to near-coast areas where flooding from the sea will reach (e.g. with the most extreme scenario plausible), excluding areas beyond high terrain or structures, or decreasing the level of detail in local areas of high elevation within the model domain.
- **Boundary locations**. Especially for coastal flooding, the type and location of the sea boundary forcing affects model extents. For instance, if overtopping of coastal structures is of main interest, the model domain may exclude near-shore areas before the structure, and the water level or flow conditions applied at the structure location. If spatially distributed water level or flow conditions along an open boundary are used as sea boundary conditions, the location of the boundary must be optimised such that computed inland flood results will not be sensitive to the location of the boundary in the flood model. In general, boundaries should be located where the boundary condition is not affected by the system, and where the interest area is not directly affected by the boundary condition.

In the Greve Case Study, the domain for the 1D-2D model was determined through several iterations of optimising the extents, mainly of the 2D component, as the number of active 2D computational points largely determined the use of computing resources in simulations. To begin with, the coverage of the 2D model was limited to only areas near the coast (Figure 9). The areas further inland, although still part of Greve Municipality, were beyond the influence of expected coastal flooding, having terrain elevations beyond the maximum sea level ever recorded in Køge Bay of 3.7 m (in 13 October of 1760) (Reference). Potential rainfall runoff contribution from the whole municipality was nevertheless considered through the runoff model component of the 1D drainage network model.

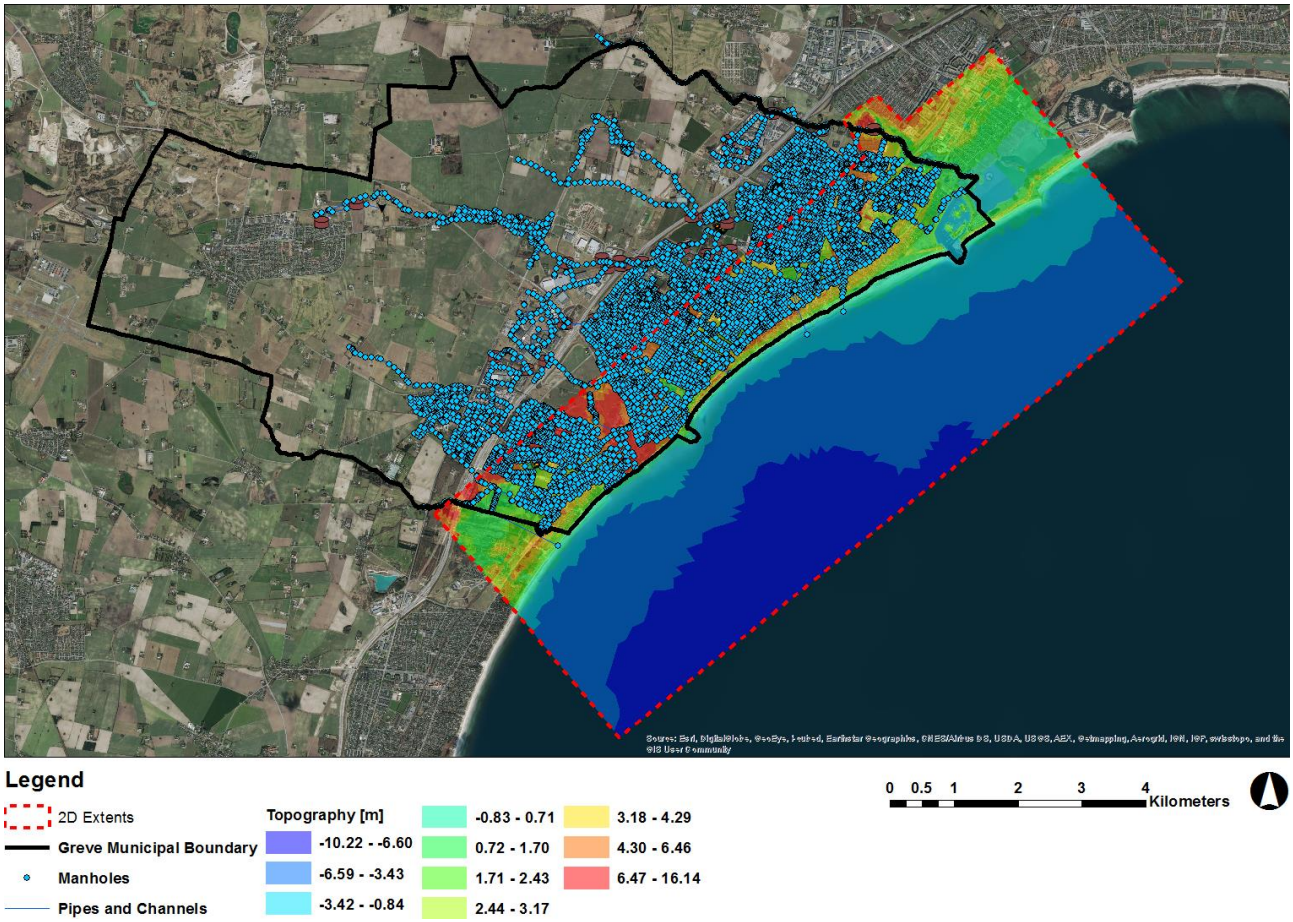


Figure 9 Map showing the municipal boundaries of Greve (black solid outline), and the extents of the integrated 1D-2D model. The 1D network is indicated by the manholes, pipes and channels, while the 2D model area is outlined in red (dashed line).

Along the coast, the demarcation of model boundaries was partly based on municipal borders as Greve was the sole municipality involved in the project. In addition, technical aspects, such as terrain elevations and presence of natural watercourses, were considered. Tests were performed using the most extreme sea level projection available (i.e. 100-year future scenario extreme sea levels) in order to estimate the maximum influence area from flooding inland. Simulations were performed with expansions of the four boundaries that delineate the 2D model (Figure 10). The sensitivity of deviation between simulated and measured sea levels at two harbour stations to the location of the sea level boundary input was analysed in order to determine the optimum location for the sea level boundary.

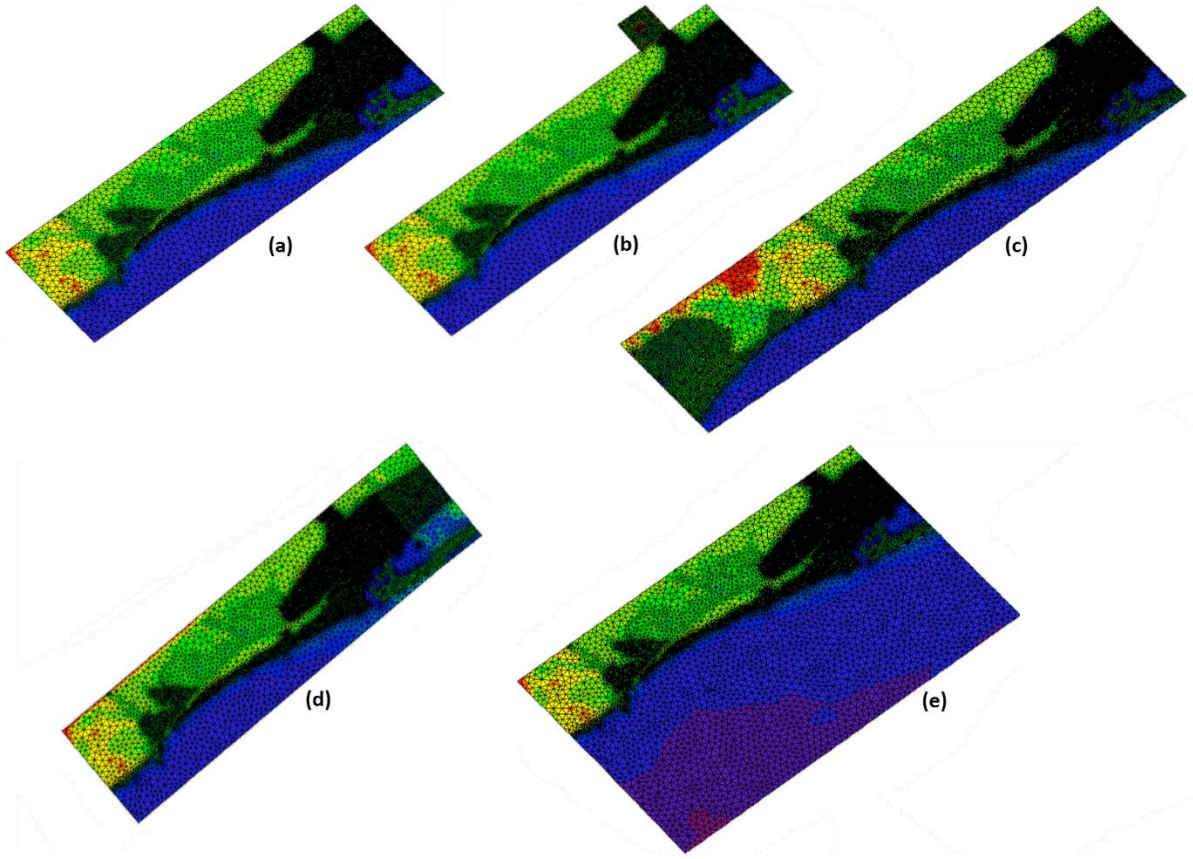


Figure 10 Tests were performed in the Greve Case Study to determine the optimum model extents and location of the sea boundary, starting from the initial model domain (a) to expansions of the northwest (b), southwest (c), northeast (d), and southeast (i.e. sea) (e) boundaries.

Based on the analysis, to properly simulate up to future scenario 100-year extreme water level events, the closed land boundaries must be expanded to avoid potential overland flow blockages along the boundaries. The northwest boundary (Figure 10 (b)) should be expanded to include a small lake area just beyond the municipal boundaries. The model should be expanded to the southwest (Figure 10 (c)) up to the boundary of Greve, and to the northeast (Figure 10 (d)) to include more of the adjacent municipality, which, although lies beyond the border of Greve, allows better representation of the terrain in the area. Finally, the sea boundary must be moved further out of the potential surge build-up zone near-shore (Figure 10 (e)). The updated 2D domain for the Greve flood model is shown in Figure 9. The 1D network model covers the whole of Greve, such that rainfall runoff from impervious areas in the whole municipality is modelled and drained by the 1D network model, which is dynamically linked to the 2D surface flow model covering the coastal and near-shore areas, simulating flooding from the sea.

2.1.3 *Selecting computational grid type*

The type of 2D computational grid used in an integrated 1D-2D model impacts the potential number of calculation points, and hence, computational resources used, during simulations. The use of flexible meshes effectively reduces computational loads in 2D hydrodynamic modelling in comparison with conventional Cartesian grid-based models for inland/floodplain applications (Sleigh et al., 1998; Namin et al., 2004; Chen et al., 2012, Kim et al., 2014). It allows for more flexibility and efficient geometric definition of surface features and complex flow domains, such as in urban areas, in turn reducing simulation times without compromising numerical accuracy.

Kim et al. (2014) compared and evaluated the use of different 2D model computational meshes in simulating different types of flow systems in terms of result accuracy and simulation speed. It was determined that mesh type is important when high modelling efficiency, balancing computational cost and model accuracy, is an objective. Different flow situations require different mesh types for maximum model efficiency; Cartesian grids are beneficial in modelling rectangular channel flows, while unstructured meshes are advantageous in applications involving natural topographies and irregular boundaries.

In the PEARL Project, the use of a flexible mesh to improve the efficiency of an urban coastal flood model was tested and evaluated in the Greve Case Study, which is described in Chapter 3.1. The aim of the study was to improve the efficiency of the coupled 1D-2D coastal flood model for the city of Greve in Eastern Denmark, for use in early warning applications. Results from the case study show that the use of a flexible mesh for the 2D surface flow model component reduced computation times by a factor of 3 to 4 compared to the use of a Cartesian grid.

2.1.4 Optimising the computational grid/mesh

Grid optimisation involves selection of appropriate grid or mesh element size, as well as the distribution or arrangement of the different grid/element sizes in the model domain. Grid coarsening and optimisation are additional techniques for improving model performance.

2.1.4.1 Grid/mesh element size

The chosen computational grid or mesh resolution must sufficiently resolve surface features that impact overland flows. The accuracy of a 2D model depends mainly on both topographical data resolution and model grid size (Hénonin, et al, 2013). Urban applications normally require a fine 2D grid to represent urban features such as buildings and streets. In addition, topographical data from which the 2D grid levels are derived, should be at least as fine as the grid size, as it is pointless to use a very fine grid size if their corresponding values are derived from coarse data.

Cartesian grid

Grid coarsening is a way of reducing computational effort in 2D modelling. However, traditional grid-coarsening, wherein grid sizes are simply enlarged, has the tendency of reducing model accuracy, as surface relief information is lost in the low-resolution grid (Chen et al., 2012; Hénonin et al., 2013) (see Figure 11). As illustrated in the figure, computational grid resolution in urban areas is ideally around 1-2 m, and less than 5 m, to properly represent urban features such as roads, buildings, etc. (Figure 11).

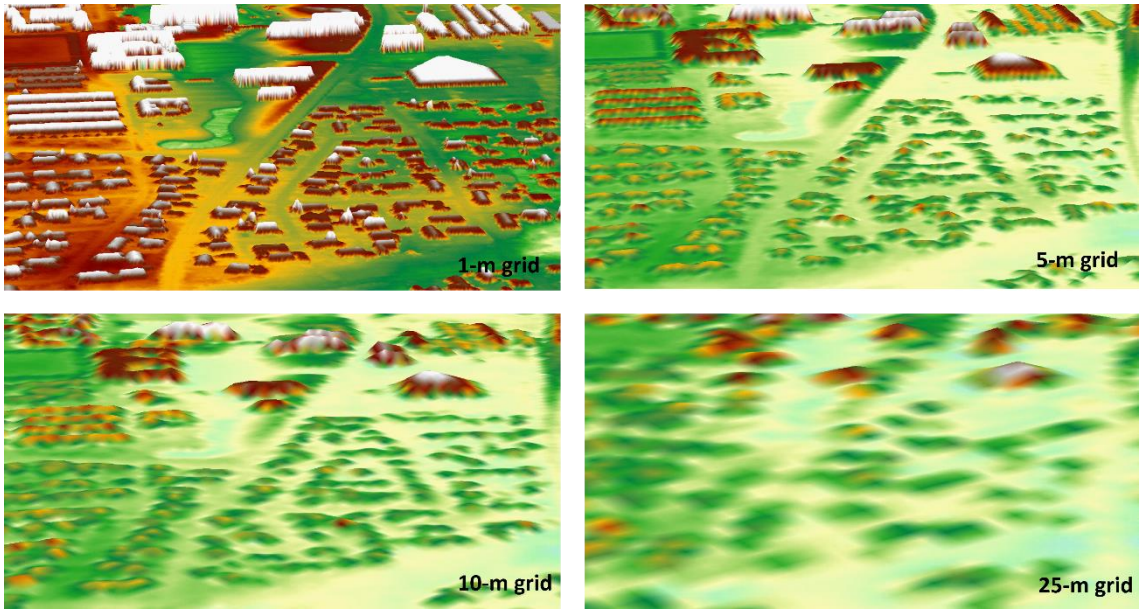


Figure 11 3D plots comparing representation of urban surface features on 2D Cartesian grids of different resolutions (i.e. grid sizes). (Source: Hénonin et al., 2013)

Nevertheless, techniques have been developed to address these limitations, which include the Multi-layered Approach by Chen et al. (2012), which considers the presence of sub-grid structures and reflects individual flow paths within a coarse grid cell, and the Multi-Cell Overland Solver approach (Hartnack et al., 2009; DHI, 2014; Hénonin, et al., 2015), which solves the governing equations on the coarse grid and then accounts for topographical variation through post-calculation of fluxes and depths on the fine grid (Figure 12).

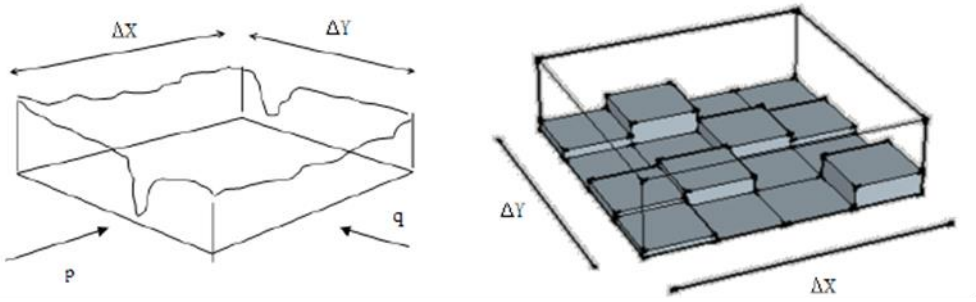


Figure 12 The Multi-Cell Overland Solver approach solves the governing equations in the control volume defined by a coarse grid cell (left), wherein the topography may vary as described by the fine grid upon which fluxes are derived (right) (Hartnack et al., 2009).

Mesh elements

As in the case of Cartesian grids, mesh element sizes should resolve important surface features affecting overland flows. For example, in urban areas, maximum element sizes of around 10 to 25 m^2 may be sought. Mesh elements can be more readily arranged to represent complex flow paths compared to rigid grids, such that bigger mesh element areas may be used compared to grid cells (Figure 13).

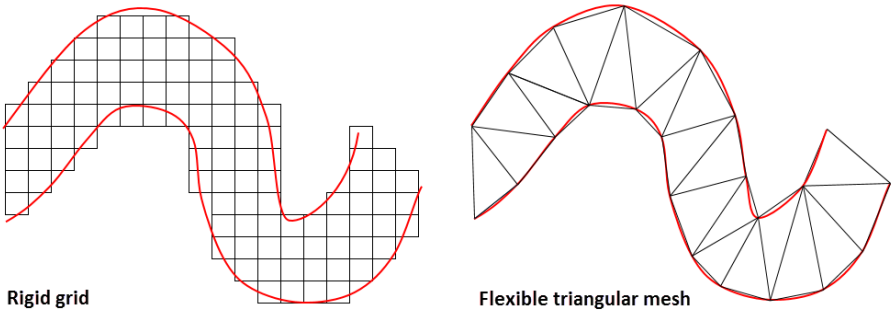


Figure 13 Illustration of complex flow path representation (red outline) using a rigid Cartesian grid (left) and a flexible triangular mesh (right).

Mesh element sizes may be varied in different zones within the domain depending on the dominant processes being analysed (e.g. finer elements along the coast for coastal flood modelling), the main overland flow paths identified, or the main area of interest. Topographical information may also be used to objectively define areas of finer mesh sizes, wherein the distribution of finer elements may be encouraged in areas of certain elevations or gradients (see Figure 14and Figure 15).

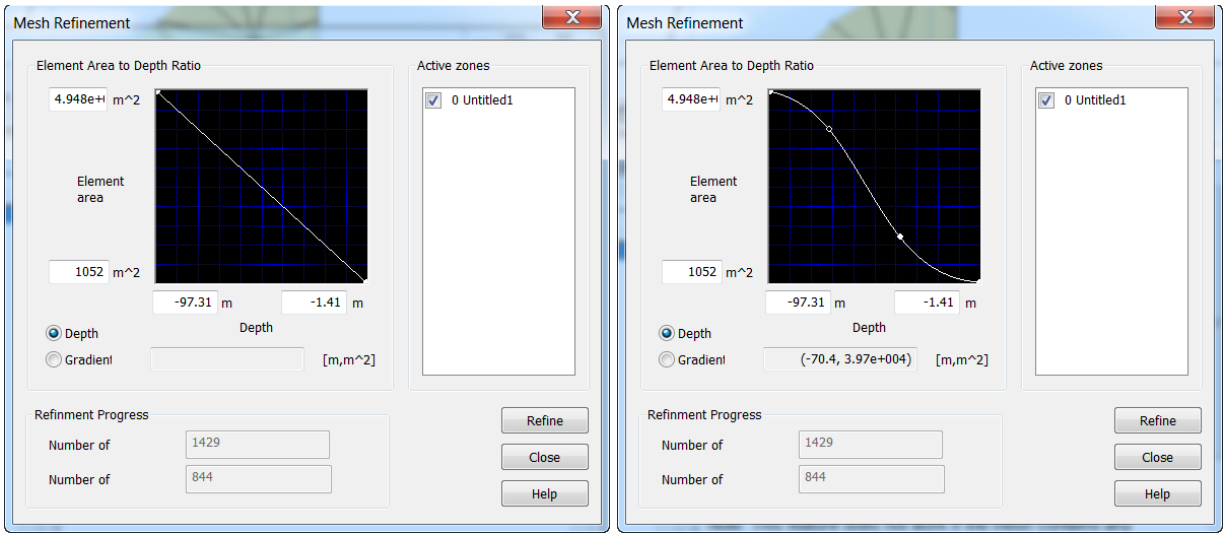


Figure 14 Example mesh refinement tool wherein the linear variation (left) of element area against terrain level in a zone could be modified (right) to promote smaller element sizes in higher-elevation areas. (Source: DHI, 2014)

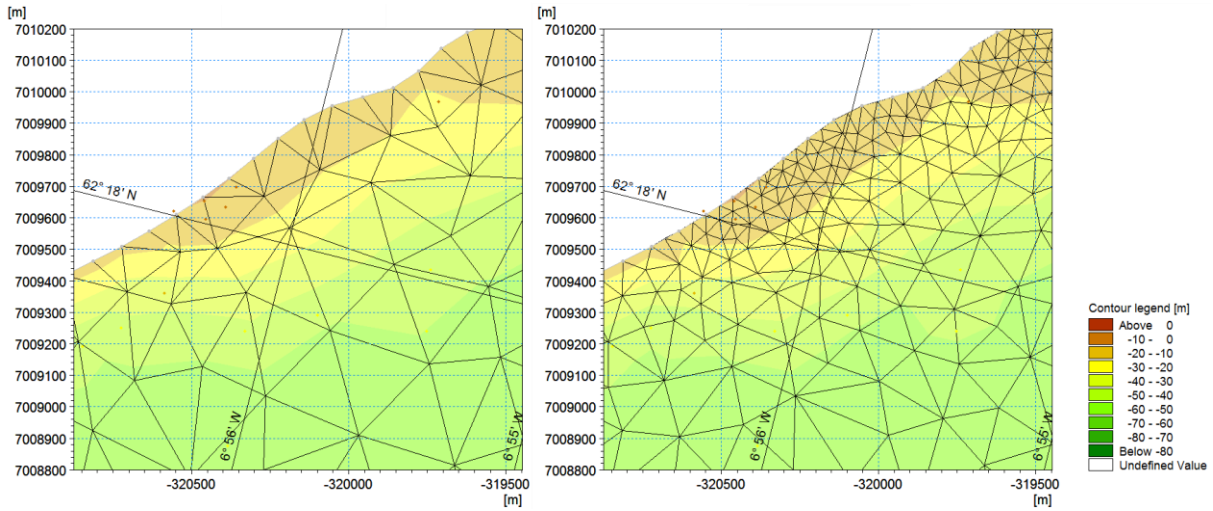


Figure 15 Mesh before (left) and after (right) mesh refinement according to terrain levels, wherein areas with higher terrain levels are represented by smaller elements.

2.1.4.2 Mesh elements arrangement

The use of flexible meshes allows for more efficient geometric definition of surface features and complex flow domains, such as in urban areas, in turn reducing simulation times without compromising numerical accuracy. To illustrate, Figure 16 compares flexible mesh and Cartesian grid representations of topography in an urban coastal area. The figure only shows part of the full meshes, but both have, in total, approximately 300 000 elements. The Cartesian grid has a uniform cell size of 49 m^2, while the flexible mesh has elements ranging in size from 5 000 m^2 (in the sea areas) to 1 m^2 (in the built-up areas). The figure shows how flexibility with the unstructured mesh can be used to adjust the level of computational detail in the model domain according to expected flow conditions, such that finer computational detail is achieved in built-up areas that have complex flow paths and are expected to be inundated, compared to rural or sea areas that have more uniform/even terrain. Moreover, surface structures (e.g. buildings) with intricate shapes could be better represented in a flexible mesh than in a rigid Cartesian grid.

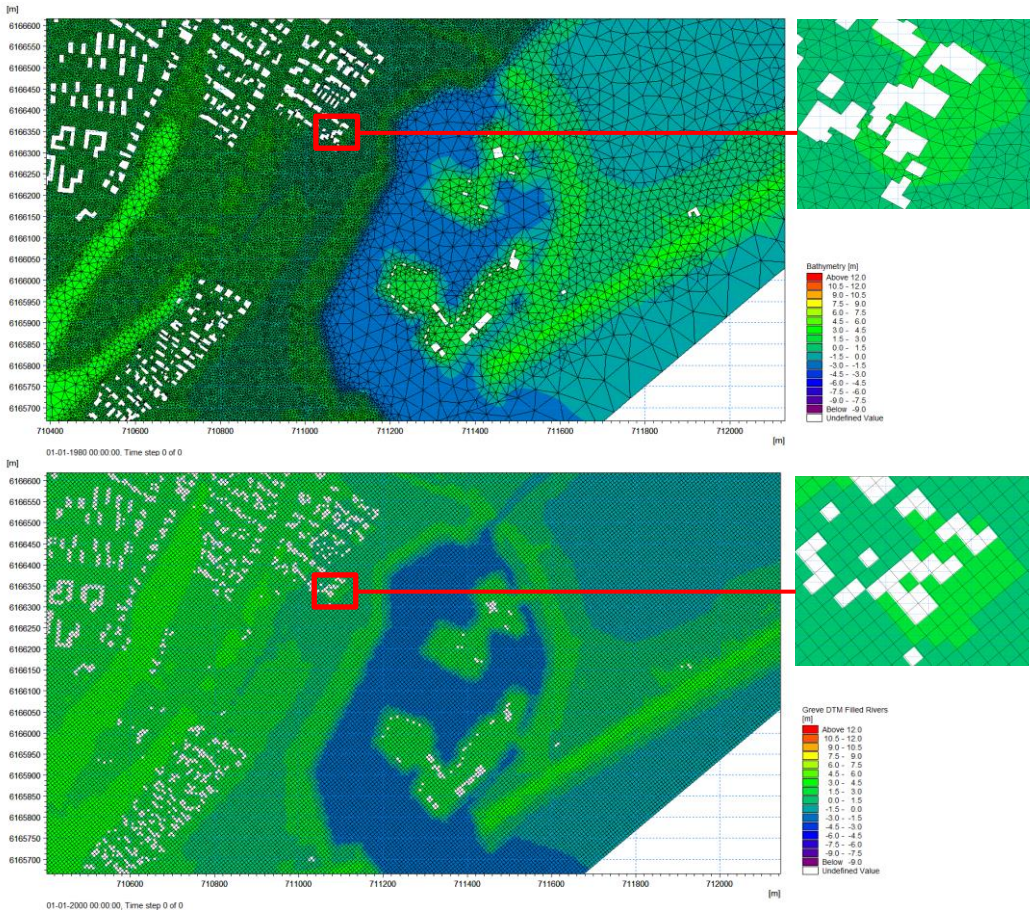


Figure 16 Complex urban coastal topography represented with flexible computational mesh (top) and Cartesian grid (bottom). The full flexible mesh has 307 890 elements, with fine elements less than 25 m² in size, while the full Cartesian grid has 332 775 cells, each with an area of 49 m² (i.e. 7 x 7 m).

2.1.4.3 Considering surface structures

Further reduction in the number of 2D computation points is afforded by the removal of areas from the model grid or mesh wherein calculations of water levels and discharges are deemed unnecessary, such as in areas assumed impervious to overland flows. Examples of these are (impervious) buildings in the model domain (Figure 17). Removal of these structures from the grid or mesh effectively reduces the number of computational points in the model, at the same time accounting for their presence in the computations, such that flows are computed to occur around and not through these features.

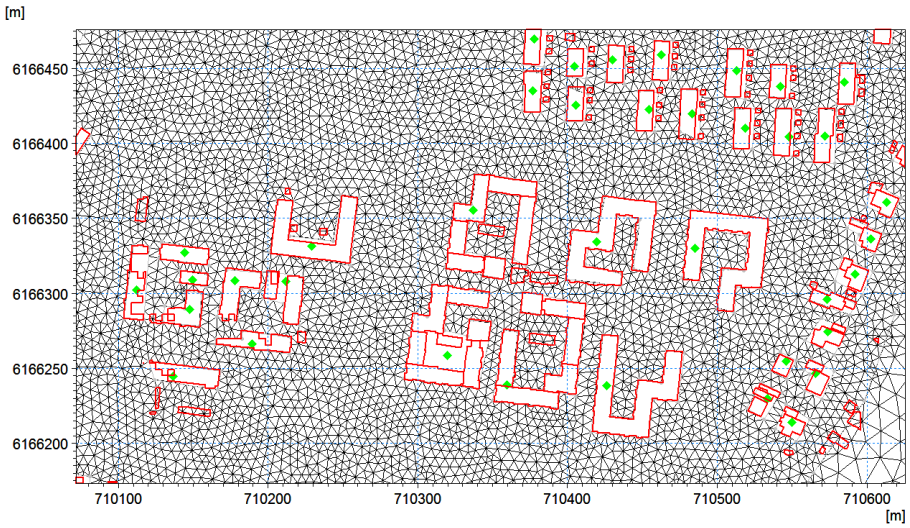


Figure 17 Illustration of how surface structures, such as buildings, are removed from a 2D computational mesh.

Alternatively, the presence of these structures may be disregarded in the grid, and elevations over these areas assumed to be at terrain level, as in cases where their ability to impede or affect overland flows is considered minimal. In this case, computed inundation around these structures (e.g. along the streets) is potentially lower than in reality, and the timing of flood propagation faster due to less impedance to flow. Surface roughness may also be modified (i.e. increased) over these areas for more realistic modelling of flood propagation (Figure 18 (b)). These surface structures may be reflected in the computational grid or mesh, wherein their actual elevations are used over the areas they occupy. However, this technique creates localised areas of high elevation and steep slopes (i.e. pyramids) (Figure 18 (c)), which are susceptible to high velocities and instabilities in 2D model computations.

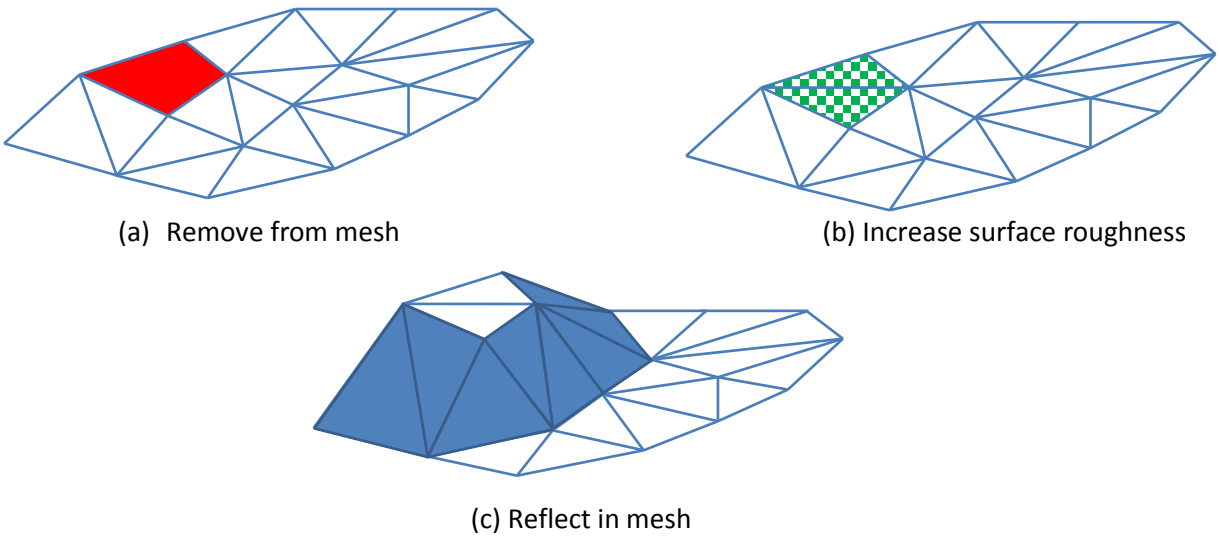


Figure 18 Illustrations of how surface structures may be considered in the 2D computational mesh: (a) Area is removed from the mesh, (b) Surface roughness increased over the area, or (c) Area is reflected in the mesh. Figure (c) illustrates how areas of steep slopes are generated when elevations of surface features are reflected in the mesh.

2.2 Maximising Computational Power

Faster model simulations may be achieved through enabling simultaneous calculations on multiple processors (i.e. parallelisation), as well as use of different types of processing units for performing calculations. Neal et al. (2010) summarizes that parallelisation has been proven to speed up various flood model simulations (Rao, 2005; Villanueva and Wright, 2006; Neal et al., 2009; Lamb et al., 2009), and that the general techniques involved include:

- Message-passing techniques (MPIF, 2012)
- Shared memory approaches, such as OpenMP (OARB, 2015; Chapman et al., 2008)
- Use of Graphics Processor Units (GPUs)

These solutions maximise the potential of recent computer hardware developments, wherein more computational cores are progressively offered (i.e. multi-core computers) and graphics cards are made to perform computational intensive calculations.

2.2.1 *Shared memory approach*

The shared memory approach divides computational tasks on a number of cores (Chapman et al., 2008). Some flood modelling systems, such as MIKE 21 FM (DHI, 2014) and LISFLOOD-FP (Bates and De Roo, 2000; Neal et al., 2010), have implemented parallelisation for shared-memory multicore computers based on Open MP (Open Multi-Processing) (OARB, 2015). The user specifies the number of threads, which should not exceed the number of cores available on the machine, and the OpenMP computing approach runs the simulation in a number of threads.

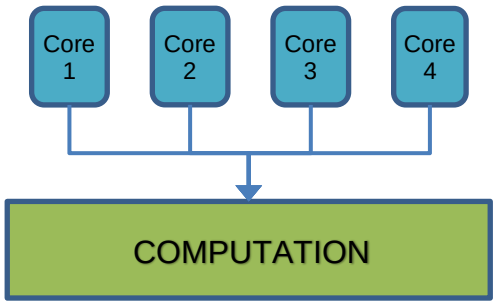


Figure 19 Illustration of shared memory approach to parallelisation.

2.2.2 *Distributed memory approach*

Some hydraulic models, including MIKE 21 FM (DHI, 2014) and TELEMAC-2D (Hervouet, 2000), implement parallelisation in distributed memory systems by subdividing the domain, distributing the work among processors, and using a message passing paradigm for communication between the processors. Domain decomposition and message passing interface (MPI) are used to run simulations in parallel.

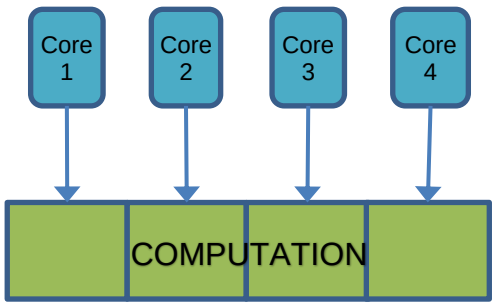


Figure 20 Illustration of distributed memory approach involving domain decomposition.

In MIKE 21 FM, the computational mesh is partitioned into a number of physical subdomains, and the work associated with each subdomain is processed by an individual processor. The data exchange between processors is based on the halo-layer approach with overlapping elements. Intermediate output are generated for each subdomain, which are eventually merged into one at the end of the simulation (DHI, 2014).

2.2.3 GPU computing

Instead of parallelisation focusing on the use of just general-purpose CPU cores, other processing units, such as Graphics Processing Units (GPUs), may be used in conjunction with CPUs to perform computationally intensive calculations. Examples are InfoWorks ICM (Innovyze, 2013), JFLOW-GPU (Lamb et al., 2009), and GPU-DASH (Smith et al., 2013), which use GPUs for parallelisation and reduction of computation time.

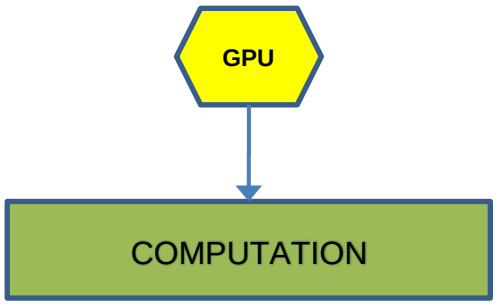


Figure 21 This approach involves the use of GPUs as co-processors in performing calculations.

The co-processor may have a more limited functionality compared to CPUs, and thus its design and use can be optimized for executing more specific tasks, such as performing floating-point calculations (Neal et al., 2010).

A benchmarking study on the use of GPU computing for modelling put forward some guidelines with respect to hardware type and installation, and model setup (DHI, 2016). One recommendation in the study is to ensure the GPU is appropriate for scientific computations, having a high number of Streaming Multiprocessors and cores (shading units), and high theoretical double precision floating-point performance (i.e. FLOPS (FLoating-point Operations Per Second) rate). In addition, some notes on the installation of the GPU include use of the fastest PCI (Peripheral Component Interconnect) Express slot available to avoid data-transfer bottlenecks, as well as allowing enough room around the GPU, or using of an extra cooling system, to avoid overheating. Finally, it was noted that optimum performance with GPU parallelisation is best achieved when the number of active (i.e. wet) elements is high in the simulation. For example, in the benchmarking test (DHI (2014), maximum performance in simulating a 2D mesh model with a GeForce GTX Titan GPU was reached only after having around 400 000 or more wet elements for each.

Some GPUs have significantly higher processing power when performing single-precision floating-point calculations, but double-precision floating-point calculations are more accurate, which affect modelling results. MIKE 21 FM offers the use of GPU parallelisation for both single- and double-precision floating-point calculations (DHI, 2014).

2.2.4 *Hybrid computing*

For efficient simulation in science and technology, fast numerical methods are of crucial importance. Graphic processing units (GPUs) have emerged during the last years, representing an important tool for increasing the performance of simulations (Lang and Rünger, 2013).
A GPU as a device can be understood as a combination of a very large collection of small and relative simple processing units and a control unit, having its own GPU memory. The large number of processing units can execute a huge amount of similar operations simultaneously, representing the most important advantage of this specific architecture: the ability of massive parallelism (NVIDIA, n.d.).

GPUs are produced by a number of manufacturers e.g. NVIDIA, which is one of the leading manufacturer for GPUs used in the field of science and technology. Furthermore, NVIDIA has invented and developed a parallel computing platform and programming model, named CUDA®, which is most widely used in the field of science and technology.

Based on the market leadership and the comprehensive experience of NVIDIA on this specific field, it was decided to use NVIDIA hardware and CUDA® technology for the development of the TUHH Kalypso Evacuation Module (see Chapter 3.3).

In the following, basics of GPU computing is described including advantages and disadvantages, leading to the approach of hybrid CPU/GPU computing.

GPU computing is the accelerated computing due to the additional use of a graphics processing units. Software, which is intended to be used for GPU computing, consists of software parts, which are executed on GPU (device) called “kernels”, in addition to software parts, running on CPU (host). The host software code overtakes the responsibility for writing the data to the device, calling the device-kernel functions and as well reading the data from the device. The kernels are sets of instructions, which are executed in parallel on all data at the same time. The parallel running processes are called threads. The threads are grouped to the thread blocks, building one grid. A kernel is executed as a grid of blocks of threads (NVIDIA, n.d.). Due to different execution times of the threads and due to execution parallelism, those threads are synchronized in case of specific barriers are reached. Conditional statements (if-statements) lead to significant increased execution times of the threads, representing a bottleneck of GPU computing architecture. Additionally, the data exchange between the host and device is very slow, compared to the calculation execution time on the device.

GPU devices have their own memories, which can be divided into two symbolic subgroups: fast shared memory (local device memory) and relative slow global memory with its components (i) constant, (ii) texture and (iii) general purpose. The host accesses only the global device memory in order to exchange data with the device. The fast shared memory can only being accessed by the device-kernels, representing an effective “working memory”. An essential limitation results from the significantly smaller size of the fast shared local memory.

The concept of hybrid CPU/GPU computing aims at the optimal utilization of the advantages of both technologies and minimization of the effects of the bottlenecks. CPUs are optimal for a low response time that means CPUs execute one task as fast as possible. The GPU, instead, is optimized for a large throughput, which means the execution of as many tasks as possible in a fixed time. Additionally, tasks with a huge amount of control flow decreases the performance of the GPU, but can be handled well by CPUs (Breß et al., 2012). The utilization of hybrid CPU/GPU computing results in increased computing power of approx. 50% to 60% (Pydiura et al., 2014), respectively an increase of the computing power by the factor 2 to 3 (Yu et al., 2011, Lu et al., 2012). Of course, the degree of computing power boost largely depends on the specific application

and hardware resources. Using latest hardware resources (e.g. latest NVIDIA graphic cards) a boost of computing boost by 200% to 300% can be gained (Pydiura et al., 2014, NVIDIA, n.d.).

Finding the right balance between workload on GPU and CPU was the essential fundament for the development of the TUHH Kalypso Evacuation Module

An example of a mesh-based software system that has a hybrid parallelisation option is MIKE 21 FM (DHI, 2014) used in the Greve Case Study (Chapter 3.1). It has the following options, which may be used in hybrid:

- Shared memory approach (OpenMP)
- Distributed memory approach (MPI)
- GPU computing approach (GPU)

The number of subdomains (i.e. MPI), and threads per subdomain (i.e. OpenMP) for which the simulation should be run may be specified. In the MPI computing approach, the model domain is divided into a number of smaller subdomains, dividing the computational work among a number of processors. The number of threads per subdomain then specifies the number of OpenMP threads to use for each subdomain, hence, a hybrid OpenMP + MPI parallelization technique may be used. In addition, it is possible to use the GPU computing approach if the computer has a supported graphics card. If more than one graphics card is available, the simulation may be run using multiple GPUs using a hybrid OpenMP + MPI + GPU parallelization technique. When using hybrid parallelisation, the specified number of subdomains or the number of threads per subdomain (or the product of these) should not exceed the number of cores available on the PC, since this will decrease performance.

3 Case Examples

3.1 Greve, Denmark

In the PEARL Project, enhancement of modelling tools and techniques for flood forecasting and early warning (Work Package 4) entailed application and testing of these developed methods in case study areas. Greve, in Eastern Denmark, is one of these case areas, and an urban coastal flood forecasting and warning system has been developed for the municipality as part of the project (Figure 22).

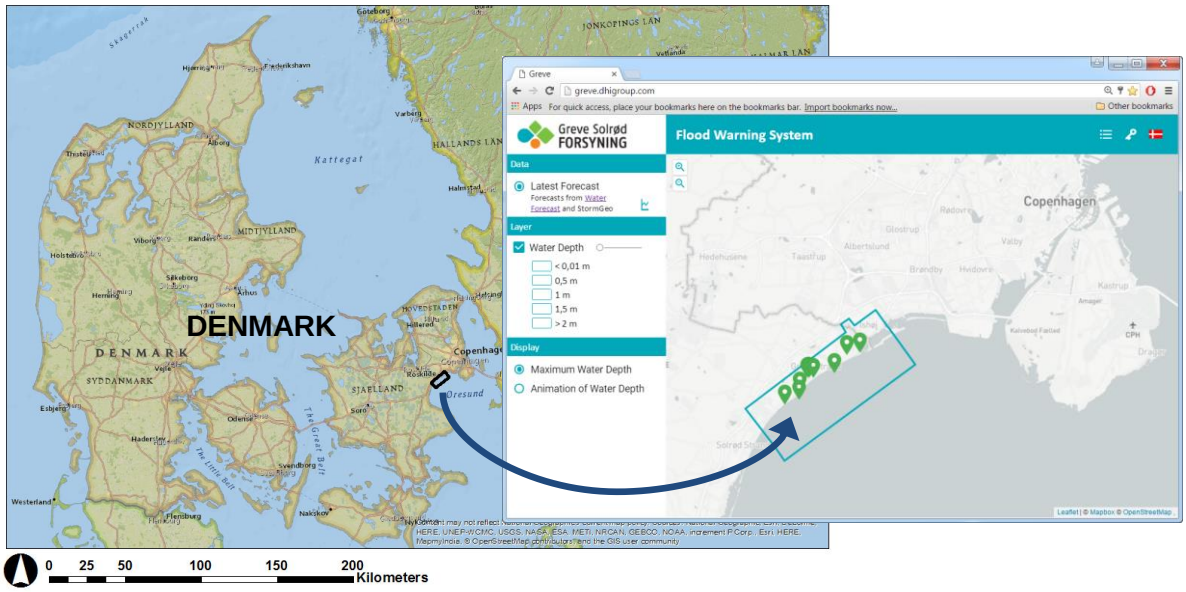


Figure 22 Homepage (www.greve.dhigroup.com) for the online coastal flood warning system in Greve, Eastern Denmark (Right). It displays the most recently calculated maximum water depths. Pre-determined critical points (see place markers) are colour-coded from green to red depending on the magnitude of forecasted flooding.

The Greve Coastal Flood Warning System employs a coupled 1D-2D hydrodynamic model for flood forecasting, which uses NWP rainfall forecasts and 3D hydrodynamic model water level forecasts as input (Figure 23). The detailed 1D network model, built with MIKE Urban, consists of around 6 000 sub-catchments, 7 000 nodes, 6 500 links and 9 outlets to the sea. The 2D model component, was built using MIKE 21 FM, a 2D modelling system based on a flexible mesh approach. The 2D model covers 53 km2 of Greve's coastal area comprising of both sea and inland regions, and its mesh is comprised of 49 000 elements varying in size from 9 m2 to 20 000 m2. Small mesh elements were used for built-up areas, roadways and flood plains, while bigger elements were used for rural and sea areas.

Forecasts for the next 24 hours are made 6 times a day every 4 hours at 00:00, 04:00, 08:00, 12:00, 16:00, 20:00 CET. Each 1-day forecast takes around 2 hours computation time for the coastal flood model.

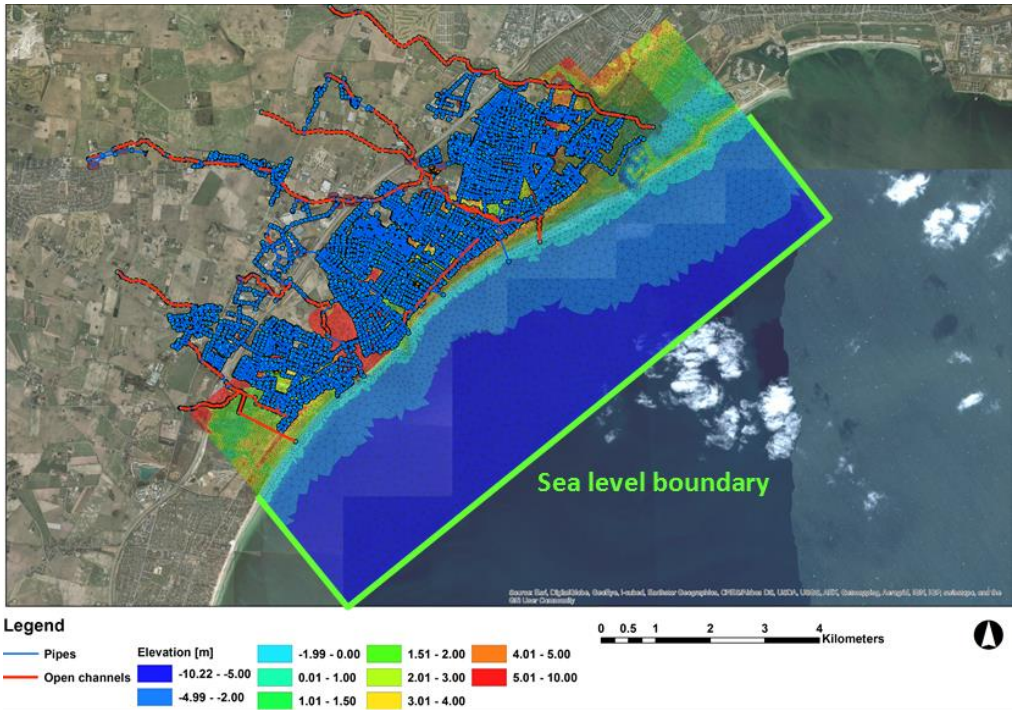


Figure 23 A plot of the 1D-2D coastal flood forecast model for Greve. A 1D model of the drainage system, comprising of streams (red lines) and sewers (blue lines and points), is linked to a 2D model of the coast (coloured areas). The triangular mesh elements and open boundaries of the 2D model are also shown.

An urban coastal flood model that performs fast enough for warning purposes without compromising result accuracy was needed in the case study. The current operational model in Greve is a result of several iterations, especially of its 2D component, applying techniques for improvement of model accuracy and computation speed. These techniques involve optimization of computational loads and machine computational power through model modification. Moreover, additional speed-up techniques involving hardware capabilities were tested and evaluated using the Greve flood forecast model. These are described in the succeeding sections.

3.1.1 Speed-up technique(s)

The following model speed-up techniques were tested and evaluated with the Greve urban coastal flood model:

1) Modification of computational grid type
2) Optimisation of computational grid size
3) Parallelisation of computations
4) Use of Graphics Processing Unit (GPU) computing

The first 2 techniques relate to computational optimization, while the last 2 concern utilisation of hardware capabilities.

Modification of computational grid type

An early version of the coupled 1D-2D urban coastal flood model for Greve used a Cartesian computational grid for its 2D model component (Sto. Domingo, et al., 2010) (See Figure 24). The 2D surface flow model used a 5m x 5m computational grid, offering a balance between model accuracy, in terms of describing main surface flow paths, and computational times.

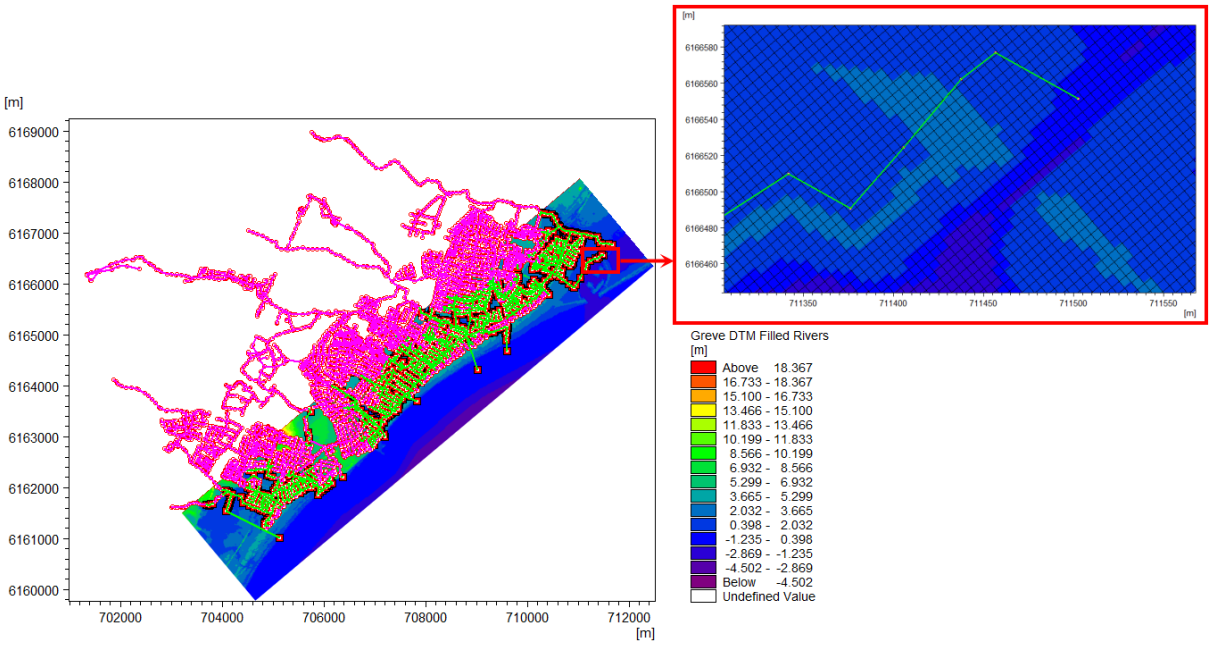


Figure 24 Early version of the 1D-2D urban coastal flood model for Greve, Denmark, using a Cartesian 2D computational grid (top right).

Coupled 1D-2D models using Cartesian grids simulate flooding from the sea and flow interactions with inland drainage systems with high resolution and reliability (Schmitt et al., 2004; Sto. Domingo et al., 2010; Hénonin et al., 2013; Leitão et al., 2013). However, these models are computationally expensive and may be too slow for real-time simulations (Yu & Lane, 2006; Chen et al., 2012a; Chen et al., 2012b). The number of computational points can be large, due to the uniform 2D grid size, leading to long computation times that limit their viability for real-time flood warning purposes.

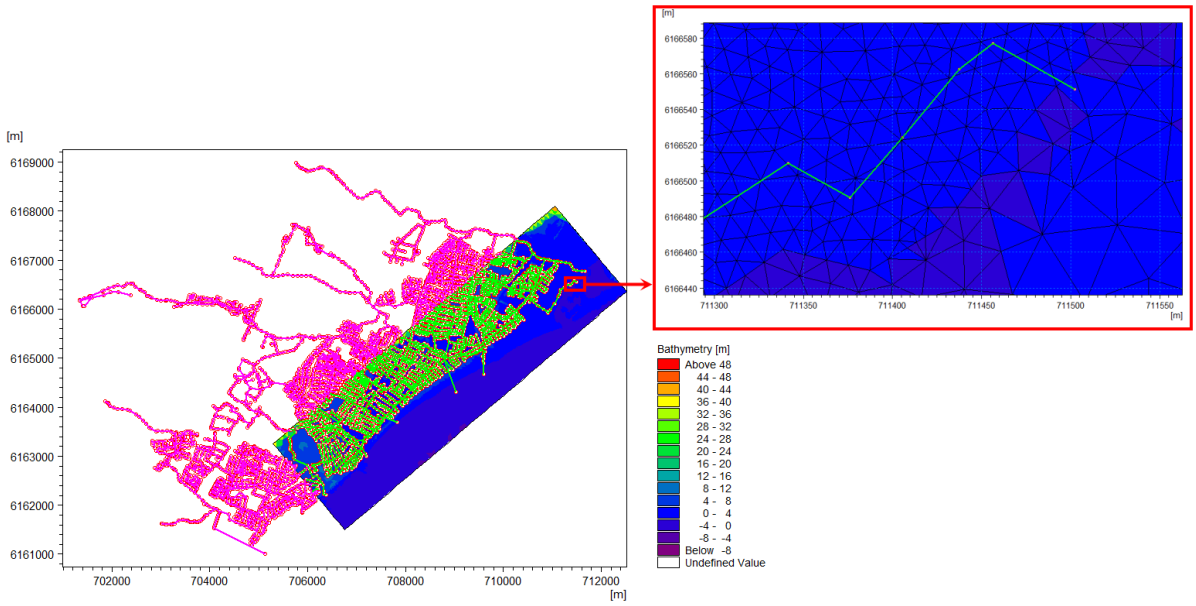


Figure 25 Modified 1D-2D urban coastal flood model for Greve, Denmark, using a mesh 2D computational grid (top right) and modified 2D model extents.

The use of flexible meshes can reduce computation times for 2D hydrodynamic models in comparison with conventional Cartesian models without compromising numerical accuracy (Sleigh et al., 1998; Namin et al., 2004). This technique was tested in Greve by building a flexible mesh-based 2D model for the area, and comparing its performance, in terms of computation time, against the grid-based model (Figure 25). The comparison tests were conducted using only 2D models instead of integrated 1D-2D models, as the computational grid being modified in the tests is mainly a component of the 2D model. The two versions of the 2D model for Greve used in the performance comparisons were:

1. Setup 1: 2D model using a Cartesian computational grid, built with MIKE 21
2. Setup 2: 2D model using a flexible mesh computational grid, built with MIKE 21 FM

Hardware and model parameters (besides the computational grid) were kept the same in the comparisons. Note that the 2D model extents for the test setups were reduced and optimized based on flood vulnerability Figure 26.

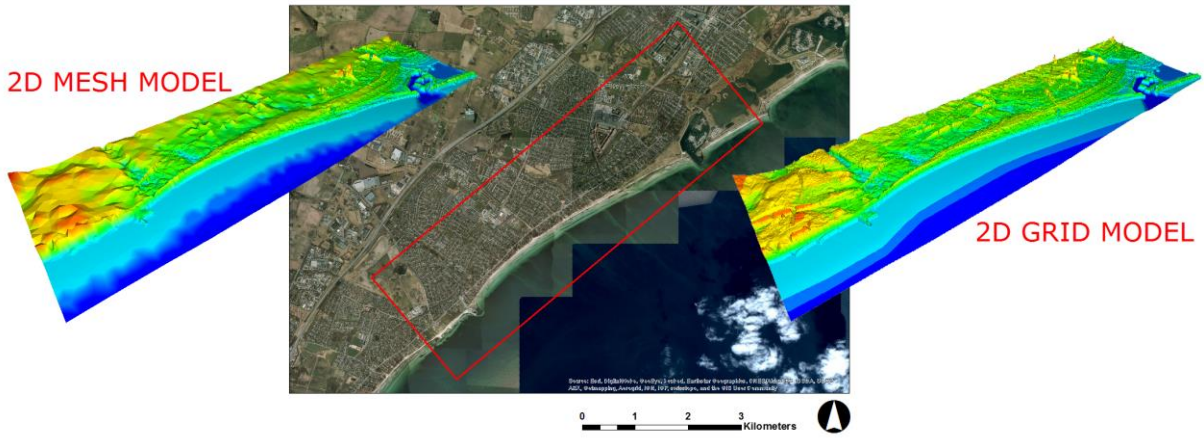


Figure 26 The centre figure shows the extents of the 2D models that were compared. The mesh surface model (Setup 2) is shown on the left, and the Cartesian grid surface model (Setup 1) is shown on the right.

The 2D models were forced with extreme sea level conditions that were obtained from a previous climate modelling study. In Sto. Domingo et al. (2010), extreme sea level time series under climate change conditions in Greve were derived based on direct application of change factors (for storm surge heights) to selected records of observed extreme water level events (See Figure 27). The expected changes in sea surges in the Danish waters were estimated through hydrodynamic simulations with a model covering the North Sea, Baltic Sea and inner Danish waters (RiskChange, 2015). In addition, local estimates of future mean sea level variation in the Danish waters were used (Edelvang, et al., 2012). More details about the derivation is found in Sto. Domingo et al. (2010).

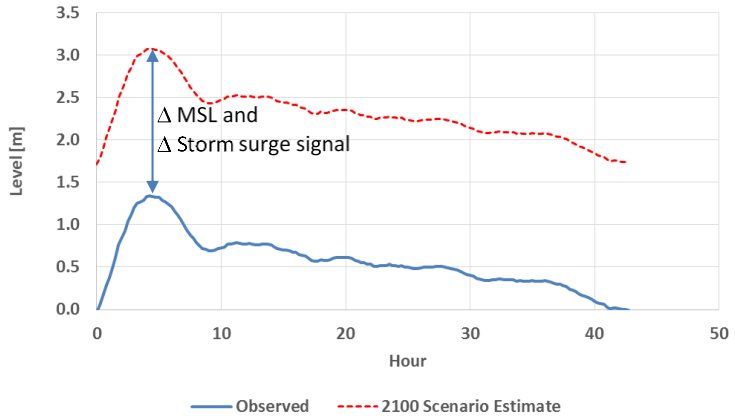


Figure 27 Future extreme water level event time series (red dotted line) estimated based on an observed extreme event pattern (blue solid line). The future event time series is derived by scaling to a given return period and adding estimates of mean sea level rise and change in storm surge signal.

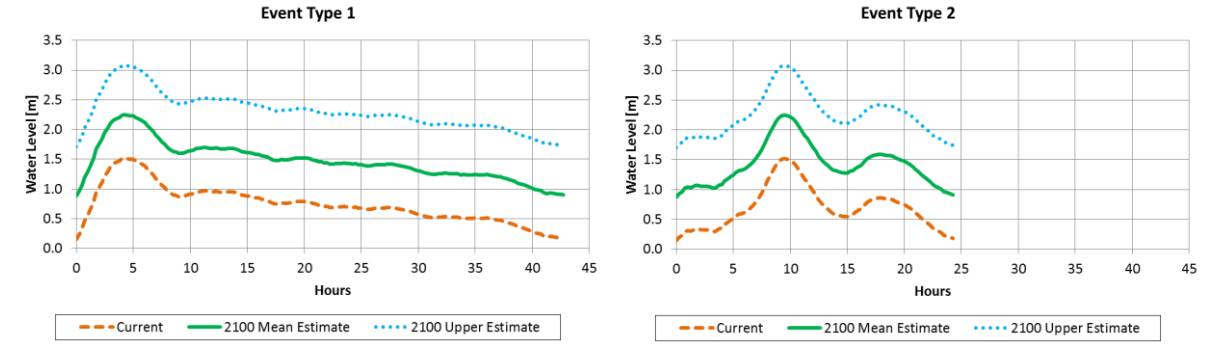


Figure 28 Two sets of 100-year return period future extreme sea level time series used as boundary conditions to the 2D coastal flood models.

Two sets of future extreme water level time series were used in the comparisons (Figure 28). Event Type 1 has a duration of 42.5 h, while Event Type 2, 24 h. For a 100-year return period, the maximum water levels are 1.52 m, 2.2.5 m, and 3.08 m for Current, Future Mean Estimate, and Future Upper Estimate Scenarios, respectively.

The grid-based model was considered the reference setup, and thus, the mesh was built ensuring results with the mesh-based 2D model remained consistent (i.e. very similar) to those of the reference model. This process involved mesh optimization, which is discussed in the following section as another model speed-up technique related to computational optimization.

Optimisation of computational grid size

The capacity to describe highly-complex geometries and surface terrain is a strong advantage of flexible meshes, as they can be locally refined in specific areas according to modelling needs (Walters et al., 2009; Danilov, 2013; Liu et al., 2013). Meshes are especially efficient in modelling coastal areas, as they can easily adapt to irregular coastlines (Shen et al., 2006). However, the generation of an optimum mesh satisfying time-efficient performance and geometrical requirements can be complex and time consuming (Sleigh et al., 1998), especially in urban model domains.

Mesh size and configuration (i.e. alignment) should be well-considered over regions, such as expected flooded areas and main surface flow paths, to ensure accurate simulation of surface flows with the resulting 2D model.

Specification of controlling boundaries during mesh generation identifies areas within the domain for which smaller mesh elements, or differently-shaped elements (e.g. quadrangular instead of triangular), should be generated. As much as possible, these controlling boundaries should be based on consistent geo-spatial data to allow for systematic generation of the mesh. Examples of such geo-spatial data include GIS vector data for:

- (Expected) Flood extents i.e. flood-prone areas
- Surface structures influencing surface flows, such as roads, buildings, etc.
- Hydrologic boundaries, such as sub-catchment boundaries and coastlines

In this study, information on flood-prone areas was obtained from previous flood analysis results—from the Cartesian (i.e. grid) model, and from GIS Terrain Analysis. Knowledge of the location of flood prone areas is important in intelligently building an efficient mesh.

An unstructured mesh with mixed triangular and quadrangular elements was developed for Greve to allow better conformation to boundaries, local refinement, and improved mesh accuracy (Figure 29). Train track areas were described with quadrangular elements, and varying spatial resolutions were specified over the domain. High mesh resolutions were defined in flood-prone areas, which were previously identified from Cartesian model results. Coarser resolutions were used over the remaining sub-regions of the domain.

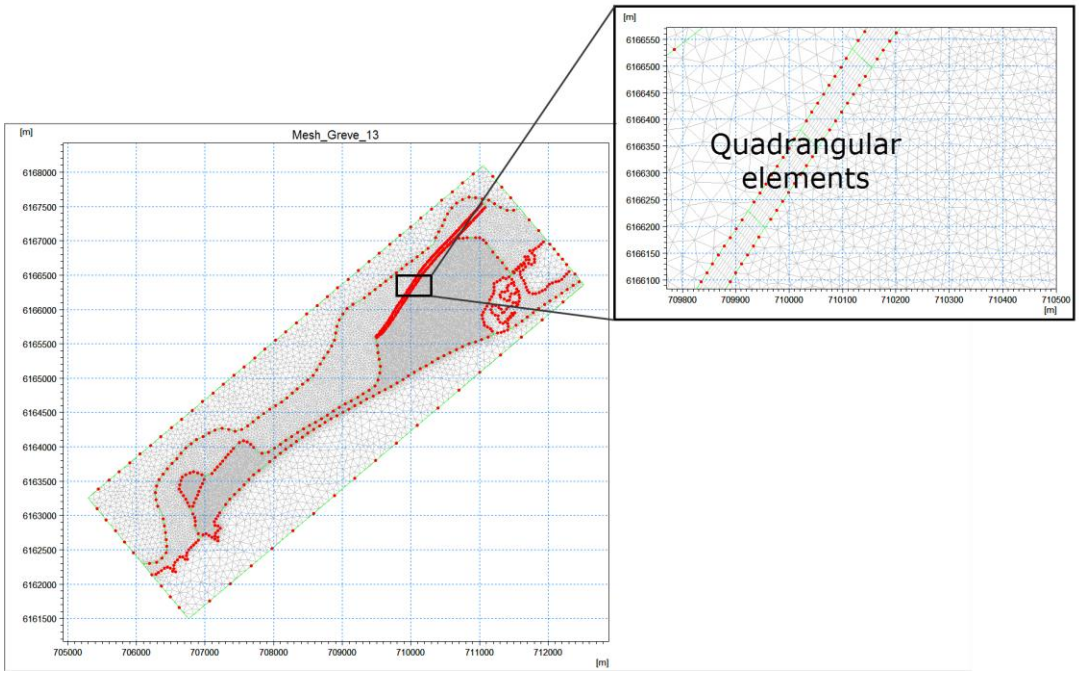


Figure 29 Mixed unstructured mesh for Greve combining triangular elements and rectangular elements (inset). The area was subdivided into regions of different maximum element sizes.

Obtaining information on flood-prone area extents from a Cartesian 2D model can be inefficient due to model computation times. A new and faster technique is to use inundation maps from GIS Terrain Analysis. This method calculates potential flooding through raster analysis subtracting the DTM (Digital Terrain Model) from an expected maximum water level value (see Figure 7). This method is especially applicable in coastal areas.

Terrain Analysis tends to overestimate flood extents, and thus, this method ensures flood-prone areas are well-covered in the estimate. Figure 30 illustrates the use of flood-prone area extent information as boundary data in mesh generation.

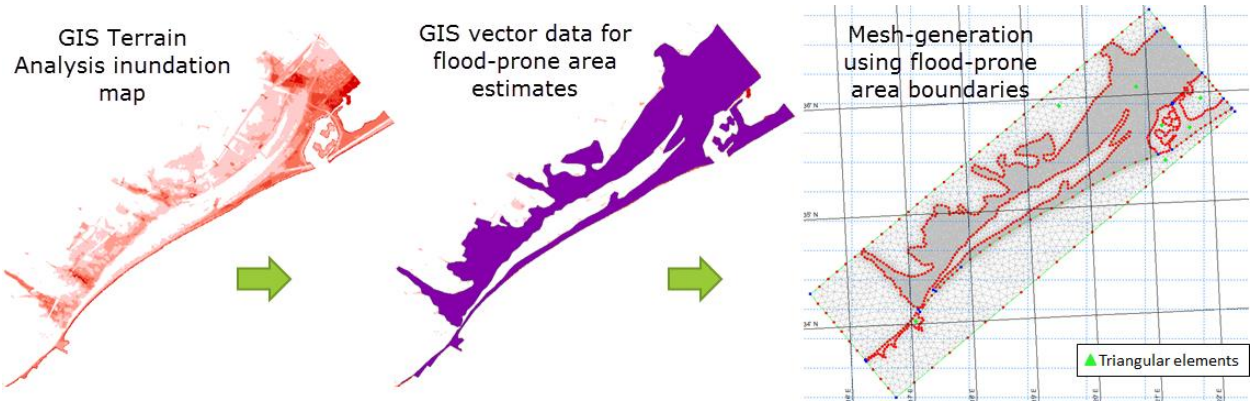


Figure 30 Flexible mesh-generation using flood-prone area extent information from Terrain Analysis.

Mesh optimisation was conducted in an iterative manner to achieve the desired quality and accuracy in the modelling results. Results from the Cartesian mesh model from previous published studies for the same study area (Sto. Domingo et al., 2010) were used to validate results obtained with flexible mesh model, and determine if the calculation mesh needs further refinement. Computation speed was also a consideration in mesh refinement, and the mesh was modified until simulation times were fast enough for warning applications.

The MIKE 21 FM model for Greve used a variable computation time step ranging from 0.01 to 1 s, keeping the critical CFL (Courant-Friedrich-Lévy) number at 0.8 (less than 1). Initial sea level conditions were defined as a uniform value based on the estimated Mean Sea Level (MSL) for the scenario of interest (e.g. consider sea level rise for climate change scenario). Finally, the 100-year sea level time series was applied along the sea boundary of the 2D model.

Maximising computational power

In this study, the use of parallelization and GPU technology were tested and evaluated in terms of reducing computation times and maintaining numerical model accuracy. Three different model configurations were used in the tests:

1. Setup 1: 2D model using a Cartesian computational grid, built with MIKE 21
2. Setup 2: 2D model using a flexible mesh computational grid, built with MIKE 21 FM
3. Setup 3: Coupled 1D-2D model using a computational mesh, built with MIKE Flood FM

Model calculations were performed on two computers:

- Computer 1: Intel Core i7-2620M with a 2.70 GHz processor, 8GB of RAM, and Windows 7 operating system
- Computer 2: Intel Xeon CPU E5-2687W with a 3.10 GHz processor, 16GB of RAM, and Windows 7 operating system

In this study, only the Open MP parallelization, and GPU computing options were tested and compared. The three model configurations were forced with the same 100-yr extreme water level event (Type 1) for the worst climate change scenario projection (Figure 28), and the same computational mesh was used in the 2D model components of Setups 2 and 3.

3.1.2 *Results and discussion*

In the modification and optimization of the computational grid, comparisons between the Cartesian grid and flexible mesh models revealed interesting results with respect to computational speed. Table 3 shows that the mesh model is almost 4 times faster than the Cartesian model. These results indicate significant potential improvement in the performance with a shift from grid-based to mesh-based flood models. Moreover, the results indicate that in terms of flood extension and water depths, the mesh model agrees well with results from the previous Cartesian model for the same study area (Sto. Domingo et al., 2010), thus ensuring model consistency with the new method. However, in this study, it is not clear if the improved computation speed is because the mesh is well-optimized, or because the number of mesh elements is much smaller than in the Cartesian grid.

Table 3 Comparison between the 2D Cartesian grid and mesh modelling approaches for the Greve case area in terms of number of computational elements, inundated area, and computation time.

	Setup 1: Cartesian Grid 2D Model	**Setup 2: Mesh 2D Model**	**Speed-up Factor (Setup 1/Setup 2)**
Number of elements	924 550	38 530	-
Inundated Area [km^2]	Event Type 1 Current: 0.638 Mean: 1.508 Upper: 3.718	Event Type 1 Current: 0.779 Mean: 1.858 Upper: 3.754	-
	Event Type 2 Current: 0.532 Mean: 1.499 Upper: 3.545	Event Type 2 Current: 0.831 Mean: 1.831 Upper: 3.643	-
Computation Time [h]	Event Type 1: ~35.8	Event Type 1: ~9.6	**~3.73**
	Event Type 2: ~13	Event Type 2: ~4.5	**~2.89**

The results of tests regarding maximisation of computational power are shown in Table 4. Model results with OpenMP and GPU computing, in terms of water depths and flood extents, were exactly the same, which is as expected since the numerical scheme is the same in both GPU and OpenMP parallelization options in Mike 21 FM.

Table 4 Comparison of computation times for the 3 model setups in the study. Parallelisation and GPU computing options were available for models with the 2D mesh components (Setups 2 and 3), but not for the Cartesian grid-based model (Setup 1).

	Computation time [h]			**Speed-up Factor (Computer 2 OpenMP/GPU)**
	Computer 1 OpenMP 4 processors	**Computer 2 OpenMP 8 processors**	**Computer 2 GPU 8 processors**	
Setup 1 Cartesian Grid 2D Model	35.8	11	-	
Setup 2 Mesh 2D Model	9.6	2.6	0.76	**3.42**
Setup 3 1D-2D Sewer-2D Mesh Model	12.2	4.9	3	**1.63**

Table 4 shows that with 2 times the number of processors, parallelisation, which allows use of multiple processors, affords a speed-up factor of around 4 for the 2D mesh model (Setup 2), and

2.5 for the 1D-2D model (Setup 3). GPU computing further reduced simulation time by more than thrice for the 2D mesh model (Setup 2). Overall, a **total speed-up factor of 47 was obtained** for the 2D model through combined modification of the calculation grid, parallelization, and GPU computing, illustrating that these techniques are effective in improving model runtimes, which is advantageous for real-time forecasting applications.

Moreover, model runtimes are also influenced by the change in the number of computation points due to 'wetting and drying', especially in inland models. Therefore, higher speed-up factors could be further achieved when simulating rapid inundations, such as flash floods. The use of GPU computing significantly reduced model runtimes for the Greve case study model, simulating 2D overland flooding for a 42.5-hour event in 0.76 instead of 11 hours. Although calculation times with coupled 1D-2D models (considering the sewer system) were longer than for 2D models, they were nevertheless fast enough for warning purposes, and with the advantage of 1D-2D models more accurately considering the influence of the sewer system in propagating flooding further inland.

3.2 Marbella, Spain

Within the PEARL Project Work Package 4 (Enhancement of modelling tools and techniques for flood forecasting and early warning) an Early Warning System (EWS) based on radar and 1D-2D hydrological model has been developed and implemented in the municipality of Marbella (Spain) pilot site.

The 1D-2D Marbella model was built using InfoWorks ICM (Innovyze, 2013), and uses both measurements of a rain gauge network and radar precipitation observations and forecasting as precipitation input information.

The model has been developed in order to assess Hazard in Marbella case study. The model covers 10.6 km^2 of the municipality land involving 68 km of sewers. A 2D unstructured mesh with more than 60 thousand cells, which was generated by the Shewchuk triangle meshing functionality (InfoWorks ICM, 2016) of the software, was created on the basis of a detailed digital terrain model (DTM). For this study, a specific 2 m^2 resolution DTM model has been used. This DTM was generated by a LIDAR provided by the National Geographic institute of Spain. More details of the model can be found in PEARL Deliverable 6.2.

The radar network provides new precipitation information (and therefore a new radar now-cast for the next 2h) every 10 minutes. The use of speed-up techniques for faster simulations, like the use of GPU processing, allowed the real-time modelling with the observed and forecasted precipitation every time that new data is available. The radar-based EWS, including the link of the radar precipitation estimation and forecasting with the hydraulic model, will be described in the PEARL Deliverable 4.5.

3.2.1 Speed-up technique(s)

In the Marbella pilot site, the two techniques that have been used to reduce computational time have been the modification of the computational grid type and optimisation of computational grid size; and the use of Graphics Processing Unit (GPU) computing. For instance, mesh zones has been implemented permitting bigger cells sizes in areas where no detail is required. Moreover, optimized buildings have been removed from the mesh in order to avoid undesired and unnecessary cell, which are important computational time consumers.

Modification of computational grid type and optimisation of computational grid size

The grid characteristics and size has been modified to speed-up the computational time without compromising the results using the build in tools of the InfoWorks ICM suite.

Maximising computational power

In order to achieve the short computational time needed to perform a real-time simulation in the EWS every time that new data is available, GPU technology has been used.

A specific computer for deploying the hydraulic model has been set with the following specifications:

- CPU: Intel Core i7-5820K 3.3GHz
- Memory: 32GB RAM DDR4 2133
- Graphics: Gigabyte GeForce GTX 980 WindForce 3X (4GB DDR5)
- Primary Hard Drive: SSD OCZ Vector 128GB SATA3
- Secondary Hard Drive: HDD 3.5" 3TB SATA3
- Cabinet to allow appropriate ventilation for the GPU heating.

This server for intensive GPU usage has been connected to the radar-based EWS and the operational scheme is the following:

1. The EWS server generates the rainfall data every time that new information is available (10 minutes) in the appropriate format for Infoworks ICM, and stores it in a shared folder among the two servers.
2. A program in the server for intensive GPU is triggered and runs the simulation with the new rainfall data.
3. The program stores the output time series for the elements (1D or 2D) in a shared folder.
4. The EWS server parses the output time series and stores the information in the internal database of the radar-based EWS. These time series could be used for issuing warnings in the web interface or for sending SMS/emails.

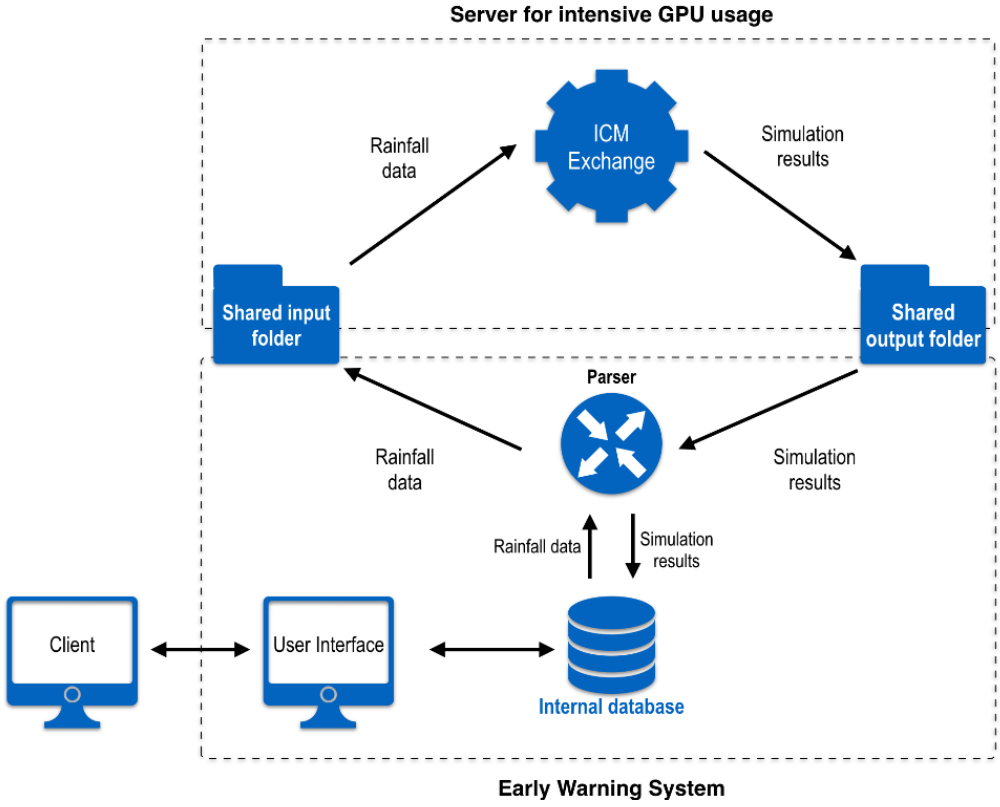


Figure 31 General schema of the process done to run a hydraulic model simulation in real time with the dedicated server with high GPU performance and the EWS.

A detailed description of the radar-based EWS and how the high-performance computer with GPU has been linked to the real-time EWS will be done in the PEARL Deliverable 4.5.

3.2.2 Results and discussion

The use of both the capabilities of InfoWorks ICM to optimize the computational grid size and the use of a GPU high-performance dedicated server for modelling allows to run the Marbella 1D-2D model in ~5 minutes. The radar update time in the Marbella pilot site is 10 minutes, thus it is possible to process the radar precipitation estimates, calculate the radar now-cast and use this information together with the rain gauges measurements to run the hydraulic 1D-2D model in real time to assess the impact of the rainfall on the sewer system and issue warning if necessary.

3.3 Hamburg, Germany

The application of the innovative technology of CPU/GPU computing especially for the realization of an evacuation module within the Kalypso model suite is explored by the TUHH aiming at the development of an evacuation model which can be run on an average NVIDIA®CUDA enabled desktop computer. A prototype for such an evacuation module is developed and applied to a selected area of the case study city of Hamburg.

3.3.1 General description of the evacuation model

The basic evacuation concept of Hamburg Wilhelmsburg

The Hamburg district of Wilhelmsburg is the favoured area for the application of the improved evacuation module. Due to its location between the Norderelbe and Süderelbe, the fact that Wilhelmsburg is like a bathtub with its ring dike and its dense population an appropriate evacuation model for planning evacuation processes is of crucial importance for the responsible authorities. Initially, the overall evacuation procedure of the island of Wilhelmsburg has been analysed with respect to the different individual evacuation processes. The overall evacuation concept encompasses pedestrian evacuation, evacuation by public transport, evacuation by private cars and the vertical evacuation. The key element is the evacuation of pedestrians to collecting points (bus stops) or to safe places (refuges). Evacuation of people by public transport take place at specified collecting points all over Wilhelmsburg (Figure 32). Public busses pick up the pedestrian waiting at the collecting points. The pedestrians are taken to the train stations Wilhelmsburg, Harburg and Veddel away from the endangered area. Evacuation by private cars is carried out via evacuation routes specified in the superordinate traffic concept of the responsible authorities. Another part of the evacuation concept of Wilhelmsburg is the vertical evacuation. It is assumed that people try to reach higher storeys in their residential houses or other buildings. In addition, there are several refuges (safe places) in Wilhelmsburg, which can be reached (personal communication, 2015).

The improved prototype is focussed on the simulation of pedestrian evacuation.

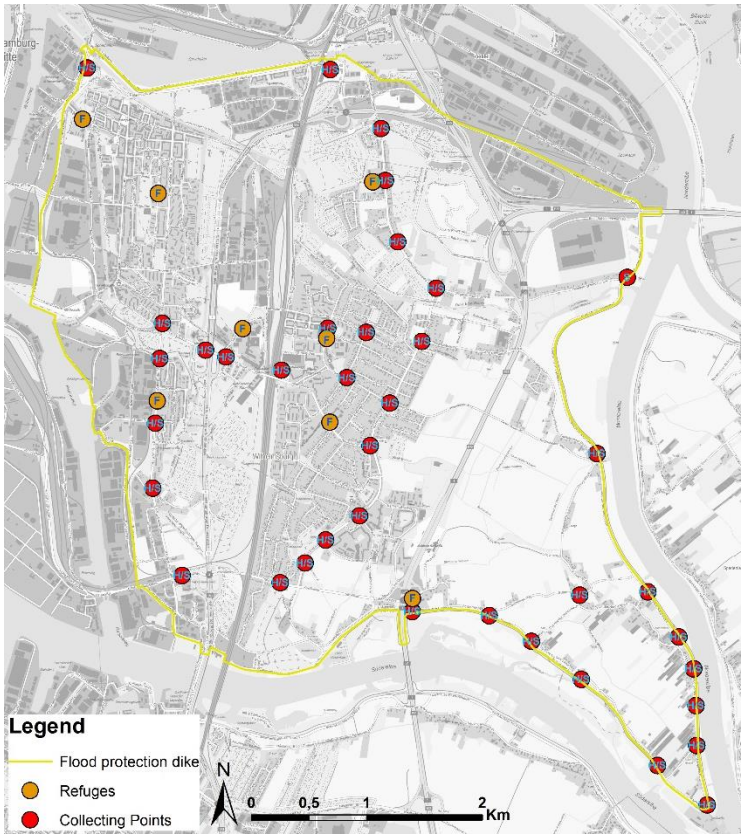


Figure 32 Collecting points and refuges in Wilhelmsburg

Mathematical-physical model for pedestrian evacuation

In a first step, the development of an evacuation model for pedestrian evacuation, which simulates the movement of the pedestrians to the collecting point is carried out based on a hybrid GPU/CPU implementation. In order to simulate the pedestrian dynamics a spatially continuous force-based model is applied, representing a robust and quantitatively verified model (Chraibi et.al. 2010). This approach couples the pedestrian dynamics with the dynamic topology. Newton's second law of dynamics is the basic principle for the applied force-based model (Chraibi et.al. 2010). According to Newton's second law of dynamics the movement of pedestrians can be described by driving and repulsive forces (Eq. 1):

$$m_i \ddot{\vec{R}}_i = \vec{F}_i = \overrightarrow{F_i^{drv}} + \sum_{j \in N_i} \overrightarrow{F_{ij}^{rep}} + \sum_{w \in W_i} \overrightarrow{F_{iw}^{rep}} \qquad \textit{Eq. 1}$$

$\overrightarrow{F_{ij}^{rep}}$ denotes the repulsive force from one pedestrian *i* acting on another pedestrian *j*, whereas $\overrightarrow{F_{iw}^{rep}}$ denotes repulsive forces from an obstacle *w* acting on a pedestrian *i*. $\overrightarrow{F_i^{drv}}$ represents the driving forces of a pedestrian *i*. m_i can be regarded as the mass of pedestrian *i*. Repulsive forces are used to represent the avoidance of possible collisions performed by pedestrians guaranteeing a certain volume exclusion of the pedestrians. Driving forces model the movement of the pedestrians form an initial position to a specified destination with a certain speed (Chraibi et.al. 2010). The applied centrifugal force model takes into account the distance between the pedestrians and their respective relative velocities. As mentioned above the movement of pedestrians can be regarded as a direct superposition of repulsive and driving forces. A detailed description of the model and the equations can be found in Yu et.al. 2005.

Data and preparatory works in ArcGIS

Adequate data need to be provided in order to apply this pedestrian dynamics approach. On the one hand, population data (number of people per age groups) are necessary. On the other hand, the location of the different collecting points and refuges (targets for the pedestrian dynamics simulation) are important as well as the building stock (obstacles for the pedestrians) in the considered area.

These data has to be processed in order to serve as the input for the pedestrian evacuation model. First, sub-evacuation areas of the collecting points and refuges have been derived based on adapted Thiessen polygons (Figure 33). There is one target, either a collecting point or a refuge, in each sub-evacuation area. This divide and conquer approach is used in order to support the efficiency of the selected mathematical-physical model and to increase the simulation efficiency. The borders of the sub-evacuation areas have been adjusted to streets and other constraint points, e.g. highways and rail tracks to create areas as simple as possible. The evacuation model simulates the movement of the pedestrians within each sub-evacuation area.

The building stock of Wilhelmsburg, available in shapefiles from the cadastral maps, are aggregated to bigger building blocks (Figure 33) aiming at the reduction of obstacles for simulation purposes for increasing the simulation efficiency. Here, the acceptable number of obstacles is given by the available hardware resources. The boundaries of the building blocks create the space for the movement of the pedestrians.

In a next step, the people living in Wilhelmsburg are to be distributed to the building blocks. For this purpose, population data, provided by the statistical office for Hamburg and Schleswig-Holstein, are used. These data is available in a shape file for bigger statistical areas containing the number of people per age groups. The distribution of the people to the buildings blocks is made on a percentage basis with respect to the areas sizes.

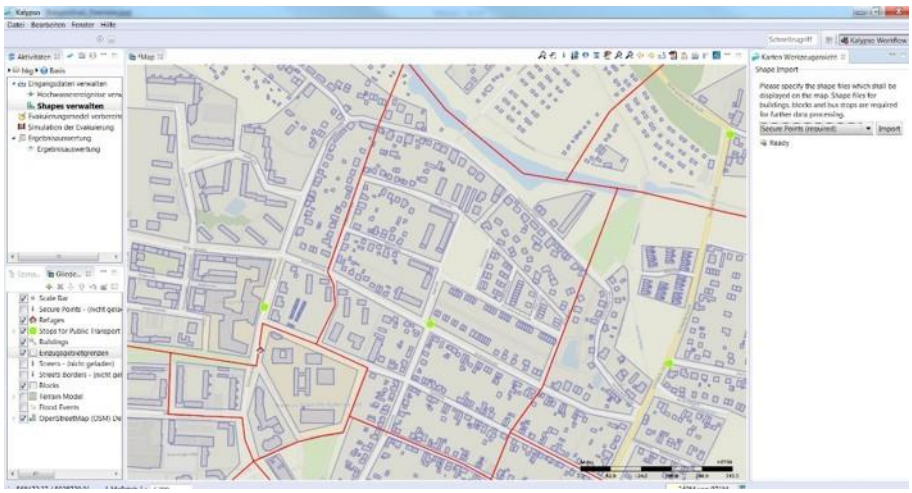


Figure 33 Screen shot from the Kalypso GUI showing "catchment areas" (red polygons), building polygons (small blue polygons) and aggregated building blocks (bigger blue polygons)

The above mentioned mathematical-physical model describes the movement of the pedestrians from the initial positions (building blocks) to the targets (collecting points or refuges). The simulation based on this model results in a distribution of the arrival times of the people at the targets, being the input for the simulation of further evacuation processes on CPU level.
The current implementation of the evacuation model is described in the following chapter.

3.3.2 *The evacuation model*

The improved evacuation model is implemented in the current version of the Kalypso model suite (TUHH, n.d.). The implemented workflow of the TUHH Kalypso Evacuation Module enables the user to set-up the evacuation model, run the model and assess the results of the evacuation simulation.

Within the pre-processing to set-up the evacuation model, the processed data (see chapter 3.3.1) can be imported into the TUHH Kalypso Evacuation Module as ArcGIS shape files (Figure 34) and viewed in the GUI. Shape files of the sub-evacuation areas, the building blocks (containing the number of people per age groups) and the collecting points, resp. the available refuges are required. Furthermore, a background map can be added to the view in the GUI. Additionally, the user can select specific sub-evacuation areas for the evacuation simulation in case that not the entire area should be considered. The selected sub-evacuation areas aggregated in so-called evacuation simulation units are the basis for the further processing within the evacuation simulation.

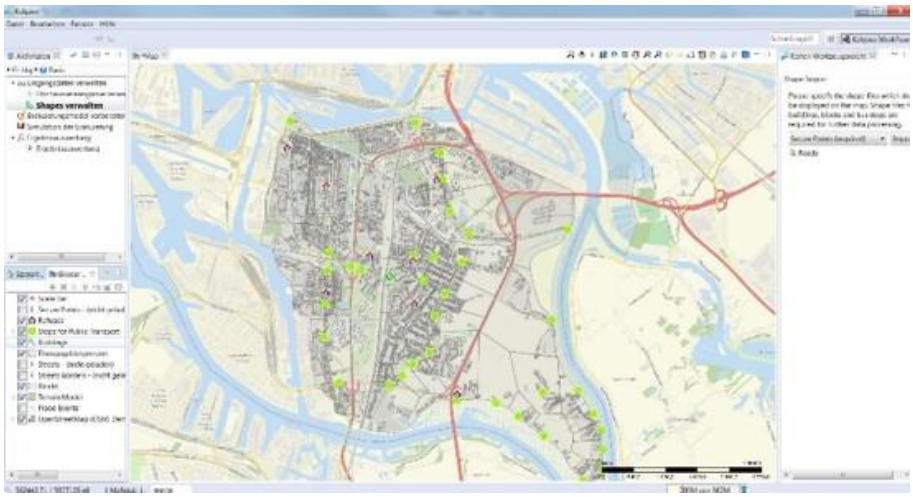


Figure 34 Screen shot from the Kalypso GUI showing the area under investigation, aggregated building blocks, collecting points (green symbols) and the refuges (house symbol)

As soon as the pre-processing is finished the pedestrian evacuation simulation is launched from the Kalypso GUI. The simulation of the pedestrian dynamics, which is the aim of the evacuation model, contains different steps carried out on either CPU level or GPU level. Here, the hybrid CPU/GPU computing approach takes effect. The first step in simulating the pedestrian dynamics is the initial distribution of the pedestrians, either around the i) buildings blocks or ii) randomly over the sub-evacuation area (can be user specific), aiming to assign an initial position to each pedestrian. In case, that the user selects the around-the-block strategy (i, see above) the distribution of the pedestrians represents different starts of movement. Later start of the movement corresponds to a longer distance between the initial position of the pedestrian to the respective target. By implication, an early start of the movement is represented by a short distance to the target. In the second case (ii, see above), the distribution is fully random over the entire area, representing e.g. a situation with pedestrians getting the evacuation alarm while moving in the public space outside of buildings.

As soon as the initial processing steps are finished on the CPU level the TUHH Kalypso Evacuation Module starts the simulation core on the GPU. It reads the input files, relevant for the simulation. The data is written to the global memory of the GPU. Both aspects, distributing the pedestrians and transferring the input data to GPU, are carried out on the CPU level. The

simulation of the pedestrian dynamics itself, using the above mentioned approach, takes place on the GPU benefitting from the mentioned advantages of the GPU computing, which is the ability of massive parallelism (see Chapter 2.2.4). The movement of the pedestrians from their initial position around the buildings blocks to the targets, the collecting points or refuges within the sub-evacuation area, using the space formed by the borders of the buildings blocks is simulated. The arrival times of the pedestrians at the targets are collected. After the simulation is finished and the collection of the arrival times is accomplished, these are written to files being read by the TUHH Kalypso Evacuation Module. A description of the workflow of the developed evacuation simulation core can be found in Figure 35.

The simulated arrival times of the pedestrians at the collecting points are used (i) for the visualization and analysis of these results or (ii) for the assessment of further successive evacuation processes, which is the evacuation of pedestrians by public transportation being carried out at CPU level. This further emphasises the hybrid CPU/GPU computing approach.

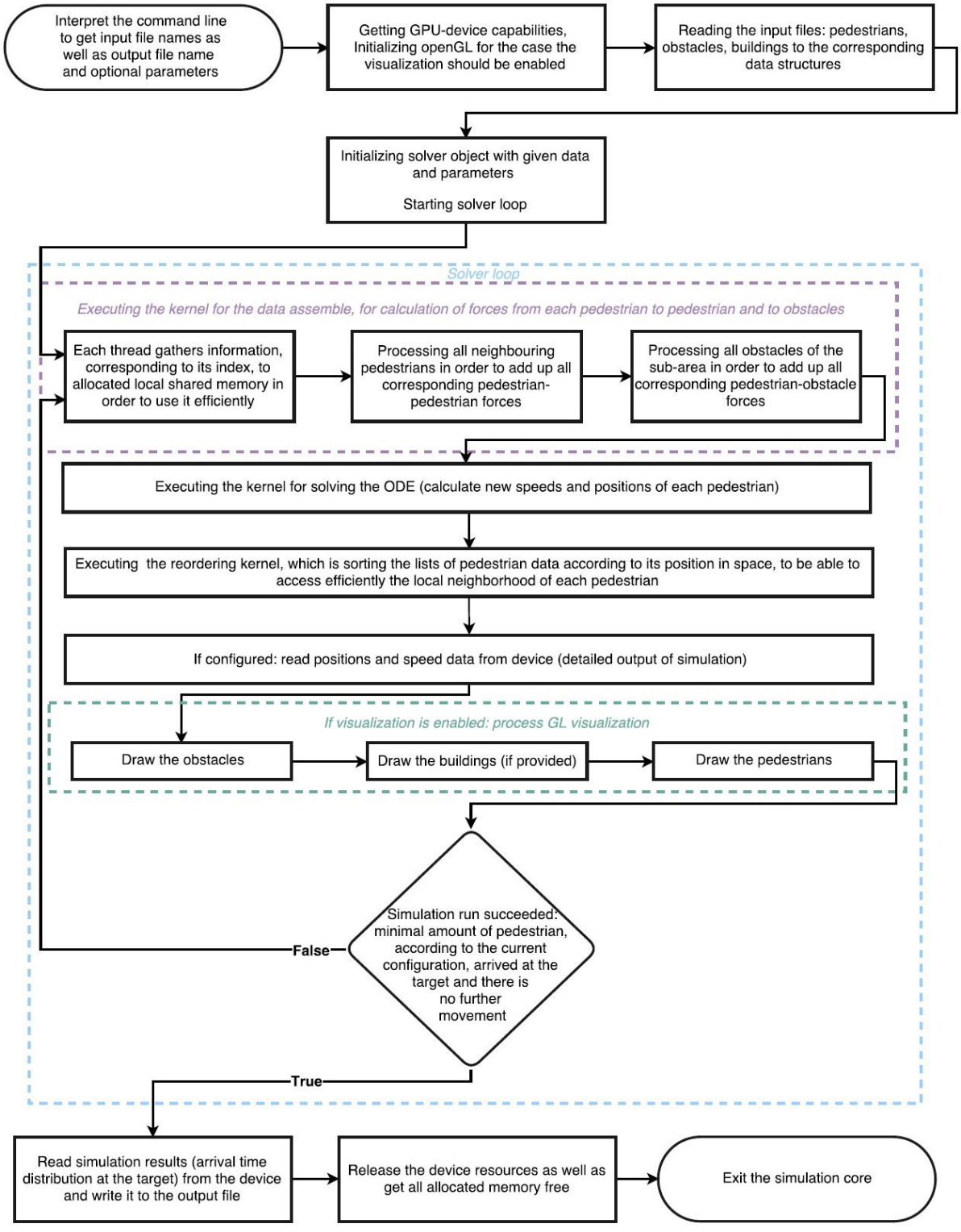


Figure 35 General workflow of the developed evacuation simulation core

3.3.3 *Summary and Outlook*

TUHH has developed a prototype for pedestrian evacuation on the basis of hybrid CPU/GPU computing using the advantages of both computing technologies, which has been implemented into the TUHH Kalypso modelling suite. On the basis of a spatially continuous force-based model, arrival times of the pedestrian at the collecting points, respectively refuges are simulated, taking into account an initial distribution of the people, possible obstacles and the interaction between the pedestrians themselves and the interaction between the pedestrians and the obstacles in the considered area. In order to create a simulation tool using the hybrid CPU/GPU approach, which is as efficient as possible, the simulation of the pedestrian dynamics is carried out on sub-evacuation areas considering as less as possible obstacles. As mentioned above arrival times of the pedestrians at the collecting points will be used as input for the further assessment of the evacuation of the pedestrians by public transportation.

A comparison of the developed simulation using the mentioned technologies against high performance computing and cloud computing, as mentioned in the PEARL DOW, is not applicable in the context of the development of an evacuation module which can be run on an average NVIDIA®CUDA enabled desktop computer. Due to the significant different dimensions of the computing capacities (desktop computer vs. high performance computer or cloud computer) it is impossible to draw a reliable comparison.

3.4 Rethymno, Greece

Rethymno, one of the PEARL case studies, was one more example of application of the various techniques suggested for achieving faster simulations. For the city of Rethymno (Figure 36), multiple stressors have always posed flood threats. The terrain and streams morphology convey volumes of storm water runoff from the upstream rural areas to the highly urbanized, flat downstream zones, pressurizing the drainage facilities and the flood defence infrastructures. The dominant strong northern and north-western winds highly affect the exposed coastal zone and result in the development of waves, which often overtop the harbour infrastructure and erode recreational beaches (Karavokiros et al., 2016). As such, Rethymno is susceptible to multiple pressures which therefore require the set-up of combined models with a complex urban topography.

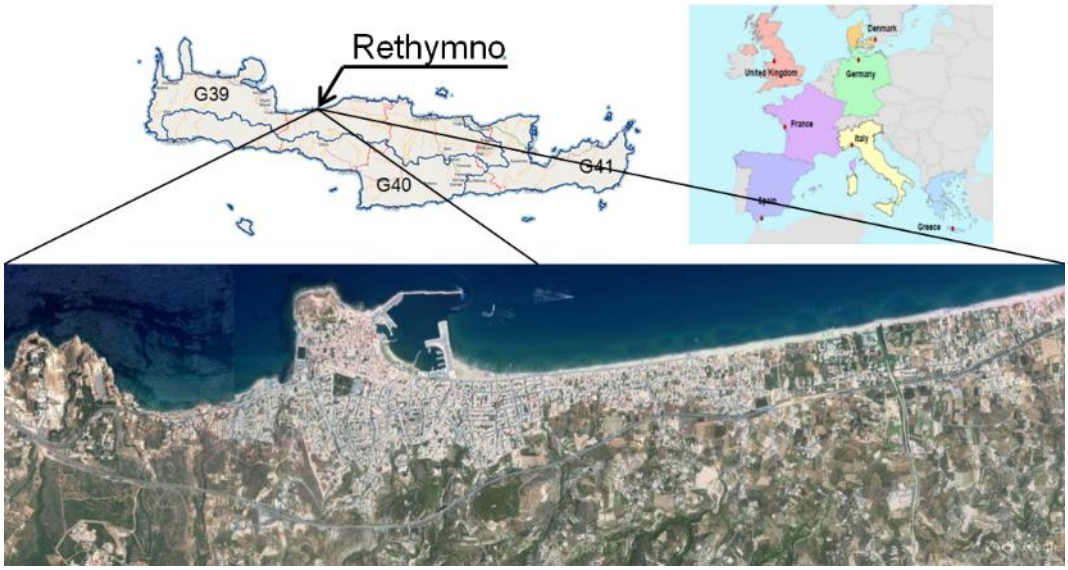


Figure 36 Location of Rethymno case study (Makropoulos et.al., 2014)

Rethymno flood models employ a 3-way coupling of 1D and 2D simulations engines, enabling analysis of urban, coastal and riverine flooding under different precipitation scenarios (past recorded events or produced under different return periods). The river network consisting with natural, open channels has been built with MIKE 11, whereas the arranged parts of streams (closed cross section) crossing the urban part of the city have been set up with MIKE Urban so that interactions with the surface runoff can be enabled. The 2D model component was build using the MIKE 21 FM, a 2D modelling system based on a flexible mesh approach. Due to the large size of the area under study, the domain was divided into two parts, hence, two different models set-ups have been created during initial model versions. Focusing on Area 1 i.e. west part of Rethymno, the 2D model covers 6.06 km2 of Rethymno's coastal area comprising of both sea and inland regions, and its mesh is comprised of 774,619 elements varying in size from 0.064 m2 to 500 m2. Small mesh elements were used for built-up areas, roadways and flood plains, while bigger elements were used for rural and sea areas.

The precipitation event which has been used as catchment loads in all model set ups, was the one recorded on November 10th, 1999, since it had major impacts to city's functions and services. The simulation period for the aforementioned event lasts 12 hours with a time step varying based on model setup versions.

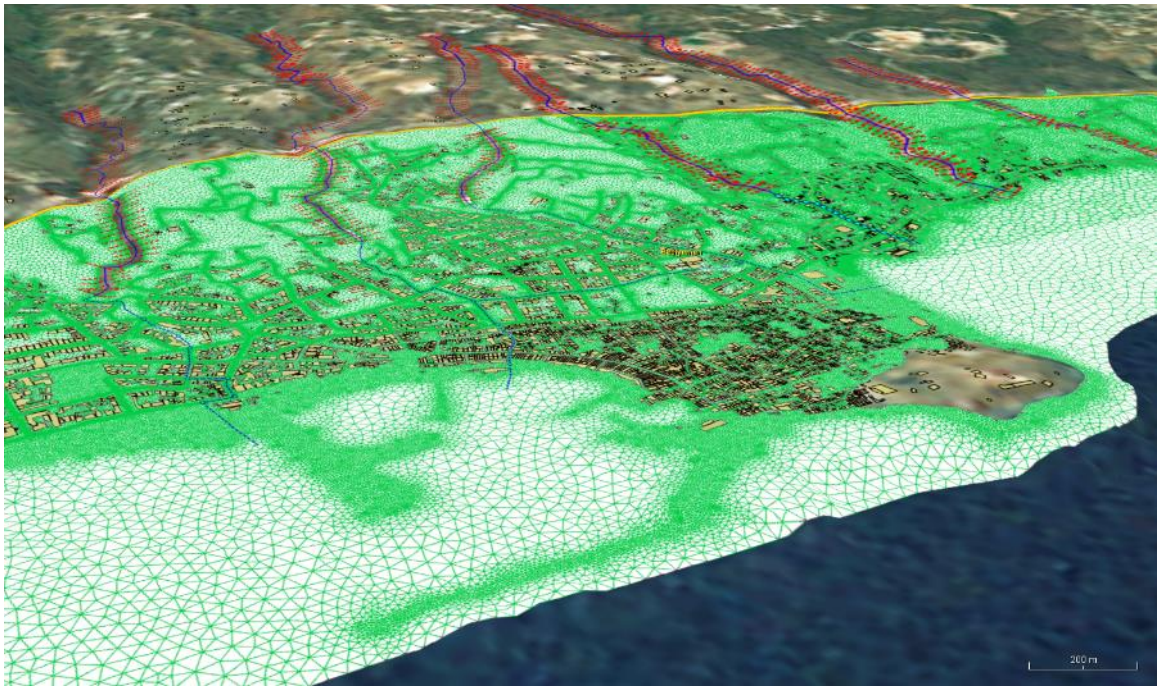


Figure 37 A plot of the 1D-2D flood model for Rethymno from the PEARL WebLP. A 1D model of the drainage system, comprising of streams with open cross sections parts (blue continuous line) and closed/arranged parts (blue dash space line), is linked to a 2D model of the inland/urban and coastal area. The triangular mesh elements (with different discretisation) of the 2D model are also shown (green triangles)

An urban coastal flood model that simulates accurately the evolution of floods in an urban environment due to multiple stressors while performs fast enough was need for the case study, so that a variety of scenarios could be examined within the content of flood risk assessment. The current model of Rethymno is a result of several iterations and a step by step approach, especially for the creation of the flexible mesh, the most important component affecting duration of simulations. While aiming to increase simulation speed a few speed-up techniques were tested in Rethymno, including model modifications, change of simulation/set up approaches and increasing hardware capabilities.

3.4.1 Speed-up technique(s)

The following model speed-up techniques were tested and evaluated with Rethymno urban coastal flood model:

1. Selection of model domain, computational grid type and elements size
2. Modification of modelling approach
3. Maximising computational power

Components 1 and 2 of the above list aim to minimise computational loads whereas the last one mainly targets to increase computation speed by maximising and utilising computational power.

Selection of model domain, computational grid type and element size

Several modelling challenges were faced while setting up the hydrodynamic models for the area under study. Those challenges were related to the large area under study, including rural and urban parts and the high resolution data that were needed to be utilised which led to long computation time. Additionally, due to the combined flood events that have been faced throughout the years, a 3way coupling of models was necessary in order to simulate accurately the physical

phenomena. Since the urban density of Rethymno is high in several parts, the creation of small grid or flexible mesh elements was unavoidable in order to produce accurately flood inundation results.

All the above reasons and challenges led to the division of the area under study into two parts, Area 1 (West area of Rethymno) and Area 2 (East area of Rethymno) as depicted in Figure 38. Furthermore, the flexible mesh was selected as computational grid type which enables the creation of different element size and the alteration of mesh resolution along domain.

Key lessons learned during mesh generation and eventually adopted in order to enable the creation of new mesh files in shorter time were:

1. Division of area under study into smaller parts and use of local maximum element area parameter depending the areas of interest and the necessary/desired resolution (type of domain and available DEM resolution)
2. Generation of mesh step by step in each sub-area since it enables to easily locate problematic areas obstructing mesh generation e.g. invalid arcs
3. Instead of defining only the maximum element area, it is advisable to control the resolution of mesh through geographical entities e.g. roads polylines or building polygons
4. Gradually increase mesh resolution and complexity and test it through initial simulation (Figure 39)

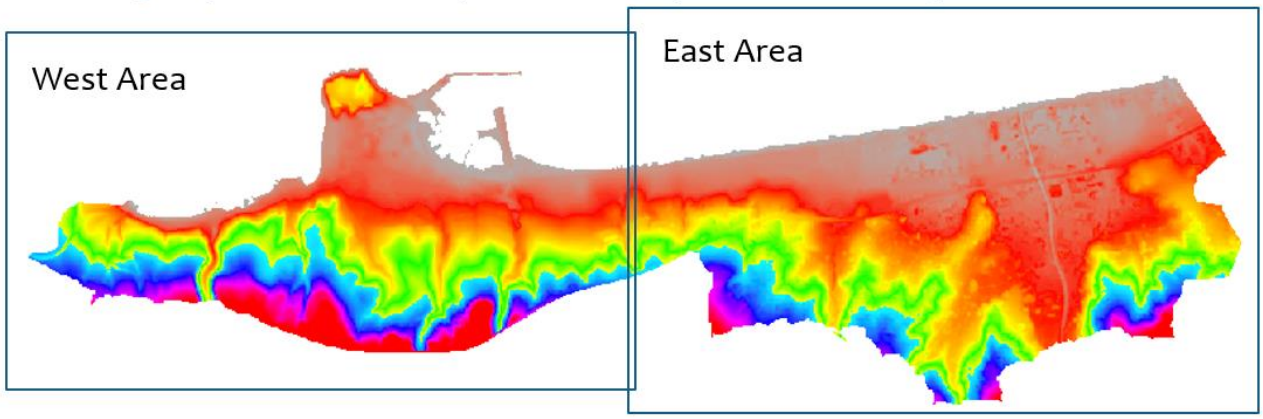


Figure 38 Area under study divided into parts for the creation of manageable set up files

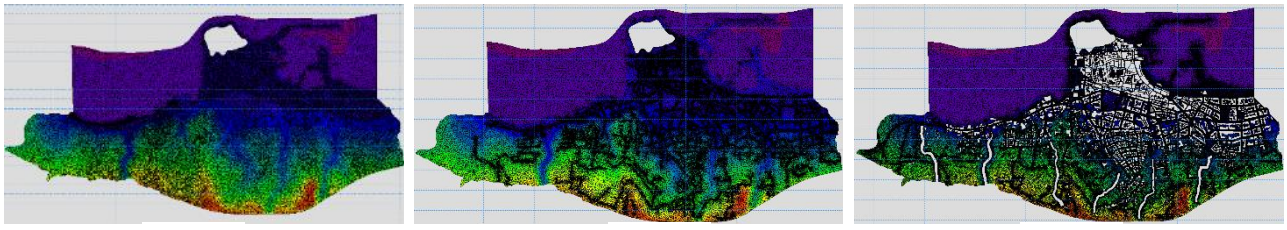


Figure 39 Step by step generation of mesh files and increase of mesh resolution. Mesh a is consisted of 32.615 elements for which river lines and sub-catchments' boundaries were used as geographical entities, mesh b is consisted of 109.950 elements while refined polylines of roads were used for mesh resolution increase and mesh c includes 369.967 elements while using building boundaries and excluding river beds and building polygons from calculations.

Modification of modelling approach

Major outcome derived during initial runs in the Old Town of Rethymno (small part of Area 1) was that when applying gridded precipitation within model setups, it was difficult to establish a stable 2D simulation due to Rethymno's complex topography. Even when normal run completions were achieved, the warning of too many CFL condition violations (more than 200) in finite elements raised doubts about the accuracy of results and therefore demanded the allocation of significant effort in order to establish a stable 2D simulation. Same wise, additional modifications are required each time the boundaries or element size is being changes and CFL violations occur.

An alternative approach, which was attempted in order to cope with the aforementioned drawbacks and also to speed the simulations up, was to consider streets as a fictive drainage network with very low capacity of links and applying the calculated run off to those nodes/links. Following the practical strategies provided in DHI's modellers' guidelines (DHI, 2015), a fictive drainage network was created along the roads of Rethymno's which receives the local runoff load and transfers it to the 2D overland model. This fictive network has nodes with invert level just a few centimetres (e.g. 5 cm) below the ground level, whereas the links of the fictive network have an insignificant transport and storage capacity. The runoff produced from the sub catchments (while using different method for runoff calculations) is being distributed to the fictive network loads. As an alternative to the surface runoff models, the precipitation load was not modelled directly by the 2D overland model, i.e. bypassing the surface runoff model. Major advantage of this method was to decrease simulations durations.

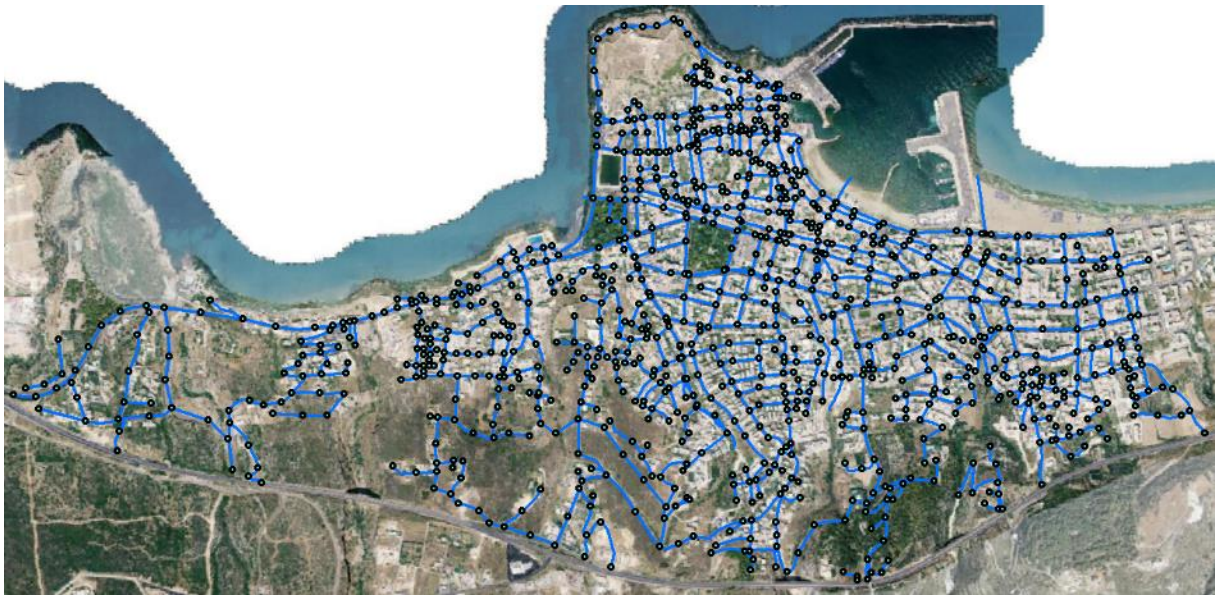


Figure 40 Alternative approach to model surface runoff through a fictive drainage network

Maximising computational power

Apart from modifying the modelling approach or the model setup itself in order to achieve faster simulations, maximising the computation power was also applied to Rethymno case study. Model calculation was performed by using three different approaches listed below:

1. Shared memory approach (OpenMP)
2. Distributed memory approach (MPI)
3. GPU computing on two computers with the below described specs:

- Computer 1: Intel(R) Core(TM) i7-4790K CPU @4.00 GHz, 32GB of RAM and Windows 10 Pro operating system (Maximum speed: 4.00 GHz, Sockets: 1, Cores: 4, Logical Processors: 8, Virtualisation: Enabled, L1 cache: 256 KB, L2 cache: 1.0 MB, L3 cache: 8.0 MB)
- Computer 2: Intel(R) Core(TM) i7-5820K CPU @3.30 GHz, 32GB of RAM and Windows 10 Pro operating system (Maximum speed: 3.30 GHz, Sockets: 1, Cores: 6, Logical Processors: 12, Virtualisation: Disabled, Hyper-V support: Yes, L1 cache: 384 KB, L2 cache: 1.5 MB, L3 cache: 15.0 MB)

The later computer enables also GPU computing option with NVIDIA GeForce GTX 750 Ti.

The model configuration were forced with the same actually recorded precipitation event of 1999 and results were compared with emphasis in simulations' duration. Regarding model configurations, the below set ups were used in the tests:

- **Setup 1**: 2way coupling of models (1D-MU and 2D-M21) with a mesh file consisting of 354322 elements, while using the alternative approach with the fictive drainage network
- **Setup 2**: 3way coupling of models (1D- MU/M11 and 2D-M21) with a mesh file consisting of 369967 elements, while applying rain on grid (varying in time, constant in domain)
- **Setup 3**: 3way coupling of models (1D- MU/M11 and 2D-M21) with a mesh file consisting of 774619 elements, while applying rain on grid (varying in time, constant in domain)
- **Setup 4:** 2D – M21 with a mesh file consisting of 774619 elements, while applying profile series of water surface fluctuation at the downstream coastal boundaries (varying in time and along boundary). No precipitation or runoff loads/boundaries were used.

3.4.2 Results and discussion

The results of tests regarding maximisation of computational power are shown in Table 5. Model results with OpenMP, MPI and GPU computing, in terms of water depths and flood extents, were exactly the same for each set-up, which is as expected since the numerical scheme is the same in all parallelization approaches in Mike 21 FM.

Taking a closer look at the results, the use of MPI instead of OpenMP approach for the same model set up and computer, leads to a speed up factor of 1.42, whereas while using a different pc with slightly better specs gives 2.23. Comparing the PCs specs and the benchmark results (Average CPU mark: 11186-Computer 1, 12979-Computer 2, available at PassMark Software, accessed on November 11th, 2016), one could argue that this acceleration in the simulation time is due to the newer and better marked CPU and the increased number of logical processors (from 8 to 12). What is remarkable though is that the overall speed up factor might be around 4, while maximising computational power and performing GPU computing as presented in the below table. In general, one should keep in mind that performance of GPU increases when the number of elements is high (e.g. 300,000 elements).

Depending on the GPU type and its ASIC quality, the results may vary significantly as shown in many studies, such as (DHI, 2014). Other techniques that increase the performance of the GPU such as overclocking will not be discussed here.

Comparing the computation time of the two approaches i.e. applying rain on grid on the domain or distributing runoff to a fictive drainage network, it is apparent that simulation lasts around 7.9 times more when using 2D rain. The readers should also note that the low performance increase while comparing durations of simulation of both computers is due to the large number of CFL violations. Model needs to be relatively stable for GPU use in order to show improvement of simulation time over CPU. Proving this statement, during setup 4 where the model was stable with no CFL violations, the GPU performance was maximised giving an overall speed-up factor around 6.17 (as presented in Table 5 below).

Table 5 Comparison computation times for the model Setup 2 and 3 in the study. Shared memory approach, distributed memory approach and GPU (NVIDIA GeForce GTX 750 Ti) computing options were tested.

	Computer 1 OpenMP	Computer 1 MPI	Computer 2 OpenMP	Computer 2 GPU	Speed-up factor
	Computation time [h]				
Setup 2 **Applying rain on grid** (369,967 elements)	101.34	-	85.71	-	~1.18
Setup 3 **Applying rain on grid** (774,619 elements)	304.30	-	-	83.84	~3.63
Setup 4 **Applying Coastal Boundary Conditions** (774,619 elements)	162.95	-	-	26.39	~6.17

Table 6 Comparison of computation times for the model Setup 1 in the study. Shared memory approach, distributed memory approach and GPU (NVIDIA GeForce GTX 750 Ti) computing options were tested.

	Computation time [h]				Speed-up factor
	Computer 1 OpenMP	Computer 1 MPI	Computer 2 OpenMP	Computer 2 GPU	
Setup 1 **Distributed approach with the fictive network** (354,322 elements)	12.79	9.00	-	-	~1.42
	-	-	5.74	3.25	~1.77
	12.79	-	5.74	-	~2.23
	12.79	-	-	3.25	~3.94

Other factors that were proven to have significant impact on the simulation duration and increased disproportionately the needs for computational power were the modification of some parameters related to the 'wetting and drying' phenomena. Using values in the order of mm instead of cm, the duration of simulation was at least doubled. Same wise, when rain on grid – constant in domain was replaced with time and spatially varying precipitation. Finally, the frequency of exporting results was one more factor with decisive influence to the simulation time (and storage needs), but its impact was not that profound.

Regarding the increased performance achieved due to the selection of model domain, the computational grid type and the element size, the researchers are not providing any actual numbers in order to quantify it, but based on personal experience while setting up the models, it was impossible to perform any simulation at all when the above suggestions (described in Section 3.4.1) were not incorporated.

4 Summary and Conclusions

This guideline document presented and discussed options for achieving faster model simulations in the context of flood early warning. These options involve different aspects of the modelling process, from model conceptualisation and construction, to computational hardware capabilities. The various techniques were organised in two categories in the discussion:

3. Techniques for minimising computational load
    - Model (concept) simplification
    - Model (construction) optimisation
4. Techniques for maximising computational power
    - CPU parallelisation
    - Hydbrid/GPU computing

A systematic process for model speed-up, presented in Figure 1, entails model optimisation techniques preceding options involving hardware improvement. This is because high-performance computing capabilities are, as yet, atypical in standard computers and would typically incur additional costs, and an optimal model setup can only be an advantage in improving model performance and maintaining accuracy.

An optimised computational grid is important in achieving a high-performing (2D) flood model for early warning. The process includes not only careful delineation of the model domain, but also use of new computational grid types affording more flexibility in terms of grid size distribution and terrain modelling. This was demonstrated in the Greve, Marbella, and Rethymno case studies, where a shift from rigid grid to flexible computational meshes brought model simulation times closer to levels that were practicable (i.e. for the case of Rethymno), and nearer targets for operational flood warning (i.e. for Marbella and Greve). Tests conducted in the Greve case study showed that a change from rigid grid to flexible mesh in the flood model provided speed-up factors of up to 3 to 4. More care is needed when using model simplification to achieve fast flood simulations. Requirements with respect to level of detail and accuracy, modelling of dominant processes, and input requirements, must be carefully considered when streamlining model type against model speed. Warning systems are subject to expectations regarding forecasting technique and accuracy, together with forecast lead-time and service delivery. Thus, speed-up techniques maintaining model accuracy are favourable in flood early warning applications.

Recent developments in computer hardware allow high-performance computing with multiple cores, and graphics cards that can perform computationally intensive calculations. Faster model simulations are achieved through parallelisation and hybrid computing techniques, and the use of GPUs (Graphics Processing Units) has been shown to significantly improve (i.e. speed up) model simulation times. In the Marbella pilot site, the use of GPU computing was also used to improve computational time for their early warning model. The use of both model setup and computational power optimisation techniques allowed the Marbella 1D-2D flood model, which covers 10.6 km2 of area with more than 60 000 mesh elements and 68 km of sewers, to run a 2-hour simulation within ~5 minutes. TUHH developed a pedestrian evacuation model, and employed hybrid CPU/GPU computing to implement it in their Kalypso Modelling Suite. In the Greve case study, an additional speed-up factor of 2 to 3 was achieved with GPU computing with the 2D flexible mesh model, and similarly in Rethymno, a speed-up factor of around two was obtained.

Various options are available for achieving fast flood model simulations suitable for flood warning applications. Significant improvement in simulation times for current state-of-the-art 1D-2D Cartesian grid flood models are possible (e.g. up to a factor of 47, as in the Greve case), from a combination of computational load, and computational power optimisation techniques that are available today.

References

Krupka, M., G. Pender, S. Wallis, P.B. Sayers and J. Mulet-Marti (2007). A Rapid Flood Inundation Model, Proc. 32th IAHR Congress, Venice, 1-6 July, paper SS05-04-O.

Neal, J. C., Fewtrell, T. J., Bates, P. D., & Wright, N. G. (2010). A comparison of three parallelisation methods for 2D flood inundation models. Environmental Modelling & Software, 25(4), 398-411.

Sto. Domingo, N. D., Paludan,B., Hansen, F., Madsen, H., Sunyer, M., Mark, O. (2010). Modeling of sea level rise and subsequent urban flooding due to climate changes. In E. &. DHI Water (Ed.), SimHydro 2010: Hydraulic modeling and uncertainty, (p. 9). Sophia Antipolis.

Message Passing Interface Forum (2012). MPI: A Message-Passing Interface Standard, Version 3.0. Available from: https://www.mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf.

OpenMP Architecture Review Board (2015). OpenMP Application Programming Interface. Available from: http://www.openmp.org/mp-documents/openmp-4.5.pdf.

Chapman, B., Jost, G., Pas, R. (2008). Using OpenMP: Portable Shared Memory Parallel Programming. MIT Press, Cambridge, Massachusetts.

Hunter, N.M., Bates, P.D., Neelz, S., Pender, G., Villanueva, I., Wright, N.G., Liang, D., Falconer, R.A., Lin, B., Waller, S., Crossley, A.J. and Mason, D. (2008). Benchmarking 2D hydraulic models for urban flood simulations. Proceedings of the Institution of Civil Engineers: Water Management, 161 (1). 13-30.

Breß, S.; Mohammad, S.; Schallehn, E. (2010): Self-tuning distribution of DB-operations on hybrid CPU/GPU Platforms, http://ceur-ws.org/Vol-850/paper_bress.pdf , Last Access: 02.02.2017

Chraibi, M.; Seyfried, A,; Schadschneider, A. (2010): Generalized centrifugal force model for pedestrian evacuation, In: Physical Review E, Vol. 82, Issue 4, DOI: 10.1103/PhysRevE.82.046111

Lang, J.; Rünger, G. (2013): Dynamic distribution of workload between CPU and GPU for a parallel conjugate gradient method in an adaptive FEM, In Procedia of Computer Science, Vol. 18, International Conference on Computational Science, ICCS 2013, DOI: 10.1016/j.procs.2013.05.193

Lu, F.; Song, J.; Coa, X.; Zhu, X. (2012): CPU/GPU computing for long-wave radiation physics on large GPU clusters, In: Computer & Geosciences, Vol. 4, DOI: 10.1016/j.cageo.2011.08.007

NVIDIA: CUDA C/C++ Basics Supercomputing 2001 Tutorial, http://www.nvidia.de/docs/IO/116711/sc11-cuda-c-basics.pdf , Last Access: 01.02.2017

NVIDIA: Gromacs 4.6 Pre-Beta Benchmark Report, http://www.nvidia.de/docs/IO/122634/GROMACS-benchmark-report.pdf , Last Access: 24.02.2017

NVIDIA: What is GPU computing, http://www.nvidia.co.uk/object/gpu-computing-uk.html , Last Access: 01.02.2017

Pydiura, N.; Kaprov, P.A.; Blume, Y. (2014): On the efficiency of CPU and hybrid CPU-GPU systems in computational biology tasks, In: Computer Science and Applications, Vol. 1, No. 1, pages 48-59.

TUHH: Kalypso Simulation hydrologischer und hydraulischer Prozesse, https://www.tuhh.de/wb/forschung/software-entwicklung/kalypso.html , Last Access: 01.02.2017

Yu, W.J.; Chen, l.; Dong, R.; Dai, S. (2005): Centrifugal force model for pedestrian dynamics, In: Physical Review E, Vol. 72, Issue 2, DOI: 10.1103/PhysRevE.72.026112

Yu, Ch. D.; Wang, W.; Pierce, D. (2011): A CPU-GPU hybrid approach for the unsymmetric mulitfrontal method, In: Parallel Computing, Vol. 37, pages 759-770, DOI: 10.1016/j.parco.2011.09.002

Innovyze (2016). InfoWorks Integrated Catchment Model (ICM) v.6.5. User manual references.

Abebe, A., & Price, R. (2005). Decision support system for urban flood management. Journal of Hydroinformatics, 7, 3-15.

Alfieri, L., Salamon, P., Pappenberger, F., Wetterhall, F., & Thielen, J. (2012). Operational early warning systems for water-related hazards in Europe. Environmental Science & Policy, 21, 35-49.

Berbel Roman, S. (2014). Modelling flooding from the sea interacting with the drainage system under the influence of combined flood hazards to develop risk management strategies for the coastal region of Greve, Denmark (Master's thesis). University of Nice Sophia Antipolis, Nice, France.

Berenguer, M., Corral, C., Sánchez-Diezma, R., & Sempere-Torres, D. (2005). Hydrological validation of a radar-based nowcasting technique. Journal of Hydrometeorology, 6(4), 532-549.

Berenguer, M., Sempere-Torres, D., & Pegram, G. G. (2011). SBMcast–An ensemble nowcasting technique to assess the uncertainty in rainfall forecasts by Lagrangian extrapolation. Journal of Hydrology, 404(3), 226-240.

Bode, L., & Hardy, T. A. (1997). Progress and recent developments in storm surge modeling. Journal of Hydraulic Engineering, 123(4), 315-331.

Booij, N., Ris, R. C., & Holthuijsen, L. H. (1999). A third-generation wave model for coastal regions: 1. Model description and validation. Journal of Geophysical Research: Oceans (1978-2012), 104(C4), 7649-7666.

Bowler, N. E., Pierce, C. E., & Seed, A. W. (2006). STEPS: A probabilistic precipitation forecasting scheme which merges an extrapolation nowcast with downscaled NWP. Quarterly Journal of the Royal Meteorological Society, 132(620), 2127-2156.

Brown, S., Nicholls, R. J., Woodroffe, C. D., Hanson, S., Hinkel, J., Kebede, A. S., ... & Vafeidis, A. T. (2013). Sea-level rise impacts and responses: a global perspective. In Coastal Hazards [C.W. Finkl (Ed.)]. Springer Netherlands, Netherlands.

Butler, D., & Davies, J. (2011). Urban drainage (3rd ed.). CRCPress, Abingdon, Oxon, UK.

Carr, R. S. & Smith, G. P. (2006). Linking of 2D and pipe hydraulic models at fine spatial scales. Paper presented in 7th International Conference on Urban Drainage Modelling and 4th International Conference on Water Sensitive Urban Design, 2-7 April 2006, Melbourne, Australia.

Cavaleri, L., Alves, J. H., Ardhuin, F., Babanin, A., Banner, M., Belibassakis, K., ... & WISE Group. (2007). Wave modelling–the state of the art. Progress in Oceanography, 75(4), 603-674.

Chen, A. S., Evans, B., Djordjevic, S. & Savic, D. A. (2012). A coarse-grid approach to representing building blockage effects in 2D urban flood modelling. Journal of Hydrology 426-427, 1-16.

Christensen, B. B., Drønen, N., Klagenberg, P., Jensen, J., Deigaard, R., & Sørensen, P. (2013). Multiscale Modelling of Coastal Flooding. Paper presented in Coastal Dynamics 2013—7th International Conference on Coastal Dynamics, 24-28 June 2013, Bordeaux, France.

pearl

Preparing for Extreme And Rare events in coastal regions