

Numerical Simulation of Flooding from Multiple Sources Using Adaptive Anisotropic Unstructured Meshes and Machine Learning Methods

A PhD Dissertation
Presented to
Faculty of Engineering

By

Rong Hu

Supervised by
Dr. Fangxin Fang
Prof. Christopher Pain

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Department of Earth Science and Engineering

Imperial College London
September 2019

Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Dedication

I here by declare that this thesis entitled “**Numerical Simulation of Flooding from Multiple Sources Using Adaptive Anisotropic Unstructured Meshes and Machine Learning Methods**” is a genuine research work carried out by me under the guidance of **Dr. Fangxin Fang** and **Prof. Christopher Pain**, Imperial College London, Department of Earth Science and Engineering, AMCG - Applied Modelling & Computation Group.

Rong Hu

Acknowledgments

It would not have been possible to write this doctoral thesis without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here.

Above all, I would like to thank my mother and sisters, who have given me their unequivocal support and great patience at all times. They always believe in me and encourage me to follow my dreams.

This thesis would not have been possible without the help, support and patience of my principal supervisor, Dr. Fangxin Fang. She's always been supportive to me. The good advice and friendship of her have been invaluable on both an academic and a personal level, for which I am extremely grateful. Without her guidance and constant feedback this PhD would not have been achievable. Also, I would like to thank my second supervisor Prof. Christopher Pain for all the support and encouragement he's given me.

Many thanks also to the support I received from the collaborative work with Dr. Pablo Salinas. During the first phase of my research work, I greatly appreciate his support, making those first few months of model development more interesting.

I am also grateful to my research team members: Dunhui Xiao and Pan Yang, for encouraging and helping me during my research work. Their friendship and the warmth they extended to me made me feel so welcome, which means a lot to me.

And finally a very special thank to Chenyang Wu, who has been by side as my dearest friend throughout her PhD at Imperial College. We have much in common and always worked out together. A big thank to her for always listening to me and helping me in whatever way she could during the challenging period.

Abstract

Over the past few decades, urban floods have been gaining more attention due to their increase in frequency. To provide reliable flooding predictions in urban areas, various numerical models have been developed to perform high-resolution flood simulations. However, the use of high-resolution meshes across the whole computational domain causes a high computational burden. In this thesis, a 2D control-volume and finite-element (DCV-FEM) flood model using adaptive unstructured mesh technology has been developed. This adaptive unstructured mesh technique enables meshes to be adapted optimally in time and space in response to the evolving flow features, thus providing sufficient mesh resolution where and when it is required. It has the advantage of capturing the details of local flows and wetting and drying front while reducing the computational cost. Complex topographic features are represented accurately during the flooding process. This adaptive unstructured mesh technique can dynamically modify (both, coarsening and refining the mesh) and adapt the mesh to achieve a desired precision, thus better capturing transient and complex flow dynamics as the flow evolves. A flooding event that happened in 2002 in Glasgow, Scotland, United Kingdom has been simulated to demonstrate the capability of the adaptive unstructured mesh flooding model. The simulations have been performed using both fixed and adaptive unstructured meshes, and then results have been compared with those published 2D and 3D results. The presented method shows that the 2D adaptive mesh model provides accurate results while having a low computational cost.

The above adaptive mesh flooding model (named as *Floodity*) has been further developed by introducing (1) an anisotropic dynamic mesh optimization technique (anisotropic-DMO); (2) multiple flooding sources (extreme rainfall and sea-level events); and (3) a unique combination of anisotropic-DMO and high-resolution Digital Terrain Model (DTM) data. It has been applied to a densely urbanized area within Greve, Denmark. Results from MIKE 21 FM are utilized to validate our model. To assess uncertainties in model pre-

dictions, sensitivity of flooding results to extreme sea levels, rainfall and mesh resolution has been undertaken. The use of anisotropic-DMO enables us to capture high resolution topographic features (buildings, rivers and streets) only where and when is needed, thus providing improved accurate flooding prediction while reducing the computational cost. It also allows us to better capture the evolving flow features (wetting-drying fronts).

To provide real-time spatio-temporal flood predictions, an integrated long short-term memory (LSTM) and reduced order model (ROM) framework has been developed. This integrated LSTM-ROM has the capability of representing the spatio-temporal distribution of floods since it takes advantage of both ROM and LSTM. To reduce the dimensional size of large spatial datasets in LSTM, the proper orthogonal decomposition (POD) and singular value decomposition (SVD) approaches are introduced. The performance of the LSTM-ROM developed here has been evaluated using Okushiri tsunami as test cases. The results obtained from the LSTM-ROM have been compared with those from the full model (Fluidity). Promising results indicate that the use of LSTM-ROM can provide the flood prediction in seconds, enabling us to provide real-time flood prediction and inform the public in a timely manner, reducing injuries and fatalities.

Additionally, data-driven optimal sensing for reconstruction (DOSR) and data assimilation (DA) have been further introduced to LSTM-ROM. This linkage between modelling and experimental data/observations allows us to minimize model errors and determine uncertainties, thus improving the accuracy of modelling. It should be noting that after we introduced the DA approach, the prediction errors are significantly reduced at time levels when an assimilation procedure is conducted, which illustrates the ability of DOSR-LSTM-DA to significantly improve the model performance. By using DOSR-LSTM-DA, the predictive horizon can be extended by 3 times of the initial horizon. More importantly, the online CPU cost of using DOSR-LSTM-DA is only $1/3$ of the cost required by running the full model.

Table of Contents

Acknowledgments	i
Abstract	iii
List of Tables	ix
List of Figures	x
Chapter 1: Introduction	1
1.1 Motivation and objectives	1
1.2 Contributions	4
1.3 Thesis outline	6
Chapter 2: Anisotropic mesh adaptivity and machine learning based reduced order models (ROMs)	8
2.1 Anisotropic mesh adaptivity	8
2.1.1 Mesh operations in anisotropic mesh adaptivity	9
2.1.2 Anisotropic dynamic mesh optimization technique (anisotropic-DMO)	11
2.2 Recurrent Neural Networks (RNNs)	13
2.3 Proper orthogonal decomposition (POD) based reduced order model (ROM)	17
2.3.1 Conceptual introduction to POD-based ROM	18

2.3.2	Extracting information using POD	19
2.3.3	Building the POD-based ROM	20
Chapter 3: Anisotropic dynamic mesh optimization for urban flood modelling .		22
3.1	Introduction	23
3.2	Governing equations	25
3.2.1	Momentum and continuity equations	26
3.2.2	Drag coefficient	27
3.2.3	Boundary conditions for the joint flooding events	28
3.3	Applicability of the anisotropic-DMO technique and its application to Glas- gow case	29
3.3.1	Descriptions of study site and data	29
3.3.2	Results and discussion	30
3.3.3	Summary	40
Chapter 4: Numerical simulation of floods from multiple sources using anisotropic dynamic mesh optimization method		41
4.1	Introduction	42
4.2	Descriptions of study site and data	43
4.3	Extreme sea-level and rainfall events	45
4.4	Model applications	47
4.5	Results and Discussion	48
4.5.1	Individual flooding events	48
4.5.2	Joint flooding events	52
4.5.3	Impact of buildings on flooding simulations	52

4.5.4	Sensitivity to the forcing inputs and mesh resolution	59
4.5.5	Performance of Floodity modelling	60
4.6	Summary	62
Chapter 5: Rapid spatio-temporal flood prediction and uncertainty quantification using a deep learning method		64
5.1	Introduction	65
5.2	Construction of the POD-based ROM	68
5.3	LSTM-ROM for predictive modelling of the spatio-temporal distribution of floods	70
5.4	LSTM in predictive and prescriptive analytics of floods	72
5.4.1	Predictive analytics: the spatio-temporal model based on LSTM-ROM	72
5.4.2	Prescriptive analytics for uncertainty quantification in flooding modelling	74
5.5	Numerical examples	76
5.5.1	Predictive analytics in flooding modelling	77
5.5.2	Prescriptive analytics for uncertainty quantification in flooding modelling	82
5.6	Summary	84
Chapter 6: An adaptive POD-LSTM predictive model using data-driven optimal sensing with data assimilation		86
6.1	Introduction	86
6.2	LSTM-ROM predictive modelling for parameterized PDEs	87
6.3	Data-driven optimal sensing data assimilation framework	89
6.3.1	Data-driven optimal sensing for reconstruction (DOSR)	90

6.3.2	DOSR-based spatio-temporal prediction using LSTM and DA	93
6.4	Numerical examples	96
6.4.1	DOSR-based reconstruction	96
6.4.2	Spatio-temporal prediction using LSTM and DA	98
6.5	Summary	102
Chapter 7: Conclusions and recommendations		104
7.1	Conclusions	104
7.2	Recommendations	107
7.2.1	Parallelism of anisotropic-DMO	107
7.2.2	Applicability and robustness of ROM	109
7.2.3	Combination of ROM with LSTM Autoencoders	110
Bibliography		132

List of Tables

3.1	Node and element number of the 2D fixed and 2D adaptive unstructured mesh and the run time of the simulations.	38
4.1	Adapting mesh schemes for the Floodity simulations.	48
4.2	Node and element number of the adaptive unstructured mesh with mesh resolution 20 m , 10 m , and 5 m for Floodity modelling.	60
5.1	The RMSE errors of free surface heights between the original full model and LSTM-ROM during predicted period $[67.5, 70]$ s	82

List of Figures

2.1	Element operations used to optimize the mesh in two dimensions. (a) Edge split; (b) Edge collapse; (c) Edge swap; (d) Node movement.	10
2.2	A simple recurrent network [59].	14
2.3	One cell in the LSTM network.	16
3.1	LiDAR DTM with buildings of modeling domain and locations of monitoring stations.	30
3.2	Multi-scale unstructured mesh with $2\text{ m}/5\text{ m}/20\text{ m}$ resolution generated by Gmsh [138].	30
3.3	Water depth obtained from 3D modelling (left column, see Zhang et al. (2016) [138]), 2D fixed unstructured mesh modelling (middle column) and 2D adaptive unstructured mesh modelling (right column) at time level $t = 20\text{ min}$ (first row), 30 min (second row), 40 min (third row) and 60 min (bottom row). Water accumulates in the low-lying area (marked with a yellow rectangle) and around buildings (marked with green rectangles). . .	32
3.4	Velocity results obtained from 3D modelling (left column, see Zhang et al. (2016) [138]), 2D fixed unstructured mesh modelling (middle column) and 2D adaptive unstructured mesh modelling (right column) at time level $t = 20\text{ min}$ (first row), 30 min (second row), 40 min (third row) and 60 min (bottom row).	33
3.5	Buildings gradually become visible as the flood water spreads west and southward. The left column shows the plane view of surface topography. The right column shows the corresponding mesh. Areas marked with green rectangles show the corresponding building areas in Fig. 3.3 (marked with green rectangles as well).	34

3.6	Error in bathymetries with the use of adaptive meshes during simulations. The left column shows the plane view of water depth for adaptive mesh modelling. The right column shows the corresponding the error of the bathymetries used at time levels (20, 25, 30, 35, 60 <i>min</i>), where the high resolution (2 <i>m</i>) topographical data is a reference data.	35
3.7	Water depth in the areas marked with green rectangles in Fig. 3.3 (first and middle row), and details of bathymetry for this building area marked with green rectangles in Fig. 3.5 (bottom row). The right column shows the corresponding mesh.	36
3.8	Flood depth time series at detector locations STA1, STA2, STA3 and STA4 simulated by 2D fixed and 2D adaptive unstructured mesh models and numerous published models.	37
3.9	The numbers of nodes and elements used in 2D adaptive mesh modelling during the simulation period [0, 120] <i>min</i>	39
4.1	(a) Situation of study area in Greve, Municipality of Denmark. (b) DTM with buildings of study area (resolution of 1.6 <i>m</i> × 1.6 <i>m</i>) - generated by the GIS (Geographical Information System) software, see ArcGIS (2010) [161].	44
4.2	(a) Forecast of water levels issued at 12:00 on 29th November 2015. (b) Correlation coefficient between observed and predicted water levels at function of forecast time [163].	45
4.3	(a) Initial water depth in a scenario of 2100 upper extreme water level. (b) Future extreme seawater levels by 2100 (considering the worst climate change scenario for 100-yr projection), see Berbel Roman (2014) [163]. (c) Extreme 24 <i>h</i> design rainfall for 2-yr, 10-yr and 100-yr return period (considering the effects of climate change), see Berbel Roman (2014) [163].	46
4.4	Water depths obtained from a MIKE 21 FM model updated from Berbel Roman (2014) [163] (left column), a Floodity model with a mesh resolution 10 <i>m</i> (middle column) and results showing the corresponding mesh (right column) based on bathymetric data without buildings in a scenario of 2100 upper extreme sea-level event at time level $t = 2\ h$ (first row), 8 <i>h</i> (middle row), and 18 <i>h</i> (bottom row).	49
4.5	Flood volume obtained from a MIKE 21 FM model and a Floodity model with a mesh resolution 10 <i>m</i> based on bathymetric data without buildings in a scenario of 2100 upper extreme sea-level event.	50

4.6	Flood depth time series at detector locations P1, P2, P3 and P4 (see Fig.4.1 (b)) simulated using MIKE 21 FM and Floodity with a mesh resolution of 20 <i>m</i> , 10 <i>m</i> , and 5 <i>m</i> respectively, based on bathymetric data without buildings in a scenario of 2100 upper extreme sea-level event.	51
4.7	Water depths at time level $t = 15\ h$ obtained from a MIKE 21 FM model (first row), a Floodity model with a mesh resolution 10 <i>m</i> (middle row) and results showing the corresponding mesh (bottom row) based on bathymetric data without buildings in scenarios of an individual 2100 upper extreme sea-level event (left column) and a joint event with 100-yr return period rainfall (right column).	53
4.8	Flood depth time series at detector locations P1, P2, P3 and P4 (see Fig.4.1 (b)) simulated using MIKE 21 FM and Floodity with a mesh resolution of 20 <i>m</i> , 10 <i>m</i> , and 5 <i>m</i> respectively, based on bathymetric data without buildings in a scenario of 2-yr return period rainfall and 2100 upper extreme sea-level event.	54
4.9	Flood velocity time series at detector locations P1, P2, P3 and P4 (see Fig.4.1 (b)) simulated using MIKE 21 FM and Floodity with a mesh resolution of 10 <i>m</i> based on bathymetric data without buildings in a scenario of 2-yr return period rainfall and 2100 upper extreme sea-level event.	54
4.10	Water depths at time level $t = 15\ h$ obtained from a MIKE 21 FM model (first row), a Floodity model with a mesh resolution 5 <i>m</i> (middle row) and results showing the corresponding mesh (bottom row) based on bathymetric data with buildings in scenarios of an individual 2100 upper extreme sea-level event (left column) and a joint event with 100-yr return period rainfall (right column).	55
4.11	Details of the areas marked with rectangles (Fig.4.10) obtained from MIKE 21 FM modelling (left column), a Floodity model with a mesh resolution 10 <i>m</i> (middle column) and 5 <i>m</i> (right column). The bottom shows more details of the anisotropic unstructured meshes.	56
4.12	Buildings gradually become visible as the flood water spreads west and northward. The left column shows the plane view of surface topography. The right column shows the corresponding mesh.	57
4.13	Flood depth time series at detector locations P1, P2, P3 and P4 (see Fig.4.1 (b)) simulated using MIKE 21 FM and Floodity with a mesh resolution of 20 <i>m</i> , 10 <i>m</i> , and 5 <i>m</i> respectively, based on bathymetric data with/without buildings in a scenario of 100-yr return period rainfall and 2100 upper extreme sea-level event.	58

4.14	Sensitivity of flood volume results to forcing inputs in scenarios: (a) individual extreme sea-level events (shown in Fig.4.3 (b)); (b) both the individual 2100 upper sea-level flood event and the joint event with 100-yr return period rainfall; and (c) mesh resolutions of 20 m , 10 m , and 5 m	59
4.15	The numbers of nodes and elements used in Floodity modelling for mesh resolution 20 m , 10 m , and 5 m during the simulation period [0, 1440] min . .	61
5.1	Framework of LSTM-ROM.	75
5.2	The bathymetry and the three gauge stations used for the Hokkaido-Nansei-Oki tsunami example [213].	76
5.3	The series of flood induced waves for training purpose.	78
5.4	Singular values of free surface heights to construct the reduced space. . . .	78
5.5	The comparison of free surface heights between the full model and LSTM-ROM.	80
5.6	Differences of free surface heights between the full model and LSTM-ROM during the whole period.	81
5.7	The RMSE errors of free surface heights between the original full model and LSTM-ROM.	81
5.8	The free surface height at Gauge 2.	83
5.9	The input (left) and output (right) in prescriptive analytics.	83
6.1	Training and execution modules for data-driven optimal sensing data assimilation framework DOSR-LSTM-DA.	89
6.2	Framework of DOSR-LSTM-DA method.	95
6.3	The bathymetry for the Hokkaido-Nansei-Oki tsunami example [213]. . . .	97
6.4	POD singular values of water depth to construct the reduced space.	98
6.5	(a) ϕ_{mean} , mean of the 30 POD modal ensemble $\{\phi_m\}_{m=1}^{30}$; (b) Sensor locations extracted from ϕ_{mean}	98

6.6	Comparison of the water depth from the full model with the DOSR-LSTM using 30 POD modes with a predictive horizon of 1, 5, 13 and 15 time steps.	99
6.7	Comparison of water depth between the full model and the DOSR-LSTM-DA (left and middle column), as well as the differences (right column) after introducing DA method.	100
6.8	The RMSE of water depth between the DOSR-LSTM-DA and full model. .	101

Chapter 1

Introduction

1.1 Motivation and objectives

Flood disasters caused the heavy economic and human losses in recent years [1] and thus make flood protection measures becoming more important in flood management and decision-making. Modelling of flood scenarios plays a central role in the implementation of flood protection measures [2]. For example, the spatial distribution of flood depths and velocities, or flooding intensity predicted by the flood inundation models are necessary for the design of flood-risk maps. They are used for identification of the risk reduction measures and thereby help to ensure people's life property safety when flood occurs.

The non-linear shallow water equations (SWEs) are widely utilised in two-dimensional (2D) flood inundation modelling. For accurate flood inundation modelling, highly accurate terrain data are often needed in the form of Digital Terrain Models (DTMs). However, the computational cost of 2D flood simulation with high mesh resolutions is very high. To reduce the computational cost, various approaches to adaptive mesh resolution for improving computational efficiency have been presented for the SWEs [3–10]. A certain number of works [6, 11–15] focus on the adaptive techniques applied to the discretization of SWEs. However, isotropic adapted meshes are generally employed in most of the literature [6, 11, 14], whereas few studies focuses on an anisotropic mesh adaptation [12, 15]. This is due to the fact that the employment of anisotropic meshes usually involves more complex setting, even though the computational advantages yielded by the use of anisotropic meshes are already well proved in the literature [12, 15]. Mesh adaptivity technique is advantageous to improve the quality of an existing mesh using mesh modification operations, namely swap,

collapse, split and relocation. For instance, these mesh modification operations have been successfully applied in the previous studies [16–19], which are governed by a desired mesh size and shape distribution for adaptivity purpose in 2D [16–18] and 3D [19]. In dynamic anisotropic mesh adaptivity, complex geometry domains (e.g. domains with complex topographic coastlines, buildings or streets) can be accounted for, making features of flow better captured around complex topography. Moreover, the spatial meshes corresponding to different time steps are refined separately, which is necessary for construction of efficient discretizations for problems with complex dynamic behavior, making significant computational savings compared with applying static meshing techniques.

In this PhD study, we have developed a 2D double control-volume finite element (DCV-FEM) flood model using adaptive unstructured mesh technology [20]. This mesh adaptivity technique has the major advantage of the fixed order mesh modification method and it can enable elements with large angles to be used, when highly anisotropic elements are used. We also use flux limiting of the free surface height based on the Normalized Variable Diagram approach of Leonard (1988) [21]. In addition, we use a non-linear two-time level theta method for the time stepping method in which the value of θ is adjusted (in space and time) so that it is second-order accurate as much as possible. $\theta = 0.5$ (Crank Nicolson time stepping) but resorts to using $\theta = 1$ (backward Euler) if an oscillation is detected because of the time stepping. This is achieved based on a Total Variational Diminishing (TVD) in time condition [22]. The non-linear iteration is based on the fixed-point iteration method described in Salinas et al. (2017) [20]. This is important as wetting and drying is highly non-linear due to the non-linear drag and inertia. The Discontinuous Galerkin (DG) method applied to momentum/velocity is very powerful as it has a natural dissipation associated with it, which is highly scale selective.

In many computational models, the physical phenomena mainly simulated are mass, momentum, and energy transfer. This transport phenomena are typically described by a set of partial differential equations (PDE), discretizing the spatial (position) and temporal

(time) of the governing equations. A reliable representations of various simulated process (e.g. heat transfer, fluid flows, chemical processes) can be provided by the computational models. However, due to the complexity caused by non-linearities and large dimension, such computational models are usually time consuming and limited to be used in real time prediction. For example, the spatial domain has to be discretized into very small grid cells so that the continuous dynamics can be approximately represented reasonably. This induces a large number of grid cells, leading to a huge number of partial differential equations (equals to the number of grid cells) need to be solved at every time step. These high order or large dimension models are generally very slow. Thus, it is desirable to derive a simpler reduced order model which approximate the original high resolution computational model. It would be beneficial that the simpler models can efficiently provide reliable estimates of the current and future process in a real time mode and considering the original physical relationships as well. As a result, another subject of this PhD thesis is how to derive such simplified predictive models while significantly minimizing the computation costs.

There exists various approaches to build a simpler reduced order model (ROM) from a complex one. The method of Proper Orthogonal Decomposition (POD) is one of the most popular reduction methods. It was first introduced to the fluid dynamics by Lumley (1967) [23] as a mathematical technique to extract coherent structures from turbulent flow fields. POD is also known as principal component analysis (PCA). By giving an orthogonal set of basis vectors with the minimal dimension, POD is useful in constructing a ROM of the flow field. Additionally, POD modes are simple to compute using the method of snapshots, which is especially attractive for high-dimensional spatial datasets. POD modes have been used to construct Galerkin projection-based reduced order models for incompressible [24–27] and compressible [28] flows. In this PhD study, we consider the POD framework in combination with Recurrent Neural Network (RNN) architecture LSTM (long short-term memory). RNNs and other machine learning (ML) algorithms have been employed in

data-driven prediction applications and are gaining more attention in the computational physics. Studies into the feasibility of ML techniques for ROMs can be found in [29, 30]. While the POD model reduction framework has been used to derive ROMs of various nonlinear systems, here we attempt to model the dynamic features of floods using a RNN-based supervised machine learning framework. Our approach can be considered as a hybrid modelling framework combining machine learning strategies and physics-based simulation tools for dynamical systems.

1.2 Contributions

A 2D control-volume and finite-element flood model using adaptive unstructured mesh technology has been developed. This adaptive unstructured mesh technique enables meshes to be adapted optimally in time and space in response to the evolving flow features, thus providing sufficient mesh resolution where and when it is required. It has the advantage of capturing the details of local flows and wetting and drying front while reducing the computational cost. Complex topographic features are represented accurately during the flooding process. For example, the high-resolution meshes around the buildings and steep regions are placed when the flooding water reaches these regions. This mesh adaptivity technique has been first introduced to urban flooding simulations and applied to a simple flooding event observed as a result of flow exceeding the capacity of the culvert during the period of prolonged or heavy rainfall. Over existing adaptive mesh refinement methods (AMR, locally nested static mesh methods), this adaptive unstructured mesh technique can dynamically modify (both, coarsening and refining the mesh) and adapt the mesh to achieve a desired precision, thus better capturing transient and complex flow dynamics as the flow evolves.

The above adaptive mesh flooding model based on 2D shallow water equations (named as *Floodity*) has been further developed by introducing (1) an anisotropic dynamic mesh optimization technique (anisotropic-DMO); (2) multiple flooding sources (extreme rain-

fall and sea-level events); and (3) a unique combination of anisotropic-DMO and high-resolution Digital Terrain Model (DTM) data. It has been applied to a densely urbanized area within Greve, Denmark. Results from MIKE 21 FM are utilized to validate our model. To assess uncertainties in model predictions, sensitivity of flooding results to extreme sea levels, rainfall and mesh resolution has been undertaken. The use of anisotropic-DMO enables us to capture high resolution topographic features (buildings, rivers and streets) only where and when is needed, thus providing improved accurate flooding prediction while reducing the computational cost. It also allows us to better capture the evolving flow features (wetting-drying fronts).

Another significant contribution of this thesis is the novel combination of reduced order model (ROM) technique with machine learning algorithm for flood prediction. While most of these studies focused mainly on time-series flooding prediction at specified sensors, rarely on spatio-temporal prediction of inundations. In this work, an integrated long short-term memory (LSTM) and reduced order model (ROM) framework has been developed. This integrated LSTM-ROM has the capability of representing the spatio-temporal distribution of floods since it takes advantage of both ROM and LSTM. To reduce the dimensional size of large spatial datasets in LSTM, the proper orthogonal decomposition (POD) and singular value decomposition (SVD) approaches are introduced. The LSTM training and prediction processes are carried out over the reduced space. This leads to an improvement of computational efficiency while maintaining the accuracy. The performance of the LSTM-ROM developed here has been evaluated using Okushiri tsunami as test cases. The results obtained from the LSTM-ROM have been compared with those from the full model (Fluidity). In predictive analytics, it is shown that the results from both the full model and LSTM-ROM are in a good agreement whilst the CPU cost using the LSTM-ROM is decreased by three orders of magnitude compared to full model simulations. Additionally, prescriptive analytics has been undertaken to estimate the uncertainty in flood induced conditions. Given the time series of the free surface height at a specified detector, the

corresponding induced wave conditions along the coastline have then been provided using the LSTM network. Promising results indicate that the use of LSTM-ROM can provide the flood prediction in seconds, enabling us to provide real-time predictions and inform the public in a timely manner, reducing injuries and fatalities. Finally, data-driven optimal sensing for reconstruction (DOSR) and data assimilation (DA) have been introduced to LSTM-ROM. This linkage between modelling and experimental data/observations allows us to minimize model errors and determine uncertainties, thus improving the accuracy of modelling.

1.3 Thesis outline

Chapter 1 is an introduction including the motivation and objectives of developing DCV-FEM flood model and LSTM-ROM framework, as well as the original contributions and outline of the thesis.

Chapter 2 demonstrates the 2D double control-volume finite element (DCV-FEM) model using adaptive unstructured mesh technology, as well as the deep learning algorithm (LSTM) with reduced order modelling approach.

In Chapter 3, the DCV-FEM flood model is validated by a simple flooding event occurred in Glasgow, UK. It is observed as a result of flow exceeding the capacity of the culvert during the period of prolonged or heavy rainfall. This Chapter is derived from:

Hu, R., Fang, F., Salinas, P., & Pain, C. C. (2018). Unstructured mesh adaptivity for urban flooding modelling. *Journal of Hydrology*, 560, 354-363.

In Chapter 4, the DCV-FEM flood model has been further developed by introducing (1) an anisotropic dynamic mesh optimization technique (anisotropic-DMO); (2) multiple flooding sources (extreme rainfall and sea-level events); and (3) a unique combination of anisotropic-DMO and high-resolution Digital Terrain Model (DTM) data. This model has been successfully applied to an urbanized area in Greve, Denmark, where joint urban flood events were caused by multiple sources (extreme rainfall and sea-level events). This Chap-

ter is derived from:

Hu, R., Fang, F., Salinas, P., & Pain, C. C., Sto.Domingo N. D., Mark O. (2019). Numerical simulation of floods from multiple sources using an adaptive anisotropic unstructured mesh method. *Advances in Water Resources*, 123, 173-188.

Chapter 5 presents an integrated long short-term memory (LSTM) and reduced order model (ROM) framework for rapid spatio-temporal flood prediction. The performance of the LSTM-ROM developed here has been evaluated using Okushiri tsunami as test cases. The results obtained from the LSTM-ROM have been compared with those from the full model (Fluidity). This Chapter is derived from:

Hu, R., Fang, F., Salinas, P., & Pain, C. C., Navon, I. M. (2019). Rapid spatio-temporal flood prediction and uncertainty quantification using a deep learning method. *Journal of Hydrology*, 575, 911-920.

A further development of the LSTM-ROM framework is demonstrated in Chapter 6. In Chapter 6, a unique integrated DOSR-LSTM-DA framework is developed. It has the features 1) use of optimal sensing data where the sensor locations are optimized using POD; (2) POD modes are updated when the data is assimilated to modelling. This is the first work to introduce both data-driven optimal sensor techniques (DOSR) and data assimilation (DA) to LSTM-ROM.

Finally, Chapter 7 draws conclusions and findings of this thesis, and future research works are proposed.

Chapter 2

Anisotropic mesh adaptivity and machine learning based reduced order models (ROMs)

2.1 Anisotropic mesh adaptivity

Anisotropic phenomena occurs in many physical problems, especially in cases such as high Reynolds number flow problems and phase change problems. In this situation, the solution fields vary differently in all directions and equilateral elements are not suitable for representation. Using isotropic elements will hence induce the generation of small elements in all directions, resulting in large size of meshes. To overcome the limitations imposed by the computational resources while maintaining a satisfied accuracy of numerical solution, dynamic adaptive remeshing techniques are employed [31]. The methods have been successfully applied in 2D [16–18] and 3D [19] numerical modelling. Mesh optimisation is available by default in 3D using the algorithm developed by Pain et al. (2001) [19], while it is available in 2D using the Ani2D library developed by Vasilevskii and Lipnikov (1999) [32]. To discretize the computational domain, there are several ways of the mesh modification: (1) *h-adaptivity*: splitting the existing elements and changing the connectivity of the nodes [33]; (2) *r-adaptivity*: moving vertices of elements to make new elements while retaining the same connectivity [34]; (3) *p-adaptivity*: varying the polynomial order of the approximation [35].

More recently, discontinuous Galerkin methods (DG) have appeared as an alternative to finite volume or high-order finite difference schemes. These methods have also proven to be very effective to solve the long-wave or shallow water approximation within a finite

element framework [36, 37]. In this thesis, we propose the numerical solution of the SWE with the discontinuous Galerkin methods (DG) method and expand it with anisotropic mesh adaptivity techniques. It allows efficient h - and r -adaptivity that optimizes the mesh resolution using a metric tensor field approach based on linear interpolation error in prognostic fields. Dynamic mesh modification operations (swap, collapse, split and relocation) on elements [38] are involved in this method, and elements sizes can be controlled by specifying the minimum and maximum edge length [19]. In this method, complex geometry domains (e.g. domains with complex topographic coastlines, buildings or streets) can be accounted for, which enables dynamic features better captured around complex topography. Moreover, the spatial meshes corresponding to different time steps are refined separately, which is necessary for construction of efficient discretizations for problems with complex dynamic behavior, making significant computational savings compared with applying static meshing techniques.

2.1.1 Mesh operations in anisotropic mesh adaptivity

The anisotropic dynamic mesh optimization algorithm is performed by applying local modifications of elements in the existing mesh:

(1) *Edge split*: Refinement by splitting existing elements and inserting a node at the centre of an edge and surrounding elements are created [33] (Fig. 2.1 (a)).

(2) *Edge collapse*: Coarsening by edge collapse, removing an existing node by collapsing an existing edge to zero length and thus replacing two vertices by a single one lying at the edge midpoint (Fig. 2.1 (b)).

(3) *Edge swap*: Face-edge and edge-face swaps, reordering the connectivity of existing elements to improve the mesh shape, while the number of nodes and elements is preserved through the operation (Fig. 2.1 (c)).

(4) *Node movement*: Repositioning a vertex within the convex hull spanned by the elements which share it [34] (Fig. 2.1 (d)).

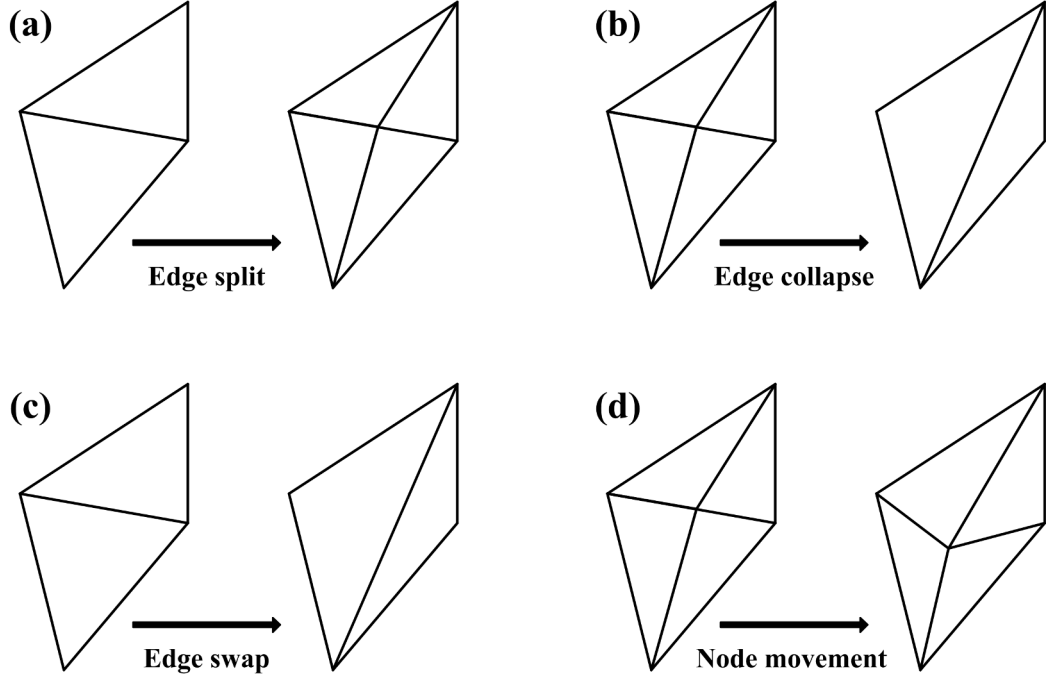


Figure 2.1: Element operations used to optimize the mesh in two dimensions. (a) Edge split; (b) Edge collapse; (c) Edge swap; (d) Node movement.

Anisotropic dynamic mesh optimisation (anisotropic-DMO) approaches based on these operations usually involve the definition of some objective function, a norm over the entire mesh of a local mesh quality estimate. By improving the worst elements through a series of elemental modifications to the mesh, the mesh optimisation aims to minimise the functional and hence improve the overall quality of the mesh. Anisotropic-DMO is achieved through the construction of error metric with which the objective functional is evaluated. The objective functional is defined to measure the quality of the mesh, namely decide which of the operations described above should be used to improve the quality of the mesh [16, 19, 39–41]. This allows local anisotropic features in a solution to be represented through a coordinate transformation. In anisotropic-DMO, element changes are made when they satisfy an acceptability criterion based on the behaviour of the estimated representation error functional. That is, when an element is selected, each of its vertices are tested to see if they can be collapsed. If no vertex collapse is performed then an edge swap is attempted

with each of the neighbouring elements. Finally, if none of these operations succeeds in reducing the functional, vertex node movement is applied. The proposed new position of a vertex is defined as:

$$\mathbf{v}_i = \frac{\sum_{j=1}^J \|\mathbf{v}_i - \mathbf{v}_j\| \mathbf{v}_j}{\sum_{j=1}^J \|\mathbf{v}_i - \mathbf{v}_j\|}, \quad (2.1)$$

where \mathbf{v}_j for $j = 1, \dots, J$ are the vertices connected to \mathbf{v}_i and $\|\cdot\|$ is the norm defined by the vertex centred metric. As with the other operations, this new position is only accepted if it improves the mesh. If the new position is rejected then the midpoint between this and the current location of the vertex is also tested.

When the algorithm terminates and a new target mesh has been calculated, the variables on the existing mesh are projected to it [42]. For more details regarding the dynamic mesh adaptivity algorithm refer to Pain et al. (2001) [19] and Piggott et al. (2009) [38]. The mesh adaptivity process must be repeated whenever the interpolation error estimate of the solution on the existing mesh grows significant.

2.1.2 Anisotropic dynamic mesh optimization technique (anisotropic-DMO)

The anisotropic-DMO has the advantage of capturing details of surface and local flows (wetting-drying front) during the process of flooding modelling. The use of anisotropic-DMO can efficiently provide a high mesh resolution where and when it is needed [43]. That is, finer meshes are placed only in specific regions where the variations of flow variable solutions are relatively large (e.g. flow around buildings and along the flooding paths), while coarser meshes are used in areas far from these regions, where inundation has not yet occurred for example.

Here, the anisotropic-DMO technique relies on the derivation of appropriate error measures, which dictate how the mesh is to be modified. The required error measure is defined in the form of a metric tensor. The metric is derived from a solution field variable and an error norm based on the interpolation error Pain et al. (2002) [19] defined to make sure that

a desired element length can be obtained while having a required interpolation error. Thus, the metric \widehat{M} is calculated from

$$\widehat{M} = \frac{\widehat{\gamma}}{\widehat{\epsilon}} |H|, \quad (2.2)$$

where $\widehat{\gamma}$ is a scalar constant, and $\widehat{\gamma} = 1$ is used here, $\widehat{\epsilon}$ is a required interpolation error, and H is the Hessian matrix for a specified field $\psi(\Omega)$ (here, the field of water depth):

$$H = \begin{pmatrix} \frac{\partial^2 \psi}{\partial x^2} & \frac{\partial^2 \psi}{\partial x \partial y} \\ \frac{\partial^2 \psi}{\partial y \partial x} & \frac{\partial^2 \psi}{\partial y^2} \end{pmatrix}. \quad (2.3)$$

The desired edge length, h_i , in the direction of the i^{th} eigenvector e_i , of the metric \widehat{M} , is defined as $h_i = 1/\sqrt{\lambda_i}$, i.e. anisotropic as well as inhomogeneous resolution results from a mesh that respects the metric \widehat{M} , where λ_i is the eigenvalue associated with e_i [38, 41].

It is advantageous to modify the metric to impose the maximum and minimum element sizes on the mesh, especially for problems that have known high-aspect ratio dynamics or domains. To impose these maximum and minimum constraints directionally, \widehat{M} is modified and defined as

$$M = V^T \wedge V, \quad (2.4)$$

where V is a rotation matrix that includes the eigenvectors of the metric \widehat{M} calculated from Eq. (2.2) and general directions for the maximum and minimum edge length can be introduced through the use of V .

To bound the aspect ratio of elements in physical space, there is a need to limit the ratio of eigenvalues. For example, to limit the aspect ratio of elements to be below a , the eigenvalues are modified as follows [19]:

$$\tilde{\lambda}_j = \max\{\lambda'_j, \frac{1}{a^2} \max_{j=1,2,3} \lambda'_j\} \forall j \in \{1, 2, 3\}, \quad (2.5)$$

where

$$\lambda'_j = \min\left\{\frac{1}{h_{min}^2}, \max\{|\lambda_j|, \frac{1}{h_{max}^2}\}\right\} \forall j \in \{1, 2, 3\}. \quad (2.6)$$

in which a is the maximum aspect ratio. h_{min} and h_{max} are the minimum and maximum element sizes. A uniform isotropic element can be transformed to an adapted anisotropic element under the transformation $S = V\sqrt{|\wedge|}^{-1}$ in the physical domain, achieving the desired interpolation error everywhere. A mesh adaptivity method similar to what described above has been implemented in Ani2D [32], which is used in this work.

The Galerkin interpolation technique [42] is used for interpolating the solutions from the previous mesh onto the newly adapted mesh. However, the interpolation error may destroy the conservation of quantities important to the physical accuracy of simulations, for example density and water height. To keep the conservation of quantities, the conservative interpolation operator with an intermediate supermesh is used, for details see Farrell and Maddison (2011) [42].

Davies et al. (2011) [44] shows that adapting every 5 time-steps incurs in a 2% extra-cost for their simulation. Since this value slightly varies when solving different problems (e.g. up to 8%), the rough figure is that the cost of adapting the mesh is around the same as doing one time-step. Therefore, adapting every 10 times-steps means an approximate 10% extra-cost. Based on heuristics for this case, the mesh is adapted every 10 time steps for the adaptive simulation, due to fact that the Courant number is never big enough such as the water front leaves the high-quality region of the mesh before the mesh is re-adapted. In general, the mesh is adapted more frequently if the flow feature is changed rapidly.

2.2 Recurrent Neural Networks (RNNs)

Machine learning is a field of statistical research for training computational algorithms that split, sort and transform a set of data to maximize the ability to classify, predict, cluster or discover patterns in a target dataset [45]. Deep learning refers to machine learning al-

gorithms that construct hierarchical architectures of increasing sophistication, for example, the artificial neural networks (ANNs) [46] with many layers. Nowadays, ANNs are employed by a large number of studies concerning training, structure design, and real world applications, ranging from classification to robot control or vision [47]. In recent years, deep-NNs (including recurrent ones) have shown significant advantages in pattern recognition and machine learning. Deep-NNs perform well in discovering intricate structures in high-dimensional data and are therefore applicable to various areas including, science, business and government [48]. They have been widely applied to image recognition [49–51], speech recognition [52–54], natural language understanding [55], language translation [56], as well as analysing particle accelerator data [57] and reconstructing brain circuits [58].

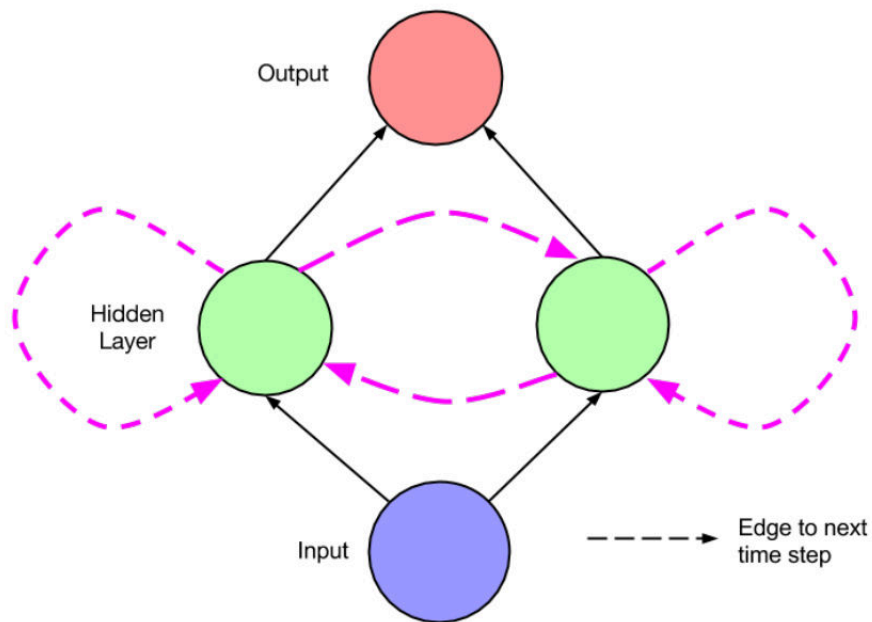


Figure 2.2: A simple recurrent network [59].

A standard neural network (NN) consists of many simple, connected neurons, each producing a sequence of real-valued activations [60]. Input neurons get activated through sensors perceiving the environment, other neurons get activated through weighted connections from previously active neurons. The process of learning is to find weights that make

the NN present desired behaviour. Usually, long chains of computational stages are required to obtain such behaviour and each stage contains transformations of the aggregate activation for the network. Recurrent neural networks (RNNs) are feed-forward neural networks augmented by the inclusion of edges that span adjacent time steps, introducing a notion of time to the model [59]. Similar to feed-forward networks, RNNs do not have cycles among conventional edges. However, edges that connect adjacent time steps (recurrent edges) form cycles, including cycles of length one that are self-connections from a node to itself across time. Fig. 2.2 shows how the network can be interpreted as a deep network with one layer per time step and shared weights across time steps. At each time step t , activation is passed along solid edges as in a feed-forward network. Dashed edges connect a source node at each time t to a target node at each following time $t + 1$. The unfolded network can be trained across many time steps using backpropagation, namely backpropagation through time (BPTT) [61]. It is a commonly used algorithm for recurrent networks.

RNNs can in principle create and process memories of arbitrary sequences of input patterns [62, 63]. More importantly, compared with traditional methods for automatic sequential programming [64–66], RNNs can learn from mixed sequential information and process in a natural and efficient way, sustaining a rapid reduction of computation cost. Over existing recurrent neural networks, LSTM (Long Short Term Memory) has an internal memory system to deal with temporal sequence inputs, which makes it a powerful tool for flooding predictions. A LSTM module is composed of a memory cell and three gates: an input gate, a forgetting gate and an output gate. The input gate controls the information transmitting from the input activations into the cell, and the forget gate and output gate select the information needed to be transmitted into next cell. Fig.1 shows how information is flowing through a LSTM cell via the cell state, which is considered as a conveyor belt in the network. The relationship between the input (x_s) and output (h_s) can be obtained as

follows:

$$\begin{aligned}
f_s &= \sigma(W_f \cdot [h_{s-1}, x_s] + b_f), \\
i_s &= \sigma(W_i \cdot [h_{s-1}, x_s] + b_i), \\
\tilde{C}_s &= \tanh(W_c \cdot [h_{s-1}, x_s] + b_c), \\
C_s &= f_s * C_{s-1} + i_s * \tilde{C}_s, \\
o_s &= \sigma(W_o \cdot [h_{s-1}, x_s] + b_o), \\
h_s &= o_s * \tanh(C_s),
\end{aligned} \tag{2.7}$$

where i_s , o_s and f_s represent the input, output and forget gates respectively, σ is the gate activation function, W_i, W_o, W_f and W_c are the weights for each of the gates and cells, b represent bias weights, C denotes the cell state, \tilde{C} is the updated cell state, x_s and h_s are the cell input and output respectively where the subscripts s and $s - 1$ denote the current and previous predictive instances respectively. With transmitting of the information, the cell states are propagated forward and weights are updated through time.

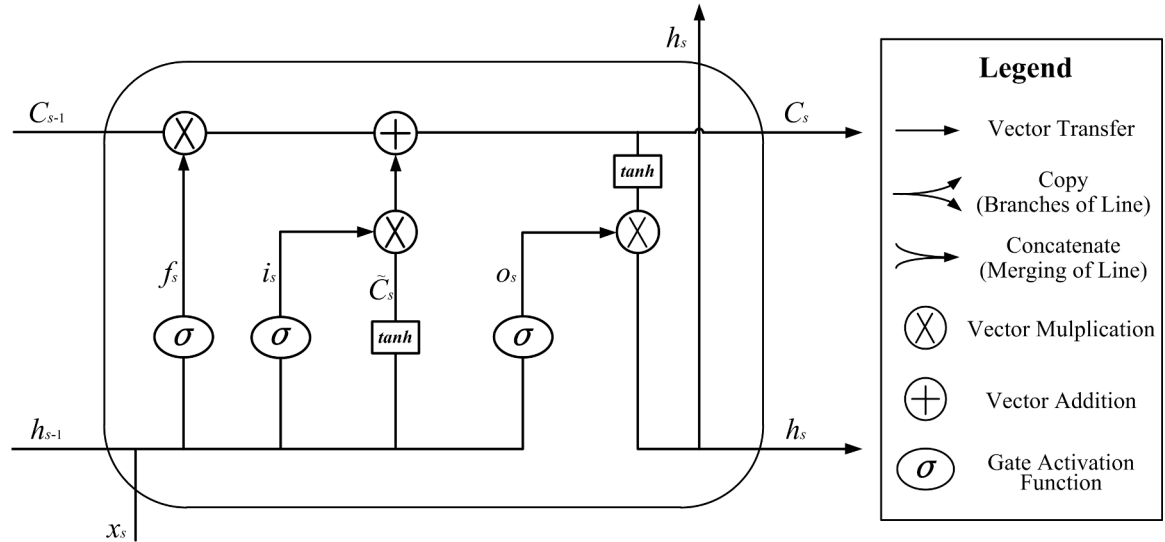


Figure 2.3: One cell in the LSTM network.

2.3 Proper orthogonal decomposition (POD) based reduced order model (ROM)

The high-dimensional models require high computational cost. Although a lot of efforts have been made to the numerical modelling of physical processes to enhance the compatibility of the real process with the simulated one, the models remain computationally intensive. This problem motivates the development of reduced order models (ROMs), which are less complex than the original computational models, especially those governed by partial differential equations (PDEs). The ROMs can be divided into two categories in terms of the dependency on the governing equations: intrusive ROM [67] and non-intrusive ROM [68]. Intrusive ROM is dependent of governing equations and source code, thus retaining much of the physical characteristics from the original system due to its intrusiveness. In contrast, the non-intrusive reduced order models (NIROMs) require no knowledge of the physical systems. It is suitable for representing high-dimensional physical systems while the source code describing the physical model is not available or the modifications required to generate the ROM are complex. Recently, the neural networks have been introduced into ROMs [69–71], which utilizes observed data to approximate an arbitrary function. A wide applications has been made in the field of fluid dynamics [72–76].

In this work, we focus on reduced order model (ROM) in which the dominant characteristics of fluid dynamics can be efficiently represented by linear combinations of a small set of basis functions. Most of the system information are projected onto a low-dimensional subspace spanned by these basis functions via Galerkin projection. Various approaches have been developed to compute the basis functions, including Krylov-subspace methods [77–79], reduced-basis methods [80, 81] and proper orthogonal decomposition (POD) [82, 83]. POD in combination with the Galerkin projection method is an effective method for deriving a reduced order model (ROM). It has proven to be an efficient means of deriving a reduced basis for high-dimensional nonlinear flow systems. The POD technique and

its variants have been applied successfully to a numerous research fields such as: ocean modelling [68, 84, 85], air pollution [86], fluids control [87–91] and data assimilation [92–95]. It is a widely used reduced order modelling approach which is employed via snapshot method, namely, the POD bases are derived from a set of the full state solutions (snapshots) obtained by solving the full order models. By applying singular value decomposition (SVD) to the snapshots, POD retains the dominant left singular vectors corresponding to the largest singular values as reduced basis. For example, for non-linear problems with the finite element framework, when considering a mesh having N nodes, one must compute N values at each time step. This means that the solution of at least one linear algebraic system of size N need to be calculated at each time step, making it computationally expensive when N increases. Thus it is reasonable to look for a reduced order model whose solution is efficiently obtained in a subspace of dimension much smaller than N . This is the main idea behind the POD reduced modelling approach, which will be explained in what follows.

2.3.1 Conceptual introduction to POD-based ROM

POD is based on patterns generated by the experiments or simulation data [96]. This data is collected from of physical process, such as variations of the temperature, chemical compositions, flow rate, and many others. Because of the regularities presented by the collected data, there exists a certain pattern which keeps repeating itself from one time period to the next time period. If the original model is governed by partial differential equations (PDEs), then the spatial domain is discretized into a high number of grid cells to approximate the continuous spatial domain as good as possible. For example, for a physical problem of temperature distribution along the slab, the temperature at every time step and every location can be expressed, or accurately approximated as functions of the patterns. Denote $T(t)$ the values of the temperature distribution along the slab at every time step, it can then be

expressed as the linear combination of N patterns:

$$\mathbf{T}(t) = a_1(t)\phi_1 + a_2(t)\phi_2 + \cdots + a_N(t)\phi_N, \quad (2.8)$$

where $\mathbf{T}(t)$ is the vector of a physical variable (here, temperature) over the whole spatial domain and at time step t and $\{\phi_i\}_{i=1}^N$ denote the spatial patterns, which are called the basis functions or the modes, and $\{a_i\}_{i=1}^N$ is time varying coefficient. Hence, the temperature is thus decomposed into a linear combination of spatial patterns. The basis functions are orthogonal to each other, namely that they are independent of each other. Associated with each pattern, there is a time-varying coefficient which varies according to the temperature distribution at time t . Thus, if the time varying coefficient is available, we can approximate the original variable based on Eq. (2.8). In POD, the basis functions are derived from the data. Assume the number of patterns can be reduced to K patterns such that $\mathbf{T}(t)$ can be expressed as a linear combination of K patterns:

$$\mathbf{T}(t) \approx a_1(t)\phi_1 + a_2(t)\phi_2 + \cdots + a_K(t)\phi_K, \quad (2.9)$$

where K is substantially smaller than N in Eq. (2.8) and $\{a_i\}_{i=1}^K$ is time varying coefficient which is slightly different from that in Eq. (2.8). By constructing a model for the first K time varying coefficient to obtain an approximate model of the process variable, the process variable can be expressed as a linear combination of very few patterns. This is the basic idea of POD-ROM.

2.3.2 Extracting information using POD

Assume that $u(\mathbf{x}, t)$ is the known field of interest at the nodes \mathbf{x}_i of a spatial mesh for discrete times $t_n = n \cdot \Delta t$, with $i \in [1, \dots, N]$ and $n \in [0, \dots, P]$. Define the notation as $u(\mathbf{x}_i, t_n) \equiv u^n(\mathbf{x}_i) \equiv u_i^n$ and $\{\mathbf{u}\}^n$ is the vector of nodal values u_i^n at time t_n . To obtain the most characteristic structure $\phi(\mathbf{x})$ among these $u_n(\mathbf{x})$ [97], the scalar quantity a need

to be maximized

$$a = \frac{\sum_{n=1}^P [\sum_{i=1}^N \phi(\mathbf{x}_i) u^n(\mathbf{x}_i)]^2}{\sum_{i=1}^N (\phi(\mathbf{x}_i))^2}, \quad (2.10)$$

namely, the following eigenvalue problem need to be solved:

$$\mathbf{c}\phi = a\phi, \quad (2.11)$$

Where the vector ϕ has i -component $\phi(\mathbf{x}_i)$, and $\{\mathbf{c}\}$ is the two-point correlation matrix

$$c_{ij} = \sum_{n=1}^P u^n(\mathbf{x}_i) u^n(\mathbf{x}_j) = \sum_{n=1}^P \mathbf{u}_n \cdot (\mathbf{u}_n)^T, \quad (2.12)$$

which is positive definite and symmetric. Define the matrix $\{\mathbf{Q}\}$ as

$$\mathbf{Q} = \begin{bmatrix} u_1^1, & u_1^2 & \cdots & u_1^P \\ u_2^1, & u_2^2 & \cdots & u_2^P \\ \vdots & \vdots & \ddots & \vdots \\ u_N^1, & u_N^2 & \cdots & u_N^P \end{bmatrix}, \quad (2.13)$$

and we have

$$\mathbf{c} = \mathbf{Q} \cdot \mathbf{Q}^T. \quad (2.14)$$

2.3.3 Building the POD-based ROM

To build a POD-based ROM, we first solve the eigenvalue problem in Eq. (2.11) and chose the K eigenvectors ϕ_i associated with the eigenvalues belonging to the interval defined by the highest eigenvalue a_1 . Normally, K is much smaller than N . These K eigenfunctions ϕ_i are then used to approximate the solution $u_n(\mathbf{x})$. To this end, define the matrix $\{\mathbf{B}\} =$

$[\phi_1, \dots, \phi_K]$ as

$$\mathbf{B} = \begin{bmatrix} \phi_1(\mathbf{x}_1), & \phi_2(\mathbf{x}_1) & \cdots & \phi_K(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2), & \phi_2(\mathbf{x}_2) & \cdots & \phi_K(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_N), & \phi_2(\mathbf{x}_N) & \cdots & \phi_K(\mathbf{x}_N) \end{bmatrix}. \quad (2.15)$$

A linear algebraic system must thus be solved to calculate the discrete solution $\{\mathbf{u}\}^{n+1}$ at time t^{n+1} . Here, we consider the equations resulting from the discretisation of a partial differential equation (PDE) in the form:

$$\mathbf{G}^n \mathbf{u}^{n+1} = \mathbf{H}^n. \quad (2.16)$$

where, \mathbf{H}^n is a vector which contains the discretized sources and the terms within the matrix system account for the solution from the previous time step t^n , \mathbf{G}^n is the discretized matrix containing mass and other matrices (e.g. advection or pressure). A POD-ROM is thus obtained by approximating $\{\mathbf{u}\}^{n+1}$ in the subspace defined by the K eigenvectors ϕ_i :

$$\{\mathbf{u}\}^{n+1} \approx \sum_{i=1}^K \phi_i \zeta^{n+1} = \mathbf{B} \zeta^{n+1}. \quad (2.17)$$

Eq. (2.16) is then rewritten:

$$\mathbf{G}^n \mathbf{B} \zeta^{n+1} = \mathbf{H}^n, \quad (2.18)$$

or equivalently written as:

$$\mathbf{B}^T \mathbf{G}^n \mathbf{B} \zeta^{n+1} = \mathbf{B}^T \mathbf{H}^n. \quad (2.19)$$

Here, the coefficients ζ^{n+1} defines the solution of the ROM. Consequently, a POD-ROM is thus obtained by solving an algebraic system of size K instead of N , when $K \ll N$.

Chapter 3

Anisotropic dynamic mesh optimization for urban flood modelling

Over the past few decades, urban floods have been gaining more attention due to their increase in frequency. To provide reliable flooding predictions in urban areas, various numerical models have been developed to perform high-resolution flood simulations. However, the use of high-resolution meshes across the whole computational domain causes a high computational burden. In this section, a 2D double control-volume finite element (DCV-FEM) flood model using adaptive unstructured mesh technology has been developed. Over existing adaptive mesh refinement methods (AMR, locally nested static mesh methods), this adaptive unstructured mesh technique can dynamically modify (both, coarsening and refining the mesh) and adapt the mesh to achieve a desired precision, thus better capturing transient and complex flow dynamics as the flow evolves. By introducing an anisotropic dynamic mesh optimization technique (anisotropic-DMO) combined with the high-resolution Digital Terrain Model (DTM) data, the high resolution topographic features (buildings, rivers and streets) are captured only where and when is needed, thus providing improved accurate flooding prediction while reducing the computational cost. It also allows us to better capture the evolving flow features (wetting-drying fronts). To demonstrate the capability of this new method, it has been successfully applied to a simple study case, a flooding event observed as a result of flow exceeding the capacity of the culvert during the period of prolonged or heavy rainfall, which happened in 2002 in Glasgow, Scotland, United Kingdom.

3.1 Introduction

Flood disaster is one of the most influential natural hazards in history [98]. Over the past few decades, the frequency of urban flooding has increased, due to the increasing urbanization, aging sewer networks and climate change threats [99]. This has drawn more attention in urban flooding research, hence increasing the effort in flood modelling [100–103]. To provide reliable flooding predictions in urban areas, high-resolution simulation is essential in order to resolve the complex urban topographic features, for example, buildings, streets and embankments. However, the high computational burden associated with full hydrodynamic models has restricted their wider applications to real-time urban flood modelling.

For efficient and accurate flood inundation modelling, numerous methods including grid coarsening methods [104], cellular automata approach [105], and speeding-up strategies (e.g. parallel processing) have been developed. Chen et al. (2012) [106] used a Building Coverage Ratio (BCR) and the Conveyance Reduction Factor (CRF) parameters to simplify the key features of building within a coarse grid. Leandro et al. (2014) [107] developed a parallelized two-dimensional diffusive wave model (P-DWave) with an adaptive time step using the Matlab parallel computing toolbox and Fortran OpenMP Application Programming. Smith et al. (2015) [108] presented a new hydrodynamic modelling framework and described how a robust finite volume Godunov-type scheme was implemented and applied it to urban flooding with a high-resolution grid. Parallel computation was achieved with either central processing units (CPU) or graphics processing units (GPU) devices.

The use of a uniform high-resolution mesh across the whole computational domain may cause the simulation to run in an unacceptably slow speed [109]. It is desirable to apply fine meshes only in specific regions for example, where complex dynamical flows (shock waves, eddies etc.) occur, while coarse meshes are used in the rest of the computational domain, especially in the area where inundation has not yet occurred [110]. Adaptive mesh

refinement (AMR) (a fine structured mesh nested within a coarse mesh) technique was developed by Berger and Oliger (1984) [111] and Berger and Colella (1989) [33]. George (2011) [112] applied the AMR technique to dam break flow modelling. An extension of AMR using the adaptive quadtree grids approach was proposed and tested using different numerical schemes [5, 113–115]. Huang et al. (2015) [116] further applied AMR to coupled flood and sediment transport modelling.

In this thesis, we have introduced an advanced optimization based adaptive mesh technique [19] to flooding modelling. In comparison to AMR (locally nested static mesh methods), this adaptive unstructured mesh technique can dynamically modify and adapt the mesh to achieve a desired level, thus better capturing transient and complex flow dynamics as the flow evolves. Using the optimization-based adaptive technique, the mesh nodes can either be increased or decreased locally in time and space (h-adaptive technique) with a good solution accuracy [40], or optimally relocated (r-adaptive technique) to resolve the small-scale flow features in a domain of interest (e.g. features of flow around buildings). This dynamically adaptive mesh technique has been applied to idealistic oceanic cases (without real bathymetry and topography), air pollution, multiphase flows and reservoir modelling [117–121].

This is the first time to apply this optimization-based adaptive mesh technique to flooding modelling. Unstructured meshes are used for optimal representation of complex domain geometries and boundaries. One of key issues in flooding modelling is the representation of wetting-drying (WD) fronts. As reviewed in Medeiros and Hagen (2013) [122], simulating the WD front over a real domain is still nontrivial, due to the fact that accurate solutions requiring high spatial and temporal resolutions are unstable and computationally expensive. The use of adaptive meshes enables the models to capture the physics of an advancing or receding wetting front better, while keeping the computational cost low.

In this thesis, our newly developed 2D double control-volume finite-element (CV-FE) shallow water model together with the adaptive unstructured mesh technique has been suc-

successfully applied to the Glasgows urban flooding event of 2002. The performance of adaptive unstructured meshes in flood modelling has been evaluated. The adaptive mesh simulations provide comparable results to the higher resolution 2D/3D fixed mesh simulations whilst reducing a 20 – 84% the computational cost.

3.2 Governing equations

Here, we have adopted the element pair P_1 DG- P_1 CV [123] (a modification of linear discontinuous velocity and continuous pressure representations) for 2D shallow water simulations. In DCV-FEM scheme used here, the pressure (free surface height) is discretized CV-wise rather than FE-wise in the classic CV-FEM [121, 124–127]. The DCV-FEM provides significant improvements in the quality of the pressure matrix that can be solved efficiently even on highly anisotropic elements. We also use flux limiting of the free surface height based on the Normalized Variable Diagram (NVD) approach of Leonard (1988) [21].

The Discontinuous Galerkin (DG) method is used for the discretization of the momentum equation. The DG approach is very powerful as it has a natural dissipation associated with it. To robustly stabilize the shallow water momentum equation and remove unwanted oscillation for complex issues (e.g. shock waves), a non-linear Petrov-Galerkin scheme [128, 129] with a DG discretization is used here, instead of using the flux limiter with a DG discretization. It is a mathematically consistent residual scheme and converges to the governing equations as the mesh and time step size are refined [130]. Due to a diffusion term proportional introduced to the residual of the momentum equation, it provides the robustness needed when there are sharp changes in velocity (usually occurs near wetting and drying fronts).

A non-linear θ method is used for the time discretization in which the value of θ (between 0.5 and 1) is adjusted in space and time. θ is calculated at each CV face based on the satisfaction of a Total Variational Diminishing (TVD) criterion [22]. The non-linear iteration is based on the fixed-point iteration method described in Salinas et al. (2017) [123].

This is important as wetting and drying (due to the non-linear drag and inertia) is highly non-linear.

3.2.1 Momentum and continuity equations

For depth averaged velocity \mathbf{u} in non-conservative form, the momentum equation or shallow water is written as:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + C_f \mathbf{u} - \mu \nabla^2 \mathbf{u} = -\nabla p + \mathbf{s}_b, \quad (3.1)$$

and the continuity equation is written as:

$$\frac{\partial h}{\partial t} + \nabla \cdot (h \mathbf{u}) = s_h, \quad (3.2)$$

where h is the water depth, C_f is the volumetric drag coefficient, μ is the dynamic viscosity, p is the depth averaged pressure, \mathbf{s}_b is the source term of velocity (unit: $m s^{-2}$), s_h is the source term of mass (from rainfall for example, unit: $m s^{-1}$), and p is calculated:

$$p = g(h + d), \quad (3.3)$$

and thus

$$h = \frac{p}{g} - d, \quad (3.4)$$

where g is the gravitational acceleration, and d is bathymetry (the height deviation from a horizontal and flat plain). The pressure and free surface height are defined CV-wise, contrary to the classical CV-FEM, where they are FE-wise.

3.2.2 Drag coefficient

A commonly used bottom stress parameterization is the Manning-Strickler formulation [131, 132]:

$$\mathbf{n} \cdot \nu \nabla \mathbf{u} = n_m^2 g \frac{|\mathbf{u}| \mathbf{u}}{h^{\frac{1}{3}}}, \quad \text{on } \Gamma_{bottom}, \quad (3.5)$$

in which \mathbf{n} is the unit normal to the bottom surface Γ_{bottom} normal, ν is the kinematic viscosity, and n_m is the Manning coefficient. The formulation for the volumetric drag coefficient C_f is:

$$C_f = n_m^2 g \frac{|\mathbf{u}|}{\max\{h, h_{min}\}^{\frac{1}{3}}}. \quad (3.6)$$

Wetting and drying: Here we use the thin film wetting and drying algorithms, which specify a minimum threshold depth that defines the categories of wet or dry in the model. Importantly, we have introduced the flux limiter to ensure the positive water depth and avoid the physical oscillation. $h_{min} = 0.01 \text{ mm}$ was finally chosen as considering a layer of water of 0.01 mm in the dry areas is physically plausible, while also reducing the non-linear behaviour which may be introduced by a smaller value (e.g. 0.001 mm). In order to enable the CVs to wet and dry freely, we perform an element wise average of the node-wise depths of water within each element. We then use this average in Eq. (3.6).

Representation of buildings: The most common methods for simulating the water flow among structures are: 1) blocking-out of the solid area; 2) local elevation rise of the solid area; 3) local increase of roughness of the solid area either via the Manning coefficient increase [133]. In this work, the elevation and shape of buildings are embedded in the realistic topography data, which is similar to the local elevation rise, namely that locations with buildings have the actual high elevation values. Without smoothing the walls, which are vertical in reality, there thus will be a jump in the bathymetry and by resolving the mesh these should get more steep and eventually discontinuous in the model. To deal with

this issue, a DG discretization field for bathymetry is used, which enables having finite resolution in an assumed continuous bathymetric field. Additionally, we increase roughness of the building areas via increasing Manning coefficient. With the use of anisotropic-DMO, the details of buildings can be captured as the water floods around them.

3.2.3 Boundary conditions for the joint flooding events

The boundary conditions need to be set for the water depth h , pressure p or velocity \mathbf{u} . We can define either pressure or velocity boundary conditions but not both [134]. Here we specify the pressure p boundary conditions and then the velocity is calculated from the pressure gradient.

h boundary conditions: For the joint flooding events involved with pluvial and coastal flooding, the water depth h boundary conditions along the coastline include: 1) time series of extreme water level heights e estimated from storm-surge models or the historical data observed; 2) bathymetry (terrain elevation) d along the coastline. Thus, the water depth h is calculated from $h = e - d$.

p boundary conditions: Once the h boundary conditions are set, the pressure p boundary conditions are specified through Eq. (3.3), and the velocity will be calculated from the pressure gradient [134] and evolve to point into or out of the domain (depending on the dynamics). The boundaries (except the coastline) over the domain can be set up as closed or open based on actual situation.

Rainfall as source term: The rainfall needs to be considered as source term s_h in Eq. (3.2). Generally, given a large domain (a large catchment), where the characteristics of the subcatchments are found to be significantly different from each other, the effect of spatial distribution of rainfall is considered. This mean that different subcatchments may receive different amounts of rainfall, namely different rainfall intensity time series are induced according to the region. However, since the computation domain in this study is relatively small, we assume that the rainfall is uniform across the domain. Due to the fact that the

effect of rainfall is relatively small compared with that of incoming waves, infiltration is not considered here. An average depth of runoff is then obtained, which contributes to the calculation of the water depth h .

3.3 Applicability of the anisotropic-DMO technique and its application to Glasgow case

3.3.1 Descriptions of study site and data

To assess the performance of adaptive meshes in flooding modelling, the new flooding model has been applied to an urban area located within the city of Glasgow, Scotland, UK, where a flood event occurred in July 2002. Here, the field of water depth is selected to be the adapted field using anisotropic-DMO technique. The whole computational domain is 1.0 km by 0.4 km (Fig. 3.1). This flood is observed as a result of flow exceeding the capacity of the culvert during the period of prolonged or heavy rainfall. The culvert is located at the northeast corner of the domain (see location Q in Fig. 3.1). Once the capacity of the culvert is exceeded, water overflows from the culvert and spreads over the west and south urban area along the main roads.

The details of hydrographic data can be found in literatures [135, 136]. The raw LiDAR data was originally collected by Infoterra Ltd., Leicester, UK, for Glasgow City Council. For hydraulic modelling, Infoterra aggregated the LiDAR data and reinserted buildings, kerbs and roads to obtain a 2 m DEM with realistic representation of urban morphologic characteristics [135]. The time series of flooding water depth at four detector locations (Fig. 3.1) obtained by Hunter et al. (2008) [135] are utilized for model verification in this study.

Fig. 3.2 shows the unstructured meshes created as 2D (triangle elements) by Gmsh [137, 138], a free finite element grid generator with a build-in CAD engine and post-processor. It can be seen how the surrounding mesh of the main street and the branches is

refined.

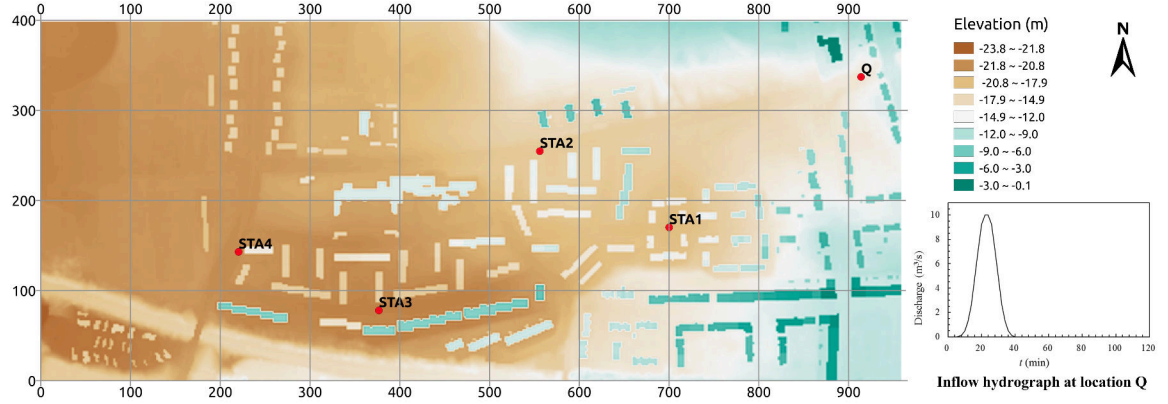


Figure 3.1: LiDAR DTM with buildings of modeling domain and locations of monitoring stations.

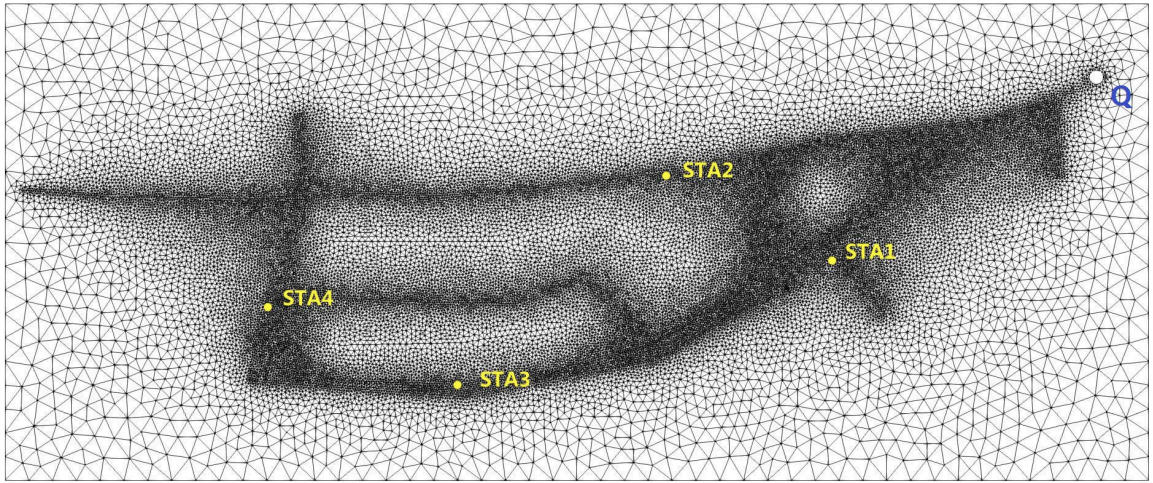


Figure 3.2: Multi-scale unstructured mesh with $2\text{ m}/5\text{ m}/20\text{ m}$ resolution generated by Gmsh [138].

3.3.2 Results and discussion

A series of model simulations using both the fixed and adaptive unstructured meshes have been carried out to assess the performance of the new flooding model developed here. In these simulations, water enters the densely urbanized area from a culvert at location Q (Fig. 3.1). The flood condition has been described earlier. As specified by the hydrographic

data used in previous studies [135, 136], the inflow discharge from the culvert started at $t = 5 \text{ min}$, peaked between 22 and 24 min , and ended at $t = 40 \text{ min}$. A fixed time step size $\Delta t = 0.15 \text{ s}$ is used in all simulations. No normal flow boundary condition is enforced at all the external boundaries. For mesh adaptivity, the aspect ratio of the elements in the adapted mesh is set to 5. For mesh size constraints, the maximum and minimum element sizes are 50 and 2 m respectively. The mesh is adapted considering the solutions of both the water depth and velocity, being the absolute interpolation errors set to 0.05 and 0.14 for the water depth and velocity respectively.

Water depth and velocity

For comparison purposes, 3D results from Zhang et al. (2016) [138] are used as a reference solution in this study. These 3D results have been proved to be consistent with 2D published results [135] in the region where the impact of 3D flow structures can be ignored.

Fig. 3.3 and 3.4 show the results of water depth and velocity from our newly developed 2D flooding model with both the fixed and adaptive unstructured meshes at time levels $t = 20, 30, 40, 60 \text{ min}$, in comparison to those results from 3D modelling [138]. They show the flood propagation process over the urban area. It can be observed that in most of the inundation area, the solutions of water depth and velocity obtained from both 2D fixed and adaptive mesh modelling are in good agreement with those of 3D modelling. As seen in Fig. 3.4, the flood propagation process is accelerating during [22, 24] min when the inflow discharge at location Q peaks. The water spreads along the main street and branches when $t = 30 \text{ min}$. During the flood recession (after $t = 40 \text{ min}$), water accumulates in the low-lying areas, especially in the southern street area marked with a yellow rectangle in Fig. 3.3. The variation of results in the speed and extent of flooding is relatively small after 45 min . We, therefore, discontinue adapting the mesh from this time level and run it with the mesh already generated at this point until the end of the simulation. This helps us to save the overhead and extra computational cost introduced during the adaptive mesh

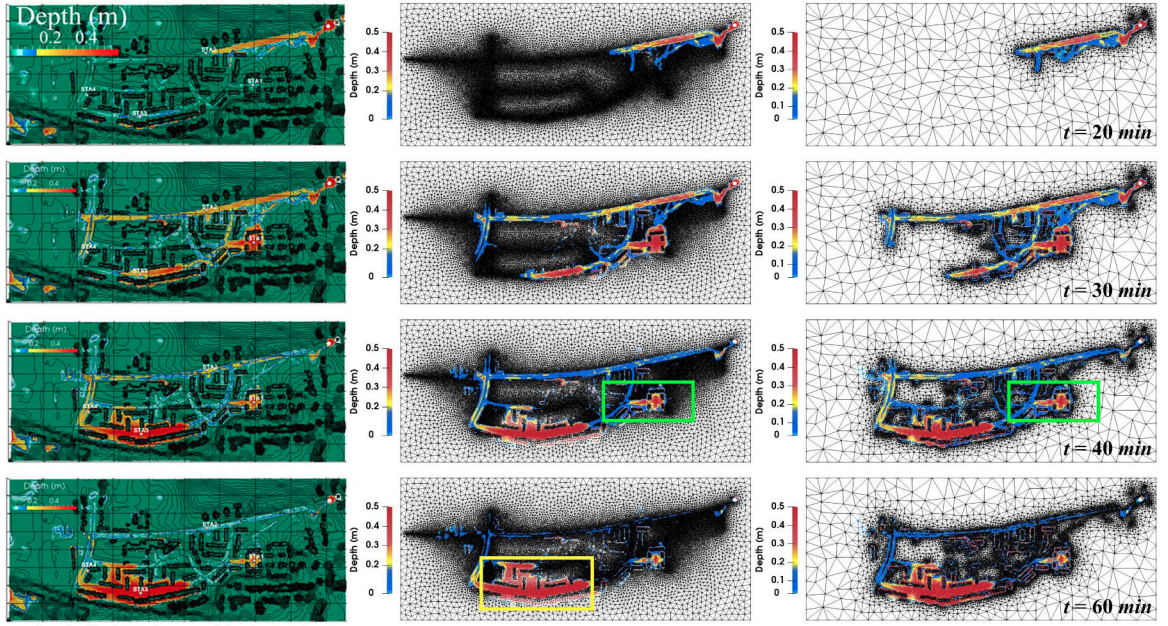


Figure 3.3: Water depth obtained from 3D modelling (left column, see Zhang et al. (2016) [138]), 2D fixed unstructured mesh modelling (middle column) and 2D adaptive unstructured mesh modelling (right column) at time level $t = 20 \text{ min}$ (first row), 30 min (second row), 40 min (third row) and 60 min (bottom row). Water accumulates in the low-lying area (marked with a yellow rectangle) and around buildings (marked with green rectangles).

procedure.

Generally speaking, the 2D adaptive mesh model provides promising results while the number of nodes/elements used during the simulation period is significantly reduced by up to 80% of that used by 2D fixed mesh modelling, thus reducing the computational cost considerably.

Accurate representation of topography using the adaptive mesh technique

Using 2D adaptive mesh technology, the mesh is dynamically adjusted during the process of flood propagation (right column in Fig. 3.5). Finer meshes are placed only in specific regions where the gradients of the flow variables are relatively steep (e.g. flow around buildings, Fig. 3.7), while coarser meshes are used in areas away from these regions, where inundation has not yet occurred. It enables the mesh-based computing to be efficient and have low computational complexity.

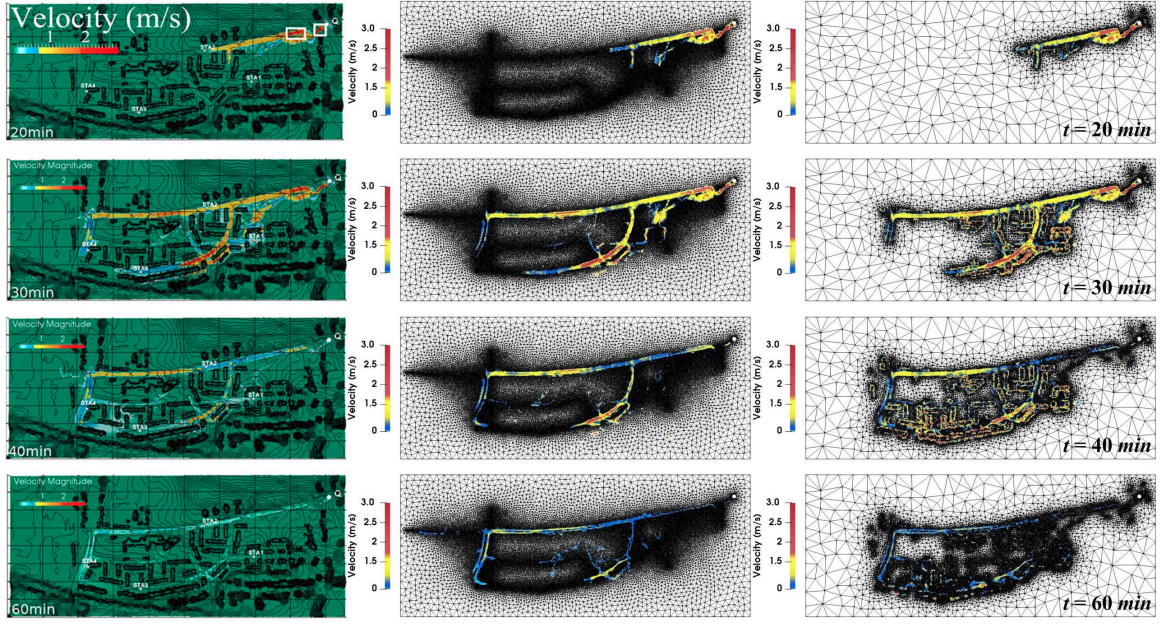


Figure 3.4: Velocity results obtained from 3D modelling (left column, see Zhang et al. (2016) [138]), 2D fixed unstructured mesh modelling (middle column) and 2D adaptive unstructured mesh modelling (right column) at time level $t = 20 \text{ min}$ (first row), 30 min (second row), 40 min (third row) and 60 min (bottom row).

In this case, the topographical data, a combination of airborne laser altimeter (LiDAR) and digital map data, is available with a high resolution of 2 m . The availability of high-resolution topographical data is important for the accurate numerical simulation of urban flood inundation. However, high-resolution topographical data requires a high computational burden, thus, resulting in a computationally demanding flood modelling. Using the adaptive mesh technology, the resolution of the topographical data can be dynamically adjusted during the process of flood propagation. The topographical data over the domain is obtained by interpolating the high resolution (2 m) data onto the adapted mesh at each time level. Therefore, the high-resolution topographical data is only used in the flooded region while the low-resolution data is used in the rest of the domain. One of the advantages of 2D adaptive unstructured mesh modelling is that the buildings can be represented accurately when/where needed during the flooding process. Fig. 3.5 shows how buildings gradually appear as the flooding water spreads across the domain.

Fig. 3.6 indicates the error in bathymetries during the simulation period when using adaptive meshes. It is seen that the high resolution (low error) data is used only over the regions along the flood pathway.

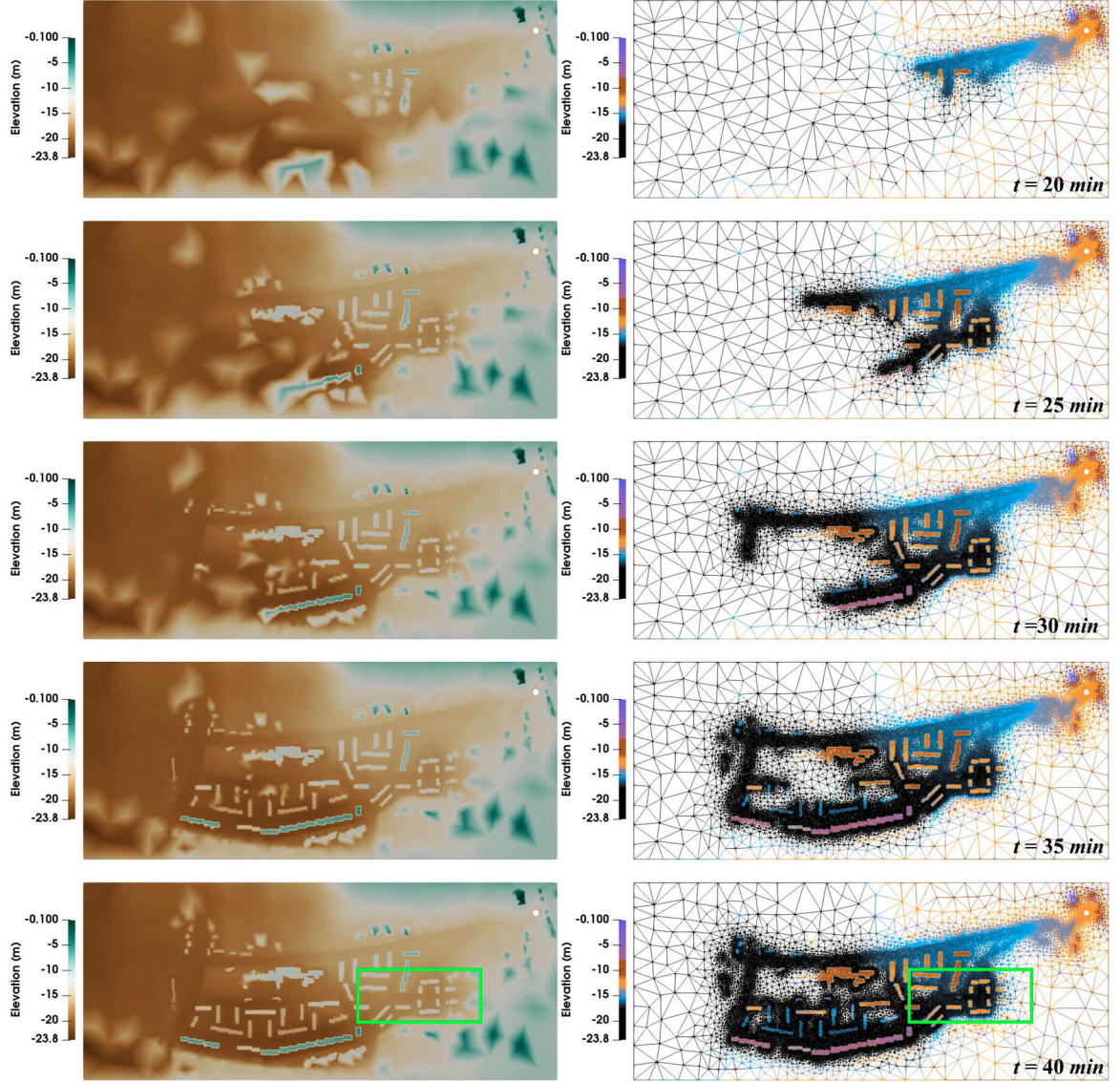


Figure 3.5: Buildings gradually become visible as the flood water spreads west and southward. The left column shows the plane view of surface topography. The right column shows the corresponding mesh. Areas marked with green rectangles show the corresponding building areas in Fig. 3.3 (marked with green rectangles as well).

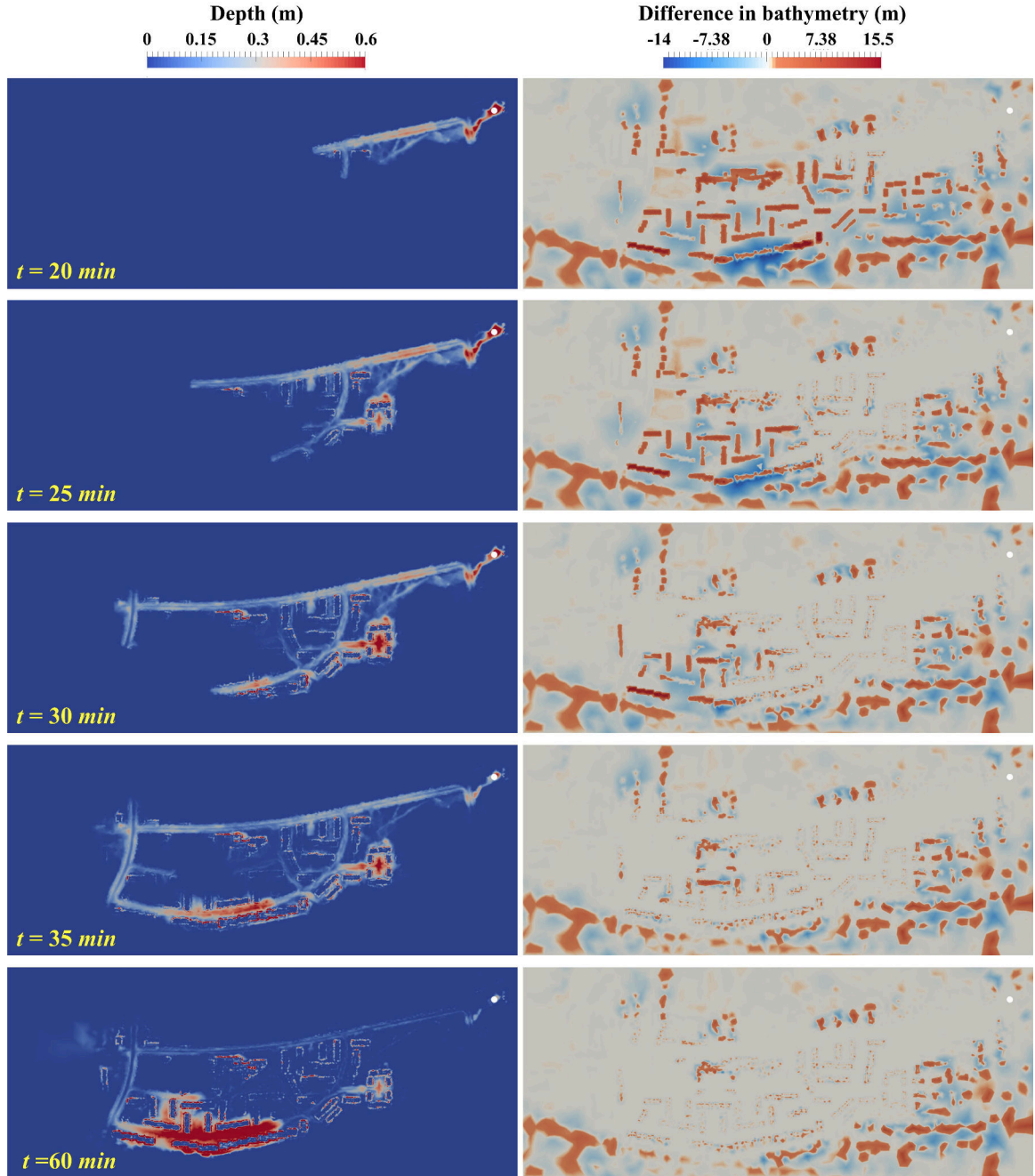


Figure 3.6: Error in bathymetries with the use of adaptive meshes during simulations. The left column shows the plane view of water depth for adaptive mesh modelling. The right column shows the corresponding the error of the bathymetries used at time levels (20, 25, 30, 35, 60 *min*), where the high resolution (2 *m*) topographical data is a reference data.

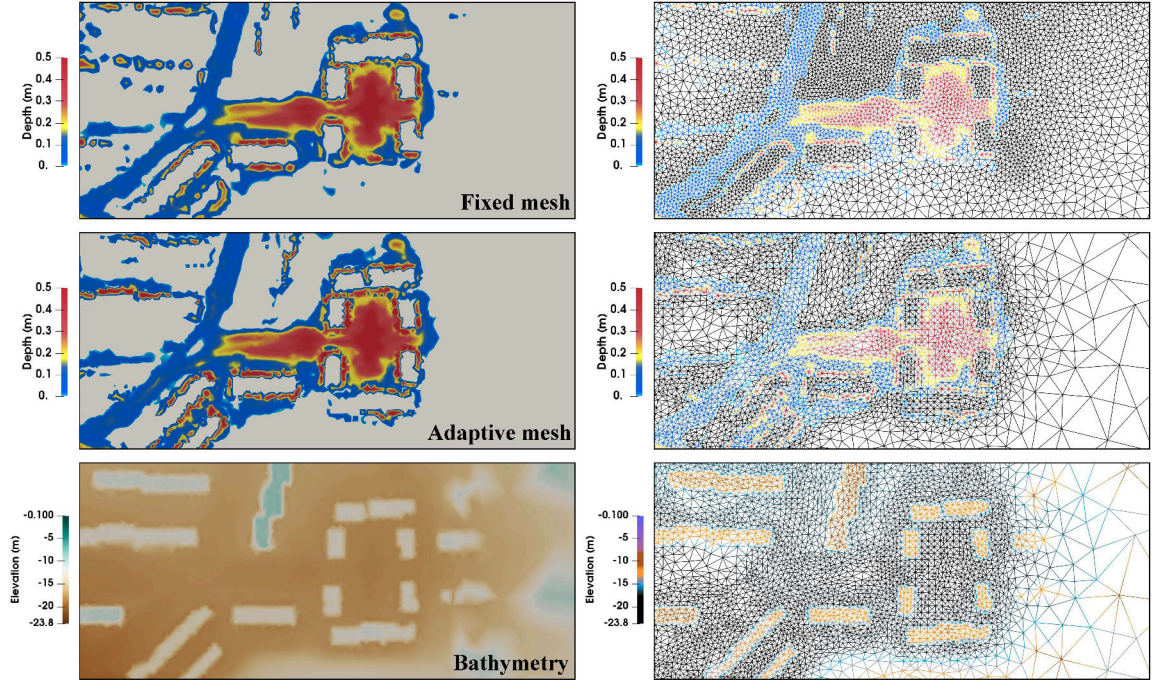


Figure 3.7: Water depth in the areas marked with green rectangles in Fig. 3.3 (first and middle row), and details of bathymetry for this building area marked with green rectangles in Fig. 3.5 (bottom row). The right column shows the corresponding mesh.

Comparison with published results at detector locations

To further validate our 2D fixed and adaptive unstructured mesh flooding models, the time series of water depth at detector locations STA1, STA2, STA3 and STA4 are compared with those of six 2D hydraulic models (DIVAST, DIVAST-TVD, JFLOW, LISFLOOD-FP, TRENT, and TUFLOW) from Hunter et al. (2008) [135] and the 3D model [138]. Fig. 3.8 shows the time series of water depth predicted by these models. STA1 represents a flat area with surrounding buildings where water accumulates fast but releases slowly during the flood process. STA2 is placed in the middle of the main road, where the flood water is shallow and moving fast. STA3 is located in the low-lying area in the southern part of the domain as marked with a yellow rectangle in Fig. 3.3. Water from the east-west oriented main street converges and ponds there in the latter part of the simulation. STA4 is sited at the side of the road where the ponding water comes from both north and south directions.

Model performance has been assessed through comparison of the results at these detector

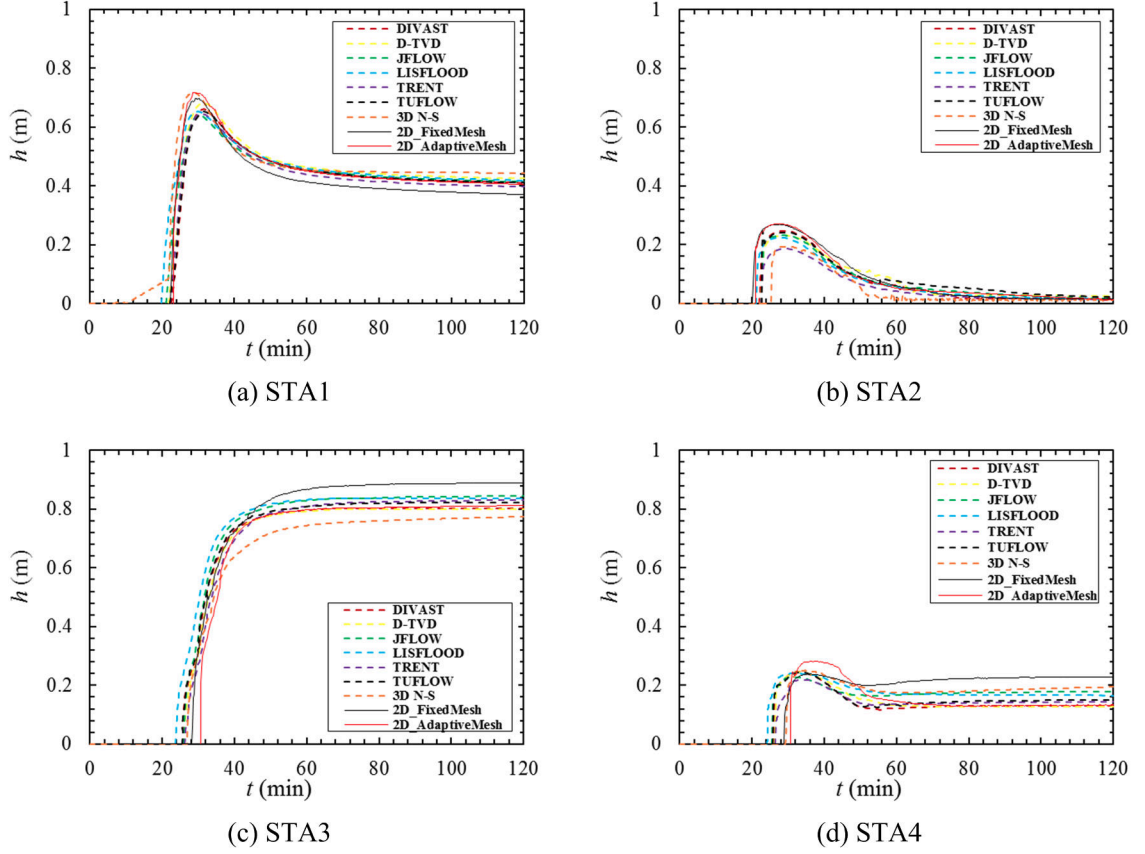


Figure 3.8: Flood depth time series at detector locations STA1, STA2, STA3 and STA4 simulated by 2D fixed and 2D adaptive unstructured mesh models and numerous published models.

Fig. 3.8, the black and red lines represent the time series of water depth predicted by the new 2D unstructured mesh flooding model with fixed and adaptive meshes respectively. It can be seen that a good agreement is achieved between the results from our 2D adaptive (and fixed) unstructured mesh models and those from other 2D models at all the detector locations. The results of our 2D unstructured mesh modelling results (especially with adaptive meshes) are very close to the 3D results [138] at STA1 (a flat and ponding area) where the vertical velocity is relatively small. However, there is a large difference between 2D and 3D modelling results at STA2 where the vertical inertial and nonhydrostatic pressure terms are large, the impact of 3D flow cannot be ignored [138]. Compared with the results

when using 2D fixed meshes, 2D adaptive mesh modelling results at STA3 match well with those from other 2D models, and are closer to the 3D results.

In general, the results between both of our 2D fixed and adaptive mesh models are in a good agreement during $t = 1 - 40 \text{ min}$ (until the flooding peak), there is then a difference between them at STAs 3 and 4 during the flooding recession (after $t = 40 \text{ min}$). Compared with the 3D results, the time series of water depth using 2D adaptive unstructured meshes is more accurate than that using 2D fixed unstructured meshes at STAs 1-3. The main reason for this is that the adaptive mesh is optimised in response to the evolving flow features while the fixed mesh is designed based on the main road network (see Zhang et al., 2016 [138]). For both fixed and adaptive mesh modelling, a high mesh resolution of 2 m is placed along the main road routes where the dominant flow features are observed. However, for fixed mesh modelling, the meshes are coarser around the building areas than that of adaptive mesh modelling. Whats more, mesh adaptivity ensures a certain precision. It is thus seen in Fig. 3.8 that at STAs 1-3, the time series of water depth using 2D adaptive unstructured meshes is more accurate than that using 2D fixed unstructured meshes while a smaller number of nodes is used during the simulation. In general, both results of using adaptive and fixed meshes are consistent with those from other 2D models.

Performance of 2D fixed and 2D adaptive unstructured mesh flood modelling

Table 3.1: Node and element number of the 2D fixed and 2D adaptive unstructured mesh and the run time of the simulations.

Meshing Type	Time Level (<i>min</i>)	Node number	Element number	Run time (<i>h</i>)
Fixed mesh	0 - 120	28508	56862	300.5
Adaptive mesh	0	28508	56862	153.5
	20	4601	8018	
	30	15153	30203	
	40	21784	43463	
	60	22331	44557	
	120	22331	44557	

As seen in Table 3.1 and Fig. 3.9, a multi-scale unstructured mesh with $2 \text{ m}/5 \text{ m}/20 \text{ m}$

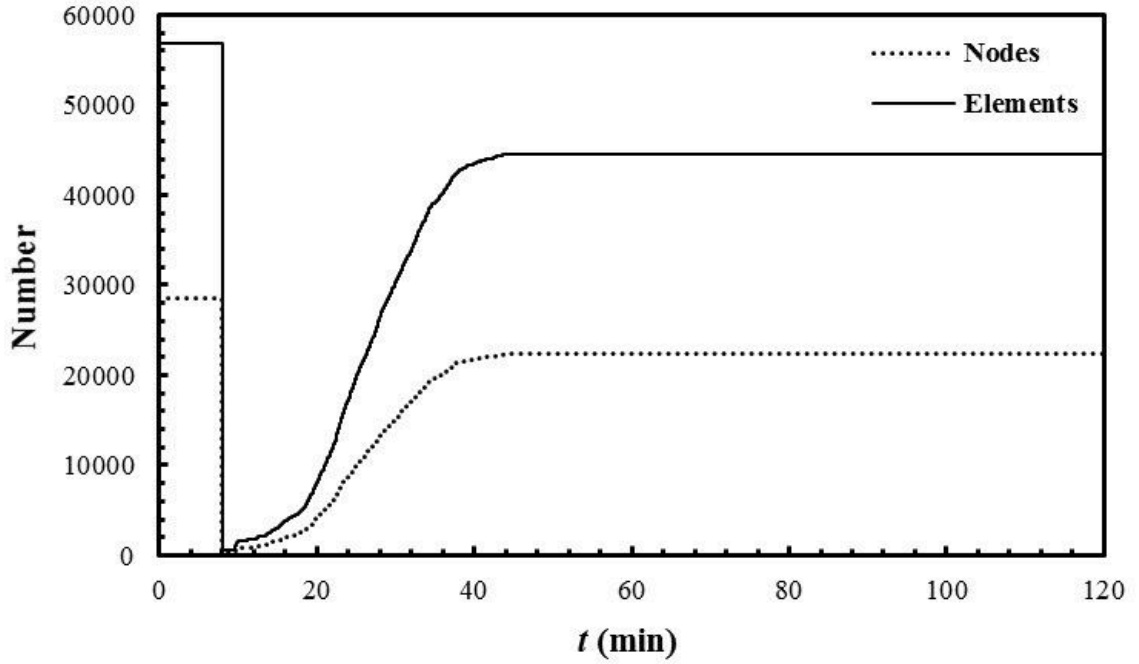


Figure 3.9: The numbers of nodes and elements used in 2D adaptive mesh modelling during the simulation period $[0, 120]$ min.

resolution generated by Gmsh (Fig. 3.2) consists of 28,508 nodes and 56,862 unstructured triangle elements. This mesh is used during the whole simulation period for 2D fixed unstructured mesh modelling, and used as the initial mesh for 2D adaptive unstructured mesh modelling. After first adapting the mesh, the numbers of nodes and elements used for 2D adaptive mesh are reduced by 84% and then gradually increased during $t = 10 - 45$ min as the flooding water spreads over the domain. At time level $t = 40$ min, the inflow at location Q ends and recession of water starts. As a result, the water accumulates in low areas and the water velocity remains small. From $t = 40$ min, the flow gradually becomes stable and this leads to a small change of element number of the meshes. To avoid the overhead and extra computational cost introduced during the adaptive mesh procedure, the mesh is fixed from $t = 45$ min. Using our 2D unstructured mesh model, the CPU time of 300.5 h is required for the fixed mesh simulation and 153.5 h for the adaptive mesh simulation, which is only half of the former. Moreover, adapting the mesh dynamically means that there is

no need to perform a priori calculations, with its corresponding computational cost, to estimate the key regions of the domain to place more resolution in those areas. Above all, 2D adaptive unstructured mesh modelling is less computational expensive than 2D fixed unstructured mesh modelling.

The use of 2D adaptive unstructured meshes improves the computational efficiency. To further reduce the computational cost, various numerical techniques can be adopted in our flood model. For example, an adaptive-time-step scheme and/or parallel computing using MPI.

3.3.3 Summary

A 2D DCV-FEM flooding model with the adaptive unstructured mesh technique has been developed and applied to a flooding event that happened in 2002 in Glasgow, Scotland, United Kingdom, where the flood is induced by a stream flow from a culvert at the north-east corner of the domain. A comparison between 2D adaptive and fixed mesh models as well as 3D model has been undertaken. It has been found that using the 2D adaptive mesh model, it is able to provide accurate results while the computational cost is reduced by 20 – 84% in comparison to 2D fixed mesh models. Another advantage of 2D adaptive unstructured mesh modelling is that urban topography can be accurately represented when/where needed by increasing the mesh resolution (around the buildings, for example) dynamically when the flooding water spreads over the urban area. This is the first time to use the dynamically adaptive mesh technique in flooding modelling and assess its performance in a relatively simple flooding event. In the following subsection, more work on flood modelling development will focus on complex realistic cases where flooding may occur from more than one source.

Chapter 4

Numerical simulation of floods from multiple sources using anisotropic dynamic mesh optimization method

The coincidence of two or more extreme events (precipitation and storm surge, for example) may lead to severe floods in coastal cities. It is important to develop powerful numerical tools for improved flooding predictions (especially over a wide range of spatial scales - metres to many kilometres) and assessment of joint influence of extreme events. Various numerical models have been developed to perform high-resolution flood simulations in urban areas. However, the use of high-resolution meshes across the whole computational domain may lead to a high computational burden. More recently, an adaptive isotropic unstructured mesh technique has been first introduced to urban flooding simulations and applied to a simple flooding event observed as a result of flow exceeding the capacity of the culvert during the period of prolonged or heavy rainfall. In this work, the above adaptive mesh flooding model based on 2D shallow water equations (named as Floodity) has been further developed by introducing (1) an anisotropic dynamic mesh optimization technique (anisotropic-DMO); (2) multiple flooding sources (extreme rainfall and sea-level events); and (3) a unique combination of anisotropic-DMO and high-resolution Digital Terrain Model (DTM) data. It has been applied to a densely urbanized area within Greve, Denmark. Results from MIKE 21 FM are utilized to validate our model. To assess uncertainties in model predictions, sensitivity of flooding results to extreme sea levels, rainfall and mesh resolution has been undertaken.

4.1 Introduction

Over the last two decades, the risk of urban flooding in heavily populated coastal regions has increased and is expected to increase further, mainly due to the urbanization and climate change. Urban flooding in coastal regions could be caused by a single source (heavy rainfall, high sea levels or storms), or several sources acting in combination [139]. Due to this increasing high risk, the combined effect and the joint probability of multiple extreme floods are gaining importance in flooding simulations [140]. Improving the predictive capabilities in such cases is critical for populated areas, especially cities. It is therefore important to develop an efficient and accurate numerical model for studying floods caused by several concurrent hazards in coastal cities.

An overview of flood inundation models has been given by Teng et al. (2017) [141]. In the past, various numerical models have been developed to simulate flood inundation [103, 106, 135, 142, 143]. These models are classified into three categories: 1) empirical methods such as measurements [144] and remote sensing [145]; 2) hydrodynamic models; and 3) conceptual models for large floodplains [146] and probabilistic flood risk assessment [147]. The hydrodynamic models include one-dimensional (1D) [148], two-dimensional (2D) [149–151] and three-dimensional (3D) models [138, 152]. MIKE Flood [153] has the capability of simulating combined river, sewer and floodplain modelling with high resolution and reliability [154]. Recently various efforts to compute overland flows by solving the shallow-water equations have been made. These studies have simulated overland flows under extreme and unsteady rainfall conditions and spatially constant infiltration rates have been taken into account [155–157]. However, due to the complexity and uncertainty of flood modelling, efficient simulation of flooding at high-resolution terrain remains a significant challenge in hydrologic and hydraulic studies. For efficient and accurate flood inundation modelling, numerous methods have been developed, such as grid coarsening methods [104], cellular automata approach [105], and speeding-up strategies

such as parallel processing [158–160]. Nguyen et al. (2016) [156] have proposed a coupled model called HiResFlood-UCI, which combines the hydrological model HL-RDHM and the hydrodynamic BreZo model while ensuring a bare minimum computational cost. HiResFlood-UCI uses HL-RDHM as a rainfall-runoff generator and BreZo as the hydrological routing scheme. This model has been successfully applied to a catchment of the Illinois river in USA.

In comparison to adaptive mesh refinement (AMR) (a fine structured mesh nested within a coarse mesh) technique [33], the DMO technique is able to adapt the mesh optimally in time and space in response to the evolving flow features, thus providing sufficient mesh resolution where and when it is required. In this work, we have further developed this adaptive unstructured mesh shallow water model with anisotropic considerations for modelling urban floods from multiple sources (rainfall and storm surge). The implicit θ -scheme has been adopted for solving the shallow water equations and applications to urban floods caused by multiple sources (rains and sea levels). In the DCV-FEM scheme, the velocity components are discretised FE-wise, while the pressure/free surface height is discretised CV-wise [123].

In this work, a DCV-FEM adaptive mesh urban flooding model for simulating the concurrent flooding and has been successfully applied to a $2.2 \text{ km} \times 1.7 \text{ km}$ densely urbanized area within Greve, Denmark. This is the first time to apply the anisotropic-DMO method to simulate urban floods caused by multiple sources based on high resolution Digital Terrain Model (DTM) data. Model validation has been performed in comparison with results from MIKE 21 FM. Sensitivity of the extreme sea levels, rainfall and mesh resolution to the flood volume has been explored to assess uncertainties in model predictions.

4.2 Descriptions of study site and data

To assess the performance of anisotropic-DMO in flooding modelling, the new flooding model has been applied to an urban area located within a $2.2 \text{ km} \times 1.7 \text{ km}$ densely urban-

ized area within Greve, Denmark. The study area is located in the northeastern part of

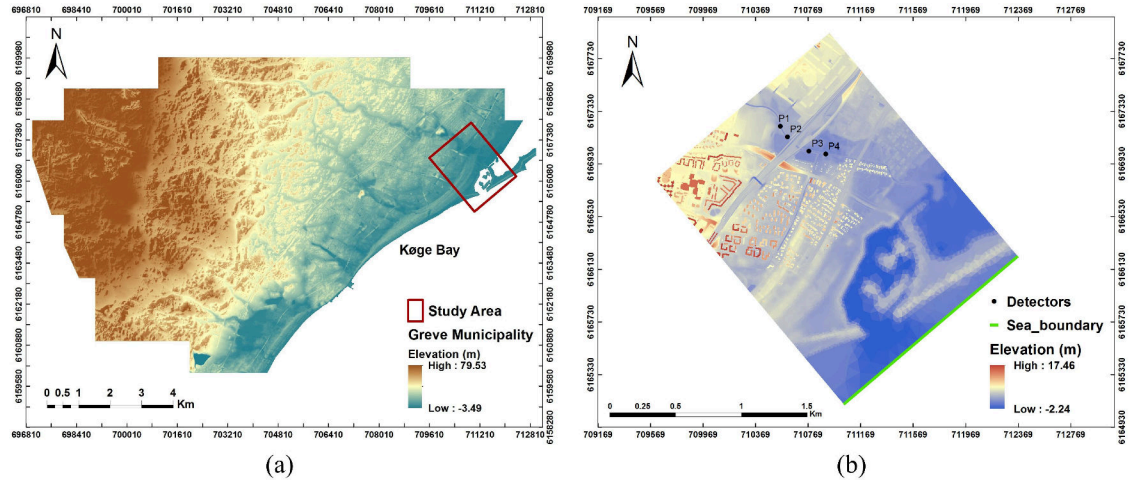


Figure 4.1: (a) Situation of study area in Greve, Municipality of Denmark. (b) DTM with buildings of study area (resolution of $1.6\text{ m} \times 1.6\text{ m}$) - generated by the GIS (Geographical Information System) software, see ArcGIS (2010) [161].

Greve, Denmark, which covers part of the coastal area (see Fig.4.1 (a)). Historical extraordinary flood events which were caused by a series of rain events have occurred in Greve. In addition to extreme rainfall events, Greve is also vulnerable to flood induced by extreme sea-level events along its coast. For example, the most extreme historical flood occurred on 13th October 1760 with a maximum water level of 3.7 m , was caused by a very serious storm surge [162]. Recurrence of any of these flood events would cause numerous damages (e.g. loss of life, direct damages to roads, railways and buildings, indirect damages including loss of income, clean-up cost, turnover loss, cost of illnesses, etc). As a consequence, it is important to develop flooding models to improve the accuracy of flood predictions. In the Greve case study, the digital elevation data provided was quality assured and buildings were incorporated into the DTM (Digital Terrain Model) data with a resolution of 1.6 m (Fig.4.1 (b)), which was detailed enough to describe topographic features (buildings, rivers and streets). Data of different extreme rainfall events as well as extreme sea-level events have been used as the boundary conditions and sources.

4.3 Extreme sea-level and rainfall events

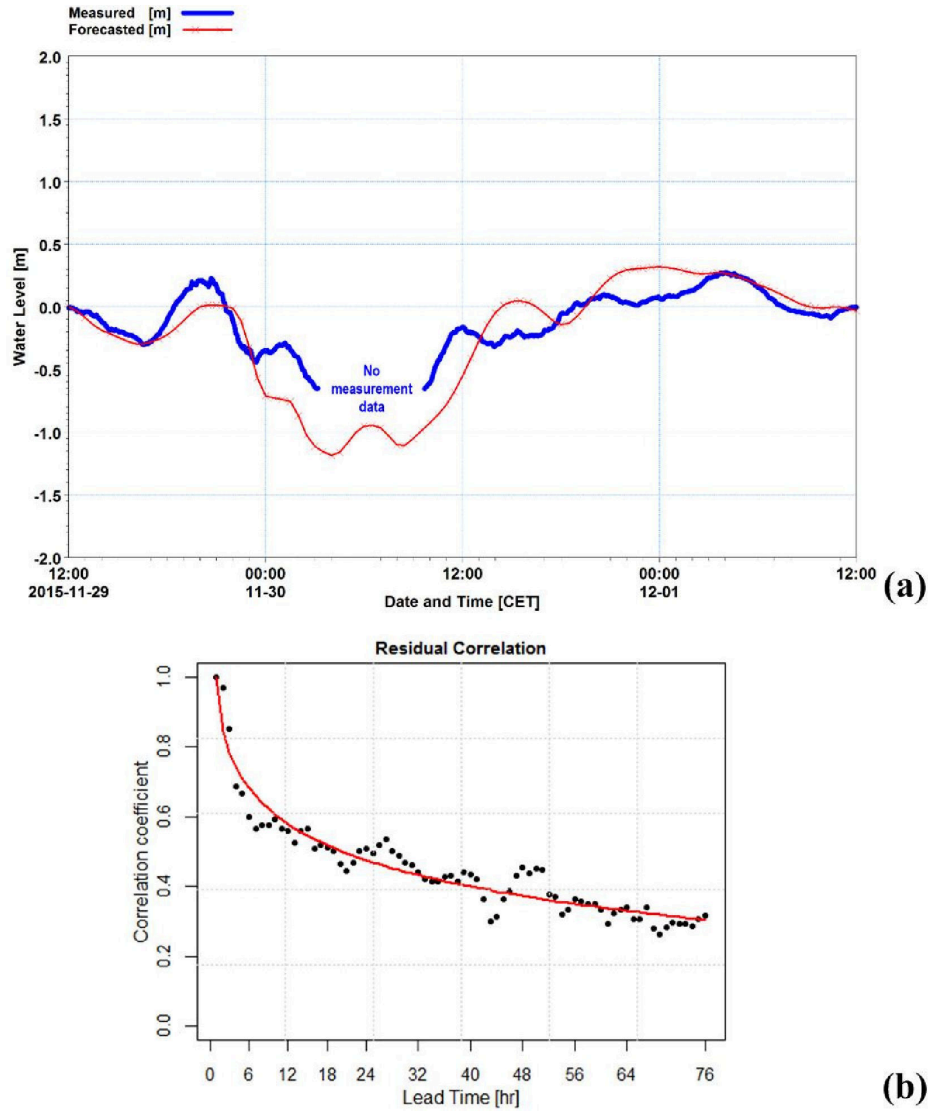


Figure 4.2: (a) Forecast of water levels issued at 12:00 on 29th November 2015. (b) Correlation coefficient between observed and predicted water levels at function of forecast time [163].

Due to the impact of climate change in the next 100 years, future climate change conditions should be taken into account to estimate the future extreme sea-level and rainfall events [163]. To forecast storm surges, a hydrodynamic model has been calibrated against historical events during 2000-2015. The capability of this built hydrodynamic model to

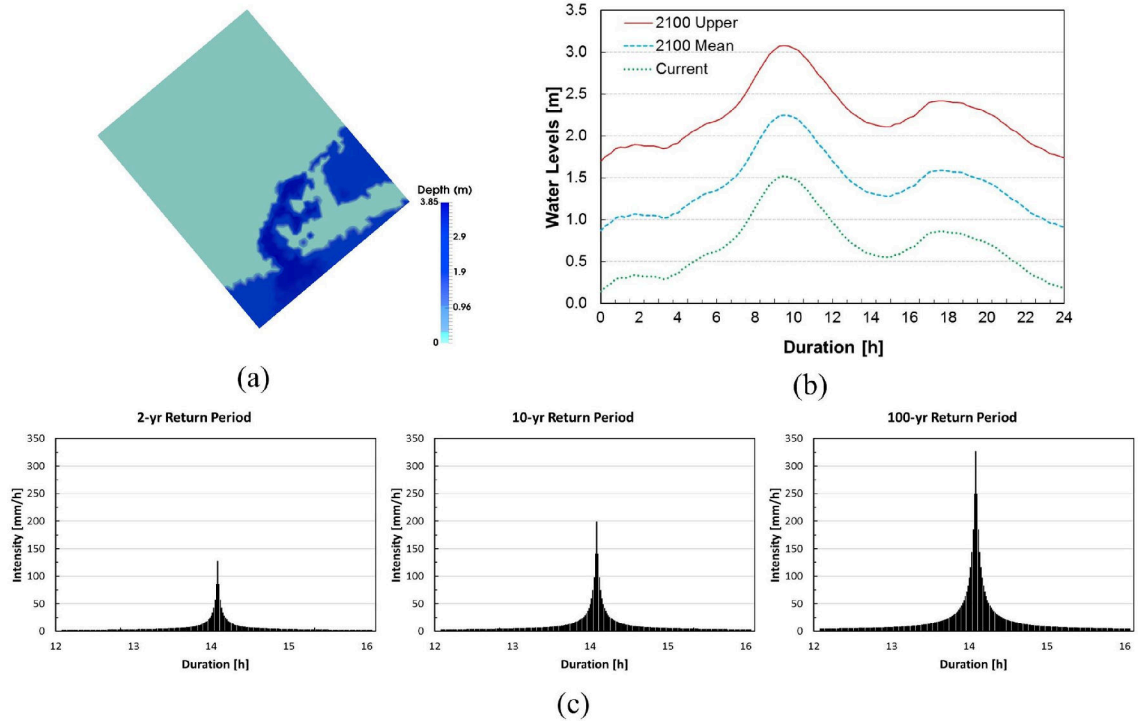


Figure 4.3: (a) Initial water depth in a scenario of 2100 upper extreme water level. (b) Future extreme seawater levels by 2100 (considering the worst climate change scenario for 100-yr projection), see Berbel Roman (2014) [163]. (c) Extreme 24 h design rainfall for 2-yr, 10-yr and 100-yr return period (considering the effects of climate change), see Berbel Roman (2014) [163].

forecast storm surges was evaluated against historical storm event during 2010-2017. An example of a real time forecast issued during the storm “Gorm” on 29th November 2015 can be seen in Fig.4.2 (a). From Fig.4.2 (b) it can be seen how the correlation coefficient between observed and predicted water levels goes down as the lead time increased.

The 3D hydrodynamic model was built using MIKE 3 FM, a software tool for modelling unsteady three-dimensional flows, which uses a flexible mesh calculation grid taking into account density variations, bathymetry and external forcing, such as meteorology, tidal elevations, currents and other hydrographic conditions [164]. Meteorological forcing for the model is obtained from a WRF (Weather Research and Forecasting) limited-area numerical weather prediction model covering Northern Europe with a resolution of 0.1 degrees, which is run by *StormGeo* in Norway. A description of the above method for estimating

storm surge event time series for climate change analysis is described in Berbel Roman (2014) [163]. For comparison purposes, the reference results from MIKE 21 FM.

Fig.4.3 (b) shows the extreme sea water levels which were used as input boundary conditions along the coastline. According to Berbel Roman (2014) [163], to estimate the expected changes in sea surges in future (up to year 2100), hydrodynamic simulations were carried out, which were driven by the wind and atmospheric pressure results from three regional climate models. A general observed extreme event pattern was identified based on the past observed extreme sea-level events. The future extreme water level event time series (Fig.4.3 (b)) were then obtained from the general observed pattern by scaling to a given return period and adding estimates of mean sea level rise and change in storm surge signal. The water level calculated with statistics projections of 100-yr return period considering climate change under a present scenario identified as ‘current’ and a future scenario identified as ‘2100 mean’ and ‘2100 upper’ depending of the climate factor considered. These extreme sea-level events (current, 2100 mean and 2100 upper) lasted 24 hours. As seen in Fig.4.3 (b), for a return period of 100 years, the maximum value for the extreme water levels is 1.52 m (current), 2.25 m (2100 mean) and 3.08 m (2100 upper).

Extreme precipitation data in this case study is obtained by a frequency analysis, where the rainfall data (over more than 10 years) from 83 stations in Denmark is used. By running a regional statistical extreme model, the intensities of rainfall for the 2-yr, 10-yr and 100-yr return period have been obtained, then multiplied by a factor of 1.1 - 1.5 due to the impact of climate change in the next 100 years [163]. The designed intensities are shown in Fig.4.3 (c). It is assumed that for a given rainfall event, a uniform rainfall is falling over the whole area of the domain ($2.2\text{ km} \times 1.7\text{ km}$).

4.4 Model applications

A series of model simulations using anisotropic-DMO have been carried out to assess the performance of the new flooding model developed here. In all cases, the field of water

depth is selected to be the adapted field using anisotropic-DMO technique. The extreme seawater level event (Fig.4.3 (b)) which lasts 24 hours is used as input boundary condition on the sea boundary (Fig.4.1 (b)), the remaining boundaries are set up as closed (no flow). In these simulations, sea water enters the densely urbanized area from the sea boundary (Fig.4.1 (b)) and the extreme rainfall events (Fig.4.3 (c)) take place simultaneously. Fig.4.3 (a) shows the initial water depth within the domain in a scenario of 2100 upper extreme water level. The adapting mesh schemes are listed in Table 4.1. The mesh is adapted to ensure an absolute error in the water depth field of 0.01 m and the aspect ratio is 100.

Table 4.1: Adapting mesh schemes for the Floodity simulations.

Minimum element size	Maximum element size	Time step Δt (s)
20	200	[5,10]
10	200	[3,10]
5	200	[1,10]

4.5 Results and Discussion

4.5.1 Individual flooding events

(1) Flooding map

Fig.4.4 shows the flood propagation process over the urban area in a scenario of an individual 2100 upper extreme sea-level event. It can be observed that in most of the inundation area, the solutions of water depths obtained from Floodity are in good agreement with those from MIKE 21 FM. The mesh is optimally adapted according to the evolving flow features in time and space, thus providing sufficient mesh resolution where and when it is required (right panel in Fig.4.4). For example, the fine mesh is located along the flood propagation path while the coarser mesh is used in the areas where inundation has not occurred yet. To further estimate and compare the flood extent, the flood volume during the flood propagation process is calculated (Fig.4.5). It is clearly that flood volume obtained

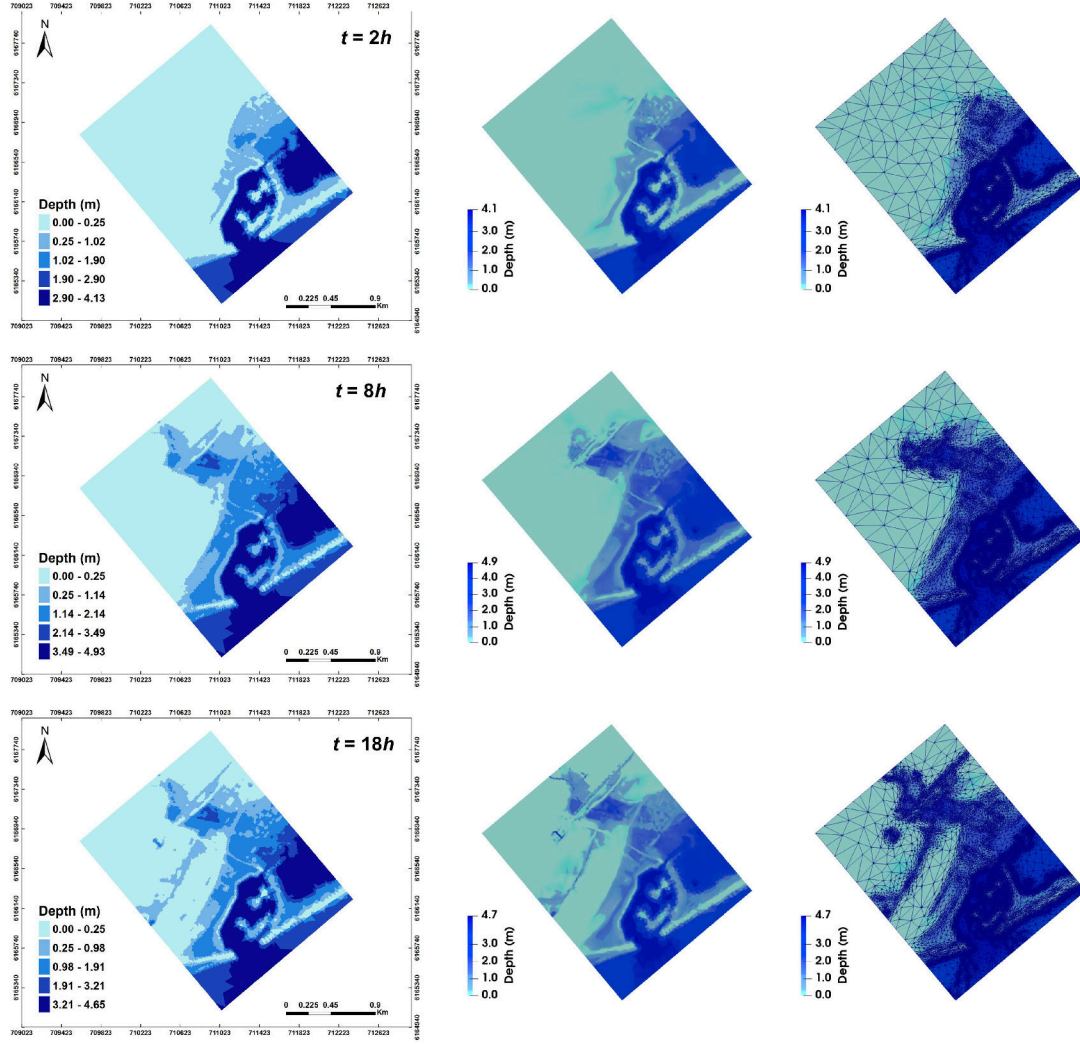


Figure 4.4: Water depths obtained from a MIKE 21 FM model updated from Berbel Roman (2014) [163] (left column), a Floodity model with a mesh resolution 10 *m* (middle column) and results showing the corresponding mesh (right column) based on bathymetric data without buildings in a scenario of 2100 upper extreme sea-level event at time level $t = 2\ h$ (first row), 8 *h* (middle row), and 18 *h* (bottom row).

from Floodity is a little higher than that from MIKE 21 FM during the whole period, whilst the general trends of both are consistent.

(2) Comparison with MIKE 21 FM results at detector locations

To further evaluate the performance of Floodity using different mesh resolutions, the time series of water depth at detector locations P1, P2, P3 and P4 are plotted in Fig.4.6, in comparison to those from MIKE 21 FM. In Fig.4.6, the blue, green and black lines represent

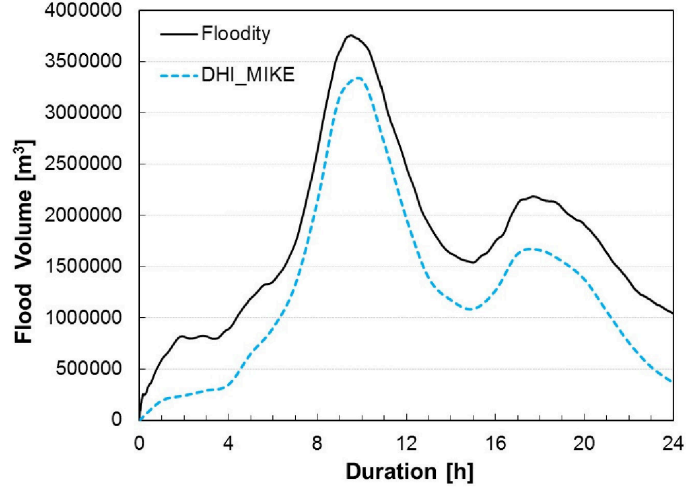


Figure 4.5: Flood volume obtained from a MIKE 21 FM model and a Floodity model with a mesh resolution 10 m based on bathymetric data without buildings in a scenario of 2100 upper extreme sea-level event.

the time series of water depth predicted by the new anisotropic unstructured mesh flooding model Floodity with a minimum mesh size of 20 m , 10 m , and 5 m respectively. It can be seen that a good agreement is achieved between the results from both the fixed and adaptive mesh simulations (except for that with a minimum mesh size of 20 m) at detectors P3 and P4 during the flooding propagation period $[6.7, 24]\text{ h}$. However, MIKE 21 FM predicts an earlier flood arrival time at P3 and P4 than that predicted by the Floodity simulations. The results of water depth at detectors P1 and P2 obtained by both the MIKE 21 FM and Floodity simulations are very close to each other when almost the same mesh resolution (10 m) is used over the inundated regions. The detectors P1 and P2 are located within a narrow open channel, where a high resolution mesh (a mesh size smaller than 10 m at least) is required to represent it. We can see that the Floodity simulation with a minimum mesh size of 5 m predicts a deeper water depth at P1 and P2 than that in the simulations with the large mesh size (10 m and 20 m). It proves that more detailed solutions can be obtained in the local areas when using an adaptive mesh instead of a fixed mesh. It is also noted that it is convergent with the increased resolution at detectors P3 and P4, not at P1 and P2. Again, the reason for this is a mesh size larger than 5 m is not able to represent the narrow channel,

thus causing an error in water depth.

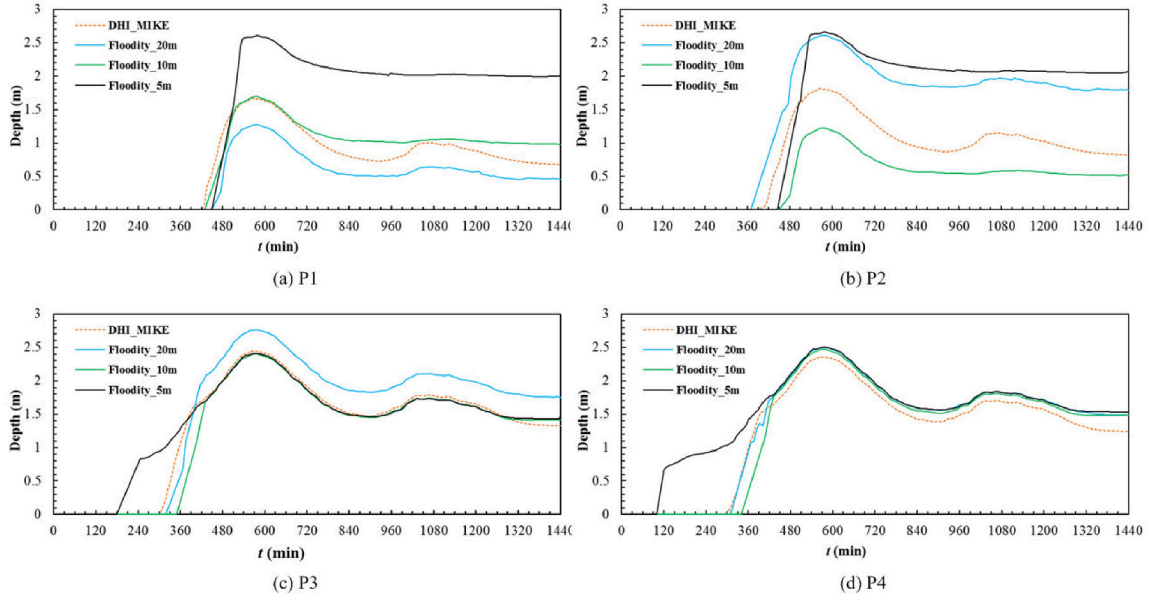


Figure 4.6: Flood depth time series at detector locations P1, P2, P3 and P4 (see Fig.4.1 (b)) simulated using MIKE 21 FM and Floodity with a mesh resolution of 20 m, 10 m, and 5 m respectively, based on bathymetric data without buildings in a scenario of 2100 upper extreme sea-level event.

For comparison purposes, MIKE 21 FM results from the model updated from Berbel Roman (2014) [163] are used as a reference solution in this study. The original model domain in Berbel Roman (2014) [163] has a dimension of $2.3 \text{ km} \times 7.5 \text{ km}$, where the northeastern region is selected as our computational domain (Fig.4.3 (a)). Berbel Roman (2014) [163] divided the domain in 9 sub-regions with the element size in range of [10,100] m, where large elements were used to represent the surroundings of harbours, the coastline and the train tracks were represented with smaller elements. Thus, a flexible mesh with the element size in range of [10,100] m is used in the MIKE 21 FM simulations while the adaptive meshes with a minimum mesh size of 20 m, 10 m, and 5 m are used in Floodity. A comparison of water depth results using the adaptive (Floodity) and fixed (MIKE 21 FM) unstructured mesh has been carried out.

4.5.2 Joint flooding events

The newly developed adaptive mesh flooding model is further used for simulating floods under the combined impacts of the events (extreme rainfall and sea level). Fig.4.7 shows the results of water depth from MIKE 21 FM and Floodity with a minimum adaptive mesh size of 10 m at time levels $t = 15\text{ h}$. It presents the flood propagation process over the urban area in scenarios of an individual 2100 upper extreme sea-level event and a joint event with 100-yr return period rainfall respectively. Due to the effect of rainfall, the joint event has larger flood areas than the individual event, as shown in the areas marked with rectangles in Fig.4.7.

Fig.4.8 and Fig.4.9 show the time series of water depth at four detector locations predicted by these models respectively. The total rainfall amount in 24 hours is 28.1 mm , 45.7 mm and 82.1 mm within the $2.2\text{ km} \times 1.7\text{ km}$ study area for the 2-yr, 10-yr, 100-yr return period respectively. Thus, in comparison to the extreme sea level, the rainfalls have a relatively smaller impact on the inundation extent. Again, it can be observed that under the scenarios of joint flooding events, the water depths and velocity obtained from Floodity using the minimum mesh size of 10 m are in good agreement with those from MIKE 21 FM simulations.

4.5.3 Impact of buildings on flooding simulations

The use of anisotropic-DMO in flooding modelling can better capture the evolving flow features and topographic features (buildings, rivers and streets), thus providing improved accurate flooding prediction. To further demonstrate the capability of the flooding model, Floodity has been applied to the joint flooding events with the bathymetric data including buildings, which were represented as impervious obstacles blocking the flow path.

Fig.4.10 presents the flooding map over the urban area at time level $t = 15\text{ h}$ in scenarios of an individual and joint flooding event respectively. It can be observed that Floodity results with a mesh resolution 5 m have a larger inundation extent than MIKE 21 FM re-

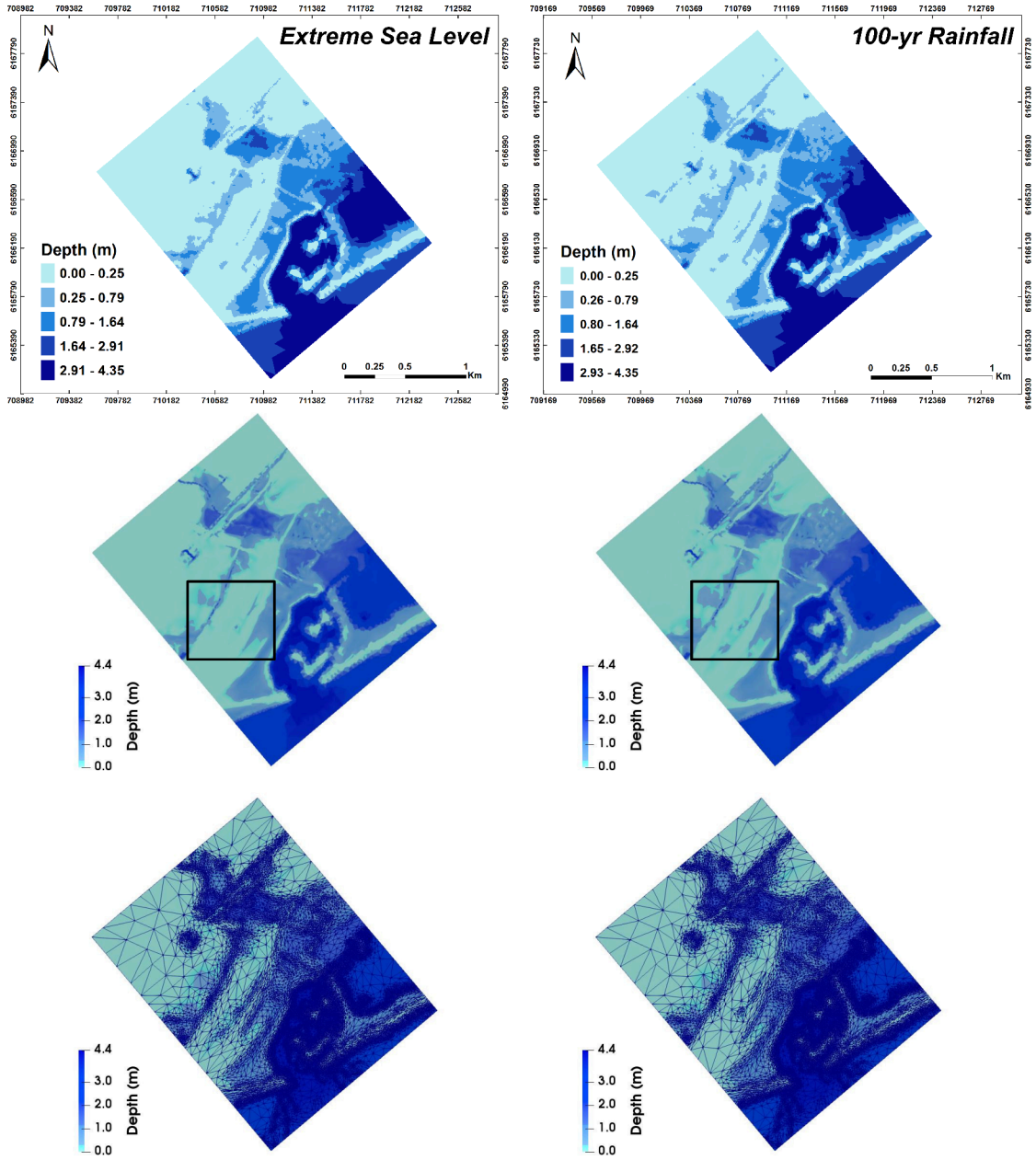


Figure 4.7: Water depths at time level $t = 15 h$ obtained from a MIKE 21 FM model (first row), a Floodity model with a mesh resolution $10 m$ (middle row) and results showing the corresponding mesh (bottom row) based on bathymetric data without buildings in scenarios of an individual 2100 upper extreme sea-level event (left column) and a joint event with 100-yr return period rainfall (right column).

sults and present more details of topographic features, including buildings and channels. The details of roads, buildings and channels can be observed clearly with an increased mesh resolution around them. Fig.4.11 provides the details of the areas marked with rect-

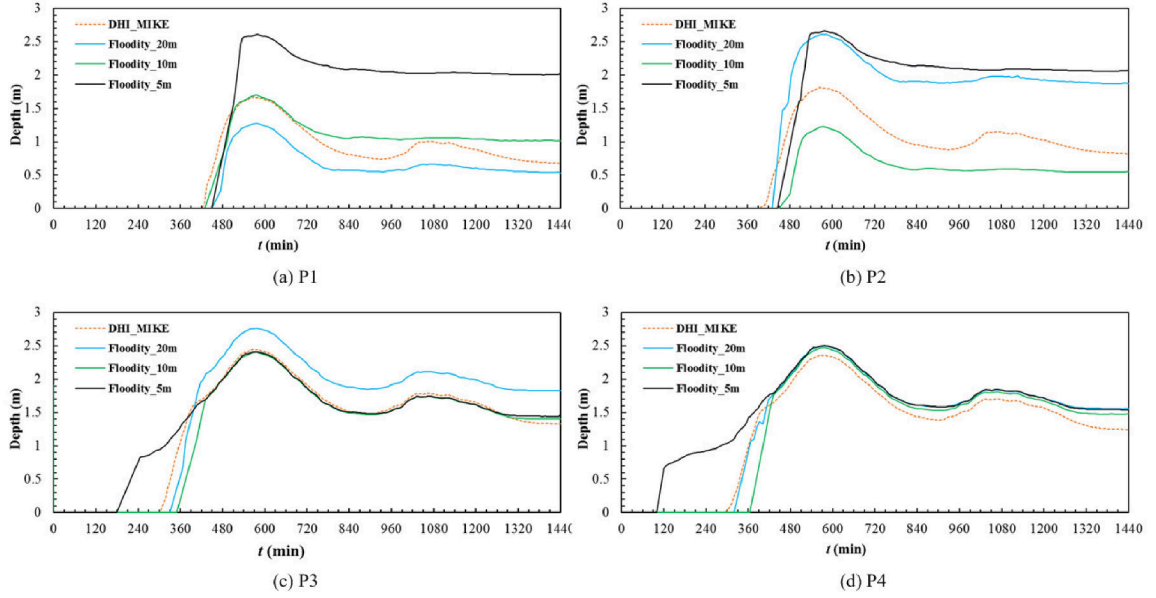


Figure 4.8: Flood depth time series at detector locations P1, P2, P3 and P4 (see Fig.4.1 (b)) simulated using MIKE 21 FM and Floodity with a mesh resolution of 20 m, 10 m, and 5 m respectively, based on bathymetric data without buildings in a scenario of 2-yr return period rainfall and 2100 upper extreme sea-level event.

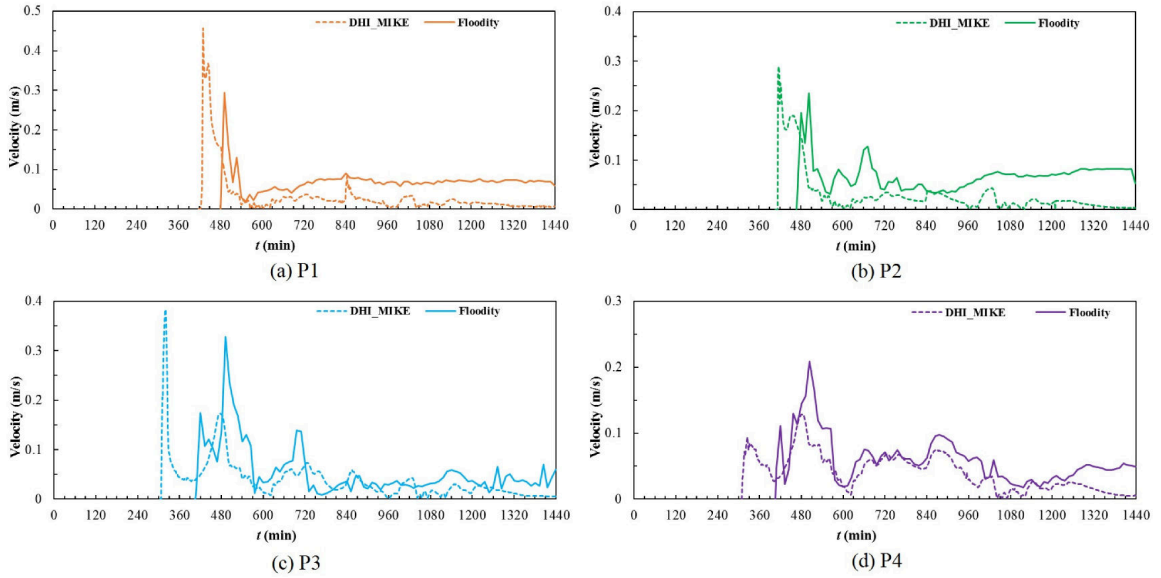


Figure 4.9: Flood velocity time series at detector locations P1, P2, P3 and P4 (see Fig.4.1 (b)) simulated using MIKE 21 FM and Floodity with a mesh resolution of 10 m based on bathymetric data without buildings in a scenario of 2-yr return period rainfall and 2100 upper extreme sea-level event.

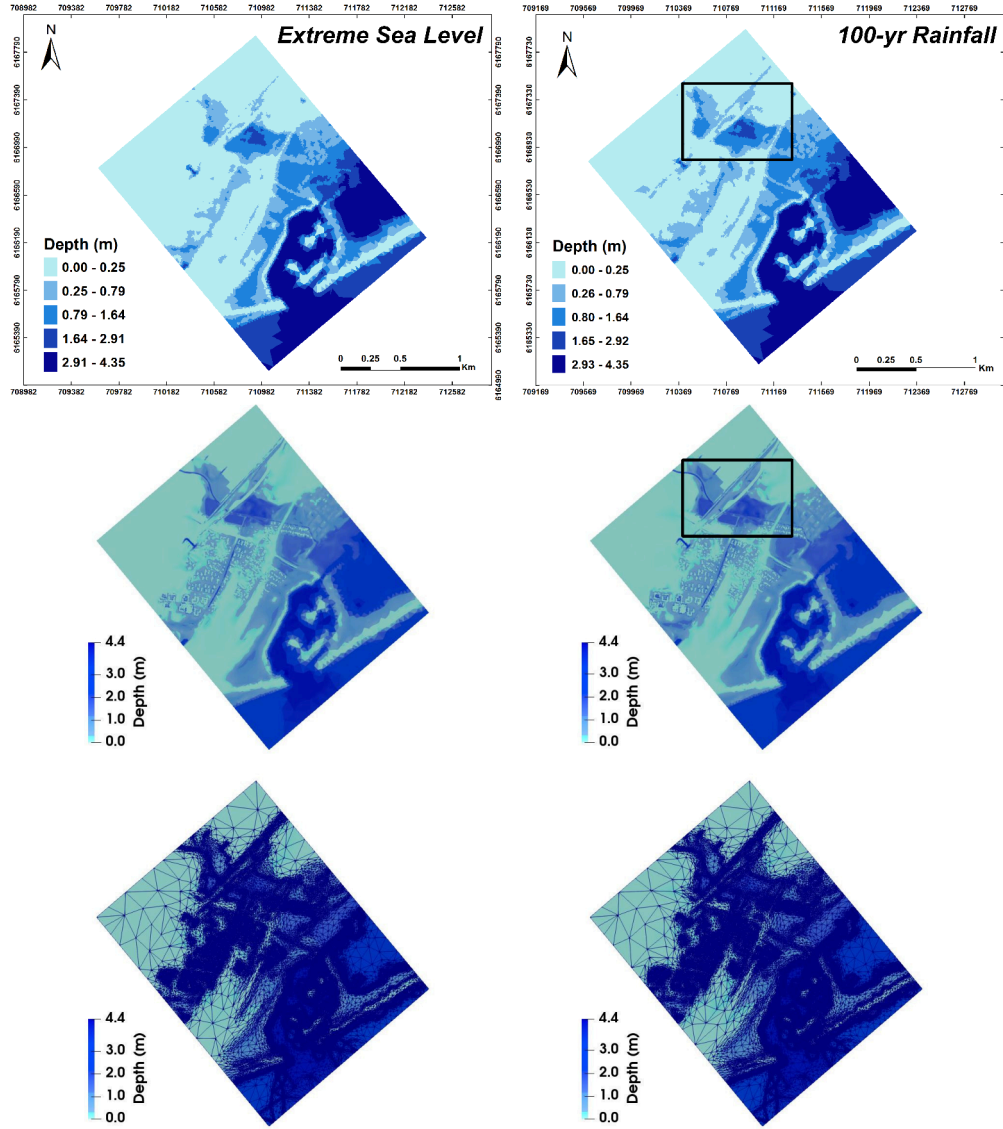


Figure 4.10: Water depths at time level $t = 15 h$ obtained from a MIKE 21 FM model (first row), a Floodity model with a mesh resolution $5 m$ (middle row) and results showing the corresponding mesh (bottom row) based on bathymetric data with buildings in scenarios of an individual 2100 upper extreme sea-level event (left column) and a joint event with 100-yr return period rainfall (right column).

angles in Fig.4.10. It is noted that the information of roads and channels has been lost in the MIKE 21 FM simulations. However, with the use of anisotropic-DMO, Floodity can capture the details of topographic features even with almost the same mesh resolution ($10 m$) as that used in the MIKE simulation. The bottom in Fig.4.11 shows more details of the anisotropic unstructured meshes, where the adapted anisotropic elements are placed

under the limitation of the aspect ratio of elements

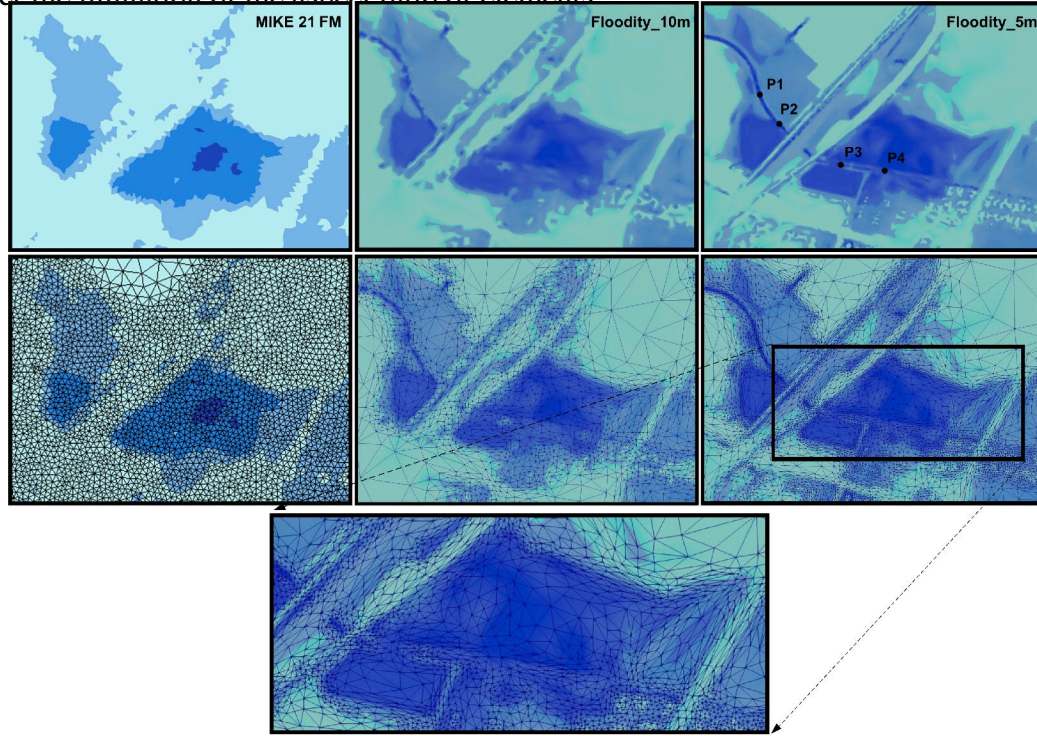


Figure 4.11: Details of the areas marked with rectangles (Fig.4.10) obtained from MIKE 21 FM modelling (left column), a Floodity model with a mesh resolution 10 m (middle column) and 5 m (right column). The bottom shows more details of the anisotropic unstructured meshes.

In these simulations, the topographical data (digital elevation data) is available with a high resolution of 1.6 m . The availability of high-resolution topographical data is important for the accurate numerical simulation of urban food inundation. However, high-resolution topographical data requires a high computational effort, thus, resulting in a computationally demanding flood modelling. Using anisotropic-DMO, the topographical data over the domain is obtained by interpolating the high resolution (1.6 m) data onto the adapted mesh at each time level. Therefore, the high-resolution topographical data is only used in the flooded region while the low-resolution data is used in the rest of the domain, thus reducing the computational cost. In addition, the details of buildings can be represented accurately as the flooding water spreads across the domain (see Figs.4.10 and 4.12).

Fig.4.13 shows the time series of water depth at four detector locations simulated using MIKE 21 FM and Floodity with three different meshes in a scenario of a joint flooding

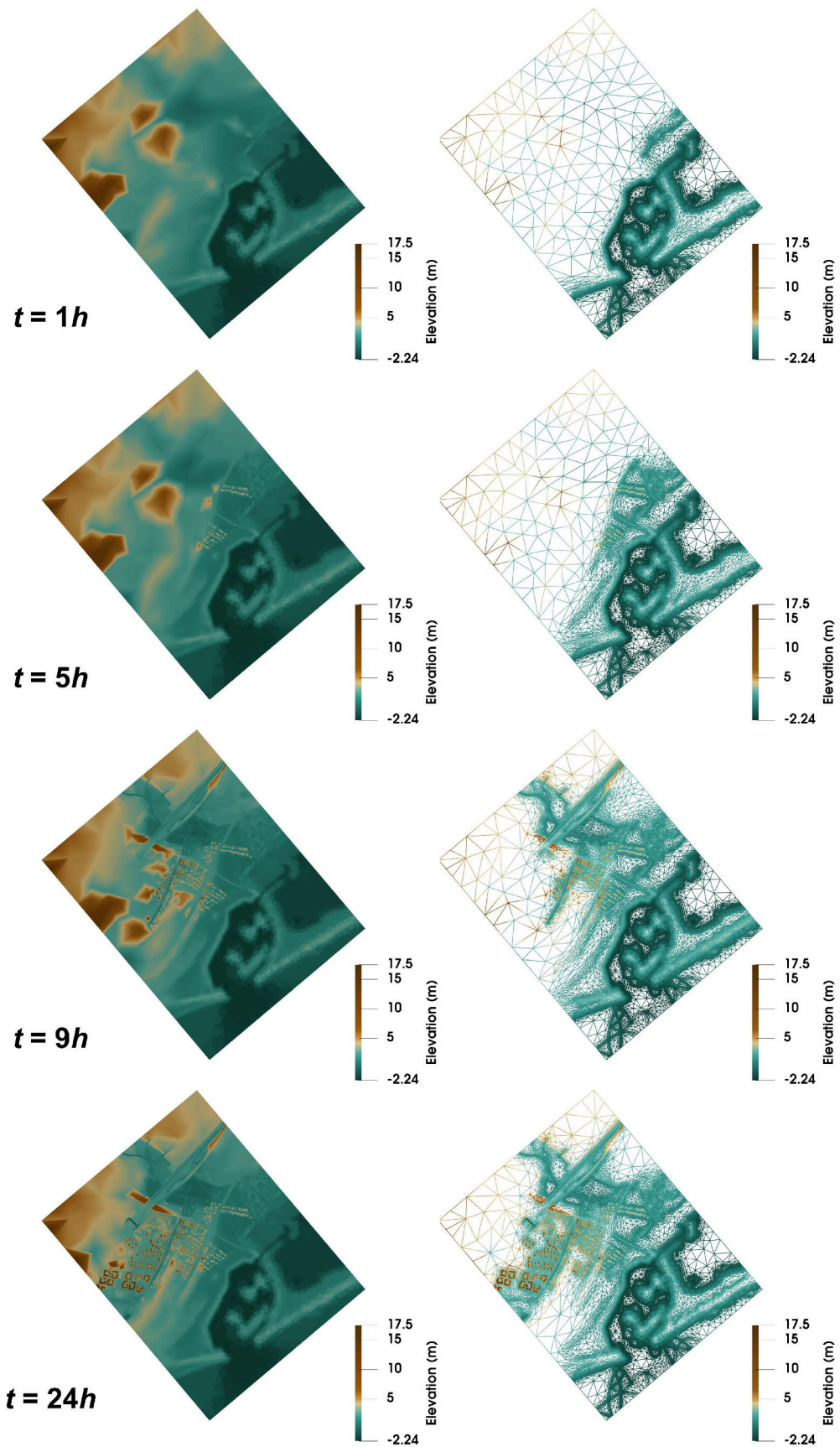


Figure 4.12: Buildings gradually become visible as the flood water spreads west and northward. The left column shows the plane view of surface topography. The right column shows the corresponding mesh.

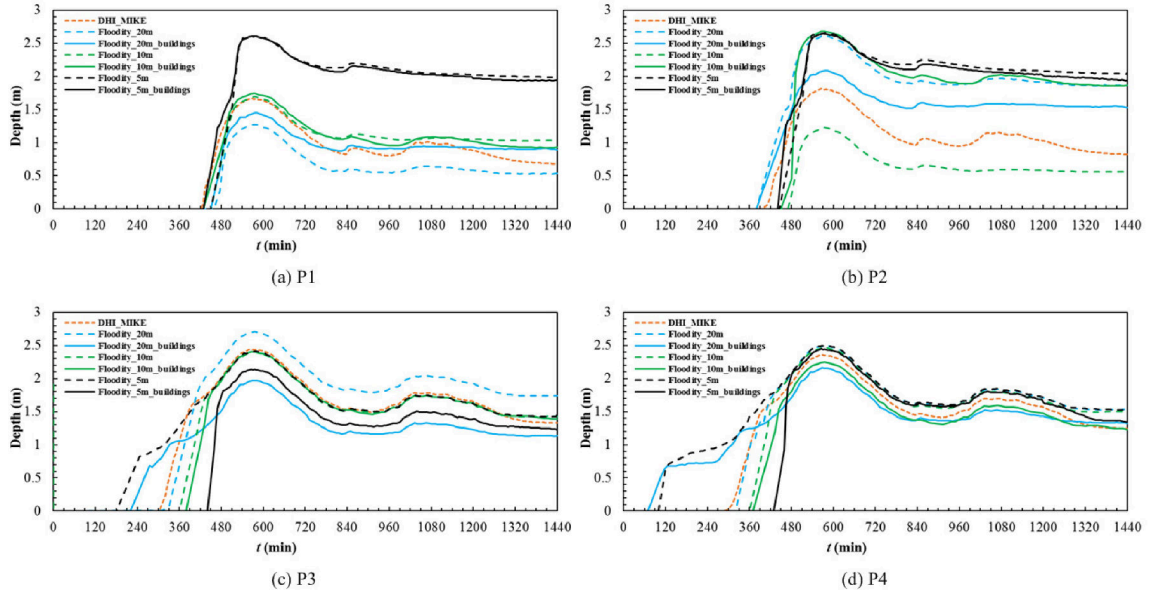


Figure 4.13: Flood depth time series at detector locations P1, P2, P3 and P4 (see Fig.4.1 (b)) simulated using MIKE 21 FM and Floodity with a mesh resolution of 20 m, 10 m, and 5 m respectively, based on bathymetric data with/without buildings in a scenario of 100-yr return period rainfall and 2100 upper extreme sea-level event.

event from extreme sea level and rainfall for a 100-yr return period. It can be observed that due to the impact of buildings, the water depths obtained from Floodity have a slight difference from that of MIKE 21 FM. A deeper water depth at P1 and P2 locations is predicted when using the 5 m resolution adaptive mesh than that using the fixed mesh (MIKE 21 FM), similarly as which is shown in Fig.4.6. This is due to the fact that the detectors P1 and P2 are located at the channel, thus having a larger water depth. Again this proves that the adaptive mesh flooding model can provide relatively accurate predictions. Also note that there is an arrival time lag at detectors if the impact of buildings is considered in the flooding simulations. As a result, the arrival time at P3 and P4 has nearly 140 minutes' time lag in Floodity with a mesh resolution of 5 m results, in comparison to MIKE 21 FM results.

4.5.4 Sensitivity to the forcing inputs and mesh resolution

Extreme joint flooding is the product of a wide range of interacting processes. Here the uncertainties from the forcing inputs are the extreme sea levels and rainfall. In addition, the mesh resolution is one of critical parameters in flooding modelling. In this section, sensitivity analysis of flood volume over inundated areas to extreme sea levels, rainfall and mesh resolution has been explored and shown in Fig.4.3 (a), (b) and (c) respectively.

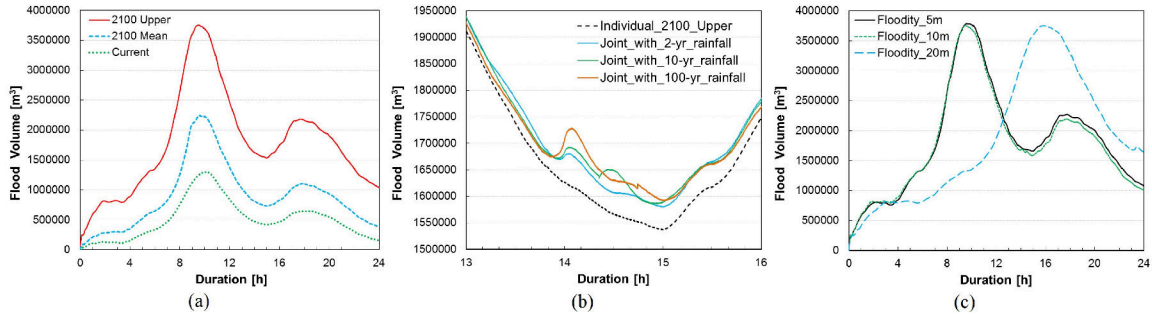


Figure 4.14: Sensitivity of flood volume results to forcing inputs in scenarios: (a) individual extreme sea-level events (shown in Fig.4.3 (b)); (b) both the individual 2100 upper sea-level flood event and the joint event with 100-yr return period rainfall; and (c) mesh resolutions of 20 *m*, 10 *m*, and 5 *m*.

The impact of the incoming sea levels on flooding results has been investigated and the corresponding results in Fig.4.14 (a). There are three scenarios of individual extreme sea-level events shown in Fig.4.3 (b), where the maximum value for the extreme incoming water levels is 1.52 *m* (current), 2.25 *m* (2100 mean) and 3.08 *m* (2100 upper) peaking at $t = 9.5$ *h*. In Fig.4.14 (a), we can see that the flood volume become large with an increased incoming wave level. The flood volume peaks approximately at $t = 9.5$ *h* when the incoming wave is reaching its extreme. There is a slight time lag in the the peak of flood volumes in the scenario of the current extreme sea-level event.

Further investigation of the effect of rainfall on flood volume has been undertaken in the scenario of the joint flood event. A comparison of flood volumes between the scenarios of the individual and joint flood events is provided in Fig.4.14 (b). The solid line is the

flood volume during the individual extreme sea-level event (2100 upper) while the dashed line represents the joint flood event with the 100-yr return period rainfall event during the rainfall period $t = 13 - 16$ h. The influence of rainfall is reflected by the divergence of the flood volume during rainfall. The largest difference in flood volume is $1.10 \times 10^5 m^3$ at $t = 14.08$ h.

Fig.4.14 (c) presents the flood volume results from the simulations using different mesh resolutions. It is found that the peak values of flood volume results are very close in all simulations, while the peak time differs greatly when using the mesh resolutions of 20 m (dotted blue line) and 10 m (dotted green line) as well as 5 m (solid black line). The reason for this is that the blocking effect of buildings cannot be represented in flood modelling with use of coarse mesh resolutions (20 m, here) due to the failure of capturing the details of buildings.

In general, the flood volume results are sensitive to both incoming sea levels and rainfall. However, in joint flood events, the effect of rainfall is relatively small in comparison to extreme incoming waves. In flooding modelling the mesh resolution is the key to capture the details of complex topography, for example, buildings and channels. One can see the blocking effect of buildings only when the buildings are captured with high mesh resolutions.

4.5.5 Performance of Floodity modelling

Table 4.2: Node and element number of the adaptive unstructured mesh with mesh resolution 20 m, 10 m, and 5 m for Floodity modelling.

Meshing Type	Time Level (min)	Node #(20 m)	Element #(20 m)	Node #(10 m)	Element #(10 m)	Node #(5 m)	Element #(5 m)
Fixed mesh	0 - 120	13033	25666	51367	101939	204980	408373
	0	13033	25666	51367	101939	204980	408373
Adaptive mesh	5	3120	6095	8166	16157	19825	39446
	200	3395	6647	9064	17952	21797	43390
	600	4904	9620	12897	25548	29289	58275
	1200	4666	9156	12520	24806	28235	56176
	1440	4581	8990	12271	24315	27485	54686

As seen in Fig.4.15 and Table 4.2, an unstructured mesh with 20 m resolution generated

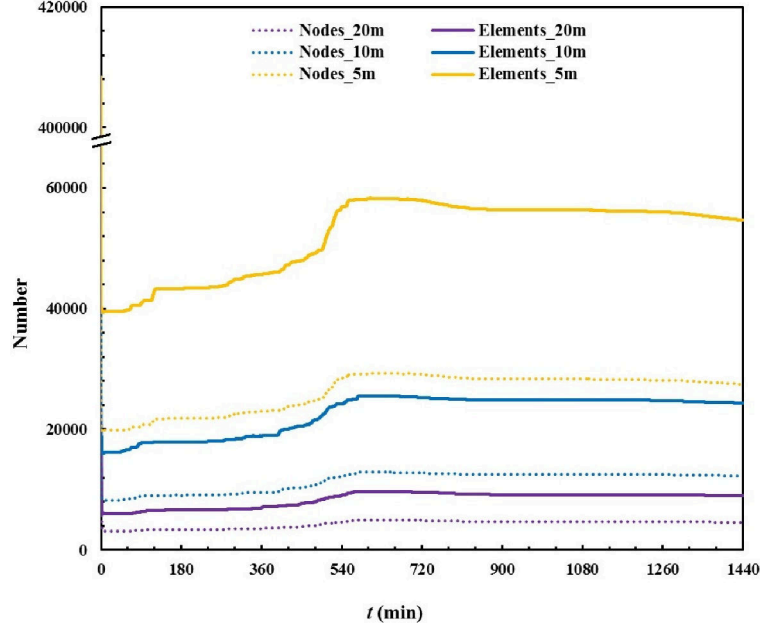


Figure 4.15: The numbers of nodes and elements used in Floodity modelling for mesh resolution 20 m , 10 m , and 5 m during the simulation period $[0, 1440]\text{ min}$.

by Gmsh [137] consists of 13,033 nodes and 25,666 unstructured triangle elements, while 10 m resolution consists of 51,367 nodes and 101,939 elements, and 5 m resolution mesh contains 204,980 nodes and 408,373 elements. These meshes are used during the whole simulation period for fixed unstructured mesh modelling, and used as the initial mesh for the adaptive mesh simulations. After first adapting the mesh, the numbers of nodes and elements used for adaptive mesh of 20 m , 10 m , and 5 m resolution are reduced by 74%, 82% and 88% respectively, then gradually increased during $t = [1, 600]\text{ min}$ and decreased during $t = [600, 1440]\text{ min}$ as the flooding water retreats. Above all, Floodity is less computationally expensive than fixed unstructured mesh modelling. The use of adaptive unstructured meshes improves the computational efficiency. To further reduce the computational cost, various numerical techniques can be adopted in the flood model Floodity, for example, parallel computing using MPI.

4.6 Summary

Realising the importance of flood coincidence risk assessments, we have further developed the adaptive unstructured mesh flooding model Floodity for the joint urban flood events caused by multiple sources (extreme rainfall and sea-level events) and successfully applied to Greve in Denmark. By introducing the anisotropic-DMO technique, the features of flooding flows (local flows around the buildings or the wetting and drying front, for example) are able to be better captured while reducing computational cost without sacrificing accuracy of flooding simulations. With a unique combination of anisotropic-DMO and high-resolution Digital Terrain Model (DTM) data, the complex urban topography can be accurately represented when/where needed by increasing the mesh resolution (around the buildings, for example) dynamically when the flooding water spreads over the urban area. This new Floodity model has been applied to several flooding scenarios that happened in Greve, Denmark, where the flood is induced by different combinations of extreme incoming sea levels and rainfall. A comparison between Floodity and MIKE 21 FM results has been undertaken. It has been found that Floodity is able to provide relatively accurate results while the computational cost is reduced by 20 - 88% in comparison to fixed mesh models. To assess uncertainties in model predictions, the sensitivity of flood volumes to extreme sea levels and rainfalls has been explored. In joint flood events, we found that the flood volume over the inundated area is more sensitive to sea levels than rainfall. Extreme sea-level events with the higher peak water levels induce higher peak flood volume while the impact of rainfall is relatively small. The sensitivity of flood results to the mesh resolution is also investigated. In flood modelling, the blocking effect of buildings on the peak time of flood volumes can be seen only when using high resolution meshes and Digital Terrain Model data.

Flood modelling is a complex and parametric problem. The input uncertainty is one of the main sources of uncertainty. In this work, we mainly focused on the simulation of

flooding from multiple sources. We have done some basic sensitivity analysis. Given its complexity, in future we will further carry out uncertainty analysis using advanced numerical techniques, for example, the adjoint sensitivity and uncertainty analysis [165]. In this work, the effect of rainfall is relatively small compared to incoming waves. So infiltration is not taken into account here, namely all amount of rainfall water becomes ponded water on ground surface. We will further introduce infiltration rate in future work. Due to the lack of optimization of codes, the CPU time is not demonstrated here. Instead, we have demonstrated the computational cost is significantly reduced by the decrease of the number of nodes used while the accuracy remains the same or better than that in fixed mesh modelling. In future work we will focus on the optimization of codes (data structures).

Chapter 5

Rapid spatio-temporal flood prediction and uncertainty quantification using a deep learning method

Recently accrued attention has been given to machine learning approaches for flooding prediction. However, most of these studies focused mainly on time-series flooding prediction at specified sensors, rarely on spatio-temporal prediction of inundations. In this work, an integrated long short-term memory (LSTM) and reduced order model (ROM) framework has been developed. This integrated LSTM-ROM has the capability of representing the spatio-temporal distribution of floods since it takes advantage of both ROM and LSTM. To reduce the dimensional size of large spatial datasets in LSTM, the proper orthogonal decomposition (POD) and singular value decomposition (SVD) approaches are introduced. The LSTM training and prediction processes are carried out over the reduced space. This leads to an improvement of computational efficiency while maintaining the accuracy. The performance of the LSTM-ROM developed here has been evaluated using Okushiri tsunami as test cases. The results obtained from the LSTM-ROM have been compared with those from the full model (*Fluidity*). In predictive analytics, it is shown that the results from both the full model and LSTM-ROM are in a good agreement whilst the CPU cost using the LSTM-ROM is decreased by three orders of magnitude compared to full model simulations. Additionally, prescriptive analytics has been undertaken to estimate the uncertainty in flood induced conditions. Given the time series of the free surface height at a specified detector, the corresponding induced wave conditions along the coastline have then been provided using the LSTM network. Promising results indicate that the use of LSTM-ROM can provide the flood prediction in seconds, enabling us to provide real-time predictions

and inform the public in a timely manner, reducing injuries and fatalities.

5.1 Introduction

Flooding causes considerable damage to people, infrastructure and economies in many countries of the world. During a flooding event, a rapid response management is critical to reducing damage resulting from a flood event. Emergency managers require timely and accurate information on the areas affected by floodwater to plan mitigation measures against damage. In recent years, the Early Warning Systems (EWS) have been widely used to better predict floods and reduce their impact on urbanized areas. Learning from the past flood events is vital to enhancing the capability of flood forecasting.

Flood modelling involves the complex and nonlinear flow and physical processes. Compared to a physically based numerical model, machine learning approaches can provide a powerful way for flooding prediction without explicitly knowing such nonlinear dynamic processes [166]. In the past, great efforts have been made to flooding prediction using machine learning methods, for example, autoencoder [167], genetic algorithm (GA) [168], artificial neural networks (ANN) [169], back propagation neural networks (BPNN) [170], radial basis function neural networks (RBFNN) [171] and recurrent neural networks (RNN) [172, 173]. Most of these studies focused mainly on time-series flooding prediction (e.g. water levels, streamflow and discharge) at specified sensors [166], but rarely addressed spatio-temporal prediction of inundations since it is difficult to handle large spatial data sets.

In this work, we propose the long short-term memory (LSTM) for flood prediction. LSTM was introduced by Sepp Hochreiter and Jurgen Schmidhuber [174]. LSTMs are able to learn non-linear functions of arbitrary-length input sequences and effectively capture long-term temporal dependencies. They have been widely used in various research areas, such as, language modelling [175], handwriting recognition [176–178], speech synthesis [179], audio analysis [180], and video recognition [181].

For efficient LSTM training and spatio-temporal prediction in flood events, ROM techniques are introduced to LSTM in this study. ROM techniques have been widely used in various research fields including fluid dynamics [182, 183], molecular dynamics [184], heat transfer [185], data assimilation [93, 186], elasticity problems [187], shape optimization [188], and aeroplane components design [189]. Proper Orthogonal Decomposition (POD) is often used for generating reduced order models. POD-ROM approaches have been widely applied to many fields [86, 184, 190–192]. There are numerous hyper-reduction POD approaches, for example, in combination with Galerkin projection [193], discrete empirical interpolation method (DEIM) for nonlinear problems [194], Gauss-Newton with approximated tensors (GNAT) [195], quadratic expansion method [196], radial basis function (RBF) [197, 198], and Smolyak sparse grid [199].

More recently, machine learning techniques have been introduced to ROM [29, 200–204]. Kerschen and Golinval (2003) [205] presented a machine learning approach for the nonlinear modal analysis. The authors combined auto-associative neural network (AANN) with nonlinear Principal Component Analysis (PCA) and this proposed method has been successfully applied in the context of model reduction for nonlinear systems [206]. San and Maulik (2018) [200] introduced the artificial neural networks to proper orthogonal decomposition-based ROMs and the newly developed ROM-ANN framework has been successfully applied to a wind-driven ocean circulation problem. The ROM-ANN satisfies the dual demands of statistical accuracy as well as low computational expense and effectively retains the dynamics of the full-order model during the simulation period. The method is proved to be robust for larger choices of time steps, which is an efficient and reliable tool in long-term predictions of geophysical turbulent flows. Swischuk et al. (2019) [201] develops a low-dimensional parametrization of high-dimensional output quantities of interest (e.g. pressure, temperature and strain fields) using POD, and combines this parametrization with machine learning methods to learn the relationship between the input parameters and the POD expansion coefficients. Two engineering examples demonstrate

the embedding physical constraints that less training data is available in engineering machine learning applications than non-engineering applications, making it essential to choose an appropriate machine learning strategy with knowledge incorporated from physical models.

The integrated LSTM and ROM framework developed here takes advantage of both ROM and RNN-LSTM. This is the first time that LSTM network based on ROM has been considered in flood forecasting and used to estimate spatial aggregation of inundations. It should be noted that having the compatibility of LSTM and ROM will enable solving predictive problems efficiently and accurately. In this work, the LSTM-ROM framework has the capability of predictive and prescriptive analytics. Predictive analysis is used to forecast the future by learning patterns from holistic/experience data. Prescriptive analytics is used for inverse problems. In prescriptive analytics the past/estimated data is used to explore dependencies among results. These previous learned dependencies are then used to estimate the cause factors given the prescribed results at a future time. This means that predictive analytics tells what will happen, while prescriptive analytics provides the cause factors and suggests what to do. Predictive and prescriptive analytics with big data are becoming more and more prevalent in industries [207, 208]. To the best of our knowledge, there has not been any particular study which incorporates these two types of analytics together to provide a framework to address flood issues. The goal of this work is to develop an advanced fast-running computational model to predict urban floods and guide effective response in the event of emergencies. It will lead to a step change in the speed of forecasting, with possible CPU reductions of several orders of magnitude compared to existing methods.

The layout of this chapter is as follows. Section 5.2 introduces the methodology of LSTM. The integrated LSTM-ROM framework for flood prediction is provided in detail in section 5.3 while LSTM for predictive and prescriptive analytics in flood prediction is described in Section 5.4. Section 5.5 demonstrates the performance of the LSTM-ROM using Okushiri tsunami as test cases. Finally in Section 5.6, a summary and future work

are presented.

5.2 Construction of the POD-based ROM

Specifically in this work, we assume that a set of optimal truncated spatial basis functions $\phi_m(x)$ (where x is the spatial coordinate) can be obtained from solution snapshots [209]. Once the dominant flow spatial basis functions (also called POD modes) are obtained, the state variables representing the evolution of the underlying flow dynamics can approximately be expressed by a linear combination of the dominant spatial modes and temporal coefficients:

$$U \approx \Phi_k \mathbf{a} \text{ or } u(x, t) \approx \sum_{k=1}^K a_k(t) \phi_k(x), \quad (5.1)$$

where u is the state variable, $\phi_k(x)$ is the POD spatial mode while the coefficients $a_k(t)$, $\mathbf{a} \in \mathbb{R}^K$, is the corresponding temporal coefficients.

Let U be the snapshot matrix containing the state solutions $u(x, t_n) = u^n(x) \in \mathbb{R}^{\mathbb{N}}$ at instances of the temporal sequence during the period $[0, t_{N_t}]$, that is, $U = [u^1, \dots, u^{N_t}] \in \mathbb{R}^{N_t \times \mathbb{N}}$, where N_t is the number of snapshots, and \mathbb{N} is the number of spatial points or nodes where data or solution is provided in the computational domain. Using SVD, the matrix U can be written:

$$U = \Phi \Sigma V^T \approx \Phi_K \Sigma_K V_K^T, \quad (5.2)$$

where the matrices Φ_K and V_K contain the first K columns of Φ and V (left and right singular vectors, respectively), and the diagonal matrix Σ_K contains the first $K \times K$ block of Σ (singular values). Alternatively, the left singular vector V_K can be obtained by solving the eigenvalue problem [83]:

$$U^T U V_K = V_K \Sigma_K^2, \quad (5.3)$$

and the first K POD spatial modes corresponding to nontrivial singular values are calcu-

lated:

$$\Phi_K = UV_K \Sigma_K^{-1}, \quad (5.4)$$

while the POD temporal coefficient \mathbf{a} is given by the orthogonal projection:

$$\mathbf{a} = (\mathbf{a}^1, \dots, \mathbf{a}^n, \dots, \mathbf{a}^{N_t}) = \Phi_K^T U, \quad (5.5)$$

where $\mathbf{a}^n = (a_1^n, \dots, a_K^n)$. Usually, K ($K \ll \mathbb{N}$) is chosen to make sure a tolerance percentage of energy of the full space to be captured [83].

The eigenvalue problem of the correlation matrix $V \Sigma U^T U \Sigma V^T = V \Sigma^2 V^T \in \Re^{N_t \times N_t}$ is solved [83]:

$$U^T U V_i = \lambda_i V_i, \quad i \in \{1, \dots, N_t\}, \quad (5.6)$$

where $\lambda_i = \sigma_i^2$ is the eigenvalue of $U^T U$. The eigenvalues are ordered $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$ while the eigenvectors are orthogonal. The K leading left eigenvectors are chosen to be the POD modes for constructing the reduced space $\Phi = (\phi_1, \dots, \phi_K) = (U_1, \dots, U_K)$. The eigenvector associated with the largest eigenvalue is the direction along which the data have the most variance of the projection. The variance of the projection can be calculated by $\sum_{m=1}^K \lambda_m$ while the percentage of the total energy captured by the POD modes is:

$$E(K) = \frac{\sum_{i=1}^K \lambda_i}{\sum_{i=1}^{N_t} \lambda_i}, \quad (5.7)$$

The number of POD modes K is chosen such that

$$K = \operatorname{argmin} \{E(k) : E(k) \geq \gamma\}, \quad (5.8)$$

where γ ($0 \leq \gamma \leq 1$) is the tolerance percentage chosen to enable most of the energy of the full system to be captured. For more details on SVD, we refer the reader to [210].

5.3 LSTM-ROM for predictive modelling of the spatio-temporal distribution of floods

As discussed above, most of existing flooding studies based on machine learning techniques are limited to either the time-series flooding prediction at sensors or the simulations of the spatial distribution of floods. The main objective of this work is to develop a new LSTM tool for predicting the spatio-temporal distribution of floods. In spatio-temporal simulations, the input dataset in Eq. (2.7) is the spatial function and can be rewritten:

$$x_s = (h_{b,s}, \mu), \quad (5.9)$$

where $h_{b,s}$ is the flood induced condition (here, an incoming wave along the coastal boundary), and μ represents the uncertainty (drag coefficient, for example) in modelling, while the output in Eq. (2.7) is the spatio-temporal distribution during the specified flooding period $[0, t_{N_t}]$, that is,

$$h_s = (h_{s,0}, \dots, h_{s,t_{n_t}}, \dots, h_{s,t_{N_t}}), \quad (5.10)$$

where $h_{s,t_{n_t}}$ is the variable spatial solution at time level t_{n_t} (here, $n_t \in \{0, \dots, N_t\}$, $t_{n_t} \in [0, t_{N_t}]$ and N_t is the number of time instances). In a discretisation form, the spatial variables in Eqs. (5.9) and (5.10) can be rewritten:

$$h_{s,0} = (h_{s,0,1}, \dots, h_{s,0,n}, \dots, h_{s,0,\mathcal{N}}), \quad h_{s,t_{n_t}} = (h_{s,t_{n_t},1}, \dots, h_{s,t_{n_t},n}, \dots, h_{s,t_{n_t},\mathcal{N}}), \quad (5.11)$$

and the variables:

- $h_{s,0}, h_{s,t_{n_t}}, h_{s,t_{N_t}} \in \mathbb{R}^{\mathcal{N}}$, where \mathcal{N} is the number of nodes where data or solution is provided in the computational domain Ω ;
- $h_b \in \mathbb{R}^{N_b}$, where N_b is the number of parameters used for defining the flood induced conditions (for example, the peak values and period in an incoming wave);

- $\mu \in \mathfrak{R}^{N_m}$, where N_m is the number of uncertainties in modelling;
- $x_s \in \mathfrak{R}^{N_b+N_m}$;
- $h_s \in \mathfrak{R}^{N_t \times \mathcal{N}}$.

One may note that for spatio-temporal prediction of floods within a large flooding area, huge spatial datasets have to be handled during the training and predictive processes. To reduce the size of data sets, a ROM based on proper orthogonal decomposition (POD) is introduced to LSTM in this work.

Let H the snapshot matrix containing flooding variables at instances of the temporal sequence during the flooding period $[0, t_{N_t}]$, that is,

$$H = (h_{s,0}^T, \dots, h_{s,t_{n_t}}^T, \dots, h_{s,t_{N_t}}^T), \quad (5.12)$$

where $H \in \mathfrak{R}^{\mathcal{N} \times N_t}$, $\mathcal{N} \gg N_t$, the flooding dataset $\{h_{s,t_{n_t}}\}$ is constructed from the output of Eq. (2.7) and the superscript T denotes the transpose.

Once the POD modes are obtained, the solution snapshot at each instance of the data sequence can be re-constructed as the linear combination of the POD modes:

$$H \approx \sum_{m=1}^K \sqrt{\lambda_m} \phi_m V_m^T, \quad (5.13)$$

where the truncated error (the residual of the projection) is $\sum_{m=K+1}^{N_t} \lambda_m$.

For any input $x_s \in \mathcal{I} \setminus \mathcal{I}_{tr}$ in Eq. (5.9) (where \mathcal{I} and \mathcal{I}_{tr} are the whole input dataset and the training input dataset respectively), the predictive solution at each time instance n_t can be expressed as a linear combination of POD modes:

$$h_{s,n_t}^T = \Phi a_{s,n_t}^T = \sum_{m=1}^K a_{s,n_t,m} \phi_m, \quad (5.14)$$

where a_{s,m,n_t} is the time-dependent coefficient associated with the corresponding POD mode ϕ_m . The coefficients $a_{s,t_{n_t}} = (a_{s,n_t,1}, \dots, a_{s,n_t,m}, \dots, a_{s,n_t,K})$ can be calculated using the LSTM described in Eq. (2.7). The output $h_s = (h_{s,0}, \dots, h_{s,t_{n_t}}, \dots, h_{s,t_{N_t}})$ in Eq. (5.10) is projected onto the low-dimensional space spanned by $\{\phi_1, \dots, \phi_m, \dots, \phi_K\}$:

$$a_s = (a_{s,1}, \dots, a_{s,t_{n_t}}, \dots, a_{s,t_{N_t}}) = (h_{s,0}\Phi, \dots, h_{s,t_{n_t}}\Phi, \dots, h_{s,t_{N_t}}\Phi). \quad (5.15)$$

Note that the dimensional size of the output dataset in Eq. (5.10) is reduced from $\mathcal{N} \times N_t$ to $K \times N_t$ ($K \ll \mathcal{N}$). The coefficient set a_s can replace h_s in Eq. (2.7) and be obtained using LSTM.

5.4 LSTM in predictive and prescriptive analytics of floods

In this section, taking an example of flooding caused by incoming waves (but not limited to), we demonstrate the way how to use LSTM for flood prediction and uncertainty quantification.

5.4.1 Predictive analytics: the spatio-temporal model based on LSTM-ROM

Predictive analytics \implies Given $\{x_s\}$, find $y_s = f(x_s, t)$. Given a set of inputs $\{x_s \in \mathcal{I}_{tr}\}$ (where \mathcal{I}_{tr} is the training input dataset), using machine learning techniques, one can find a mathematical predictive model, $f(x_s, t)$, which is used for providing the variable values at a future time t . In flooding simulations, for a set of given training input-output pairs $(x_s, y_s) \in (\mathcal{I}_{tr}, \mathcal{O}_{tr})$, one first uses the past flooding data and machine learning techniques to construct the predictive model. For a new flood induced condition $x_s \in \mathcal{I} \setminus \mathcal{I}_{tr}$, one can forecast the future floods. To tackle the challenge of large spatio-temporal datasets, the ROM/POD described above can be used to reduce the dimensional size of data. Both the training and predictive processes are then undertaken in a reduced space, thus reducing significantly the computational cost.

The input dataset is the incoming wave boundary condition:

$$x_s = (h_{b,s}^{cr}, h_{b,s}^{dr}, T_{b,s}), \quad (5.16)$$

where, $h_{b,s}^{cr}$ and $h_{b,s}^{dr}$ are the crest and trough wave heights respectively, and $T_{b,s}$ is the wave period. Three input variables $h_{b,s}^{cr}$, $h_{b,s}^{dr}$ and $T_{b,s}$ are used to define the incoming wave [211].

Given S_w as the number of incoming wave series used for training purpose, the input data x_{s_w} in the training input dataset \mathcal{I}_{tr} are:

$$x_{s_w} = (h_{b,s_w}^{cr}, h_{b,s_w}^{dr}, T_{b,s_w}), \quad s_w \in \{1, \dots, S_w\}. \quad (5.17)$$

The corresponding output data h_{s_w} in the training dataset \mathcal{O}_{tr} are:

$$h_{s_w} = (h_{s_w,0}^T, \dots, h_{s_w,t_{n_t}}^T, \dots, h_{s_w,t_{N_t}}^T), \quad s_w \in \{1, \dots, S_w\}. \quad (5.18)$$

To reduce the dimensional size of $h_{s_w} \in \mathbb{R}^{N \times N_t}$, using POD/SVD a set of POD modes is obtained from the snapshots $(h_1, \dots, h_{s_w}, \dots, h_{S_w})$ and used for constructing the reduced space $\mathfrak{R}_m = \text{span}\{\phi_1, \dots, \phi_K\}$. Projecting the training output data h_{s_w} onto the reduced space yields:

$$a_{s_w} = (a_{s_w,1}, \dots, a_{s_w,t_{n_t}}, \dots, a_{s_w,t_{N_t}}) = (h_{s_w,0}\Phi, \dots, h_{s_w,t_{n_t}}\Phi, \dots, h_{s_w,t_{N_t}}\Phi), \quad (5.19)$$

where $a_{s_w} \in \mathbb{R}^{K \times N_t}$. The input-output dataset in Eq. (2.7) is (x_{s_w}, a_{s_w}) , $s_w \in \{1, \dots, S_w\}$. The LSTM training process is thus undertaken in the reduced space. In LSTM, most of input-output pairs $(x_{s_w}, a_{s_w}) \in (\mathcal{I}_{tr}, \mathcal{O}_{tr})$ are used for training while the rest for validation.

Once the LSTM training and validation processes are completed, one can find the relationship between the input-output pair (x_s, a_s) in the reduced space. By using LSTM based on ROM, the predictive solutions during the flooding period $[0, t_{N_t}]$ for any input

data $x_s \in \mathcal{I} \setminus \mathcal{I}_{tr}$ can be obtained in Eq. (5.14).

The framework of LSTM-ROM methodology is demonstrated in Fig. 5.1.

5.4.2 Prescriptive analytics for uncertainty quantification in flooding modelling

Prescriptive/inverse analytics: \implies Given y_s , find $x_s = f^{-1}(y_s, t)$. Given a prescribed outcome y_s at time level t , using the previously learned dependencies, one can find an inverse function, $f^{-1}(y_s, t)$, which is used for optimising the inputs (or uncertainties in modelling). Flood modelling is a complex and parametric problem. The flood induced condition is the main factor of uncertainties in flood risk management. In this work, LSTM is used to conduct the prescriptive analytics to estimate the uncertainty in flood induced conditions (incoming waves) for given historical/estimated flood results.

To construct the LSTM network in the prescriptive analytics, the training input is the temporal variation of one flood characteristic (e.g. free surface height, water depth, quantity of flow, or velocity) at specified sensor locations, while the corresponding flood induced condition (e.g. waves or rainfall) is the output. Subsequently, the LSTM network is built through learning with the input-output pairs. Finally, given a new estimated temporal variation of the flood characteristic, the corresponding flood induced condition will be provided using the LSTM network.

As an example, for a series of induced waves, the input dataset is the prescribed free surface height at specified sensors:

$$h_{s_w}(\mathbf{x}_d) = (h_{s_w,0}\mathbf{x}_d, \dots, h_{s_w,t_{n_t}}\mathbf{x}_d, \dots, h_{s_w,t_{N_t}}\mathbf{x}_d), \quad s_w \in \{1, \dots, S_w\}, \quad d \in \{1, \dots, \mathcal{N}_d\}, \quad (5.20)$$

where \mathcal{N}_d is the number of sensors, while the corresponding induced wave is uncertainty to be determined, that is, the output data is the parameters used for defining the induced wave $x_s = h_{b,s} = (h_{b,s}^{cr}, h_{b,s}^{dr}, T_s)$. Using LSTM described in Eq. (2.7), the relationship $x_s = h_{b,s}(\mathbf{x}_d) = f^{-1}(h_{s_w}(\mathbf{x}_d))$ can be found and used to estimate the induced wave condition for

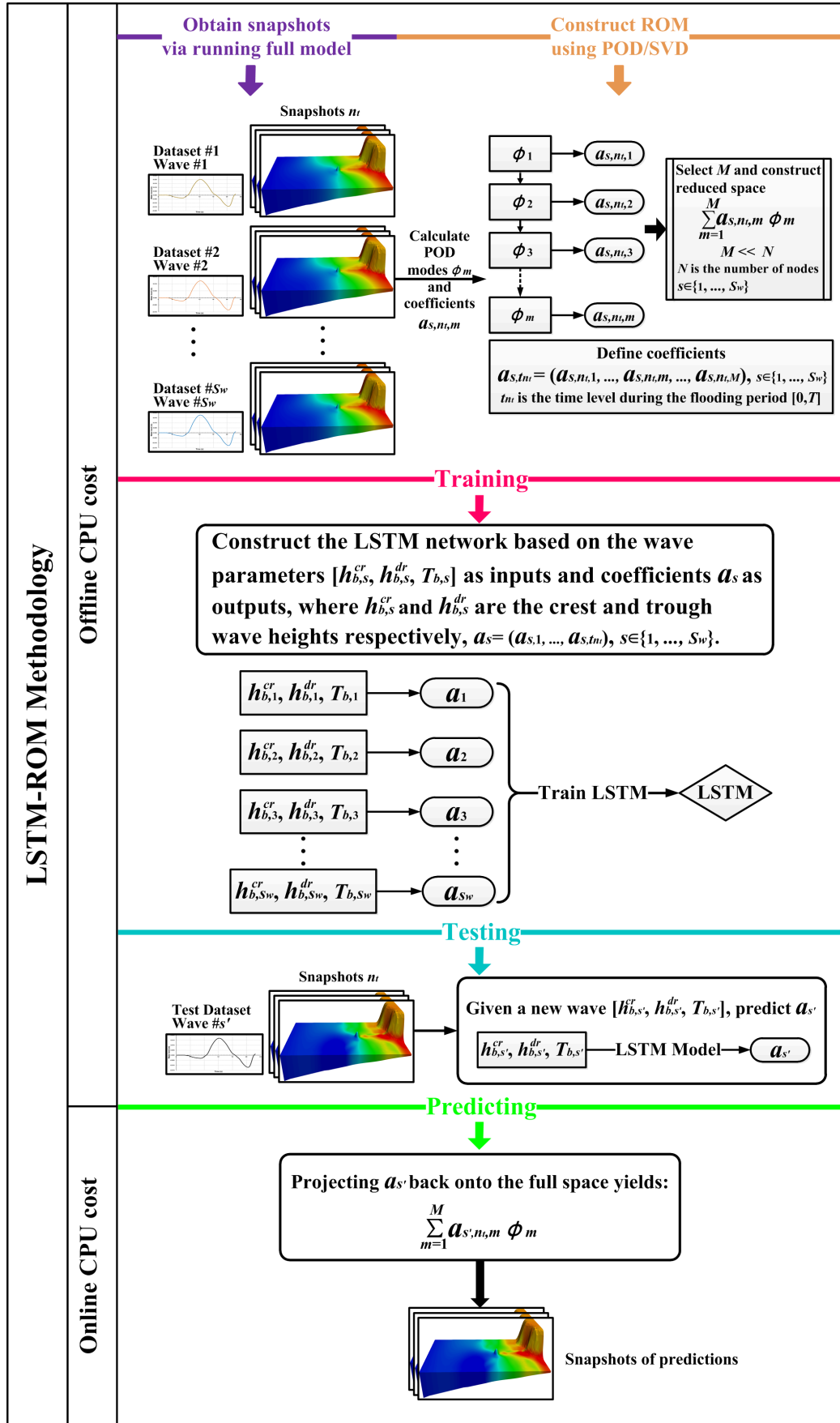


Figure 5.1: Framework of LSTM-ROM.

given observational data at specified sensors.

5.5 Numerical examples

The example used for validation of the new LSTM-ROM is the Okushiri tsunami test case. The event is the Okushiri tsunami in 1993 caused by the Hokkaido Nansei-Oki earthquake offshore of southwestern Hokkaido Island, Japan. The resulting tsunami hit a sparsely populated part of the Okushiri Island, Japan with a runup height of up to 30 *m*. To investigate the danger of such extreme events, the Research Institute for Electric Power Industry (CRIEPI) in Abiko, Japan constructed a scaled-down laboratory model of the area around the island [212]. The inundated area measures 5.448 *m* \times 3.402 *m* with walls on each side except the left where the water level is enforced (see Fig 5.2). Fig. 5.2 shows the bathymetry and the three gauge stations, as well as the actual induced wave data used for the example case.

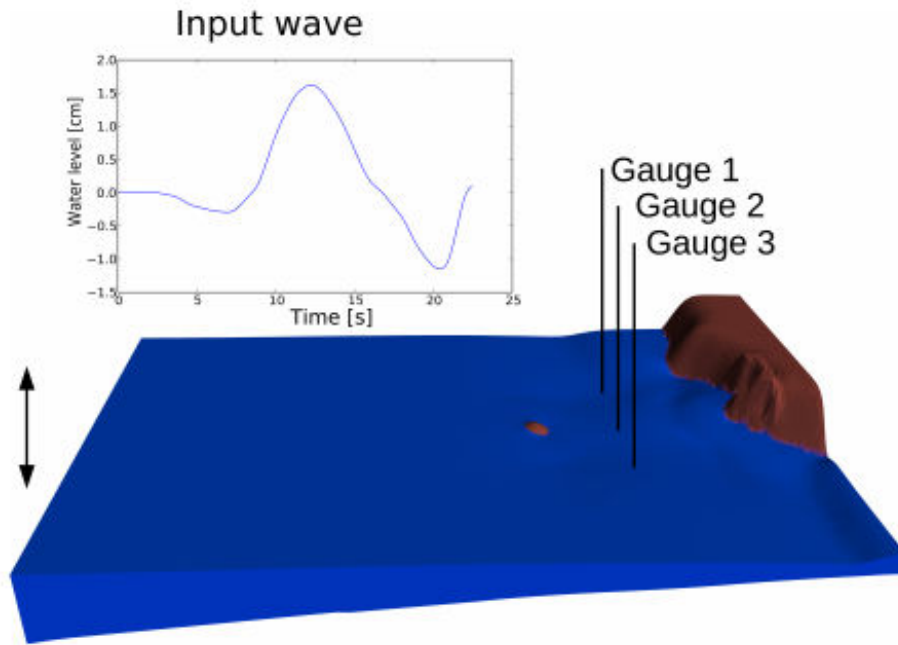


Figure 5.2: The bathymetry and the three gauge stations used for the Hokkaido-Nansei-Oki tsunami example [213].

The training data is from the existing 3D simulation in (*Fluidity*) [213], and the mesh used for the simulation is a 2D single layer horizontally unstructured mesh. *Fluidity* can extrude this 2D mesh to create a layered 3D mesh. The extrusion is downwards (in the direction of gravity), and the top of the domain is always flat. It is performed according to bathymetry such that the depth of the extrusion conforms to bathymetric data. The extruded 3D mesh is 2-layer mesh with a top surface and bottom surface, in which all the nodes line up in vertical lines. In this case study, only water depth solution is included in the training set since 1) water depth is the main concern in 2D flood modelling, and 2) the laboratory data of surface heights is available at three gauges. The equations are solved with the $P_1 - P_1$ finite element pair and a backward Euler time discretisation. The threshold value of wetting and drying (d_0) is set to be 0.5 mm in dry area to prevent non-physical flows in numerical simulations. A Manning-Strickler drag is used at the bottom with $n = 0.002 \text{ sm}^{-1/3}$. The acceleration of gravity magnitude is 9.81 ms^{-2} . In this case, the simulation period is $[0, 70] \text{ s}$, and a time step size is 0.1 s is used. Thus, the training data were obtained by running the full model (*Fluidity*), with an unstructured mesh of 6894 nodes. A fixed mesh scheme is used since the same dimensional size of snapshots is required for constructing POD modes.

5.5.1 Predictive analytics in flooding modelling

Calculation of snapshots via running the full model: As seen in Fig. 5.1, for training purpose, given a series of incoming waves (here, $s_w = 1, \dots, 10$), the corresponding solution snapshots ($h_{s_w} \in \mathcal{O}_{tr}$) are obtained from the full model simulations. By scaling the actual induced wave (in Fig. 5.1) with a factor of $0.7 - 1.25$, 10 series of incoming waves (in Fig. 5.3) are given as inputs. Thus, for each incoming wave x_{s_w} , the collection of 100 snapshots during the period $[25, 35] \text{ s}$ for the free surface height is the output training dataset h_{s_w} . The dimensional size of the output dataset is large, i.e. $\mathcal{N} \times N_t = 6894 \times 100 = 689400$. Reduction of the size of the training dataset is necessary for LSTM prediction.

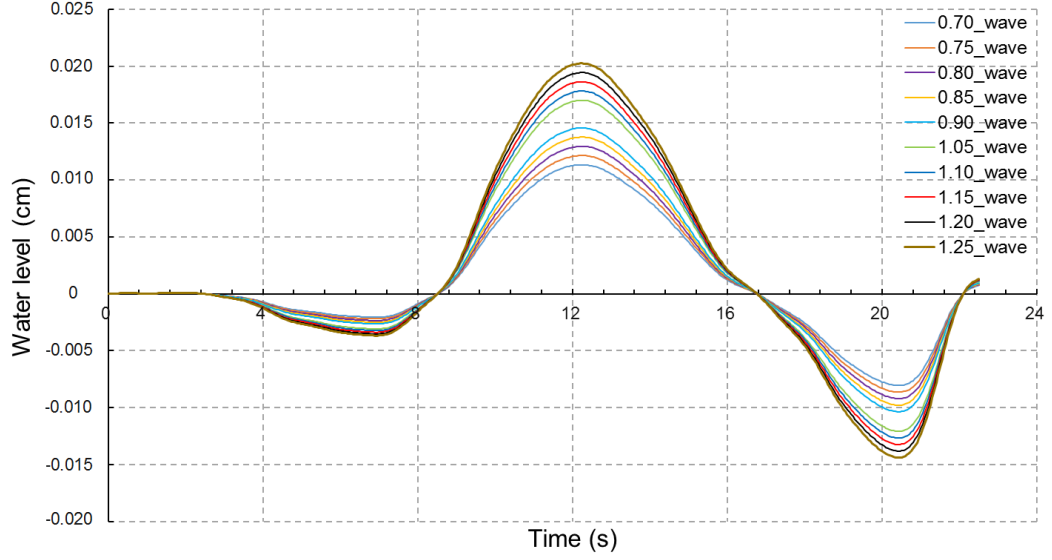


Figure 5.3: The series of flood induced waves for training purpose.

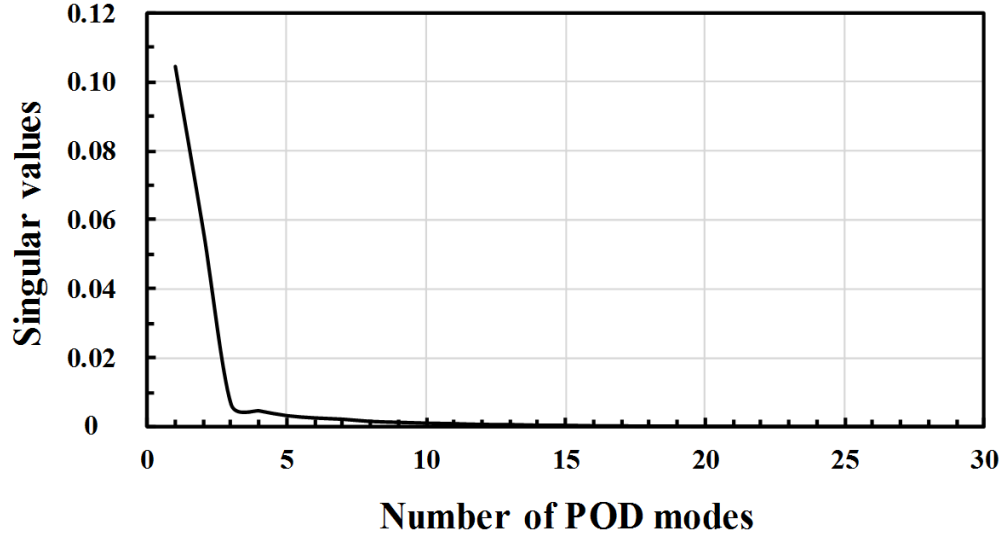


Figure 5.4: Singular values of free surface heights to construct the reduced space.

Construction of LSTM-ROM: With the datasets of snapshots obtained above, a LSTM-ROM model is generated using POD/SVD techniques. Fig. 5.4 shows that the singular values decreases dramatically by two orders of magnitude between the first 3 leading POD modes and 99.5% of the total energy is captured by 20 POD modes. In this work, the selected number of POD modes equals to the number of LSTM models need to be trained and

fitted. Less LSTM models results in less predictions of temporal coefficients for each step, which could cause over simplification and low accuracy when projecting back to generate snapshot. Thus, the first 20 leading POD modes ($K = 20$) were chosen for constructing the reduced space, thus ensuring the predictive accuracy. The training and prediction processes were then undertaken over the reduced space. The training input dataset is the parameters used to define the incoming waves, $x_{s_w} = (h_{b,s_w}^{cr}, h_{b,s_w}^{dr}, T_{b,s_w})$, while the corresponding output dataset $a_{s_w} \in \mathcal{R}^{K \times N_t}$ (where $K = 20$ and $N_t = 100$) is obtained by projecting $h_{s_w} \in \mathcal{R}^{\mathcal{N} \times N_t}$ (where $\mathcal{N} = 6894$) onto the reduced space. By using ROM, the dimensional size of the output dataset is reduced from 689400 to 2000. In LSTM model, the epochs is a measure of the number of times passing through the entire training dataset. The number of epochs should be large enough to allow the learning model fit optimally to the data. Here, the number of epochs is set to 1000 and RMSprop optimizer [214] is used here. The LSTM model contains 3 layers: 1 input layer, 1 hidden layer, and 1 output layer.

Prediction of floods using LSTM-ROM: To evaluate the predicability of the LSTM-ROM, for the given actual input wave ($x_s \in \mathcal{I} \setminus \mathcal{I}_{tr}$ shown in Fig. 5.1), a comparison between the LSTM-ROM and full modelling results is carried out (see Fig. 5.5). It is worth noting that results from the full model and LSTM-ROM are in very good agreement with each other. As shown in Fig. 5.6, the differences of the free surface height between the full model and LSTM-ROM are quite small over the whole domain, which suggests that the accuracy of LSTM-ROM is maintained whilst the CPU cost is reduced by several orders of magnitude in comparison with the full model.

To further estimate the performance of the LSTM model, the root mean square error (RMSE) is calculated for each time step by,

$$RMSE_j = \sqrt{\frac{\sum_{i=1}^{\mathcal{N}} (S_{full}^{i,j} - S_{LSTM-ROM}^{i,j})^2}{\mathcal{N}}}, \quad i = 1, 2, \dots, \mathcal{N}, j = 1, 2, \dots, N_t. \quad (5.21)$$

where $\mathcal{N} = 6894$ denotes the number of nodes on the full mesh; $N_t = 20$ is the number

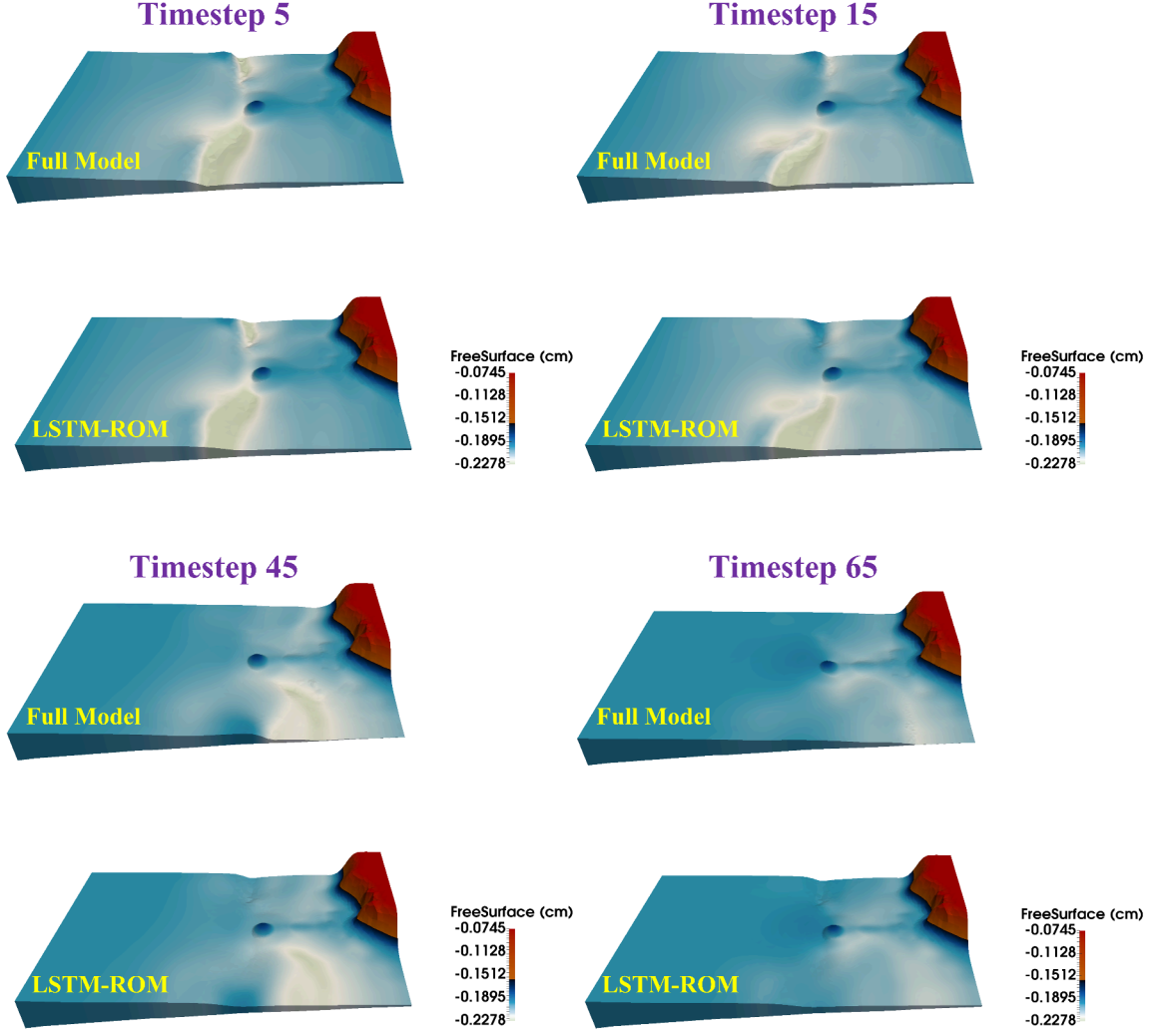


Figure 5.5: The comparison of free surface heights between the full model and LSTM-ROM.

of time step; $S_{full}^{i,j}$ and $S_{LSTM-ROM}^{i,j}$ represent the solution on node i at time step j from the original full model and LSTM-ROM, respectively. As shown in Fig. 5.7, the errors are quite small, demonstrating that LSTM-ROM performs very well and the predicted solutions are in good agreement with the true solutions.

The LSTM prediction includes both offline and online procedures. The offline CPU cost is defined as the time for pre-computing, that is, the calculation of snapshots and POD modes and LSTM training. The online CPU cost is the time required for prediction of floods using LSTM-ROM for any given flood induced condition. In this study, the

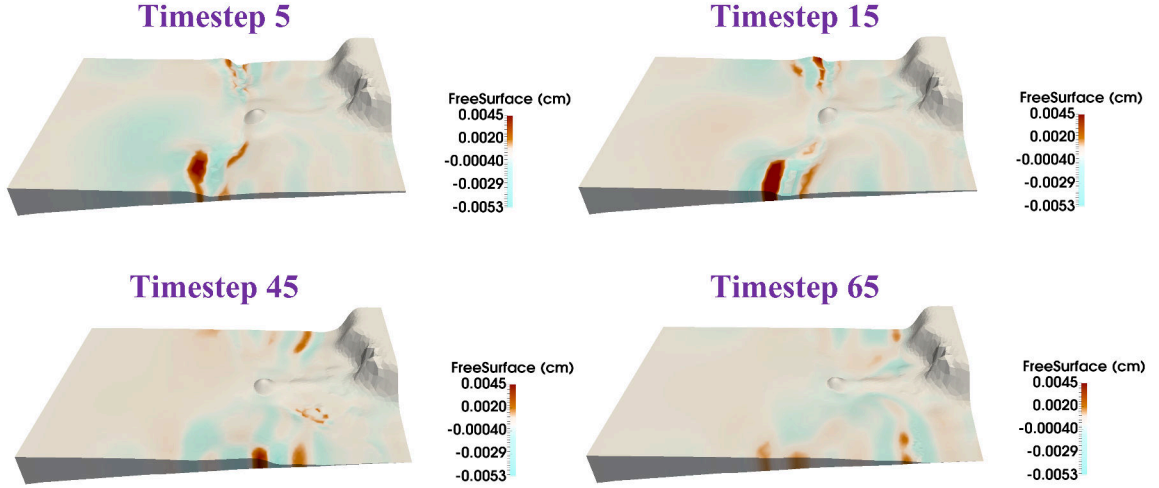


Figure 5.6: Differences of free surface heights between the full model and LSTM-ROM during the whole period.

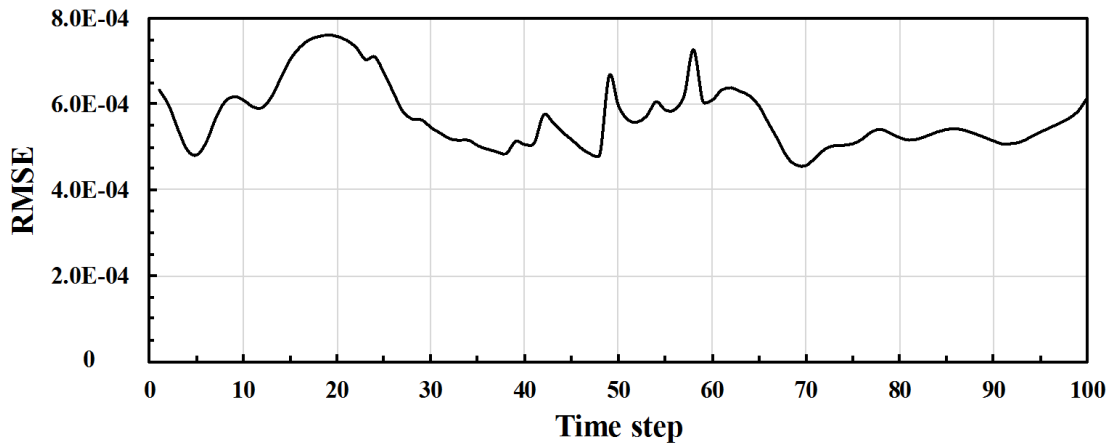


Figure 5.7: The RMSE errors of free surface heights between the original full model and LSTM-ROM.

simulations were performed on a computer with 24 cores (Intel Xeon(R) CPU@ 2.90GHz) and 16GB RAM. Only one of the cores was used when running the model. It is worth mentioning that the online CPU cost for running LSTM-ROM during one time-step is only 0.02 s, while the full model is 30.8 s. Although the offline CPU cost of training the LSTM network is somewhat expensive (331 s), it is much less than the simulation time of the full model (6160 s). Results obtained from this test case show that the novel LSTM-ROM can

provide reasonable accuracy in flood prediction whilst the CPU time is reduced by three orders of magnitude.

In summary, for any given input $x_s \in \mathcal{I} \setminus \mathcal{I}_{tr}$, the LSTM-ROM can provide a better prediction of floods at a future time $t \in [0, t_{N_t}]$. However, it is also noted that the predictive accuracy of LSTM-ROM decays when the predictive lead-time is extended to beyond the period $[0, t_{N_t}]$. The longer the predictive lead-time, the larger the RMSE of LSTM results (see Table 5.1, similarly in the work of Babovic et al. (2001) [215] and Sun et al. (2010) [216]). This indicates that capturing dependencies of features between successive time period in the model becomes more intractable as the prediction horizon extends in time. In this case, it is suggested that the use of data assimilation methods will help the improvement of model performance [215]. Using data assimilation techniques, only the reliable information from observations will be extracted to modify/correct the input datasets during the prediction process. It helps extending the prediction horizon and enables LSTM-ROM predictions in a real-time mode, which will be the focus of our future work.

Table 5.1: The RMSE errors of free surface heights between the original full model and LSTM-ROM during predicted period $[67.5, 70]$ s.

Time (s)	67.5	68	68.5	69	69.5	70
RMSE ($\times 10^{-3}$)	2.16	2.67	4.03	5.99	8.39	11.13

5.5.2 Prescriptive analytics for uncertainty quantification in flooding modelling

The uncertainty quantification is critical in flood prediction. Can one estimate the incoming wave through LSTM prescriptive analytics if the temporal measurements of free surface heights at specified locations? In this study, ten flood scenarios are set up with the induced waves shown in Fig. 5.3. The corresponding pseudo-data at Gauge 2 are shown in Fig. 5.8 which are obtained by running the full model. Here 12 key values are selected from each incoming wave (values marked with red circle in Fig. 5.8) as the input data for training, while choosing 3 key values from the corresponding wave time series as the output data for

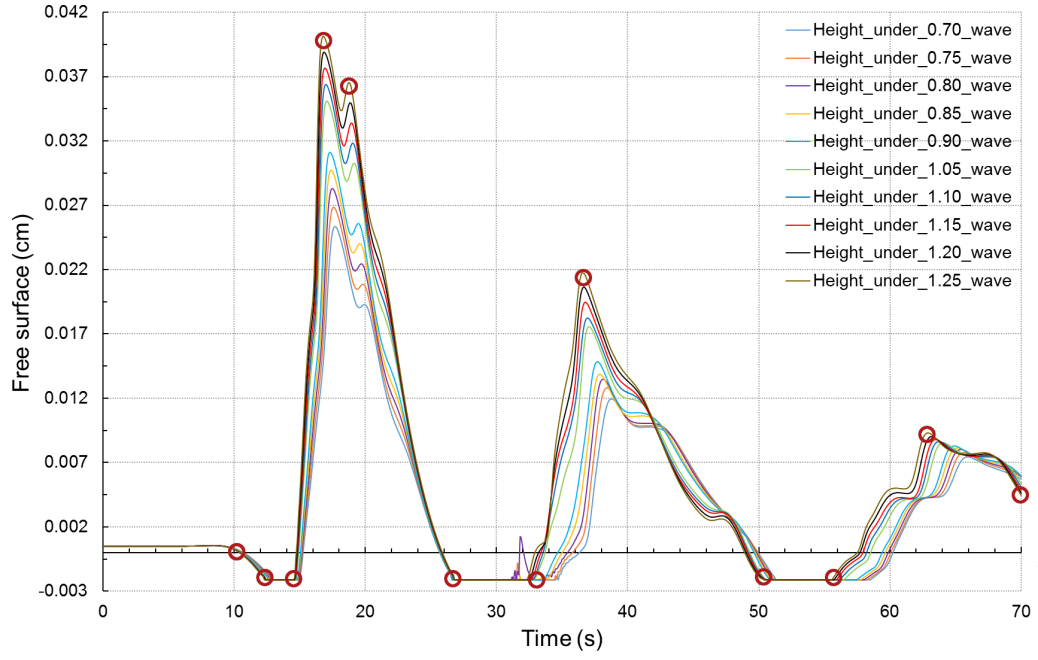


Figure 5.8: The free surface height at Gauge 2.

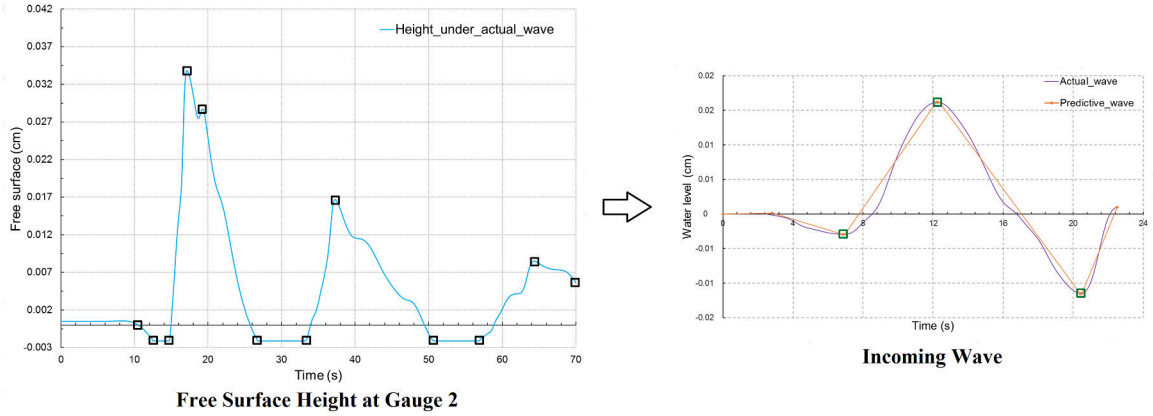


Figure 5.9: The input (left) and output (right) in prescriptive analytics.

training. The LSTM network would thus be constructed through the training process and used to estimate the flood induced condition (here, the incoming wave condition) for any given time series of free surface height at Gauge 2.

For a given measurement of free surface heights at Gauge 2 shown in Fig. 5.9 (left), the corresponding flood induced condition was estimated using the established LSTM network (shown in the right of Fig. 5.9). It is seen that the percent relative errors (%) between

the estimated and actual wave peak values (marked with green rectangles) are (0.01, -1.56), (0.12, 0.31), (0.05, -0.23), respectively. It is clear that in the prescriptive analytics, the corresponding flood induced condition can be accurately provided using the LSTM network.

5.6 Summary

In this work, a RNN architecture LSTM in combination with POD/ROM has been proposed for flood prediction, which simultaneously takes advantage of both LSTM and ROM. This is the first time that the ROM based LSTM network has been considered in flood forecasting and used to estimate spatial aggregation of inundations. This novel integrated LSTM-ROM framework which consists of both predictive and prescriptive analytics leads to a step change in the speed of forecasting, thus providing effective response management in emergencies.

The performance of the new method has been illustrated using Okushiri tsunami test cases. To estimate the accuracy of the LSTM-ROM, a comparison of results obtained from the LSTM-ROM and full model has been undertaken. It is shown that the accuracy of solutions from the LSTM-ROM is maintained while the CPU cost is reduced by three orders of magnitude. An error analysis has also been performed for the validation and accuracy of the LSTM-ROM through RMSE. Comparing LSTM-ROM results with those from the full model, the LSTM-ROM exhibits a good agreement with the full model. Subsequently in the prescriptive analytics, the corresponding flood induced conditions have been accurately provided using the LSTM network. Having the compatibility of LSTM and ROM, the integrated framework will enable predictive problems to be solved rapidly and accurately.

In addition, further evaluation of LSTM-ROMs in flood forecast will be carried out with different forecasting configurations including optimized training parameters, the predictive lead-time and the type of inputs. For example, the selection of training parameters in the networks (e.g. values for time delays and embedding dimension) is critical for LSTM

model predictive performance optimization. With optimal architectures, the underlying structure of nonlinear time series can be better viewed [216]. To further improve predictive accuracy, data assimilation methods can be adopted. The data assimilation techniques can improve the accuracy of model prediction by updating output variables based on the observed variables during a forecast period. For instance, given historical/simulated datasets of snapshots from floods, the target value (e.g. water depth) of the whole domain for the next few time steps in the future will be predicted. Usually, the performance of the model decays as the forecasting horizon extends beyond these few time steps, while data assimilation may be the most suitable approach to address this issue.

Chapter 6

An adaptive POD-LSTM predictive model using data-driven optimal sensing with data assimilation

6.1 Introduction

High fidelity numerical modelling plays an important role in providing accurate predictive information for decision makers/engineers, design and real-time operation, especially in industrial applications. However it often suffers from not only computationally intensive, but also significant uncertainties in the controlling parameters used as inputs when predicting the performance. Reduced order modelling (ROM) is a rapidly growing discipline, with significant potential advantages in predictive, real-time modelling: interactive use, emergency response, ensemble calculations, control, error norm quantification and in data assimilation. In ROM, the computational cost can be reduced by decreasing the dimensions of the control space thus ensuring that the minimisation of the cost function (or error covariances) is carried out within a low-dimensional space.

Chapter 5 demonstrates the integrated LSTM-ROM developed for both prescriptive and predictive analytics [217]. The LSTM-ROM has been applied to simulation of temporal-spatial distribution of floods. To reduce the dimensional size of large spatial datasets in LSTM, the proper orthogonal decomposition (POD) and singular value decomposition (SVD) approaches are introduced. The results showed that given a unseen input dataset the predictive accuracy of results using LSTM-ROM remains within the given simulation period. However the performance of the LSTM-ROM decays as the forecasting horizon extends beyond the given period. To tackle this challenging issue, data-driven optimal sensing

for reconstruction (DOSR) and data assimilation (DA) are here introduced to LSTM-ROM. This linkage between modelling and experimental data/observations allows us to minimize model errors and determine uncertainties, thus improving the accuracy of modelling.

In this work, a unique integrated DOSR-LSTM-DA framework is developed. It has the features 1) use of optimal sensing data where the sensor locations are optimized using POD; (2) POD modes are updated when the data is assimilated to modelling, which is named as adaptive POD method here. To the best of our knowledge, this is the first work to introduce both DA and optimal sensor techniques to LSTM-ROM.

The layout of the chapter is as follows. Details of constructing a LSTM-ROM predictive model are presented in Section 6.2. Section 6.3 illustrates how data driven optimal sensing and data assimilation are introduced to LSTM-ROMs to establish the new DOSR-LSTM-DA framework, which largely extends the predictive horizon and improves the predictive ability. Finally in Section 6.5, a summary is presented.

6.2 LSTM-ROM predictive modelling for parameterized PDEs

As a widely used data-driven dimensionality reduction technique, POD extracts dominant features from a data set over a full space and efficiently constructs a low-dimensional approximation representation. Here the POD approach is used for 1) reduction of large dataset; 2) construction of LSTM-ROM (Section 6.2); and 3) optimization of sensor locations (Section 6.3).

In this section, we provide the details of constructing a LSTM-ROM for representing the physical dynamics of the parameterized PDEs. In general, the parameterized PDEs at time level t_i can be written:

$$u(x, t_n, \mu) = F(u(x, t_{n-1}), x, t_n, \mu). \quad (6.1)$$

By using Eq. (5.1) and projecting Eq. (6.1) onto the POD sub-space, a set of ROMs

f_m can be constructed to represent the physical dynamics of the original PDEs over the reduced space::

$$a_k^n = f_k(a_1^{n-1}, a_2^{n-1}, \dots, a_K^{n-1}), \quad k \in (1, \dots, K). \quad (6.2)$$

The LSTM technique is here selected to construct the set of ROMs f_k since it explicitly considers the memory in a sequence without the vanishing gradient problem [218]. In the LSTM network, the input is the temporal coefficients $(a_1^{n-1}, \dots, a_K^{n-1})$ at the previous time level $n - 1$ while the output is the temporal coefficient a_m^n associated with the k^{th} POD basis function Φ_k ($k \in (1, \dots, K)$). The relationship function (f_m) between the input \mathbf{a}^{n-1} and output a_k^n can be obtained using the following equations:

$$\begin{aligned} I^i &= \varrho(W_{ia}\mathbf{a}^{n-1} + W_{ih}h^{n-1} + W_{iCe}Ce^{n-1} + b_i), \\ f^n &= \varrho(W_{fa}\mathbf{a}^{n-1} + W_{fh}h^{n-1} + W_{fCe}Ce^{n-1} + b_f), \\ o^n &= \varrho(W_{oa}\mathbf{a}^{n-1} + W_{oh}h^{n-1} + W_{oCe}Ce^n + b_o), \\ Ce^n &= r^n \odot Ce^{n-1} + i^n \odot Ce_i(W_{ceu}\mathbf{a}^{n-1} + W_{ceh}h^{n-1} + b_{Ce}), \\ h^n &= o^n \odot Ce_o(Ce^n), \\ a_k^n &= \zeta(W_{rh}h^n + b_r), \end{aligned} \quad (6.3)$$

where I , f and o denote the input, forget and output gate vectors respectively, Ce is the cell activation vector, b is the bias vector, ϱ is the activation function, W denotes the weight matrices (e.g. W_{ia} is the weight matrix from the input gate to the input), \odot is the element wise product of the vectors, Ce_o and Ce_i are the cell output and cell input activation functions respectively and ζ is the network output activation function.

Given the input-output training data set during the training period $[0, t_{N_t}]$, using Eq. (6.4), a set of LSTM-ROMs f_m for representing the low-dimensional physical dynamics is constructed during the offline training process. The LSTM-ROMs can then be used to

predict the temporal coefficients at time level $t_n \in [0, t_{N_t}]$ and beyond. The predicted variable solution will then be calculated using Eq. (5.1). However it is noted that the predictive accuracy of solutions is decayed beyond the training period. To tackle this issue, data assimilation techniques in conjunction with optimal sensing data are introduced to the LSTM-ROMs.

6.3 Data-driven optimal sensing data assimilation framework

Our newly developed predictive model has three characteristics that distinguish it significantly from our previous work [217]: (1) Optimal point sensor locations are selected specifically to reconstruct high-dimensional dynamic systems; (2) POD modes are updated according to the prediction accuracy; (3) Extension of predictive horizon is achieved in data assimilation procedures, where these optimal sensor locations provide observations. Fig. 6.1 demonstrates a schematic representation of our procedure.

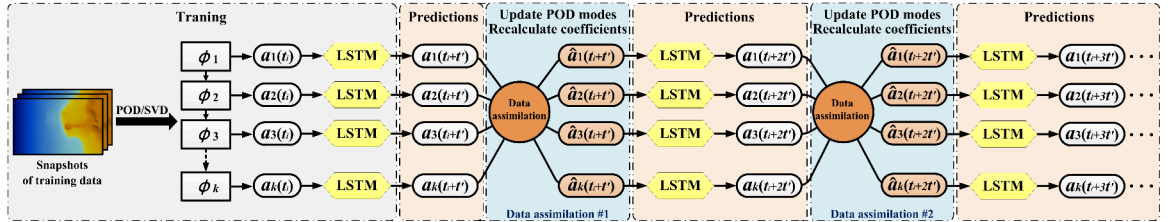


Figure 6.1: Training and execution modules for data-driven optimal sensing data assimilation framework DOSR-LSTM-DA.

Recall that the full order solution of state variable at time level n can be approximately computed:

$$\tilde{u}^n = \sum_{k=1}^K a_k^n \phi_k(x), \quad (6.4)$$

where $u^n \in \mathbb{R}^N$. The temporal coefficient a_k^n is obtained by the LSTM network in Eq.

(6.3). The absolute error of solutions in Eq. (6.4) can be expressed:

$$\xi = \| u^n - \tilde{u}^n \|_2^2, \quad (6.5)$$

Our aim to alleviate the error in Eq. (6.4) by assimilating optimal sensing data into the LSTM-ROM.

6.3.1 Data-driven optimal sensing for reconstruction (DOSR)

Theory of data-driven sparse sensing using POD

Inspired by recent advances in sparse sensor placement optimization techniques [219–221], we present a data-driven optimal sensing framework (DOSR) to reconstruct data from few point sensors (measurements). DOSR aims to find the locations of optimal sensors where the data is assimilated to the LSTM-ROM for improving the accuracy of solutions.

Let $\hat{U} = (\hat{u}^{p_1}, \dots, \hat{u}^{p_n}, \dots, \hat{u}^{p_t})^T \in \mathbb{R}^{p_t \times p}$ denote the sparse measurements, and $\hat{u}^{p_n} \in \mathbb{R}^p$ is a compressed version consisting of only the measured data from p selected sensors at time level p_n :

$$\hat{u}^{p_n} = C u^{p_n} \quad (6.6)$$

where

- $C \in \mathbb{R}^{p \times \mathbb{N}}$ is a masked matrix with entries of 1s and 0s to obtain a filtered data matrix

$\hat{U} \in \mathbb{R}^{p_t \times p}$, which is defined as follows:

$$\left\{ \begin{array}{l} C = (e_{x_1}, \dots, e_{x_j}, \dots, e_{x_p})^T, \in \mathbb{R}^{\mathbb{N}} \\ e_{x_j}^n = 1 \quad \text{at the element where the sensor is located at node } j, \\ e_{x_j}^n = 0 \quad \text{at the rest of elements,} \end{array} \right. \quad (6.7)$$

- $(p_1, \dots, p_n, \dots, p_t) \subset (1, \dots, n, \dots, N_t)$ denote the time levels when the measurements available;

- $(x_1, \dots, x_j, \dots, x_p) \subset (1, \dots, n, \dots, \mathbb{N})$ denote the sensor locations;
- p is the number of optimal sensors;
- p_t is the number of time levels when measurements are available.

Using POD, the sparse measurements at time level p_n can be written:

$$\hat{u}^{p_n} \approx e^{p_n} \sum_{k=1}^K \hat{a}_k^{p_n} \phi_k = \sum_{k=1}^K \hat{a}_k^{p_n} \hat{\phi}_k = \hat{\Phi}_K \hat{\mathbf{a}}^{p_n}, \quad (6.8)$$

where $\hat{a}_k^{p_n}$ ($\hat{\mathbf{a}}^{p_n} \in \mathbb{R}^{K \times \mathbb{N}}$) is the POD coefficient while the corresponding $\hat{\phi}_k$ ($\hat{\Phi}_K \in \mathbb{R}^{\mathbb{N} \times K}$) is masked POD modes.

The error in the reconstructed solutions can be written:

$$\hat{\xi} = \left\| \hat{u}^{p_n} - \sum_{k=1}^K \hat{a}_k^{p_n} \hat{\phi}_k \right\|_2^2. \quad (6.9)$$

By differentiating Eq. (6.9) with respect to the temporal coefficient $\hat{a}_k^{p_n}$, one can find how sensitive the error in solutions is to the change of temporal coefficients. For minimization of errors in Eq. (6.9), $\hat{\mathbf{a}}$ can be obtained by solving the least squares problem below:

$$\mathbf{M} \hat{\mathbf{a}} = \mathbf{f}, \quad (6.10)$$

where $\mathbf{M} = \hat{\Phi}_k^T \hat{\Phi}_k$ and $\mathbf{f} = \hat{\Phi}_k^T \hat{U}$.

Sensor placement optimization in DOSR

To maximally improve the accuracy of numerical solutions by DA, one has to consider the problem of where the best locations to put the sensors. For example, given p sensors and l possible locations, how to select the locations that will determine the POD coefficients most accurately? Mathematically, one way to solve this issue is to minimize the condition number of \mathbf{M} . Considering Eqs. (6.8) - (6.10), the optimal sensor location problem is

defined as:

$$\begin{aligned}
& \mathbf{min} \quad \kappa(\mathbf{M}) \\
& \mathbf{s.t.} \quad C_{i,j} \in 0, 1, \quad j = 1, \dots, l \\
& \quad \sum_{j=1}^l C_{i,j} = p,
\end{aligned}$$

where $\kappa(\mathbf{M})$ is the condition number of \mathbf{M} . By minimizing the condition number $\kappa(\mathbf{M})$ [222], this sensor location optimization problem is solved. According to [223], the locations of the POD spatial modes $\hat{\phi}_m$ that experience relatively large modal amplitudes (e.g. locations of maxima/minima values) can be used for sensor placement. Consequently, here we define the mean of the ensemble $\{\hat{\phi}_m\}_{m=1}^k$ as ϕ_{mean} ($\phi_{mean} = \frac{1}{k} \sum_{m=1}^k \hat{\phi}_m$) and choose locations where the absolute values are relatively large as sensor locations. These identified locations will be used for DA procedure demonstrated in the following section. To summarize, the DOSR method consists of following steps:

- (1) Select the training datasets of N snapshots and build data matrix U ;
- (2) Compute SVD of U to obtain the POD spacial modes Φ ;
- (3) Select the number of modes $k(k \ll n)$ to be employed in the reconstruction;
- (4) Design a masked matrix C to obtain a filtered data matrix \hat{U} ;
- (4) Construct the matrix $\mathbf{M} = \hat{\Phi}_k^T \hat{\Phi}_k$ and vector $\mathbf{f} = \hat{\Phi}_k^T \hat{U}$;
- (5) Solve the least square problem: $\mathbf{M}\hat{\mathbf{a}} = \mathbf{f}$ for new temporal coefficients $\hat{\mathbf{a}}$;
- (6) Reconstruct the solution field using $\hat{U} = \hat{\Phi}_k \hat{\mathbf{a}}$ with the new coefficients $\hat{\mathbf{a}}$ and determined number of modes k ;
- (7) Compare full data U with approximated solution \hat{U} in validation procedure to guarantee accuracy of $\hat{\mathbf{a}}$ and $\hat{\Phi}_k$;
- (8) Select p sensor locations according to the maxima/minima values from the POD modes $\hat{\Phi}_k$.

6.3.2 DOSR-based spatio-temporal prediction using LSTM and DA

Utilizing DOSR outlined above, the first goal of modelling the spatio-temporal dynamics without computing the governing equations is achieved, thus reducing computational and data handling costs. In this work, we further explore a DOSR-based approach combined with deep learning (LSTM) and data assimilation (DA) to predict dynamic features at future time instants.

Spatio-temporal prediction using LSTM

In LSTM modelling, the historical temporal coefficients $\hat{a}_m(t_i)$ of spatial modes $\hat{\phi}_m$ are used to train and test the LSTM model, which is then utilized to predict the temporal behaviour in the next few time instants. Subsequently, the snapshots during predicted period will be constructed using Eq. (6.8). Assume that $\hat{a}_m(t_i)$ are calculated from N snapshots during period $[t_1, t_N]$ with time step size Δt . Set the predicted time period $t' = [t_N, t_P]$. The DOSR-based LSTM approach is outlined below:

- (1) Split temporal coefficients $\hat{a}_m(t_i)$ from period $[t_1, t_v]$ and $[t_v, t_N]$ into training and test data respectively, namely that $\{\hat{a}_m(t)\}_{t=t_1}^{t_v}$ are training data, and $\{\hat{a}_m(t)\}_{t=t_v}^{t_N}$ are test data;
- (2) Train the LSTM model with $\{\hat{a}_m(t)\}_{t=t_1}^{t_v}$;
- (3) Use the trained LSTM model to predict $\{\hat{a}_m(t)\}_{t=t_v}^{t_N}$ and compare them with the true $\{\hat{a}_m(t)\}_{t=t_v}^{t_N}$ from the test data to guarantee the model performance;
- (4) Given the validated LSTM model from step (3), use the $\hat{a}_m(t_N)$ as input to predict $\hat{a}_m(t_N + \Delta t)$, which is the temporal coefficients for the next time step;
- (5) Compute the predicted solution field using $\hat{u} = \sum_{m=1}^k \hat{a}_m(t_N + \Delta t) \hat{\phi}_m(x)$, thus obtaining snapshot at time level $t_N + \Delta t$;
- (6) Use previous predicted $\hat{a}_m(t_N + \Delta t)$ as input to predict $\hat{a}_m(t_N + 2\Delta t)$ and repeat step (5) to obtain snapshot at time level $t_N + 2\Delta t$.
- (7) Repeat steps (4)-(5) to compute all predicted snapshots during $[t_N, t_P]$.

Extension of predictive horizon using data assimilation

Previously developed DOSR-based LSTM approach is employed to predict the spatio-temporal dynamics efficiently. However, the performance of the model decays as the predictive horizon extends beyond a few time steps, while DA method may be the most suitable approach to address this issue. It is found that the predicted results (15 steps) remain good even without retraining the LSTM after updating the POD, due to the reason that the variation of POD is small. Thus, the LSTM models have not been retrained. In this work, we integrate DOSR-based LSTM with DA approach to extend the predictive horizon, which helps to improve the predictive ability.

In most cases, the difference between the predicted coefficients $\{\hat{a}_m(t)\}_{t=t_N}^{t_P}$ and the true ones is increasing with time. Here, to illustrate the DA procedure, assume that the observed data at $t = t_P$ (true snapshot at $t = t_P$) is known and a relatively large difference exists between the predicted snapshot and the true one at time level t_P . There's thus a need to modify/correct the predicted snapshot at $t = t_P$ and update POD modes $\hat{\phi}_m(x)$ according to this corrected snapshot. Subsequently, $\hat{a}_m(t_P)$ are re-calculated, which is used as the input for prediction of next time step. This updating procedure enables a longer predictive horizon compared to the original t' mentioned above. This means that, the predictive horizon is able to be extended to d size of t' when d times of updating are preformed. In this work, we use the optimal interpolation (OI) algorithm of DA [224], which is operated by directly assimilating data into the predicted snapshot given the observed data. Given the observed data (known snapshot at $t = t_P$), the observations derive from the sensor locations identified using DOSR (Section 6.3) provide data to be assimilated in OI analysis. Finally, to demonstrate the overall integration of DOSR-based POD-LSTM-DA method, the outline of algorithmic procedure is shown in Fig. 6.2.

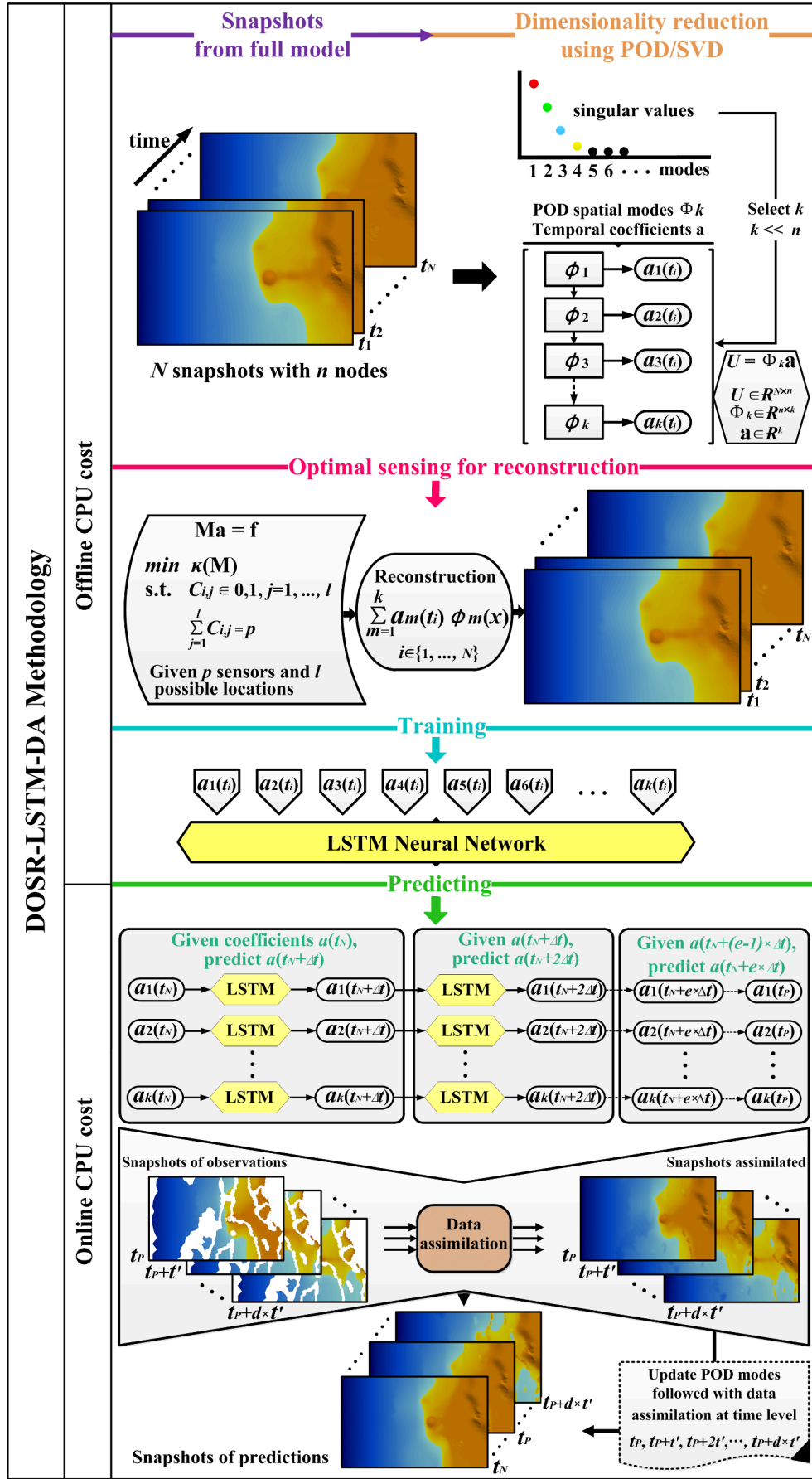


Figure 6.2: Framework of DOSR-LSTM-DA method.

6.4 Numerical examples

The example used for validation of the new DOSR-LSTM-DA method is the Okushiri tsunami test case. The event is the Okushiri tsunami in 1993 caused by the Hokkaido Nansei-Oki earthquake offshore of southwestern Hokkaido Island, Japan. The resulting tsunami hit a sparsely populated part of the Okushiri Island, Japan with a runup height of up to 30 m . To investigate the danger of such extreme events, the Research Institute for Electric Power Industry (CRIEPI) in Abiko, Japan constructed a scaled-down laboratory model of the area around the island [212]. The tsunami problem is simulated using the 3D unstructured mesh finite element fluid model (*Fluidity*) [225]. As shown in Fig. 6.3, the considered domain measures $5.448\text{ m} \times 3.402\text{ m}$ with a tsunami wave is enforced on the left boundary while the remaining boundaries were solid. The simulation period is $[0, 70]\text{ s}$, and a time step size is $\Delta t = 0.1\text{ s}$. The snapshots as training data were thus obtained by running the full model (*Fluidity*), with an unstructured mesh of 6894 nodes.

6.4.1 DOSR-based reconstruction

In this Hokkaido-Nansei-Oki tsunami example, here we select $N = 200$ snapshots during period $[10, 30]\text{ s}$ from the full model as the training dataset. Each snapshot is a spatial measurement of the system at a given time t_i ($i = 1, \dots, 200$), where $t_1 = 10\text{ s}$, $t_{200} = 30\text{ s}$. Using SVD technique, the POD modes of the dataset reflect its oscillatory dynamics. As shown in Fig. 6.4, the singular values decay rapidly and most of the spectral energy in the dataset is captured by the first 30 POD modes. Thus, we determine number of POD modes $k = 30$ for reconstruction of the system, which yields a dramatic reduction from the initial state dimension of $n = 6894$ spatial grid points. It should be noted that the solution field (water depth) reconstructed using Eq. (6.8) from the first 30 POD modes successfully captures nearly 100% of the system's energy, which is the normalized sum of the singular values.

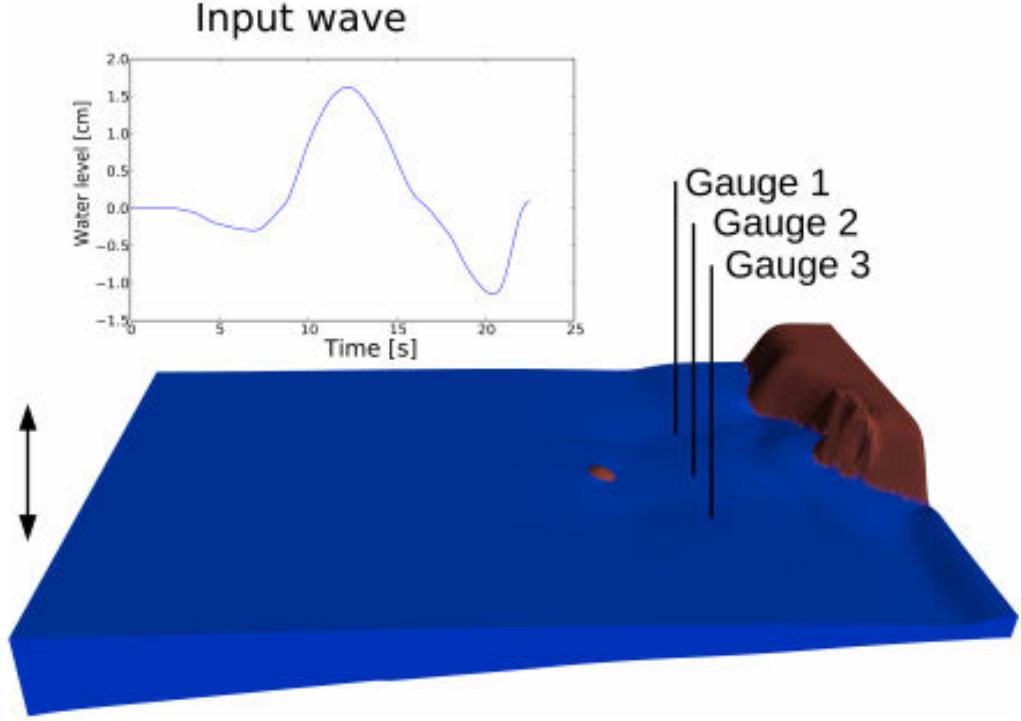


Figure 6.3: The bathymetry for the Hokkaido-Nansei-Oki tsunami example [213].

As mentioned in DOSR procedure (Section 6.3), the locations of the POD spatial modes that experience relatively large modal amplitudes (e.g. locations of maxima/minima values) should be used for sensor placement. Consequently, here we calculate the mean of the 30 modal ensemble $\{\phi_m\}_{m=1}^{30}$ as $\phi_{mean} = \frac{1}{30} \sum_{m=1}^{30} \hat{\phi}_m$ and choose the locations where the absolute values are relatively large as sensor locations. These identified locations will be used for data assimilation (DA) procedure demonstrated in the following section. Fig. 6.5 (b) demonstrates the sensor locations extracted from ϕ_{mean} (Fig. 6.5 (b)). Selecting the number of sensors p for Hokkaido-Nansei-Oki tsunami example illustrates the ability of optimized sensing to significantly reduce the number of sensors required for a given performance.

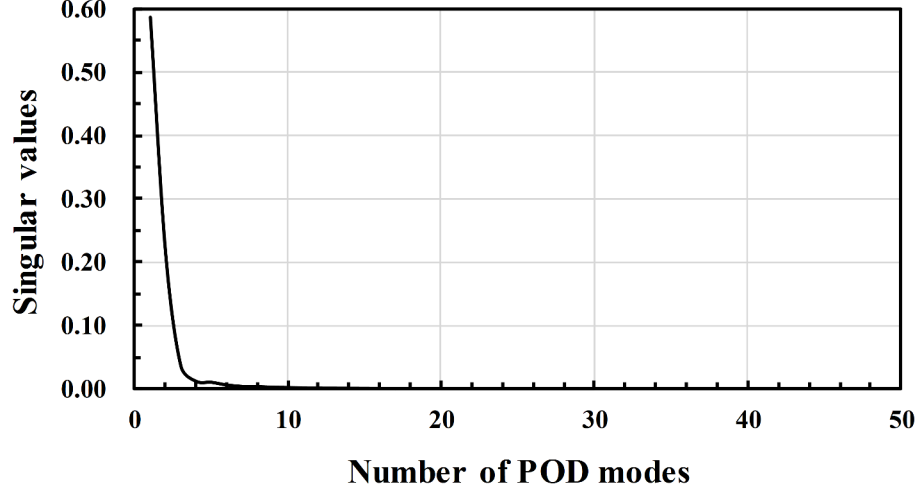


Figure 6.4: POD singular values of water depth to construct the reduced space.

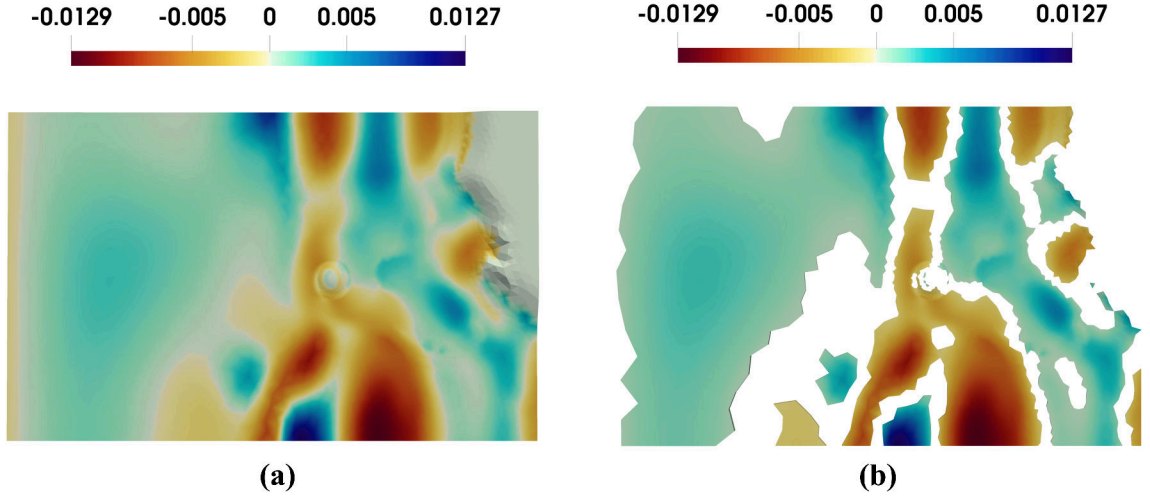


Figure 6.5: (a) ϕ_{mean} , mean of the 30 POD modal ensemble $\{\phi_m\}_{m=1}^{30}$; (b) Sensor locations extracted from ϕ_{mean} .

6.4.2 Spatio-temporal prediction using LSTM and DA

Following the reconstruction demonstrated in Section 6.4.1, LSTM modelling is performed given the calculated temporal coefficients. Here, the temporal coefficients $\{a_k(t)\}_{t=t_1}^{t_{200}}$ are used to train and test the LSTM model, which is then utilized to predict the temporal coefficients in the next few time instants $\{a_m(t)\}_{t=t_1}^{t_{200}+t'}$. Given the historical period $[t_1, t_{200}]$ ($t_1 = 10s$, $t_{200} = 30s$), here we set the predicted time period $t' = [t_{200}, t_{215}] =$

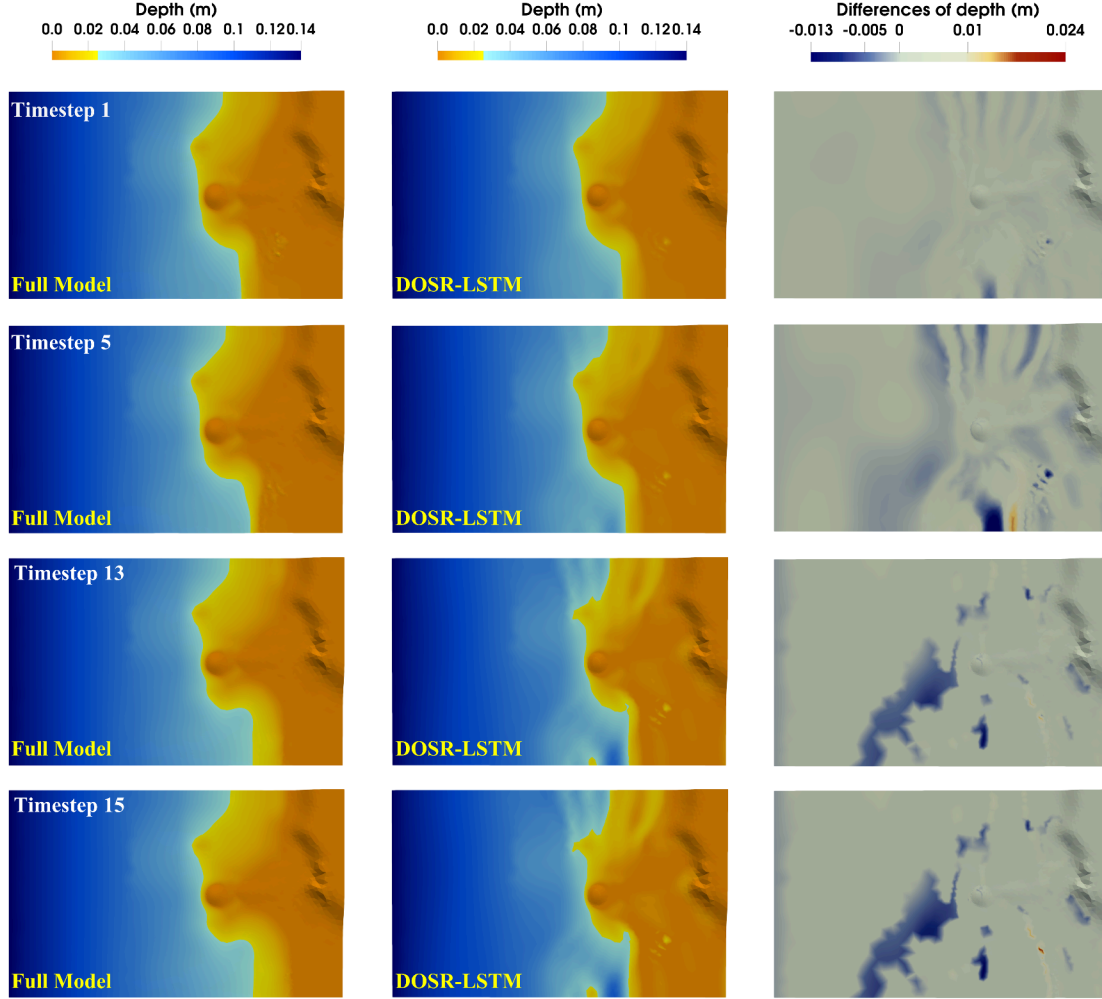


Figure 6.6: Comparison of the water depth from the full model with the DOSR-LSTM using 30 POD modes with a predictive horizon of 1, 5, 13 and 15 time steps.

$1.5s$ ($t_{200} = 30s$, $t_{215} = 31.5s$). In this case, the coefficients $\{a_m(t)\}_{t=t_{200}}^{t_{215}}$ are predicted using the trained and tested LSTM models. Subsequently, the 15 predicted snapshots during period $[t_{200}, t_{215}]$ ($[30, 31.5]s$) are constructed using Eq. (6.8). Fig. 6.6 provides a comparison of the water depth from the full model with the DOSR-LSTM using 30 POD modes with a predictive horizon of 15 time steps. It shows that the DOSR-LSTM solutions with 30 POD modes are in close agreement with the full state solutions while the predictive accuracy decays with the extending predictive lead-time. The longer the predictive lead-time, the larger difference exists between the results. To im-

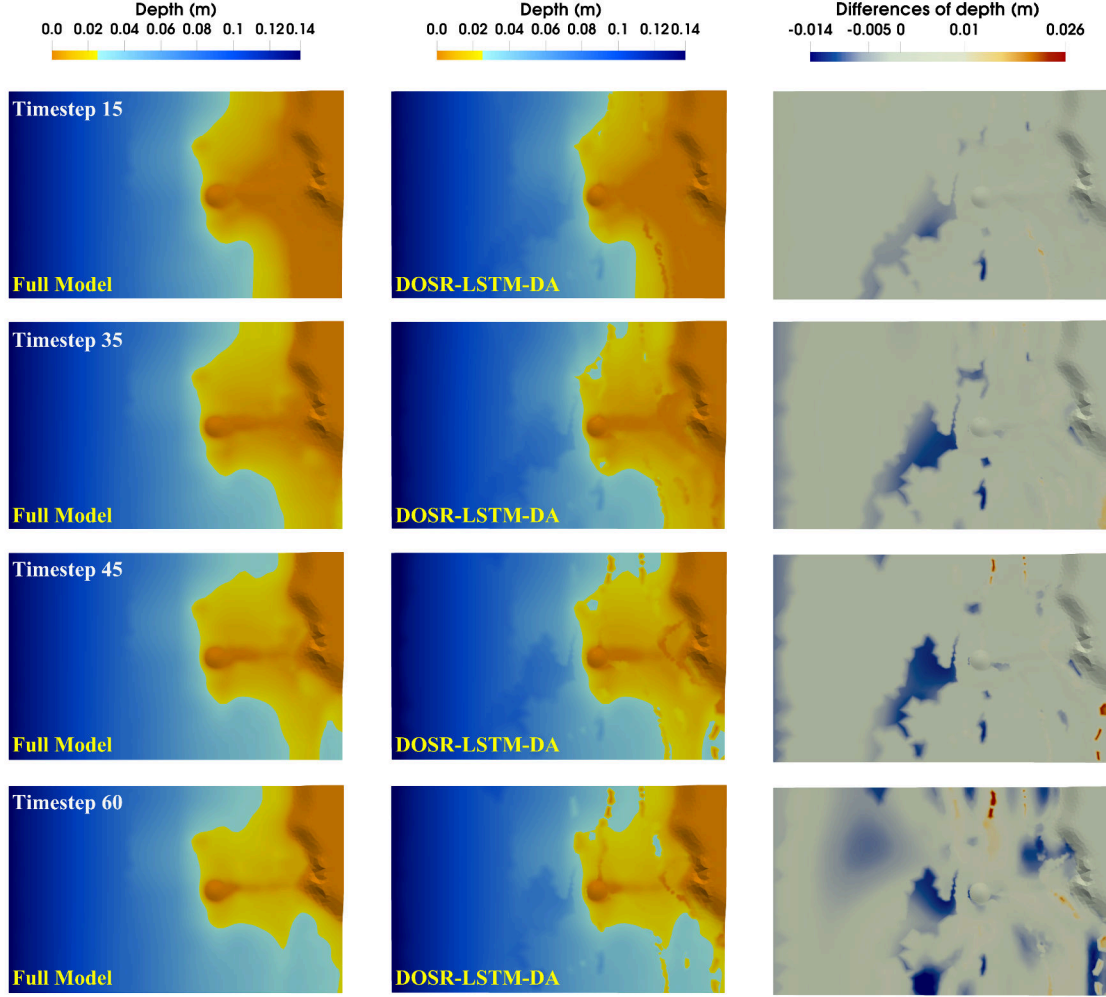


Figure 6.7: Comparison of water depth between the full model and the DOSR-LSTM-DA (left and middle column), as well as the differences (right column) after introducing DA method.

prove the predictive ability, we introduced the DA approach to extend the predictive horizon. Given the observed data from the sensor locations with a predictive horizon of 15, 30, 45 and 60 time steps, namely the known snapshot at time level $t = t_{215} = 31.5s$, $t = t_{230} = 33s$, $t = t_{245} = 34.5s$, and $t = t_{260} = 36s$, we use the optimal interpolation (OI) algorithm of DA [224] to directly assimilating data into the predicted snapshot at the same time levels. Following each assimilation procedure at time level $t = t_{215}, t_{230}, t_{245}$ and t_{260} , coefficients $a_m(t_{215})$, $a_m(t_{230})$, $a_m(t_{245})$, and $a_m(t_{260})$ are recalculated and used as the new input for prediction of next time steps. Simultaneously,

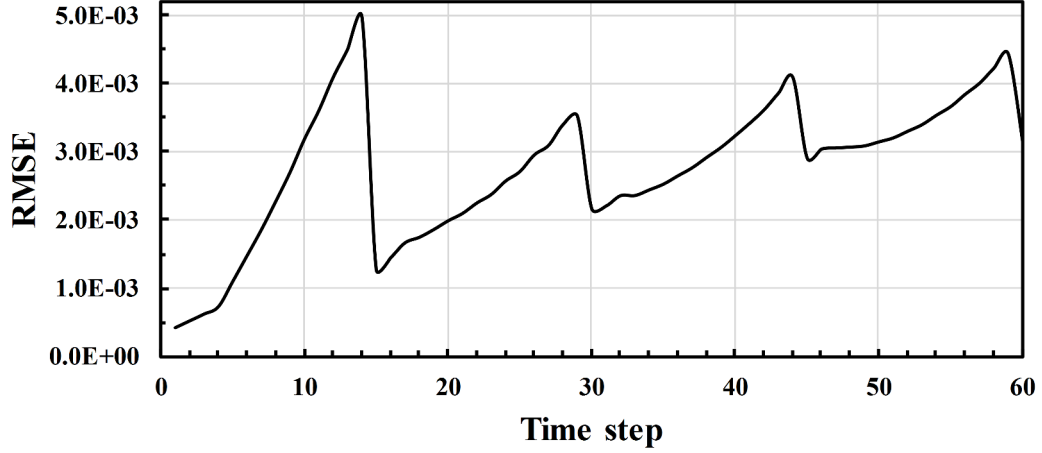


Figure 6.8: The RMSE of water depth between the DOSR-LSTM-DA and full model.

POD modes $\{\phi_k\}_{k=1}^{30}$ are updated and recalculated base on the the newly modified/corrected snapshots. This means that $d = 4$ times of updating procedure enables the predictive horizon extended to $d \times t' = 4 \times t' = 4 \times 1.5s = 6s$ (60 time steps), which is 4 times of the original predictive lead-time $t' = 1.5s$. Fig. 6.7 demonstrates the comparison between the DOSR-LSTM-DA and full modelling results. It is worth noting that results from the DOSR-LSTM-DA and full model are in very good agreement with each other. As shown in Fig. 6.7, the differences of the water depth between the DOSR-LSTM-DA and full model are quite small over the whole domain, which proves the predictive ability improvement of DOSR-LSTM-DA.

The error analysis of DOSR-LSTM-DA has been further carried out using the root-mean-square error (RMSE) of solutions (water depth) between the DOSR-LSTM-DA and full model. The RMSE is calculated for each time step by,

$$RMSE_j = \sqrt{\frac{\sum_{i=1}^n (S_{full}^{i,j} - S_{DOSR}^{i,j})^2}{n}}, \quad i = 1, 2, \dots, n, j = 1, 2, \dots, N_p, \quad (6.11)$$

where $n = 6894$ denotes the number of nodes on the full mesh; $N_p = 60$ is the number of time step; $S_{full}^{i,j}$ and $S_{DOSR}^{i,j}$ represent the solution on node i at time step j from the

original full model and DOSR-LSTM-DA, respectively. Obviously from Fig. 6.8, it can be seen that the errors gradually increase with the extending predictive horizon, demonstrating the decay of predictive accuracy during prediction. However, each assimilation produce at time level $t = t_{215}, t_{230}, t_{245}$ and t_{260} (a predictive horizon of 15, 30, 45, and 60 time steps) significantly reduces the prediction errors, which illustrates the ability of DOSR-LSTM-DA to significantly improve the model performance.

The DOSR-LSTM-DA algorithm includes both offline and online procedures. The offline CPU cost is defined as the time for pre-computing, that is, the calculation of snapshots and POD modes and LSTM training. The online CPU cost is the predictive time required using LSTM and DA. In this study, the simulations were performed on a computer with 24 cores (Intel Xeon(R) CPU@ 2.90GHz) and 16GB RAM. Only one of the cores was used when running the model. It is worth mentioning that the online CPU cost for running DOSR-LSTM-DA during one time-step is only 10.8 *s*, while the full model is 30.8 *s*. Although the offline CPU cost of training the LSTM network is somewhat expensive (213 *s*), it is much less than the simulation time of the full model (1848 *s*). Results obtained from this test case show that the novel DOSR-LSTM-DA approach can provide reasonable accuracy in flood prediction whilst the CPU time is largely reduced.

6.5 Summary

In this work, we addressed the challenging issue of how to extend the predictive horizon of LSTM-ROM [217] beyond the given period with a good predictive accuracy remained. To this end, we first employed the dimensionality reduction technique POD for reconstruction of high-dimensional dynamic systems. The dominant features from a data set over a full space is efficiently extracted and hence a handful of POD modes is sufficient to describe the essential dynamics. In particular, we selected optimal point sensor locations for the reconstruction (DOSR) based on POD modes. Subsequently, we explored the predictive ability by introducing deep learning (LSTM) and data assimilation (DA) to predict dynamic

features at future instants. During the DA procedure, POD modes are updated when the data is assimilated to modelling. This, in turn, takes the advantages of DA technique, with which the accuracy of predictions can be improved with the extending predictive lead-time. To the best of our knowledge, this is the first work to introduce both DA and optimal sensor techniques to LSTM-ROM.

To assess the performance of the integrated DOSR-LSTM-DA framework, we've applied the method to the Okushiri tsunami test case successfully. To estimate the accuracy of the DOSR-LSTM-DA, a comparison of results obtained from the DOSR-LSTM-DA and full model has been undertaken. It is found that the DOSR-LSTM solutions with 30 POD modes are in close agreement with the full state solutions while the predictive accuracy decays with the extending predictive lead-time. The longer the predictive lead-time, the larger difference exists between the results. It should be noting that after we introduced the DA approach, the prediction errors are significantly reduced at time levels when an assimilation procedure is conducted, which illustrates the ability of DOSR-LSTM-DA to significantly improve the model performance. By using DOSR-LSTM-DA, the predictive horizon can be extended by 3 times of the initial horizon. More importantly, the online CPU cost of using DOSR-LSTM-DA is only $1/3$ of the cost required by running the full model. We conclude that the method presented here improves the performance of LSTM-ROMs and satisfies the dual demands of required accuracy as well as low computational cost in predictions in long-term evolution of dynamical systems.

Chapter 7

Conclusions and recommendations

7.1 Conclusions

In this thesis, I have developed a 2D control-volume and finite-element (DCV-FEM) flood model using adaptive unstructured mesh technology. This model has been further developed by introducing (1) an anisotropic dynamic mesh optimization technique (anisotropic-DMO); (2) multiple flooding sources (extreme rainfall and sea-level events); and (3) a unique combination of anisotropic-DMO and high-resolution Digital Terrain Model (DTM) data. Additionally, an integrated long short-term memory (LSTM) and reduced order model (ROM) framework has been developed for rapid spatio-temporal flood prediction. Finally, a unique integrated DOSR-LSTM-DA framework has been developed by introducing both data-driven optimal sensor techniques (DOSR) and data assimilation (DA) to LSTM-ROM.

In Chapter 3, a 2D DCV-FEM flooding model with the adaptive unstructured mesh technique has been developed and applied to a flooding event that happened in 2002 in Glasgow, Scotland, United Kingdom, where the flood is induced by a stream flow from a culvert at the northeast corner of the domain. A comparison between 2D adaptive and fixed mesh models as well as 3D model has been undertaken. It has been found that using the 2D adaptive mesh model, it is able to provide accurate results while the computational cost is reduced by 20 – 84% in comparison to 2D fixed mesh models. Another advantage of 2D adaptive unstructured mesh modelling is that urban topography can be accurately represented when/where needed by increasing the mesh resolution (around the buildings, for example) dynamically when the flooding water spreads over the urban area. This is the first time to use the dynamically adaptive mesh technique in flooding modelling and assess

its performance in a relatively simple flooding event. In the following subsection, more work on flood modelling development will focus on complex realistic cases where flooding may occur from more than one source.

Realising the importance of flood coincidence risk assessments, in Chapter 4, we have further developed the adaptive unstructured mesh flooding model Floodity for the joint urban flood events caused by multiple sources (extreme rainfall and sea-level events) and successfully applied to Greve in Denmark. By introducing the anisotropic-DMO technique, the features of flooding flows (local flows around the buildings or the wetting and drying front, for example) are able to be better captured while reducing computational cost without sacrificing accuracy of flooding simulations. With a unique combination of anisotropic-DMO and high-resolution Digital Terrain Model (DTM) data, the complex urban topography can be accurately represented when/where needed by increasing the mesh resolution (around the buildings, for example) dynamically when the flooding water spreads over the urban area. This new Floodity model has been applied to several flooding scenarios that happened in Greve, Denmark, where the flood is induced by different combinations of extreme incoming sea levels and rainfall. A comparison between Floodity and MIKE 21 FM results has been undertaken. It has been found that Floodity is able to provide relatively accurate results while the computational cost is reduced by 20 - 88% in comparison to fixed mesh models. To assess uncertainties in model predictions, the sensitivity of flood volumes to extreme sea levels and rainfalls has been explored. In joint flood events, we found that the flood volume over the inundated area is more sensitive to sea levels than rainfall. Extreme sea-level events with the higher peak water levels induce higher peak flood volume while the impact of rainfall is relatively small. The sensitivity of flood results to the mesh resolution is also investigated. In flood modelling, the blocking effect of buildings on the peak time of flood volumes can be seen only when using high resolution meshes and Digital Terrain Model data.

In Chapter 5, a RNN architecture LSTM in combination with POD/ROM has been

proposed for flood prediction, which simultaneously takes advantage of both LSTM and ROM. This is the first time that the ROM based LSTM network has been considered in flood forecasting and used to estimate spatial aggregation of inundations. This novel integrated LSTM-ROM framework which consists of both predictive and prescriptive analytics leads to a step change in the speed of forecasting, thus providing effective response management in emergencies. The performance of the new method has been illustrated using Okushiri tsunami test cases. To estimate the accuracy of the LSTM-ROM, a comparison of results obtained from the LSTM-ROM and full model has been undertaken. It is shown that the accuracy of solutions from the LSTM-ROM is maintained while the CPU cost is reduced by three orders of magnitude. An error analysis has also been performed for the validation and accuracy of the LSTM-ROM through RMSE. Comparing LSTM-ROM results with those from the full model, the LSTM-ROM exhibits a good agreement with the full model. Subsequently in the prescriptive analytics, the corresponding flood induced conditions have been accurately provided using the LSTM network. Having the compatibility of LSTM and ROM, the integrated framework will enable predictive problems to be solved rapidly and accurately.

In Chapter 6, we addressed the challenging issue of how to extend the predictive horizon of LSTM-ROM [217] beyond the given period with a good predictive accuracy remained. To this end, we first employed the dimensionality reduction technique POD for reconstruction of high-dimensional dynamic systems. The dominant features from a data set over a full space is efficiently extracted and hence a handful of POD modes is sufficient to describe the essential dynamics. In particular, we selected optimal point sensor locations for the reconstruction (DOSR) based on POD modes. Subsequently, we explored the predictive ability by introducing deep learning (LSTM) and data assimilation (DA) to predict dynamic features at future instants. During the DA procedure, POD modes are updated when the data is assimilated to modelling. This, in turn, takes the advantages of DA technique, with which the accuracy of predictions can be improved with the extending predictive lead-time.

To assess the performance of the integrated DOSR-LSTM-DA framework, we've applied the method to the Okushiri tsunami test case successfully. To estimate the accuracy of the DOSR-LSTM-DA, a comparison of results obtained from the DOSR-LSTM-DA and full model has been undertaken. It is found that the DOSR-LSTM solutions with 30 POD modes are in close agreement with the full state solutions while the predictive accuracy decays with the extending predictive lead-time. The longer the predictive lead-time, the larger difference exists between the results. It should be noting that after we introduced the DA approach, the prediction errors are significantly reduced at time levels when an assimilation procedure is conducted, which illustrates the ability of DOSR-LSTM-DA to significantly improve the model performance. By using DOSR-LSTM-DA, the predictive horizon can be extended by 3 times of the initial horizon. More importantly, the online CPU cost of using DOSR-LSTM-DA is only $1/3$ of the cost required by running the full model. We conclude that the method presented here improves the performance of LSTM-ROMs and satisfies the dual demands of required accuracy as well as low computational cost in predictions in long-term evolution of dynamic systems.

7.2 Recommendations

7.2.1 Parallelism of anisotropic-DMO

In this thesis, the newly developed 2D DCV-FEM flooding model with the anisotropic-DMO technique has been validated by two test cases. However, the relatively high CPU time required is an issue which should be further improved by using parallel computing.

Highly accurate terrain data in the form of Digital Terrain Models (DTMs) are often needed in 2D flood inundation modelling. For example, a typical DTM data at $1m$ horizontal resolution contains detailed terrain features such as individual buildings, streets or open channels. Researchers have to simulate flooding at this scale [159]. Unfortunately, the computational cost of simulations at $1m$ horizontal resolution is very high. For exam-

ple, when a $1m$ mesh resolution corresponds to 10^6 elements per km^2 and a time step of $0.1s$ or less is required, this means that nearly 10^{12} operations per km^2 is needed and the computation becomes even more complex when the study area is $100km^2$ or more. The running time required for simulations may far exceed the period of the flood events. Although the anisotropic-DMO technique performs well in meshing and largely reduces the meshes involved in computation at each time step, we expect our model generate results in a matter of minutes. This requires parallel computing or parallelism to address the running time challenge.

There are two fundamental types of parallel computing architectures [159]:

(1) *Shared memory systems*: where calculations are performed in parallel using workstations with multiple processor cores on a single motherboard or with numerous graphical processing units (GPUs). Implementation of shared memory systems is usually aided by the OpenMP directives [226]. Moreover, the recent advances in GPUs provide a promising cost efficient way for implementing over a hundred processors in a shared memory environment [227]. However, this approach is presently limited a specialized coding language and the maximum amount of memory available on a single motherboard is also limited.

(2) *Distributed memory systems*: where multiple servers are networked to enable fast communication between servers or nodes. Distributed memory systems are limited by the number of available nodes and the efficiency of communication between nodes, but the potential memory limitations are avoided because each additional node includes additional memory capacity. In the future work, we can focus on explicit, distributed-memory parallelism where the task decomposition, mapping of tasks to processors and the communication structure are determined by the programmer [228]. Implementation of distributed memory parallelism is aided by the Message Passing Interface (MPI), with which a set of processes of data exchange is executed. Each process refers to a task assigned to a processor, namely one process is assigned per processor, resulting in a significant reduction in running time compared to sequential computing.

7.2.2 Applicability and robustness of ROM

In this PhD thesis, the reduced order modelling techniques are applied for flood modelling. However, this does not imply that the techniques developed in this thesis are restricted to the domain of flood modelling only. The LSTM-ROM techniques can be applied to reduce models in other application domains such as aerodynamics, ocean modelling, air pollution, and large-scale mechanical applications. In this thesis we've developed non-intrusive reduced order models (NIROMs) where inputs (e.g. input wave conditions) vary in time. To generate a robust ROM for different parameters/inputs to represent the physical dynamics of PDEs with the varying model parameters, in the future, it will be beneficial to have a more precise detail about the validity range of the physical parameters (e.g. precipitations, weather, viscosity, manning coefficients) and their individual or combined contribution to the stability and validity of the ROM. For example, it will be important to know whether the ROM will still be a valid model if the parameters such as manning coefficients or viscosity have specified minimum and maximum values. Since the change of spatial distribution is dependent on changes of the physical parameters, in nonlinear PDEs, the POD basis functions may not be suitable to approximate the new spatial distribution any more. If this limitation can be addressed by determining the variations of physical parameter, it will thus help engineers to decide whether a new ROM has to be built or whether the existing models are already adequate. Moreover, since non-intrusive reduced order modelling based on snapshot method has been applied here, the effectiveness of POD-ROM depends on the accuracy and smoothness of the data collected from the measurements or simulations (snapshots). The POD-ROM approach may thus not work well if the studied problem exhibits discontinuous dynamics, since it is difficult to obtain a POD basis numerically from the data with discontinuous dynamics [96]. To tackle this issue, an intrusive ROM approach is suggested since the complex fluid flow dynamics is considered in the physical PDEs. Further work will be the development of an intrusive ROM in combination with machine learning techniques.

7.2.3 Combination of ROM with LSTM Autoencoders

In recent years, the abundance of data have facilitated the construction of data-driven models of systems lacking high-fidelity governing laws. Data-driven methods have become an increasingly popular approach to tackle the challenge of data abundance. These methods provide solutions which are learned from measurements or numerical solutions [30, 229, 230]. Reduced order models (ROMs) have thus been constructed to capture the large-scale system behaviour while retaining physical fidelity. More recently, machine learning strategies have been applied to aid in modelling low-dimensional dynamics of complex systems, such as identification for truncated generalized POD coordinates [202, 231, 232]. The normal approaches for the treatment of nonlinearities include ROMs based on discrete empirical interpolation (DEIM) [233], or Gauss-Newton with approximated tensors (GNAT) [195]. Gonzalez and Balajewicz (2018) [234] have proposed a modular model consisting of a deep convolutional autoencoder and a modified LSTM network to model the evolution of low-dimensional data representations while avoiding costly state reconstructions at every step. The model performance have been validated via three examples for fluid systems with large parameter-variations and the stability in long-term prediction have also been proved. In another paper, Nitish Srivastava et al. (2015) [235] describe the LSTM Autoencoder as an extension or application of the Encoder-Decoder LSTM. Their model uses an encoder LSTM to map an input sequence into a fixed length representation. This representation is decoded using single or multiple decoder LSTMs to perform different tasks, such as reconstructing the input sequence, or predicting the future sequence. The best performing model was the Composite Model that combined an autoencoder and a future predictor. The model was able to persistently generate motion well beyond the time scales it was trained for.

In this work, the ROM in combination with LSTM is constructed based on POD, where the physical variables are represented by a linear combination of POD modes. Despite the successes of LSTM-ROM developed here, there exists issues of instability for complex nonlinear flow simulations, particularly when dealing with high-Reynolds number

fluid flows. POD-ROMs are biased towards large energy-producing scales and are not endowed with the small energy-dissipating scales that maybe dynamically significant [234]. To address these issues, in future work, instead of POD, autoencoder techniques can be introduced to construct ROMs for nonlinear fluid flow problems. We can develop a new LSTM Autoencoders ROM model using a completely data-driven approach to identify and evolve a low-dimensional representation of a spatio-temporal system. More efficient filtering strategies would be included in decoder designs and the design of the LSTM network should be changeable according to spatial parameter variations. LSTM Autoencoders are used to learn the dynamics of low-dimensional representation on its underlying nonlinear manifold, offering specific advantages over POD-ROMs.

Bibliography

- [1] K. Smith, *Environmental hazards: assessing risk and reducing disaster*. Routledge, 2003.
- [2] J. Ernst, B. J. Dewals, S. Detrembleur, P. Archambeau, S. Erpicum, and M. Pirotton, “Micro-scale flood risk analysis based on detailed 2D hydraulic modelling and high resolution geographic data,” *Natural Hazards*, vol. 55, no. 2, pp. 181–209, 2010.
- [3] P. Sleigh, P. Gaskell, M. Berzins, and N. Wright, “An unstructured finite-volume algorithm for predicting flow in rivers and estuaries,” *Computers & Fluids*, vol. 27, no. 4, pp. 479–508, 1998.
- [4] B. Rogers, M. Fujihara, and A. G. Borthwick, “Adaptive Q-tree godunov-type scheme for shallow water equations,” *International Journal for Numerical Methods in Fluids*, vol. 35, no. 3, pp. 247–280, 2001.
- [5] Q. Liang, A. Borthwick, and G. Stelling, “Simulation of dam-and dyke-break hydrodynamics on dynamically adaptive quadtree grids,” *International journal for numerical methods in fluids*, vol. 46, no. 2, pp. 127–162, 2004.
- [6] T. Krámer and J. Józsa, “Solution-adaptivity in modelling complex shallow flows,” *Computers & Fluids*, vol. 36, no. 3, pp. 562–577, 2007.
- [7] Q. Liang, J. Zang, A. G. Borthwick, and P. H. Taylor, “Shallow flow simulation on dynamically adaptive cut cell quadtree grids,” *International Journal for Numerical Methods in Fluids*, vol. 53, no. 12, pp. 1777–1799, 2007.
- [8] Z. Skoula, A. Borthwick, and C. Moutzouris, “Godunov-type solution of the shallow water equations on adaptive unstructured triangular grids,” *International Journal of Computational Fluid Dynamics*, vol. 20, no. 9, pp. 621–636, 2006.
- [9] S. A. Ivanenko and G. V. Muratova, “Adaptive grid shallow water modeling,” *Applied Numerical Mathematics*, vol. 32, no. 4, pp. 447–482, 2000.
- [10] M. E. Hubbard and N. Dodd, “A 2D numerical model of wave run-up and overtopping,” *Coastal Engineering*, vol. 47, no. 1, pp. 1–26, 2002.
- [11] P. Lamby, S. Müller, and Y. Stiriba, “Solution of shallow water equations using fully adaptive multiscale schemes,” *International journal for numerical methods in fluids*, vol. 49, no. 4, pp. 417–437, 2005.

- [12] J.-F. Remacle, S. S. Frazao, X. Li, and M. S. Shephard, “An adaptive discretization of shallow-water equations based on discontinuous Galerkin methods,” *International journal for numerical methods in fluids*, vol. 52, no. 8, pp. 903–923, 2006.
- [13] G. M. Porta, S. Perotto, and F. Ballio, “Anisotropic mesh adaptation driven by a recovery-based error estimator for shallow water flow modeling,” *International Journal for Numerical Methods in Fluids*, vol. 70, no. 3, pp. 269–299, 2012.
- [14] J. Dietrich, R. Kolar, and K. Dresback, “Mass residuals as a criterion for mesh refinement in continuous Galerkin shallow water models,” *Journal of Hydraulic Engineering*, vol. 134, no. 5, pp. 520–532, 2008.
- [15] Q. Liang and A. G. Borthwick, “Adaptive quadtree simulation of shallow flows with wet–dry fronts over complex topography,” *Computers & Fluids*, vol. 38, no. 2, pp. 221–234, 2009.
- [16] G. C. Buscaglia and E. A. Dari, “Anisotropic mesh optimization and its application in adaptivity,” *International Journal for Numerical Methods in Engineering*, vol. 40, no. 22, pp. 4119–4136, 1997.
- [17] M. Castro-Díaz, F. Hecht, B. Mohammadi, and O Pironneau, “Anisotropic unstructured mesh adaption for flow simulations,” *International Journal for Numerical Methods in Fluids*, vol. 25, no. 4, pp. 475–491, 1997.
- [18] W. G. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, and M.-G. Vallet, “Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. part I: general principles,” *International Journal for Numerical Methods in Fluids*, vol. 32, no. 6, pp. 725–744, 2000.
- [19] C. Pain, A. Umpheby, C. De Oliveira, and A. Goddard, “Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 29-30, pp. 3771–3796, 2001.
- [20] P Salinas, D Pavlidis, Z Xie, C. Pain, M. Jackson, *et al.*, “A double control volume finite element method with dynamic unstructured mesh optimization,” in *SPE Reservoir Simulation Conference*, Society of Petroleum Engineers, 2017.
- [21] B. P. Leonard, “Simple high-accuracy resolution program for convective modelling of discontinuities,” *International Journal for Numerical Methods in Fluids*, vol. 8, no. 10, pp. 1291–1318, 1988.
- [22] D. Pavlidis, J. L. Gomes, Z. Xie, J. R. Percival, C. C. Pain, and O. K. Matar, “Compressive advection and multi-component methods for interface-capturing,” *Inter-*

national Journal for Numerical Methods in Fluids, vol. 80, no. 4, pp. 256–282, 2016.

- [23] J. L. Lumley, “The structure of inhomogeneous turbulent flows,” *Atmospheric turbulence and radio wave propagation*, 1967.
- [24] N. Aubry, P. Holmes, J. L. Lumley, and E. Stone, “The dynamics of coherent structures in the wall region of a turbulent boundary layer,” *Journal of Fluid Mechanics*, vol. 192, pp. 115–173, 1988.
- [25] L Ukeiley, L Cordier, R Manceau, J Delville, M Glauser, and J. Bonnet, “Examination of large-scale structures in a turbulent plane mixing layer. part 2. Dynamical systems model,” *Journal of Fluid Mechanics*, vol. 441, pp. 67–108, 2001.
- [26] B. R. Noack, K. Afanasiev, M. MORZYŃSKI, G. Tadmor, and F. Thiele, “A hierarchy of low-dimensional models for the transient and post-transient cylinder wake,” *Journal of Fluid Mechanics*, vol. 497, pp. 335–363, 2003.
- [27] B. R. Noack, P. Papas, and P. A. Monkewitz, “The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows,” *Journal of Fluid Mechanics*, vol. 523, pp. 339–365, 2005.
- [28] C. W. Rowley, T. Colonius, and R. M. Murray, “Model reduction for compressible flows using POD and Galerkin projection,” *Physica D: Nonlinear Phenomena*, vol. 189, no. 1-2, pp. 115–129, 2004.
- [29] A. Moosavi, R. Stefanescu, and A. Sandu, “Efficient construction of local parametric reduced order models using machine learning techniques,” *arXiv preprint arXiv:1511.02909*, 2015.
- [30] O. San and R. Maulik, “Neural network closures for nonlinear model order reduction,” *Advances in Computational Mathematics*, vol. 44, no. 6, pp. 1717–1750, 2018.
- [31] P. Mostaghimi, J. R. Percival, D. Pavlidis, R. J. Ferrier, J. L. Gomes, G. J. Gorman, M. D. Jackson, S. J. Neethling, and C. C. Pain, “Anisotropic mesh adaptivity and control volume finite element methods for numerical simulation of multiphase flow in porous media,” *Mathematical Geosciences*, vol. 47, no. 4, pp. 417–440, 2015.
- [32] Y. V. Vasilevskii and K. Lipnikov, “An adaptive algorithm for quasioptimal mesh generation,” *Computational mathematics and mathematical physics*, vol. 39, no. 9, pp. 1468–1486, 1999.
- [33] M. J. Berger and P. Colella, “Local adaptive mesh refinement for shock hydrodynamics,” *Journal of computational Physics*, vol. 82, no. 1, pp. 64–84, 1989.

- [34] C. J. Budd, W. Huang, and R. D. Russell, “Adaptivity with moving grids,” *Acta Numerica*, vol. 18, pp. 111–241, 2009.
- [35] I. Babuška and M. Suri, “The p and h-p versions of the finite element method, basic principles and properties,” *SIAM review*, vol. 36, no. 4, pp. 578–632, 1994.
- [36] A. Ern, S. Piperno, and K. Djadel, “A well-balanced Runge–Kutta discontinuous Galerkin method for the shallow-water equations with flooding and drying,” *International journal for numerical methods in fluids*, vol. 58, no. 1, pp. 1–25, 2008.
- [37] P. Tassi, O. Bokhove, and C. Vionnet, “Space discontinuous Galerkin method for shallow water flowskinetic and HLLC flux, and potential vorticity generation,” *Advances in water resources*, vol. 30, no. 4, pp. 998–1015, 2007.
- [38] M. Piggott, P. Farrell, C. Wilson, G. Gorman, and C. Pain, “Anisotropic mesh adaptivity for multi-scale ocean modelling,” *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 367, no. 1907, pp. 4591–4611, 2009.
- [39] R. B. Simpson, “Anisotropic mesh transformations and optimal error control,” *Applied Numerical Mathematics*, vol. 14, no. 1-3, pp. 183–198, 1994.
- [40] M. Piggott, C. Pain, G. Gorman, P. Power, and A. Goddard, “H, r, and hr adaptivity with applications in numerical ocean modelling,” *Ocean modelling*, vol. 10, no. 1-2, pp. 95–113, 2005.
- [41] P. Power, M. Piggott, F Fang, G. Gorman, C. Pain, D. Marshall, A. Goddard, and I. Navon, “Adjoint goal-based error norms for adaptive mesh ocean modelling,” *Ocean modelling*, vol. 15, no. 1-2, pp. 3–38, 2006.
- [42] P. Farrell and J. Maddison, “Conservative interpolation between volume meshes by local Galerkin projection,” *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 1-4, pp. 89–100, 2011.
- [43] H. Hiester, M. Piggott, and P. Allison, “The impact of mesh adaptivity on the gravity current front speed in a two-dimensional lock-exchange,” *Ocean Modelling*, vol. 38, no. 1, pp. 1–21, 2011.
- [44] D. R. Davies, C. R. Wilson, and S. C. Kramer, “Fluidity: a fully unstructured anisotropic adaptive mesh computational modeling framework for geodynamics,” *Geochemistry, Geophysics, Geosystems*, vol. 12, no. 6, 2011.
- [45] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, *et al.*, “Deep learning and process understanding for data-driven earth system science,” *Nature*, vol. 566, no. 7743, p. 195, 2019.

- [46] M. H. Hassoun *et al.*, *Fundamentals of artificial neural networks*. MIT press, 1995.
- [47] K. Khan and A. Sahai, “A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context,” *International Journal of Intelligent Systems and Applications*, vol. 4, no. 7, p. 23, 2012.
- [48] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, p. 436, 2015.
- [49] A Krizhevsky, I Sutskever, and G Hinton, “//Proc. Advances in Neural Inform.,” *Proces. Systems. 2012. V. 25.*, vol. 25, p. 1090, 2012.
- [50] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, “Learning hierarchical features for scene labeling,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2012.
- [51] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, “Joint training of a convolutional network and a graphical model for human pose estimation,” in *Advances in neural information processing systems*, 2014, pp. 1799–1807.
- [52] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Černocký, “Strategies for training large scale neural network language models,” in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, IEEE, 2011, pp. 196–201.
- [53] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, *et al.*, “Deep neural networks for acoustic modeling in speech recognition,” *IEEE Signal processing magazine*, vol. 29, 2012.
- [54] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, “Deep convolutional neural networks for LVCSR,” in *2013 IEEE international conference on acoustics, speech and signal processing*, IEEE, 2013, pp. 8614–8618.
- [55] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” *Journal of machine learning research*, vol. 12, no. Aug, pp. 2493–2537, 2011.
- [56] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in neural information processing systems*, 2014, pp. 3104–3112.
- [57] T Ciodaro, D Deva, J. De Seixas, and D Damazio, “Online particle detection with neural networks based on topological calorimetry information,” in *Journal of physics: conference series*, IOP Publishing, vol. 368, 2012, p. 012 030.

- [58] M. Helmstaedter, K. L. Briggman, S. C. Turaga, V. Jain, H. S. Seung, and W. Denk, "Connectomic reconstruction of the inner plexiform layer in the mouse retina," *Nature*, vol. 500, no. 7461, p. 168, 2013.
- [59] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.
- [60] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.
- [61] R. J. Williams and J. Peng, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural computation*, vol. 2, no. 4, pp. 490–501, 1990.
- [62] J. Schmidhuber, "Dynamische neuronale netze und das fundamentale raumzeitliche lernproblem," PhD thesis, Technische Universität München, 1990.
- [63] H. T. Siegelmann and E. D. Sontag, "Turing computability with neural nets," *Applied Mathematics Letters*, vol. 4, no. 6, pp. 77–80, 1991.
- [64] R. Balzer, "A 15 year perspective on automatic programming," *IEEE Transactions on Software Engineering*, no. 11, pp. 1257–1268, 1985.
- [65] Y. Deville and K.-K. Lau, "Logic program synthesis," *The Journal of Logic Programming*, vol. 19, pp. 321–350, 1994.
- [66] E. Soloway, "Learning to program= learning to construct mechanisms and explanations," *Communications of the ACM*, vol. 29, no. 9, pp. 850–858, 1986.
- [67] M. P. Mignolet, A. Przekop, S. A. Rizzi, and S. M. Spottswood, "A review of indirect/non-intrusive reduced order modeling of nonlinear geometric structures," *Journal of Sound and Vibration*, vol. 332, no. 10, pp. 2437–2460, 2013.
- [68] D Xiao, F Fang, A. Buchan, C. Pain, I. Navon, and A Muggeridge, "Non-intrusive reduced order modelling of the Navier–Stokes equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 293, pp. 522–541, 2015.
- [69] R. Noori, A. Karbassi, K. Ashrafi, M. Ardestani, and N. Mehrdadi, "Development and application of reduced-order neural network model based on proper orthogonal decomposition for BOD 5 monitoring: Active and online prediction," *Environmental progress & sustainable energy*, vol. 32, no. 1, pp. 120–127, 2013.
- [70] M. Guénot, I. Lepot, C. Sainvitu, J. Goblet, and R. Filomeno Coelho, "Adaptive sampling strategies for non-intrusive POD-based surrogates," *Engineering computations*, vol. 30, no. 4, pp. 521–547, 2013.

- [71] M Winter and C Breitsamter, *Reduced-order modeling of unsteady aerodynamic loads using radial basis function neural networks*. Deutsche Gesellschaft für Luft- und Raumfahrt-Lilienthal-Oberth eV, 2014.
- [72] F. L. Pena, V. D. Casás, A Gosset, and R. Duro, “A surrogate method based on the enhancement of low fidelity computational fluid dynamics approximations by artificial neural networks,” *Computers & Fluids*, vol. 58, pp. 112–119, 2012.
- [73] A. Gholami, H. Bonakdari, A. H. Zaji, and A. A. Akhtari, “Simulation of open channel bend characteristics using computational fluid dynamics and artificial neural networks,” *Engineering Applications of Computational Fluid Mechanics*, vol. 9, no. 1, pp. 355–369, 2015.
- [74] S. Baghalian, H. Bonakdari, F. Nazari, and M. Fazli, “Closed-form solution for flow field in curved channels in comparison with experimental and numerical analyses and artificial neural network,” *Engineering Applications of Computational Fluid Mechanics*, vol. 6, no. 4, pp. 514–526, 2012.
- [75] M. De Giorgi, D Bello, and A Ficarella, “An artificial neural network approach to investigate cavitating flow regime at different temperatures,” *Measurement*, vol. 47, pp. 971–981, 2014.
- [76] K. Elsayed and C. Lacor, “CFD modeling and multi-objective optimization of cyclone geometry using desirability function, artificial neural networks and genetic algorithms,” *Applied Mathematical Modelling*, vol. 37, no. 8, pp. 5680–5704, 2013.
- [77] P. Feldmann and R. W. Freund, “Efficient linear circuit analysis by padé approximation via the Lanczos process,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 5, pp. 639–649, 1995.
- [78] K Gallivan, E Grimme, and P. Van Dooren, “Padé approximation of large-scale dynamic systems with Lanczos methods,” in *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, IEEE, vol. 1, 1994, pp. 443–448.
- [79] E. Grimme, “Krylov projection methods for model reduction,” PhD thesis, 1997.
- [80] A. K. Noor and J. M. Peters, “Reduced basis technique for nonlinear analysis of structures,” *Aiaa journal*, vol. 18, no. 4, pp. 455–462, 1980.
- [81] J. S. Peterson, “The reduced basis method for incompressible viscous flow calculations,” *SIAM Journal on Scientific and Statistical Computing*, vol. 10, no. 4, pp. 777–786, 1989.
- [82] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press, 2012.

- [83] L. Sirovich, “Turbulence and the dynamics of coherent structures. I. Coherent structures,” *Quarterly of applied mathematics*, vol. 45, no. 3, pp. 561–571, 1987.
- [84] Y. Cao, J. Zhu, Z. Luo, and I. M. Navon, “Reduced-order modeling of the upper tropical pacific ocean model using proper orthogonal decomposition,” *Computers & Mathematics with Applications*, vol. 52, no. 8-9, pp. 1373–1386, 2006.
- [85] F Fang, C. Pain, I. Navon, M. Piggott, G. Gorman, P. Farrell, P. Allison, and A. Goddard, “A POD reduced-order 4D-Var adaptive mesh ocean modelling approach,” *International Journal for Numerical Methods in Fluids*, vol. 60, no. 7, pp. 709–732, 2009.
- [86] F Fang, T Zhang, D Pavlidis, C Pain, A. G. Buchan, and I. M. Navon, “Reduced order modelling of an unstructured mesh air pollution model and application in 2D/3D urban street canyons,” *Atmospheric Environment*, vol. 96, pp. 96–106, 2014.
- [87] S. Ahuja and C. W. Rowley, “Feedback control of unstable steady states of flow past a flat plate using reduced-order estimators,” *Journal of fluid mechanics*, vol. 645, pp. 447–478, 2010.
- [88] I. Akhtar, “Parallel simulations, reduced-order modeling, and feedback control of vortex shedding using fluidic actuators,” PhD thesis, Virginia Tech, 2008.
- [89] S. Bagheri, L. Brandt, and D. S. Henningson, “Input–output analysis, model reduction and control of the flat-plate boundary layer,” *Journal of Fluid Mechanics*, vol. 620, pp. 263–298, 2009.
- [90] A. Barbagallo, D. Sipp, and P. J. Schmid, “Closed-loop control of an open cavity flow using reduced-order models,” *Journal of Fluid Mechanics*, vol. 641, pp. 1–50, 2009.
- [91] M. Bergmann, L. Cordier, and J.-P. Brancher, “Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model,” *Physics of fluids*, vol. 17, no. 9, p. 097 101, 2005.
- [92] D. Daescu and I. Navon, “A dual-weighted approach to order reduction in 4DVAR data assimilation,” *Monthly Weather Review*, vol. 136, no. 3, pp. 1026–1041, 2008.
- [93] G. Dimitriu, N. Apreutesei, and R. Ștefănescu, “Numerical simulations with data assimilation using an adaptive POD procedure,” in *International Conference on Large-Scale Scientific Computing*, Springer, 2009, pp. 165–172.
- [94] J. Du, I. Navon, J. Zhu, F. Fang, and A. Alekseev, “Reduced order modeling based on POD of a parabolized Navier–Stokes equations model II: Trust region POD 4D

- VAR data assimilation,” *Computers & Mathematics with Applications*, vol. 65, no. 3, pp. 380–394, 2013.
- [95] R. Ștefănescu, A. Sandu, and I. M. Navon, “POD/DEIM reduced-order strategies for efficient four dimensional variational data assimilation,” *Journal of Computational Physics*, vol. 295, pp. 569–595, 2015.
 - [96] P. Astrid, “Reduction of process simulation models: A proper orthogonal decomposition approach,” 2002.
 - [97] D. Ryckelynck, F. Chinesta, E. Cueto, and A. Ammar, “On the a priori model reduction: Overview and recent developments,” *Archives of Computational methods in Engineering*, vol. 13, no. 1, pp. 91–128, 2006.
 - [98] F. C. B. Mascarenhas, K. Toda, M. G. Miguez, K. Inoue, *et al.*, *Flood risk simulation*. WIT, 2005.
 - [99] A. Semadeni-Davies, C. Hernebring, G. Svensson, and L.-G. Gustafsson, “The impacts of climate change and urbanisation on drainage in Helsingborg, Sweden: Sub-urban stormwater,” *Journal of hydrology*, vol. 350, no. 1-2, pp. 114–125, 2008.
 - [100] T. G. Schmitt, M. Thomas, and N. Ettrich, “Analysis and modeling of flooding in urban drainage systems,” *Journal of hydrology*, vol. 299, no. 3-4, pp. 300–311, 2004.
 - [101] E. van Dijk, J. van der Meulen, J. Kluck, and J. Straatman, “Comparing modelling techniques for analysing urban pluvial flooding,” *Water science and technology*, vol. 69, no. 2, pp. 305–311, 2013.
 - [102] R. Borsche and A. Klar, “Flooding in urban drainage systems: coupling hyperbolic conservation laws for sewer systems and surface flow,” *International Journal for Numerical Methods in Fluids*, vol. 76, no. 11, pp. 789–810, 2014.
 - [103] A.-L. Son, B. Kim, and K.-Y. Han, “A simple and robust method for simultaneous consideration of overland and underground space in urban flood modeling,” *Water*, vol. 8, no. 11, p. 494, 2016.
 - [104] J. N. Hartnack, H. G. Enggrob, and M. Rungø, *2D overland flow modelling using fine scale DEM with manageable runtimes*. Taylor & Francis Group, London, 2009.
 - [105] F. Dottori and E. Todini, “Developments of a flood inundation model based on the cellular automata approach: testing different methods to improve model performance,” *Physics and Chemistry of the Earth, Parts A/B/C*, vol. 36, no. 7-8, pp. 266–280, 2011.

- [106] A. S. Chen, B. Evans, S. Djordjević, and D. A. Savić, “Multi-layered coarse grid modelling in 2D urban flood simulations,” *Journal of Hydrology*, vol. 470, pp. 1–11, 2012.
- [107] J Leandro, A. Chen, and A Schumann, “A 2D parallel diffusive wave model for floodplain inundation with variable time step (P-DWave),” *Journal of Hydrology*, vol. 517, pp. 250–259, 2014.
- [108] L. S. Smith, Q. Liang, and P. F. Quinn, “Towards a hydrodynamic modelling framework appropriate for applications in urban flood assessment and mitigation using heterogeneous computing,” *Urban Water Journal*, vol. 12, no. 1, pp. 67–78, 2015.
- [109] A. S. Chen, S. Djordjevic, J. Leandro, and D. Savic, “The urban inundation model with bidirectional flow interaction between 2D overland surface and 1D sewer networks,” *NOVATECH 2007*, 2007.
- [110] F. Zhou, G. Chen, Y. Huang, J. Z. Yang, and H. Feng, “An adaptive moving finite volume scheme for modeling flood inundation over dry and complex topography,” *Water Resources Research*, vol. 49, no. 4, pp. 1914–1928, 2013.
- [111] M. Berger, “Adaptive mesh refinement for hyperbolic partial differential equations[ph. d. thesis],” 1982.
- [112] D. George, “Adaptive finite volume methods with well-balanced Riemann solvers for modeling floods in rugged terrain: Application to the Malpasset dam-break flood (france, 1959),” *International Journal for Numerical Methods in Fluids*, vol. 66, no. 8, pp. 1000–1018, 2011.
- [113] Q. Liang, “A simplified adaptive Cartesian grid system for solving the 2D shallow water equations,” *International Journal for Numerical Methods in Fluids*, vol. 69, no. 2, pp. 442–458, 2012.
- [114] J.-P. Wang and Q Liang, “Testing a new adaptive grid-based shallow flow model for different types of flood simulations,” *Journal of Flood Risk Management*, vol. 4, no. 2, pp. 96–103, 2011.
- [115] S. Popinet, “Adaptive modelling of long-distance wave propagation and fine-scale flooding during the Tohoku tsunami,” 2012.
- [116] W. Huang, Z. Cao, G. Pender, Q. Liu, and P. Carling, “Coupled flood and sediment transport modelling with adaptive mesh refinement,” *Science China Technological Sciences*, vol. 58, no. 8, pp. 1425–1438, 2015.

- [117] J. Du, J. Zhu, F. Fang, C. Pain, and I. Navon, “Ensemble data assimilation applied to an adaptive mesh ocean model,” *International Journal for Numerical Methods in Fluids*, vol. 82, no. 12, pp. 997–1009, 2016.
- [118] J. Zheng, J. Zhu, Z. Wang, F. Fang, C. Pain, and J. Xiang, “Towards a new multiscale air quality transport model using the fully unstructured anisotropic adaptive mesh technology of Fluidity (version 4.1. 9),” *Geoscientific Model Development*, vol. 8, no. 10, p. 3421, 2015.
- [119] K. Su, J.-P. Latham, D. Pavlidis, J. Xiang, F. Fang, P. Mostaghimi, J. R. Percival, C. C. Pain, and M. D. Jackson, “Multiphase flow simulation through porous media with explicitly resolved fractures,” *Geofluids*, vol. 15, no. 4, pp. 592–607, 2015.
- [120] Z. Xie, G. F. Hewitt, D. Pavlidis, P. Salinas, C. C. Pain, and O. K. Matar, “Numerical study of three-dimensional droplet impact on a flowing liquid film in annular two-phase flow,” *Chemical Engineering Science*, vol. 166, pp. 303–312, 2017.
- [121] M. Jackson, J. Percival, P. Mostaghimi, B. Tollit, D. Pavlidis, C. Pain, J. Gomes, A. H. Elsheikh, P. Salinas, A. Muggeridge, *et al.*, “Reservoir modeling for flow simulation by use of surfaces, adaptive unstructured meshes, and an overlapping-control-volume finite-element method,” *SPE Reservoir Evaluation & Engineering*, vol. 18, no. 02, pp. 115–132, 2015.
- [122] S. C. Medeiros and S. C. Hagen, “Review of wetting and drying algorithms for numerical tidal flow models,” *International Journal for Numerical Methods in Fluids*, vol. 71, no. 4, pp. 473–487, 2013.
- [123] P. Salinas, D. Pavlidis, Z. Xie, C. Jacquemyn, Y. Melnikova, M. D. Jackson, and C. C. Pain, “Improving the robustness of the control volume finite element method with application to multiphase porous media flow,” *International Journal for Numerical Methods in Fluids*, vol. 85, no. 4, pp. 235–246, 2017.
- [124] P. A. Forsyth *et al.*, “A control volume finite element method for local mesh refinement,” in *SPE Symposium on Reservoir Simulation*, Society of Petroleum Engineers, 1989.
- [125] S. Geiger, S. Roberts, S. Matthäi, C. Zoppou, and A. Burri, “Combining finite element and finite volume methods for efficient multiphase flow simulations in highly heterogeneous and structurally complex geologic media,” *Geofluids*, vol. 4, no. 4, pp. 284–299, 2004.
- [126] J. Gomes, D. Pavlidis, P. Salinas, Z. Xie, J. R. Percival, Y. Melnikova, C. C. Pain, and M. D. Jackson, “A force-balanced control volume finite element method for multi-phase porous media flow modelling,” *International Journal for Numerical Methods in Fluids*, vol. 83, no. 5, pp. 431–445, 2017.

- [127] S. K. Matthai, A. A. Mezentsev, M. Belayneh, *et al.*, “Finite element-node-centered finite-volume two-phase-flow experiments with fractured rock represented by unstructured hybrid-element meshes,” *SPE Reservoir Evaluation & Engineering*, vol. 10, no. 06, pp. 740–756, 2007.
- [128] R. Muñoz-Carpena, C. T. Miller, and J. E. Parsons, “A quadratic petrov-galerkin solution for kinematic wave overland flow,” *Water Resources Research*, vol. 29, no. 8, pp. 2615–2627, 1993.
- [129] R. Wait and M. Parsaei, “Petrov-galerkin methods for two dimensional flow calculations,” in *Finite Elements in Water Resources*, Springer, 1982, pp. 885–893.
- [130] S. R. Merton, “Discontinuous galerkin methods for computational radiation transport,” 2012.
- [131] G. H. Keulegan, *Laws of turbulent flow in open channels*. National Bureau of Standards US, 1938, vol. 21.
- [132] H. Rouse, *Fluid mechanics for hydraulic engineers*. New York: Dover Publications, Inc., 1961, p. 246.
- [133] V. Bellos, “Ways for flood hazard mapping in urbanised environments: a short literature review,” *Water Utility Journal*, vol. 4, pp. 25–31, 2012.
- [134] P. Gresho and R. Sani, “Incompressible Flow and the Finite Element Method, vol 1: Advection-Diffusion and Isothermal Laminar Flow,” *Journal of Fluid Mechanics*, vol. 411, no. 1, pp. 378–381, 2000.
- [135] N. M. Hunter, P. D. Bates, S. Neelz, G. Pender, I. Villanueva, N. G. Wright, D. Liang, R. A. Falconer, B. Lin, S. Waller, *et al.*, “Benchmarking 2D hydraulic models for urban flood simulations,” in *Proceedings of the institution of civil engineers: water management*, Thomas Telford (ICE publishing), vol. 161, 2008, pp. 13–30.
- [136] D. Liang, R. A. Falconer, and B. Lin, “Coupling surface and subsurface flows in a depth averaged flood wave model,” *Journal of Hydrology*, vol. 337, no. 1-2, pp. 147–158, 2007.
- [137] C. Geuzaine and J.-F. Remacle, “Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities,” *International journal for numerical methods in engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.
- [138] T. Zhang, P. Feng, Č. Maksimović, and P. D. Bates, “Application of a three-dimensional unstructured-mesh finite-element flooding model and comparison with two-dimensional approaches,” *Water resources management*, vol. 30, no. 2, pp. 823–841, 2016.

- [139] R. J. Dawson, L Speight, J. W. Hall, S Djordjevic, D Savic, and J Leandro, "Attribution of flood risk in urban areas," *Journal of Hydroinformatics*, vol. 10, no. 4, pp. 275–288, 2008.
- [140] J. J. Lian, K Xu, and C Ma, "Joint impact of rainfall and tidal level on flood risk in a coastal city with a complex river network: a case study of Fuzhou City, China," *Hydrology and Earth System Sciences*, vol. 17, no. 2, p. 679, 2013.
- [141] J. Teng, A. Jakeman, J. Vaze, B. F. Croke, D. Dutta, and S Kim, "Flood inundation modelling: a review of methods, recent advances and uncertainty analysis," *Environmental Modelling & Software*, vol. 90, pp. 201–216, 2017.
- [142] S. Soares-Frazão, J. Lhomme, V. Guinot, and Y. Zech, "Two-dimensional shallow-water model with porosity for urban flood modelling," *Journal of Hydraulic Research*, vol. 46, no. 1, pp. 45–64, 2008.
- [143] E. van Dijk, J. van der Meulen, J. Kluck, and J. H. M. Straatman, "Comparing modelling techniques for analysing urban pluvial flooding," *Water science and technology*, vol. 69, no. 2, pp. 305–311, 2014.
- [144] J. E. O'Connor and J. E. Costa, *The world's largest floods, past and present: their causes and magnitudes*. US Geological Survey, 2004, vol. 1254.
- [145] L. C. Smith, "Satellite remote sensing of river inundation area, stage, and discharge: a review," *Hydrological processes*, vol. 11, no. 10, pp. 1427–1439, 1997.
- [146] J Teng, J Vaze, D Dutta, and S Marvanek, "Rapid inundation modelling in large floodplains using LiDAR DEM," *Water Resources Management*, vol. 29, no. 8, pp. 2619–2636, 2015.
- [147] H. Apel, A. H. Thieken, B. Merz, and G. Blöschl, "Flood risk assessment and associated uncertainty," *Natural Hazards and Earth System Science*, vol. 4, no. 2, pp. 295–308, 2004.
- [148] W. C. Huber, R. E. Dickinson, and T. O. Barnwell, *Storm water management model, version 4: user's manual*. Environmental Research Laboratory, Office of Research and Development, US Environmental Protection Agency, 1988.
- [149] DHI, "MIKE 21 FLOW MODEL – Hydrodynamic Module, Scientific Documentation," *MIKE BY DHI Software*, Hørsholm, Denmark, 2009.
- [150] E Mignot, A Paquier, and S Haider, "Modeling floods in a dense urban area using 2D shallow water equations," *Journal of Hydrology*, vol. 327, no. 1, pp. 186–199, 2006.

- [151] X. Xia, Q. Liang, X. Ming, and J. Hou, “An efficient and stable hydrodynamic model with novel source term discretization schemes for overland flow and flood simulations,” *Water Resources Research*, vol. 53, no. 5, pp. 3730–3759, 2017.
- [152] M. Prakash, K. Rothauge, and P. W. Cleary, “Modelling the impact of dam failure scenarios on flood inundation using SPH,” *Applied Mathematical Modelling*, vol. 38, no. 23, pp. 5515–5534, 2014.
- [153] DHI, “MIKE Flood – 1D-2D Modelling, User Manual,” *MIKE BY DHI Software, Hørsholm, Denmark*, 2014.
- [154] N. D. Sto. Domingo, B. Paludan, F. Hansen, H. Madsen, M. Sunyer, and O. Mark, “Modeling of Sea Level Rise and Subsequent Urban Flooding Due To Climate Changes,” in *Conference: SimHydro: Hydraulic Modeling and Uncertainty*, 2010.
- [155] V. Bellos and G. Tsakiris, “A hybrid method for flood simulation in small catchments combining hydrodynamic and hydrological techniques,” *Journal of Hydrology*, vol. 540, pp. 331–339, 2016.
- [156] P. Nguyen, A. Thorstensen, S. Sorooshian, K. Hsu, A. AghaKouchak, B. Sanders, V. Koren, Z. Cui, and M. Smith, “A high resolution coupled hydrologic–hydraulic model (HiResFlood-UCI) for flash flood modeling,” *Journal of Hydrology*, vol. 541, pp. 401–420, 2016.
- [157] J. Singh, M. S. Altinakar, and Y. Ding, “Numerical modeling of rainfall-generated overland flow using nonlinear shallow-water equations,” *Journal of Hydrologic Engineering*, vol. 20, no. 8, p. 04014089, 2014.
- [158] X. Hu and L. Song, “Hydrodynamic modeling of flash flood in mountain watersheds based on high-performance GPU computing,” *Natural Hazards*, vol. 91, no. 2, pp. 567–586, 2018.
- [159] B. F. Sanders, J. E. Schubert, and R. L. Detwiler, “ParBreZo: A parallel, unstructured grid, Godunov-type, shallow-water code for high-resolution flood inundation modeling at the regional scale,” *Advances in Water Resources*, vol. 33, no. 12, pp. 1456–1467, 2010.
- [160] S. Zhang, Z. Xia, R. Yuan, and X. Jiang, “Parallel computation of a dam-break flow model using OpenMP on a multi-core computer,” *Journal of hydrology*, vol. 512, pp. 126–133, 2014.
- [161] ArcGIS, “Technical Reference and Documentation,” *Esri Software*, 2010.
- [162] H. Madsen, *Vandstandsstatistik i Køge Bugt under klimaændringer. Report by Danish Hydraulic Institute for Greve Kommune*. <http://www.klarforsyning.dk/sites/default>

/files/fakta_filer/vandstandsstatistik_i_koege_bugt_under_klimaaendringer_dec08.pdf.
2008.

- [163] S. Berbel Roman, *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), 2014.
- [164] DHI, "Mike21 & MIKE3 Flow model FM. Hydrodynamic and Transport Module Scientific Documentation," *DHI: Hørsholm, Denmark*, 2016.
- [165] D. G. Cacuci, M. Ionescu-Bujor, and I. M. Navon, *Sensitivity and uncertainty analysis*. Chapman & hall/CRC Boca Raton, Florida, 2003, vol. 1.
- [166] A. Mosavi, P. Ozturk, and K.-w. Chau, "Flood prediction using machine learning models: Literature review," *Water*, vol. 10, no. 11, p. 1536, 2018.
- [167] J. M. M. Alves *et al.*, "Autoencoders: application to forecasting," 2010.
- [168] C. Wu and K. Chau, "A flood forecasting neural network model with genetic algorithm," *International journal of environment and pollution*, 2006.
- [169] A. Mukerji, C. Chatterjee, and N. S. Raghuwanshi, "Flood forecasting using ANN, neuro-fuzzy, and neuro-GA models," *Journal of Hydrologic Engineering*, vol. 14, no. 6, pp. 647–652, 2009.
- [170] G.-F. Lin and G.-R. Chen, "A systematic approach to the input determination for neural network rainfall–runoff models," *Hydrological Processes: An International Journal*, vol. 22, no. 14, pp. 2524–2530, 2008.
- [171] G.-F. Lin and L.-H. Chen, "A non-linear rainfall-runoff model using radial basis function network," *Journal of Hydrology*, vol. 289, no. 1–4, pp. 1–8, 2004.
- [172] F.-J. Chang, P.-A. Chen, Y.-R. Lu, E. Huang, and K.-Y. Chang, "Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control," *Journal of Hydrology*, vol. 517, pp. 836–846, 2014.
- [173] P.-A. Chen, L.-C. Chang, and F.-J. Chang, "Reinforced recurrent neural networks for multi-step-ahead flood forecasts," *Journal of Hydrology*, vol. 497, pp. 71–79, 2013.
- [174] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

- [175] W. Zaremba, I. Sutskever, and O. Vinyals, “Recurrent neural network regularization,” *arXiv preprint arXiv:1409.2329*, 2014.
- [176] R Bertolami, H Bunke, S Fernandez, A Graves, M Liwicki, and J Schmidhuber, “A Novel Connectionist System for Improved Unconstrained Handwriting Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, 2009.
- [177] P. Doetsch, M. Kozielski, and H. Ney, “Fast and robust training of recurrent neural networks for offline handwriting recognition,” in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, IEEE, 2014, pp. 279–284.
- [178] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, “Dropout improves recurrent neural networks for handwriting recognition,” in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, IEEE, 2014, pp. 285–290.
- [179] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, “TTS synthesis with bidirectional LSTM based recurrent neural networks,” in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.
- [180] E. Marchi, G. Ferroni, F. Eyben, L. Gabrielli, S. Squartini, and B. Schuller, “Multi-resolution linear prediction based features for audio onset detection with bidirectional LSTM neural networks,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, IEEE, 2014, pp. 2164–2168.
- [181] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.
- [182] Y. Wang, I. M. Navon, X. Wang, and Y. Cheng, “2D Burgers equation with large Reynolds number using POD/DEIM and calibration,” *International Journal for Numerical Methods in Fluids*, vol. 82, no. 12, pp. 909–931, 2016.
- [183] M. Alotaibi, V. M. Calo, Y. Efendiev, J. Galvis, and M. Ghommem, “Global–local nonlinear model reduction for flows in heterogeneous porous media,” *Computer Methods in Applied Mechanics and Engineering*, vol. 292, pp. 122–137, 2015.
- [184] K. Hoang, Y. Fu, and J. Song, “An hp-proper orthogonal decomposition–moving least squares approach for molecular dynamics simulation,” *Computer Methods in Applied Mechanics and Engineering*, vol. 298, pp. 548–575, 2016.
- [185] B. Peherstorfer and K. Willcox, “Data-driven operator inference for nonintrusive projection-based model reduction,” *Computer Methods in Applied Mechanics and Engineering*, vol. 306, pp. 196–215, 2016.

- [186] G. Dimitriu and N. Apreutesei, “Comparative study with data assimilation experiments using proper orthogonal decomposition method,” in *International Conference on Large-Scale Scientific Computing*, Springer, 2007, pp. 393–400.
- [187] K. C. Hoang, P. Kerfriden, and S. P. A. Bordas, “A fast, certified and tuning free two-field reduced basis method for the metamodeling of affinely-parametrised elasticity problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 298, pp. 121–158, 2016.
- [188] G. Le Quilliec, B. Raghavan, and P. Breitkopf, “A manifold learning-based reduced order model for springback shape characterization and optimization in sheet metal forming,” *Computer Methods in Applied Mechanics and Engineering*, vol. 285, pp. 621–638, 2015.
- [189] A. Manzoni, F. Salmoiraghi, and L. Heltai, “Reduced basis Isogeometric Methods (RB-IGA) for the real-time simulation of potential flows about parametrized NACA airfoils,” *Computer Methods in Applied Mechanics and Engineering*, vol. 284, pp. 1147–1180, 2015.
- [190] A. G. Buchan, A. Calloo, M. G. Goffin, S. Dargaville, F. Fang, C. C. Pain, and I. M. Navon, “A POD reduced order model for resolving angular direction in neutron/photon transport problems,” *Journal of Computational Physics*, vol. 296, pp. 138–157, 2015.
- [191] A. Corigliano, M. Dossi, and S. Mariani, “Model order reduction and domain decomposition strategies for the solution of the dynamic elastic–plastic structural problem,” *Computer Methods in Applied Mechanics and Engineering*, vol. 290, pp. 127–155, 2015.
- [192] R. Ștefănescu, A. Sandu, and I. M. Navon, “Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations,” *International Journal for Numerical Methods in Fluids*, vol. 76, no. 8, pp. 497–521, 2014.
- [193] M Couplet, C Basdevant, and P Sagaut, “Calibrated reduced-order POD-Galerkin system for fluid flow modelling,” *Journal of Computational Physics*, vol. 207, no. 1, pp. 192–220, 2005.
- [194] S. Sargsyan, S. L. Brunton, and J. N. Kutz, “Nonlinear model reduction for dynamical systems using sparse sensor locations from learned libraries,” *Physical Review E*, vol. 92, no. 3, p. 033 304, 2015.
- [195] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem, “The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows,” *Journal of Computational Physics*, vol. 242, pp. 623–647, 2013.

- [196] J. Du, F. Fang, C. C. Pain, I. Navon, J. Zhu, and D. A. Ham, “POD reduced-order unstructured mesh modeling applied to 2D and 3D fluid flow,” *Computers & Mathematics with Applications*, vol. 65, no. 3, pp. 362–379, 2013.
- [197] D. Xiao, F. Fang, A. Buchan, C. Pain, I. Navon, and A. Muggeridge, “Non-intrusive reduced order modelling of the Navier–Stokes equations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 293, pp. 522–541, 2015.
- [198] D. Xiao, F. Fang, C. Pain, and G. Hu, “Non-intrusive reduced-order modelling of the Navier–Stokes equations based on RBF interpolation,” *International Journal for Numerical Methods in Fluids*, vol. 79, no. 11, pp. 580–595, 2015.
- [199] D. Xiao, Z. Lin, F. Fang, C. C. Pain, I. M. Navon, P. Salinas, and A. Muggeridge, “Non-intrusive reduced-order modeling for multiphase porous media flows using Smolyak sparse grids,” *International Journal for Numerical Methods in Fluids*, vol. 83, no. 2, pp. 205–219, 2017.
- [200] O. San and R. Maulik, “Extreme learning machine for reduced order modeling of turbulent geophysical flows,” *Physical Review E*, vol. 97, no. 4, p. 042 322, 2018.
- [201] R. Swischuk, L. Mainini, B. Peherstorfer, and K. Willcox, “Projection-based model reduction: Formulations for physics-based machine learning,” *Computers & Fluids*, vol. 179, pp. 704–717, 2019.
- [202] Z. Wang, D. Xiao, F. Fang, R. Govindan, C. Pain, and Y. Guo, “Model identification of reduced order fluid dynamics systems using deep learning,” *International Journal for Numerical Methods in Fluids*, vol. 86, no. 4, pp. 255–268, 2018.
- [203] K. Worden and P. Green, “A machine learning approach to nonlinear modal analysis,” *Mechanical Systems and Signal Processing*, vol. 84, pp. 34–53, 2017.
- [204] D. Xiao, F. Fang, J. Zheng, C. Pain, and I. Navon, “Machine learning-based rapid response tools for regional air pollution modelling,” *Atmospheric environment*, vol. 199, pp. 463–473, 2019.
- [205] G. Kerschen and J.-C. Golinval, “Feature extraction using auto-associative neural networks,” *Smart Materials and Structures*, vol. 13, no. 1, p. 211, 2003.
- [206] G. Kerschen and J.-C. Golinval, “A model updating strategy of non-linear vibrating structures,” *International Journal for Numerical Methods in Engineering*, vol. 60, no. 13, pp. 2147–2164, 2004.
- [207] R. Soltanpoor and T. Sellis, “Prescriptive analytics for big data,” in *Australasian Database Conference*, Springer, 2016, pp. 245–256.

- [208] G.-Y. Vahn, “Business analytics in the age of Big Data,” *Business Strategy Review*, vol. 25, no. 3, pp. 8–9, 2014.
- [209] G. Berkooz, P. Holmes, and J. L. Lumley, “The proper orthogonal decomposition in the analysis of turbulent flows,” *Annual review of fluid mechanics*, vol. 25, no. 1, pp. 539–575, 1993.
- [210] C. Eckart and G. Young, “The approximation of one matrix by another of lower rank,” *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.
- [211] M. Arhan and R. Plaisted, “Non-linear deformation of sea-wave profiles in intermediate and shallow water,” *Oceanologica Acta*, vol. 4, no. 2, pp. 107–115, 1981.
- [212] P. L. F. Liu, H. Yeh, and C. Synolakis, “Advanced Numerical Models for Simulating Tsunami Waves and Runup,” *Advances in Coastal and Ocean Engineering*, 2008.
- [213] Fluidity, “Fluidity manual v4.1.12,” *Applied Modelling and Computation Group, Imperial College*, https://figshare.com/articles/Fluidity_Manual/1387713, manual 2015.
- [214] T. Tieleman and G. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.
- [215] V. Babovic, R. Cañizares, H. R. Jensen, and A. Klinting, “Neural networks as routine for error updating of numerical models,” *Journal of Hydraulic Engineering*, vol. 127, no. 3, pp. 181–193, 2001.
- [216] Y. Sun, V. Babovic, and E. S. Chan, “Multi-step-ahead model error prediction using time-delay neural networks combined with chaos theory,” *Journal of Hydrology*, vol. 395, no. 1-2, pp. 109–116, 2010.
- [217] R Hu, F Fang, C. Pain, and I. Navon, “Rapid spatio-temporal flood prediction and uncertainty quantification using a deep learning method,” *Journal of Hydrology*, 2019.
- [218] S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.
- [219] B Yildirim, C Chrysosostomidis, and G. Karniadakis, “Efficient sensor placement for ocean measurements using low-dimensional concepts,” *Ocean Modelling*, vol. 27, no. 3-4, pp. 160–173, 2009.

- [220] I. Bright, G. Lin, and J. N. Kutz, “Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements,” *Physics of Fluids*, vol. 25, no. 12, p. 127 102, 2013.
- [221] S. L. Brunton, J. H. Tu, I. Bright, and J. N. Kutz, “Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems,” *SIAM Journal on Applied Dynamical Systems*, vol. 13, no. 4, pp. 1716–1732, 2014.
- [222] M. A. Grepl and A. T. Patera, “A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations,” *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 39, no. 1, pp. 157–181, 2005.
- [223] K. Cohen, S. Siegel, and T. McLaughlin, “Sensor placement based on proper orthogonal decomposition modeling of a cylinder wake,” in *33rd AIAA Fluid Dynamics Conference and Exhibit*, 2003, p. 4259.
- [224] I. D. Rutherford, “Data assimilation by statistical interpolation of forecast error fields,” *Journal of the Atmospheric Sciences*, vol. 29, no. 5, pp. 809–815, 1972.
- [225] C. Pain, M. Piggott, A. Goddard, F Fang, G. Gorman, D. Marshall, M. Eaton, P. Power, and C. De Oliveira, “Three-dimensional unstructured mesh ocean modelling,” *Ocean Modelling*, vol. 10, no. 1-2, pp. 5–33, 2005.
- [226] J. C. Neal, T. J. Fewtrell, P. D. Bates, and N. G. Wright, “A comparison of three parallelisation methods for 2D flood inundation models,” *Environmental Modelling & Software*, vol. 25, no. 4, pp. 398–411, 2010.
- [227] R. Lamb, M. Crossley, and S. Waller, “A fast two-dimensional floodplain inundation model,” in *Proceedings of the Institution of Civil Engineers-Water Management*, Thomas Telford Ltd, vol. 162, 2009, pp. 363–370.
- [228] R. Buyya, “High performance cluster computing: programming and applications, vol. 2,” *Pre ticeHallPTR, NJ*, vol. 29, 1999.
- [229] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Inferring solutions of differential equations using noisy multi-fidelity data,” *Journal of Computational Physics*, vol. 335, pp. 736–746, 2017.
- [230] S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

- [231] Q. Wang, J. S. Hesthaven, and D. Ray, “Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem,” *Journal of computational physics*, vol. 384, pp. 289–307, 2019.
- [232] J. N. Kani and A. H. Elsheikh, “DR-RNN: A deep residual recurrent neural network for model reduction,” *arXiv preprint arXiv:1709.00939*, 2017.
- [233] S. Chaturantabut and D. C. Sorensen, “Nonlinear model reduction via discrete empirical interpolation,” *SIAM Journal on Scientific Computing*, vol. 32, no. 5, pp. 2737–2764, 2010.
- [234] F. J. Gonzalez and M. Balajewicz, “Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems,” *arXiv preprint arXiv:1808.01346*, 2018.
- [235] N. Srivastava, E. Mansimov, and R. Salakhudinov, “Unsupervised learning of video representations using lstms,” in *International conference on machine learning*, 2015, pp. 843–852.