UNIVERSITY OF CALIFORNIA,
IRVINE

The Effect of Urban Texture on Flood Behavior

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Civil and Environmental Engineering

by

Sarah Balaian

Thesis Committee:
Assistant Professor Mohammad Javad Abdolhosseini Qomi, Chair
Professor Brett Sanders
Assistant Professor Joel Lanning

2021

DEDICATION

To my parents, Katrin and Edward, who have always encouraged me to pursue my dreams and without whom I would not be who I am today. Through the highs and lows, they have walked beside me and their love and support is forever in my heart.

To my friends who motivate, inspire, and uplift me every day.

And finally, to my advisor, Professor M.J. Qomi, for his endless support and involvement in my academic development.

TABLE OF CONTENTS

Page

LIST OF FIGURES **v**

LIST OF TABLES **vi**

ACKNOWLEDGMENTS **vii**

ABSTRACT OF THE THESIS **viii**

1 Introduction **1**
- 1.1 Flood Hazards and Urbanization . . . 1
- 1.2 Flood Modeling . . . 4
- 1.3 Urban Form . . . 5
- 1.4 Research Objective . . . 8
- 1.5 Thesis Outline . . . 8

2 Fluid Dynamics Theory **10**
- 2.1 Introduction . . . 10
- 2.2 Shallow Water Equations . . . 11
- 2.3 Reynolds Number . . . 12
- 2.4 Froude Number . . . 13
- 2.5 Summary . . . 16

3 Synthetic City Modeling **17**
- 3.1 Introduction . . . 17
- 3.2 Porosity . . . 18
- 3.3 Mermin Order Parameter . . . 20
- 3.4 Other Model Parameters . . . 25
  - 3.4.1 Length/Width of Channel . . . 25
  - 3.4.2 Slope . . . 25
- 3.5 Hybrid Reverse Monte Carlo Algorithm . . . 26
- 3.6 Model Refinement . . . 33
  - 3.6.1 Merging Overlapping Buildings . . . 33
  - 3.6.2 Expand Configuration to Channel Length . . . 34
- 3.7 Computational Grid/Mesh . . . 35
- 3.8 Summary . . . 37

4 BreZo: An Adaptive Godunov-Based Model for Flood Simulation **39**
4.1 Introduction . . . 39
4.2 Methodology . . . 40
4.3 Summary . . . 43

5 Flood Inundation Results **44**
5.1 Introduction . . . 44
5.2 Convergence Analysis . . . 45
5.2.1 Channel Length . . . 45
5.2.2 Run Time . . . 47
5.3 Final Simulations . . . 50
5.4 Dimensional Analysis . . . 57
5.5 Effects of Volumetric Flow Rate . . . 59
5.6 Effects of Porosity and Spatial Order . . . 62
5.7 Summary . . . 71

6 Conclusion **72**

Bibliography **75**

Appendix A MATLAB Codes **80**

LIST OF FIGURES

Page

3.1 Porosity Examples 19
3.2 Difference Between C_n Values 22
3.3 Inner Boundary for Mermin Calculation 23
3.4 Mermin Spatial Order Examples 24
3.5 Energy Graph 29
3.6 Workflow of the HRMC Process 31
3.7 HRMC Convergence 32
3.8 Merge Overlapping Buildings 33
3.9 Remove Holes and Outliers 34
3.10 Duplication of Configuration 35
3.11 Computational Mesh/Grid 38

5.1 Entire Workflow of Study 44
5.2 Convergence Analysis for Channel Lengths 46
5.3 Non-Uniform Flow 47
5.4 Run Time for Three Configurations 49
5.5 Plot of 20 Different Configurations. 51
5.6 Boundary Conditions 53
5.7 Flow Profiles of 60 Configurations 54
5.8 Upstream Flood Height Measurement 56
5.9 Relationship Between Upstream Flood Heights and Volumetric Flow Rate . . 60
5.10 Relationship Between Upstream Flood Heights and Porosity 63
5.11 Determination of a and n 64
5.12 Relationship Between Upstream Flood Heights and Mermin Order 65
5.13 Determination of b and m 67
5.14 Contour Plot of Water Heights for Case 1. 68
5.15 Contour Plot of Water Heights for Case 2. 69
5.16 Contour Plot of Water Heights for Case 3. 70

LIST OF TABLES

Page

5.1 List of Porosity and Mermin Order Values 52
5.2 Dimensional Matrix 58
5.3 The Three Flood Cases 68

ACKNOWLEDGMENTS

This work would not have been possible without the generous support of my advisors, committee members, colleagues, friends and family. I would first like to express my deepest gratitude and appreciation to my advisor and committee chair, Professor Mohammad Javad Abdolhosseini Qomi whose passion and intellect has motivated me endlessly. His incredible tenacity and profound wisdom has inspired me to challenge the unknown and strive for greatness in all that I do. Without his guidance, my journey through the world of academia would not have been nearly as enjoyable or impactful. I can say with complete confidence that I couldn't have asked for a better advisor, mentor, and friend.

I would also like to thank my committee members Professor Brett Sanders and Professor Joel Lanning. The remarkable work done by Professor Brett Sanders in the field of flood modeling has inspired me greatly throughout the course of this study. His contributions have brought life to this project. Professor Joel Lanning's incredible influence in the classroom and beyond has greatly shaped my experience as a student. It has been an honor and a privilege to grow as an engineer and scientist under their direction. I would also be remiss not to mention the incredible help of many of my colleagues, namely Ali Morshedifard and Jochen Schubert, whose expertise and insight greatly aided in the success of this work.

Lastly, I gratefully acknowledge that this research was funded in part by the University of California, Irvine, Henry Samueli School of Engineering through the PhD Bridge Fellowship.

ABSTRACT OF THE THESIS

The Effect of Urban Texture on Flood Behavior

By

Sarah Balaian

Master of Science in Civil and Environmental Engineering

University of California, Irvine, 2021

Assistant Professor Mohammad Javad Abdolhosseini Qomi, Chair

Flood damages around the world have been increasing at a daunting rate, causing extreme socioeconomic harm to entire communities. With the increase in urbanization and population growth, a framework to understanding the effects that the changing urban landscape has on natural conditions such as flooding has become crucial. While many different approaches to modeling urban flooding have been introduced, few have considered the effects of spatial orientation and form of urban areas on flood behavior. Here, we employ a parameter that is commonly used in statistical and molecular science, called an order parameter, to aid in the quantification of urban texture. A hybrid reverse Monte Carlo algorithm is implemented to produce synthetic cities for the purpose of flood modeling; synthetic cities allow for better control of morphology parameters and reduce the number of independent variables that could affect resulting trends. By subsequently inundating these models with different flood possibilities, the results show that there is a connection between the form of the urban layout and resulting flood water heights. The interaction between porosity and spatial order in the resulting flood heights prove that they both should be considered in tandem when studying urban morphology. These results can be used in future flood hazard mitigation and urban planning programs to account for effects of urban layout on flood levels.

Chapter 1

Introduction

1.1 Flood Hazards and Urbanization

Despite the tremendous amount of money and effort that has been dedicated to controlling floods, flood damages in the U.S. have increased over the past decades and are projected to increase even more into the future [12, 17, 18, 20, 23]. From 1928 through 2000, flood control expenditures have amounted to $122 billion and in that same time, flood damages have consistently increased with relatively high damage years (above $5 billion in damages) becoming more common [12]. Natural hazards in general are on the rise in the United States, causing more than half trillion dollars in direct economic losses over the past 50 years; flooding and coastal hazards alone account for one quarter of these losses [16]. This trend coincides with the increased numbers of people and assets exposed to flood risk. Currently, over 40 million people, or 0.6 percent of the global population, are exposed to a 1 in 100 year coastal flood event. One study showed that the total value of assets exposed to coastal flooding in 2005 in the world's largest post cities is estimated to be $3 trillion, or 5 percent of the global GDP for that year [18]. This same study suggested that by 2070, the total

exposed population could grow more than threefold, with asset exposure increasing to more than ten times current levels [18].

There are many causes of increased flood exposure and damage, including climate change, population and socioeconomic growth, increased urbanization, and violations of flood management guidelines [12, 17, 18]. Climate change, in particular, has been a topic of great discussion in today's society and the data supports the claims that it is one of the driving factors of urban flooding. Of the many environmental changes that are caused by climate change, the largest contribution to population exposure is sea level rise [18, 19]. It is likely that the rate of sea level rise (SLR) globally in the 21st century will exceed rates observed in the past, with estimates projecting SLR of 0.3 meters up to 1 meter by 2100 mainly due to land-ice contributions. This could cause up to 4.6% of the global population to be flooded annually [14, 20]. As sea level rise boosts the average water height, once in a decade floods will begin to reach further inland, which will not only intensify floods in regions already exposed but could also introduce new hazards in areas unfamiliar with them [16, 23]. Along with climate change, flood losses are further exacerbated by increased urbanization and wealth concentration, especially in floodplains [12, 31]. Continued capital investment in concentrated areas leads to greater losses when infrastructure and systems of higher economic wealth are inundated [31]. One study showed that in certain cases the role of socioeconomic drivers is actually larger than that of climate change; even without projected sea level rise, asset exposure could already grow eightfold [18]. Any marginal improvements in hazard mitigation over the years have become insignificant due to the increased concentration of people and goods in hazardous areas, along with the emergence of new vulnerabilities and hazards [16]. With urbanization and concentrated wealth also comes an increase in impermeable surfaces such as concrete. Since 2001, the area of impermeable surfaces in urban and suburban areas has risen by 22% [31]. This, along with mismanagement of efficient sewerage infrastructure systems, also worsens flood damages in urban areas.

Although public interest in the dangers of climate change has grown and direct links have been made between increased urbanization and flood damages, little preparation has been done for the management of extreme events and their long term implications [18, 19]. In the years after the 2009 Copenhagen meeting, which was one of many international events geared towards climate change negotiations, the growth rate of new homes inside America's ten percent flood risk zones outpaced the growth rate outside of those areas in a third of the country's coastal states [23]. Since most major cities are developed along seas and waterways, they follow trends towards building in more hazardous areas as they increase in size and wealth. This causes flood risk to threaten more people than other natural catastrophes [18]. Millions of people already live in big metropolitan centers with projections of 6.5 billion people, or around 68% of the global population, residing in urban areas by 2050 [2, 18]. Compounded with climate change issues, this level of mass urbanization could increase the frequency and severity of natural disasters. Instead of addressing these trends at the root, sea level rise and increased urbanization in hazardous areas has resulted in over-reliance on applied protections such as dams and dikes which increase the risk of massive catastrophes in the case of failure [20].

The apparent inability of cities to control and mitigate flood damages is due to many factors, including bureaucratic conflicts and the technical complexity of understanding urban flooding. A serious shortcoming of existing hazard reduction policies is the inability to empirically quantify economically devastating impacts of disasters [16]. Factors such as a lack of data on populations and developments in floodplains, and poor understanding of flood control measures and policies contribute to this issue [12]. Loss reduction efforts need to be evaluated and re-assessed in terms of their effectiveness [16], but there seems to be resistance towards making these improvements. This is partially due to the very nature of extreme weather events: they happen relatively rapidly, are impossible to determine due to a lack of early warning signs, and are assumed to be very infrequent. These factors have shaped stakeholders' perception of risk, fostering complacency and affecting risk management deci-

sions [2]. To improve the mitigation efforts of societies across the globe, it is imperative to incorporate sustainable development and vulnerability reduction as priorities in urban planning [16]. Risk must be addressed explicitly to inform decision-making, utilizing knowledge between experts and stakeholders to tackle complex problems [2].

1.2 Flood Modeling

With the increase in urban flood events across the world and the degree of socioeconomic harm they cause to entire communities, it has become crucial to produce accurate and effective flood models that can be used by everybody including politicians, stakeholders, and everyday citizens alike. One of the leaders in this production, distribution and implementation of flood maps is the Federal Emergency Management Agency (FEMA). However, the results from some of FEMA's flood mitigation attempts, including the Flood Insurance Rate Maps (FIRMs), fell short of their expectations. The FIRMs had the unintended consequence of providing residents with a false sense of certainty that homes outside the designated flood zones were safe from harm [39]. It is apparent that along with technical accuracy of maps and tools, flood modeling must also consider factors such as people's perceptions and future changes. Sanders et. al. [39] proposes that true reductions in consequences from flooding stem from the application of collaborative flood modeling between local stakeholders and experts. There is value in the production of detailed, progressive, and innovative flood modeling techniques.

The work of flood modeling has extended past the efforts of agencies such as FEMA into the realm of academia in fields such as urban planning, engineering, and even sociology. Important advancements have included the incorporation of projected climate change and urbanization trends into flood maps to account for unforeseen circumstances [3]. However, much of the existing research on these trends focus on real-world case studies and not more

generic configurations [10]. While these real-world studies and flood models are useful in their own right, more generic models can provide a better insight into how isolated topographic trends and patterns in urban form affect flood behavior.

The creation of more generic urban form and flood inundation models has introduced innovative and valuable modeling techniques into the conversation of flood risk mitigation. One major trend that is studied often is urban expansion, such as transitions between nonurban to urban land use. The current policies for flood damage mitigation most commonly focus on controlling expansion into flood prone zones. This expansion into hazardous zones not only exposes more infrastructure, assets, and people to risk but also alters the natural flow of the floods that do pass through, leading to even greater problems. Huang et. al. [21] presented both physical experimentation and computational modeling which proved the impact of building coverage on flow behavior. By considering only the urban expansion process however, models may fail to estimate flood damages properly because it neglects the densities and patterns of urban cells [28]. In this regard, recent studies have introduced urban models that focus on patterns of already urbanized areas instead of models that simply observe a switch from nonurban to urban land. For instance, Mustafa et. al. [28] investigates different urban development scenarios of densification and their impacts on flood damages. Along this path, Bruwier et. al. [10] also presents an analysis on the influence of multiple urban characteristics on surface flow in the case of pluvial flooding. Studies such as these have contributed to a new multidisciplinary approach to flood modeling: a merger between fluid dynamics and urban form analysis.

1.3 Urban Form

The creation of a novel type of landscape through intense urbanization has brought with it a new set of challenges to the urban environment. As urban areas have become more

populated, there has been a push to understand the connections between urban form and its effects on an array of different conditions. For instance, the urban heat island, which is a phenomenon that creates urban micro-climates which experience increased temperatures relative to their surroundings, is a direct consequence of increased urbanization [42]. Land use and land cover composition have also been linked to land surface temperature changes [53]. Due to building heights and density of urban areas, cities take on unique characteristics of high-rise compact building blocks and deep street canyons. Yuan et. al. [51] recognized that this is a major source of concern in regard to natural ventilation in urban areas. Reduced permeability has caused stagnant air to worsen outdoor thermal comfort and air pollution dispersion. Different wind patterns and vortex regimes have been known to form in response to this unique urban landscape [22], further contributing to reduced ventilation. Urban flood damages have also been exacerbated by the increased density of cities and dramatic changes in hydrological conditions as urbanization continues to grow [28].

Many recent studies have moved beyond simply recognizing this relation and have attempted to quantify the connection between urban morphology and its adverse effects. Traditionally, cities were characterized by different urban typologies, or spatial compositions of buildings and land use types [52]. These compositions were the main factors of urban morphology considered when trying to understand the effects of urban structure. For instance, many urban flood mitigation efforts have focused on regulating land use and encroachment [11] but fail to recognize the effects that configuration can also have on flood disasters. Attributing urban conditions to not only the composition of the region, as in the land use designation, but also the configuration introduces new parameters to the discussion of city modeling. While the composition of land cover is of great importance to urban conditions such as land surface temperature and flood damages, spatial arrangements may also play a key role in increasing or decreasing these effects [53]. Permeability and porosity are two notable parameters that have been emphasized in this field. Ng et. al. [30] utilized these parameters to recommend "air paths" which can increase ventilation in high-density cities. Tadi et. al. [45] presents

porosity as a fundamental morphological characteristic of urban systems that integrates two basic components of urban space: volumes and voids. Along with porosity, other morphological indicators of environmental performance including density, rugosity (height variations of the urban canopy), and sinuosity (angular differences between fluid flow and open spaces) have been used to characterize city layouts [1]. Some models go even further to introduce multiple parameter systems which intricately recreate urban forms. Mustafa et. al. [29] presents a model that introduces up to 10 parameters for generating urban layouts capable of reducing water depth during urban floods. All of these findings have recognized that it is crucial to develop a framework within which the spatial information about urban form can be understood and analyzed.

A novel concept for the characterization of urban geometry that has been employed in recent years is the application of methods commonly used in molecular and statistical physics to quantitatively analyze urban form. This is justified by the parallels that can be drawn between urban morphology and molecular physics; at the right scale, the structure of cities closely resembles the molecular structure of materials [42, 43, 48]. Sobstyl et. al. [42] observed that the distinct textures of cities could be compared to the molecular structure of crystals or liquids. In particular, these studies adopt the radial distribution function (RDF) to analyze spatial characteristics, which is a tool that describes the probability of finding a point at a specific distance from the reference point. The radial distribution function is an important and widely used tool which was shown to extract crucial information about building clusters in the context of urban cities. Another tool that was briefly mentioned by Sobstyl [42] is the Mermin order parameter which was used to further describe city texture.

Order parameters are tools used in statistical physics that capture angular distortion between points, whether it be particles in a molecular structure or buildings in a city. These parameters determine the degree of spacial ordering within a configuration and can act as a single statistical measure of the arrangement of points, preserving the heterogeneous nature

of the structure. Presenting spatial information through a single value allows for the simplified measurement of its effects on any selected physical response. Order parameters have commonly been used in molecular physics to analyze a wide array of issues, including stress intensity in the design of composite materials [4] and physio-chemical behavior of complex systems [35]. Its integration into the world of urban modeling, although new, demonstrates great potential.

1.4 Research Objective

First and foremost, this project aims at establishing a clear and concise connection between the form/morphology of urban areas and its effect on flood behavior, specifically water heights. A probabilistic automated algorithm is introduced for generating synthetic cities using morphology parameters that are representative of features of urban form. One important variable being tested for is the spatial order of the system, a parameter that we propose to be a key factor in urban form. By subsequently inundating these synthetic cities, we can obtain crucial information about the correlation between urban morphology and resulting flood heights. The tools and methods presented in this work can further aid the development of urban modeling and flood hazard mitigation efforts.

1.5 Thesis Outline

Following this introductory chapter, there are five chapters that make up this thesis. Chapter two discusses important fluid dynamics theory that is necessary in both the modeling and resulting data analysis of this study. In chapter three, the algorithm for producing synthetic cities is explained, along with the parameters that are used for said creation. This chapter also highlights the spatial order parameter that is crucial to the novelty of this project.

Chapter four is dedicated to describing the methods behind the finite-volume model BreZo, which is used to inundate the synthetic cities. Chapter five presents the final results produced in this study. The resulting data from the inundated synthetic cities are analyzed and important trends pertaining to the effects of urban morphology are extracted. This thesis concludes with chapter six, which presents some final thoughts and recommendations for future work.

Chapter 2

Fluid Dynamics Theory

2.1 Introduction

Fluid dynamics is the branch of science that deals with the movement of gases and liquids. Understanding the basics of fluid dynamics is crucial to the successful modeling of any type of flow, whether it be the flow of flood waters through a city or the flow of air along an airplane wing. This chapter addresses some of the most critical aspects of fluid dynamics when discussing hydrodynamic open channel flow, including the shallow water equations and the dimensionless Reynolds number and Froude number. A brief discussion of the famous Navier-Stokes equations gives context to the derivation of the shallow water equations which are used for computing flood models. Beyond the modeling of this flow, the dimensionless Reynolds number and Froude number can be used to analyze resulting flow behavior.

2.2 Shallow Water Equations

The fluid flow of shallow water systems can be modeled using a specific set of equations called the shallow water equations. These equations are applicable to systems in which the horizontal length scale tends to be much larger than the vertical length scale. Certain types of phenomena such as tsunami propagation, river flow, and flooding can be modeled using these equations.

The shallow water equations are a coupled system of partial differential equations (PDEs) that are derived from the Navier-Stokes equations and consists of a fluid continuity equation and two momentum equations. The Navier-Stokes equations are a set of two partial differential equations which express the conservation of mass and the conservation of momentum and can be written as follows:

$$\begin{aligned} \nabla \cdot \mathbf{v} &= 0 \\ \frac{\partial}{\partial t}\rho\mathbf{v} + \nabla \cdot (\rho\mathbf{v}\otimes\mathbf{v}) &= -\nabla p + \rho\mathbf{g} + \nabla \cdot \overline{\mathbf{T}} \end{aligned} \tag{2.1}$$

Here, v is the three-dimensional velocity vector $v = (u\, v\, w)^T$, ρ is the density of the fluid, ∇p is the pressure gradient, and $\mathbf{T}$ is a matrix of stress terms. Also, the following assumptions are made about the fluid: it is incompressible, *i.e.* the density ρ is constant, and it is a Newtonian fluid [15]. In the case of shallow flow where the horizontal length scale is much greater than the vertical length scale, the vertical dimension can be neglected and the previous equations can be averaged over the depth, producing the simplified two-dimensional shallow water equations (SWEs). The depth-integrated continuity equation and the depth-integrated x- and y-momentum equations combine to form the two-dimensional SWEs in conservative form as:

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(h\overline{u}) + \frac{\partial}{\partial y}(h\overline{v}) = 0$$
$$\frac{\partial}{\partial t}(h\overline{u}) + \frac{\partial}{\partial x}(h\overline{u}^2) + \frac{\partial}{\partial y}(h\overline{uv}) = -gh\frac{\partial z}{\partial x} + \frac{1}{\rho}[\tau_{sx} - \tau_{bx} + F_x] \tag{2.2}$$
$$\frac{\partial}{\partial t}(h\overline{v}) + \frac{\partial}{\partial x}(h\overline{uv}) + \frac{\partial}{\partial y}(h\overline{v}^2) = -gh\frac{\partial z}{\partial y} + \frac{1}{\rho}[\tau_{sy} - \tau_{by} + F_y]$$

Here, h is the total depth of the water and z is the elevation of the free surface relative to the mean water level. Also, τ_{sx} and τ_{sy} are the surface shear stress, τ_{bx} and τ_{by} are the bottom friction, and F_x and F_y are the external forces on the system [15]. These values are determined based on the system that is being considered. Often, the shallow water equations can appear in different forms depending on its application by performing differentiation and canceling out terms [49]. Also, to solve for the shallow water equations, different numerical methods are implemented to obtain specific solutions of the flow. The nonconservative form of the previous formulation and a numerical method of solving for it will be discussed in Chapter 4.

2.3 Reynolds Number

After flow variables are obtained through computations, they are analyzed using important dimensionless parameters to understand the behavior of the flow and how it is characterized. Many dimensionless parameters have been commonly used for this purpose, providing simple and effective tools for flow characterization. No discussion about dimensionless flow parameters is complete without the Reynolds number, which is arguably the most important parameter in fluid mechanics. This parameter was introduced by the British engineer Osborne Reynolds in 1883 [50] and it correlates the inertial forces to the viscous forces of a fluid in motion. This correlation describes the transport properties of a fluid, distinguishing

between laminar flow and turbulent flow [36]. This number is defined as follows:

$$Re = \frac{\rho D v}{\mu} \tag{2.3}$$

where Re is the dimensionless Reynolds number, ρ is the density of the fluid, D is the diameter of the passage way, v is the velocity of the fluid, and μ is the viscosity of the fluid. When the Reynolds number is 2,000 and below, the flow can be characterized as laminar. In laminar flow, the layers of water flow over each other at different speeds with little to no mixing and the fluid particles move in definite and observable paths, making laminar flow much easier to study and understand. When the Reynolds number is around 4,000 and above, the flow is then considered turbulent [24]. Turbulent flow is much more irregular and does not possess many of the simple patterns that laminar flow does, making it more difficult to deal with. At a Reynolds number between 2,000 and 4,000, the flow is in the transition region. Each type of flow is desirable for some circumstances and problematic for different ones, so it is important to understand the flow characterization using this parameter.

2.4 Froude Number

Another very important dimensionless parameter that is only applicable in free-surface flows is the Froude number, which is named after the British naval architect William Froude [50]. This parameter is the ratio of channel velocity to speed of propagation of surface waves. For a rectangular channel, the Froude number is defined as follows:

$$Fr = \frac{\text{flow velocity}}{\text{surface wave speed}} = \frac{u}{\sqrt{gy}} \tag{2.4}$$

where y is the water depth. The resulting value of this ratio can characterize the behavior of free-surface flow into three different flow regimes:

$$\begin{aligned} Subcritical flow \quad & Fr < 1 \\ Critical flow \quad & Fr = 1 \\ Supercritical flow \quad & Fr > 1 \end{aligned}$$

Critical flow occurs when the velocity of the flow is the same as the wave speed, causing $Fr = 1$. At critical flow, the normal flow depth is equal to the critical depth y_c which represents the depth of flow where energy is at a minimum. This regime is extremely unlikely and is unstable because any disturbance in energy will shift the flow into one of the other two regimes. The two types of flow that are more likely are subcritical and supercritical flow.

Subcritical flow occurs when the water velocity is less than the wave speed, meaning that it behaves in a slow and steady manner. The actual flow depth in this regime is higher than the critical depth ($y > y_c$) and it is common to a mild slope. On the other hand, flow is supercritical when $Fr > 1$, or the flow velocity is greater than the wave speed. This flow identifies with a higher flow rate and lower depth than the critical flow and behaves as rapid or unstable flow. Along with differing flow rates and depths, the regimes also differ in their flow profiles. The flow profile is a measure of how the flow depth changes longitudinally. Changes in the flow regime can occur because of many different reasons. The most common reasons are a change in the channel bed or channel geometry, changes in the volumetric flow rate, or changes in friction. At this instance, a hydraulic jump could possibly occur due to increased available energy from head loss [13].

When modeling open channel flow, it is important to determine which flow regime the flow falls into. This is because it could have a direct effect on how the flow is modeled and how the results are interpreted. To determine whether a flow can be considered subcritical or

supercritical, the normal flow depth must be compared to the critical depth. The critical depth of flow is calculated as follows:

$$y_c = (\frac{q^2}{g})^{1/3} \tag{2.5}$$

where $q = Q/w$ with Q = volumetric flow and w = channel width. This equation is derived from the equation for the Froude number at critical flow ($Fr = 1$), in which the normal depth is the same as the critical depth. It is a function of both volumetric flow and channel geometry. By calculating this value for critical depth, it can then be compared to the observed depth of the flow in question to determine the flow regime.

Since the Froude number is a function of channel velocity and wave speed, it is crucial to defining wave directions and necessary boundary conditions. In shallow water problems, two separate waves are admitted and the SWEs have two characteristic equations:

$$\frac{d}{dt}(u + \sqrt{gh}) = 0 \qquad along \ \frac{dx}{dt} = u + \sqrt{gh} \tag{2.6}$$

$$\frac{d}{dt}(u - \sqrt{gh}) = 0 \qquad along \ \frac{dx}{dt} = u - \sqrt{gh} \tag{2.7}$$

The boundary conditions for shallow water flow are determined by the celerity of these characteristic curves. The two celerities are:

$$\frac{dx}{dt} = u + c \qquad \frac{dx}{dt} = u - c \tag{2.8}$$

where the wave speed $c = \sqrt{gh}$. In the case of subcritical flow, the Froude number $Fr = u/c$ is less than one, meaning $u < c$. When this is the case, the celerity of one characteristic curve is $u + c > 0$ and the other is $u - c < 0$. Since the celetrity is the rate that a wave propagates, this shows one wave is propagating positively and the other is propagating negatively. In terms of boundary conditions, this explains the need for one upstream boundary condition

and one downstream boundary condition. For supercritical flow however, the Froude number is greater than one, meaning $u > c$. The celerity of the characteristic curves are $u + c > 0$ and $u - c > 0$ meaning both waves propagate positively. Because of this, the model will need two upstream boundary conditions and no downstream boundary conditions [47]. This is crucial to accurate computational modeling of fluid flow as incorrect boundary conditions can cause large errors in results.

2.5 Summary

By establishing the important fluid dynamics theories that will be utilized in this study, we have set the foundation for our computations moving forward. The shallow water equations are the governing equations for the flood inundation models, and the dimensionless parameters can be utilized to understand the resulting flow. However, fluid dynamics is only one part of this study; it is coupled with city science and modeling to create a comprehensive understanding of urban flooding.

Chapter 3

Synthetic City Modeling

3.1 Introduction

While many flood inundation models have focused on real-world case studies and city configurations to perform flow analysis, there is a benefit to computationally generating a wide range of urban configurations using specified morphology parameters. Having control over these parameters allows for more consistency when observing trends in urban physics. These configurations must represent combinations of spatial order and porosity values that may be difficult to find in real-life city configurations. While it is possible to analyze city geometries and fit specific order values to each city, a more extensive study can be accomplished with computer-generated "synthetic" city configurations. Although ignoring some important factors of city layouts such as streets and specific land usage, this method of generating synthetic cities avoids the limitations of existing city configurations and provides clear and concise models to work with. This simplification reduces the number of independent variables that are not of importance to this study and allows for a systemic analysis on the influence of selected factors. Synthetic models were created using a novel probabilistic pro-

cedural algorithm which takes user-defined input parameters and uses them to create city configurations. The algorithm follows the hybrid reverse Monte Carlo model which is a reconstruction technique aimed at producing a model by implementing constraints [8]. Monte Carlo methods are widely-used computational algorithms that employ the process of random sampling to make numerical estimations of certain values [34]. The importance and novelty of the algorithm presented here is that it is able to produce hundreds of configurations without manual interference because of its fully-automated nature. The entire process is also randomized to ensure no internal or unintended influence affects the models. This allows for extremely extensive studies to be done on a wide range of configurations, along with future implementation of more parameters.

3.2 Porosity

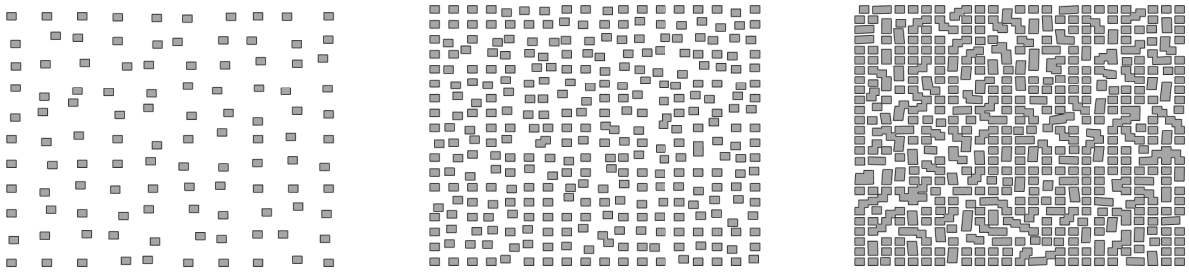
Porosity is a very important parameter that is utilized to understand phenomena such as rock permeability, groundwater flow, separation processes and much more. While this parameter is most commonly employed when discussing fluid motion through systems, it also appears in different forms as well since it is simply a representation of void space. Classic methods of defining random microstructural particle arrangements in composites use a packing fraction, which is a function of the space occupied by the particles, as a complementary representation of porosity. It is also used in the calculations of the radial distribution function, which is a common tool employed in statistical physics [4]. It is not a coincidence that porosity is also a major factor in understanding urban forms and their influence on flow behavior. When discussing urban form, porosity is often presented as "urban void space" [45] or as a percentage of building/built-up coverage, which is the fraction of the total area occupied by buildings [10, 29]. In their study on the influence of urban forms on surface flow in pluvial flooding, Bruwier et. al. [10] determined that flooding severity is mostly influenced by this

building coverage parameter. In our study, we decided to adopt a porosity parameter, which is the complement of the building coverage, because it provides a more direct representation of the space that the flood is passing through instead of the space that is occupied by the buildings.

The porosity parameter in this study is calculated from the ratio of area covered by all building footprints to the area of the entire domain:

$$\Phi = 1 - \frac{Nb^2}{wL} \tag{3.1}$$

where N is the number of buildings in the domain, b^2 is the area of each square building footprint, w is the width of the domain, and L is the length of the channel. In our analysis, the number of buildings is determined by the porosity value and the assigned size of building footprints. Changing the porosity value is done simply by keeping the size of the domain and the area of the building footprints the same, while changing the number of buildings created. Figure 3.1 shows three different configurations, all with the same spatial order but with varying porosity. As the value of porosity Φ becomes smaller, the configurations become more densely packed and the percentage of built-up area is increased. This aids in the visualization of the effects that porosity has on flow behavior.



(a) $\phi = 0.9, \chi_4 = 0.7$ (b) $\phi = 0.7, \chi_4 = 0.7$ (c) $\phi = 0.4, \chi_4 = 0.7$

Figure 3.1: Examples of three different configurations, all with the same spatial order χ_4 value but different porosity Φ values. The porosity progresses from the highest porosity (left) to the lowest porosity (right). As porosity decreases, the area of building coverage increases.

In this study, we create configurations with incrementally changing values of porosity in order to determine its effect on flood behavior. From our understanding, many of the other studies done on the effects of urban form have explored configurations with porosity values of 80% to 50% [10, 28]. To obtain a comprehensive view of the effects of porosity, we have considered a wider range of values, from 90% down to 10% porosity.

3.3 Mermin Order Parameter

While porosity is an important parameter in the understanding of urban form, here we introduce another parameter which will go beyond porosity and describe the angular orientation of buildings. Order parameters are used in statistical physics to quantify the patterns and orientations of particles in different configurations. They have a wide variety of applications within molecular science, and vary depending on the characteristics of the configuration [35]. Studies in different areas such as fracture mechanics have utilized these order parameters to quantify the effects of microstructural spatial order on observed trends in material behavior [4]. As mentioned before, the similarities previously established between structures on the micro-scale and the city-scale provide a precedent for the implementation of these parameters in the quantification of urban form. Of the many order parameters that have been introduced, the Mermin order parameter, which was originally created to quantify order in a two dimensional crystal of circular disks [27], most closely relates to our study of two dimensional urban form. The use of order parameters has been chosen above other previously studied methods of quantifying urban form because it provides new insight into spatial orientation and is more streamlined than other multi-parameter approaches.

The Mermin order parameter quantifies the spatial degree of order of a system by determining the orientation of n neighboring points with respect to a single point of reference, repeating this for all N points, and finally calculating the average of these values. This is adopted to

the current study by considering the centroid of each building to be a point in the domain. The orientation of n neighboring buildings with respect to one building j is calculated as:

$$\chi_{n,j} = \frac{1}{C_{n,j}} \sum_{k=1}^{C_{n,j}} \exp(iC_{n,j}\theta_{j,k}) \tag{3.2}$$

This is done for all N regular points inside the configuration, and the average is then taken as:

$$\chi_{C_n} = \frac{1}{N} \sum_{j=1}^{N} \chi_{n,j} \; = \frac{1}{N} \sum_{j=1}^{N} [\frac{1}{C_{n,j}} \sum_{k=1}^{C_{n,j}} \exp(iC_{n,j}\theta_{j,k})] \tag{3.3}$$

where N is the total number of points within the domain and $\theta_{j,k}$ is the angle between the line defined by the main building j and the k-th neighboring building and the horizontal axis. The variable $C_{n,j}$ is the number of nearest neighbors with respect to the main building j, and determines the type of pattern the parameter is calculating for. For instance, if we are considering a perfect close-packing triangular/hexagonal lattice, this means the number of nearest neighbors is six. This instance is often used in molecular physics applications, where a space-filling structure is considered a perfectly ordered system [4]. In this study of urban form, the most spatially ordered configurations are considered to be regular grids that form urban blocks [52], commonly referred to as grid planes. This format is often used to study a wide range of urban phenomena as it is historically and commonly used in planning mega cities around the world [51]. Understanding this, we consider a perfectly ordered configuration in this study to be a grid lattice, making $C_{n,j}$ equal to four for all future calculations. Figure 3.2 shows the patterns that correspond to each value of $C_{n,j}$ and their Mermin values $\chi_{C_{n,j}}$.

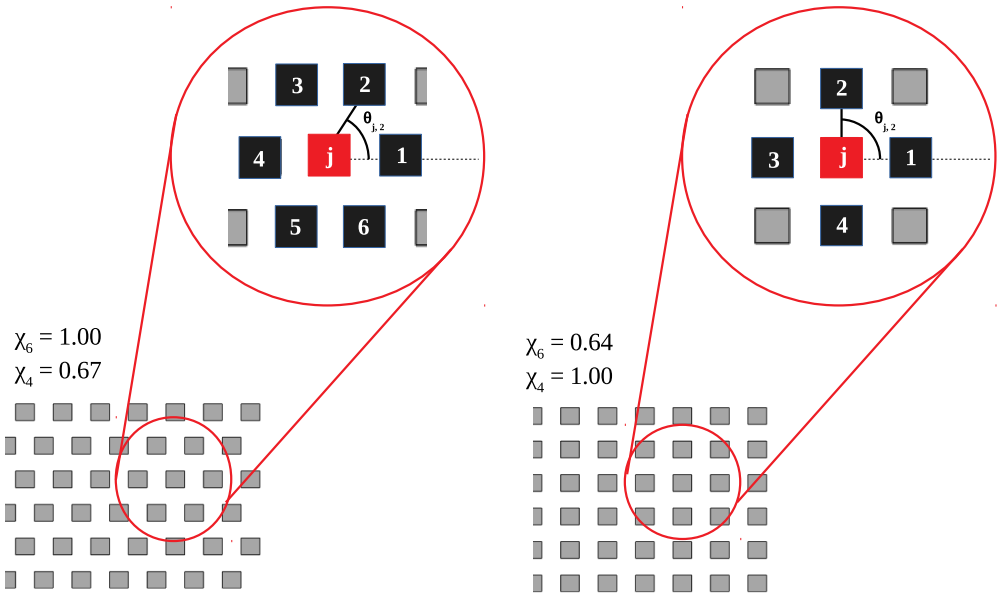


Figure 3.2: Corresponding Mermin order values for two different configurations. The configuration on the left is a perfectly ordered hexagonal system and the configuration on the right is a perfectly ordered grid system. The central atom is shown in red, while the first neighbors are shown in black. The determination of $\theta_{j,2}$ is shown in both systems.

To solve for the Mermin order value with the number of nearest neighbors being four, the term $C_{n,j}$ will be replaced by four in Equation 3.3 to define the final Mermin order parameter χ_4 as follows:

$$\chi_4 = \frac{1}{N}\sum_{j=1}^{N}[\frac{1}{4}\sum_{k=1}^{4}\exp(i4\theta_{j,k})] \tag{3.4}$$

The final Mermin order parameter ranges from a value of zero for extremely disordered systems, to a value of one for perfectly aligned grid systems.

Since buildings at the boundaries experience a lack of neighbors, the calculation of an order parameter for these points will be inaccurately lower than the ones away from the boundary,

making the Mermin order value of the entire system inaccurate. To eliminate this surface effect caused by the under-coordinated buildings at the boundaries, these boundary buildings are not considered as part of the N regular buildings; they are only used as neighboring buildings. An inner box is used to distinguish regular points and strictly neighboring points. This means that, in reference to Equation 3.4, the points inside the inner box are included in $j = 1, 2, \ldots, N$. The outer box represents all the points which are located within one neighboring shell of the reference points and are included in the calculations as $k = 1 \ldots 4$. While all the points in the domain can be considered neighboring points, only points inside the inner box are used as reference points. Figure 3.3 distinguishes between these two boxes and the points that fall within each box.

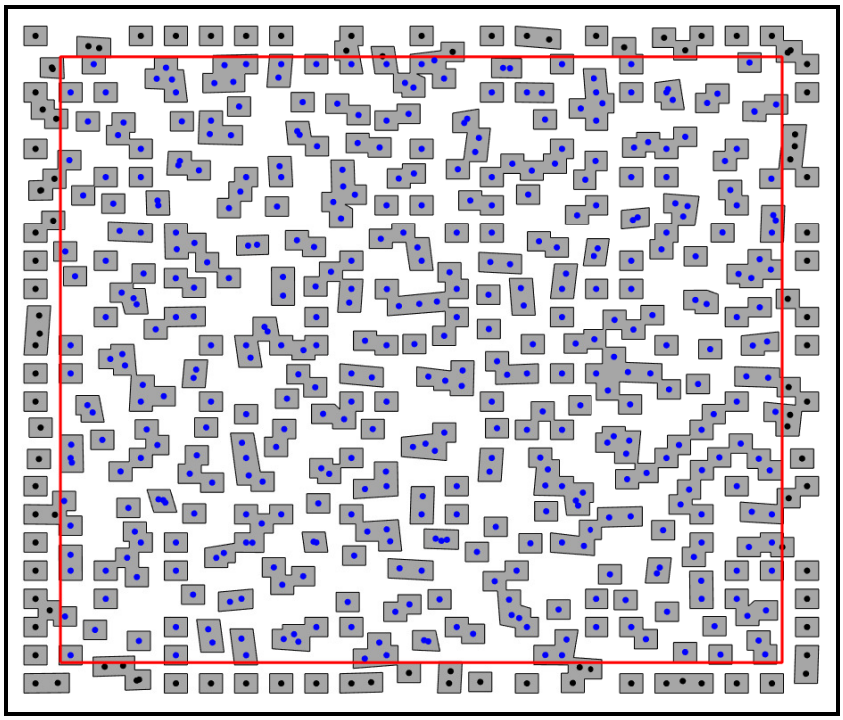
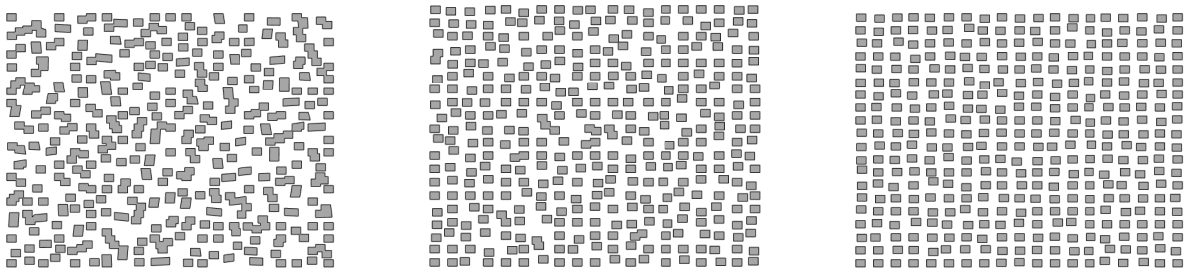


Figure 3.3: The determination of regular points and strictly neighboring points. The red line represents the inner boundary which separates the regular points (shown in blue dots) from the strictly neighboring points (shown in black dots).

Figure 3.4 shows three examples of configurations which all have the same porosity but different Mermin order values. Here, you can see the effect that changing the Mermin order value has on the overall configuration.



(a) $\phi = 0.65, \chi_4 = 0.4$ (b) $\phi = 0.65, \chi_4 = 0.7$ (c) $\phi = 0.65, \chi_4 = 0.9$

Figure 3.4: Examples of three different configurations, all with the same porosity Φ value but different spatial order χ_4 values. The spatial order progresses from the lowest order (left) to the highest order (right).

Although there is very little research on common spatial order values in city texture, the work done by Sobstyl [43] that introduced the use of this method for quantifying urban textures found that local Mermin order parameters in cities range between 0.5 – 0.9. Similar to our integration of porosity in our synthetic cities, we incrementally change the Mermin order value used to generate the synthetic cities in order to understand its effects. The range of Mermin order values tested is from 0.3 – 1. However, it should be noted that as we reach lower porosity values past 50%, we start to create packed arrangements which cannot contain as much randomness or disorder as more porous arrangements. The minimum possible order values change with respect to porosity, with only the configurations of 50% to 90% porosity reaching the most disordered states of $\chi_4 = 0.3$. This causes an empty zone in our configuration landscape of unrealistic or computationally expensive models which are avoided.

3.4 Other Model Parameters

3.4.1 Length/Width of Channel

In this study, we ran multiple tests to determine the optimal channel length that ensured uniform flow. Since the ratio of width to length of the channel is of importance in this case, the width of the rectangular channel was kept constant at 500 meters while the length was changed incrementally, ranging from 1,000 meters (1:2) to 10,000 meters (1:20). These convergence tests will be discussed in further detail in Chapter 5. The results from the tests showed that for the most complex configurations that are being considered for this study, which are densely packed and highly disordered, we reach uniform flow with a channel length of 10,000 meters. This domain size of 500 meters wide and 10,000 meters long will be applied for all configurations.

3.4.2 Slope

When studying hydrodynamic behavior within a domain, one parameter that should be considered is the bed slope, since it can have a dramatic effect on flow depth, water surface profile, and overall flow classification. In order to comprehensively understand the behavior of floods in the configurations created, we have done an analysis with slopes of -0.1% and -1% in the left-to-right direction. Simplifying the channel bed to a uniform slope is a strong assumption, but it is motivated by the intention to focus our analysis on the influence of the configurations alone. The slope is represented by assigning a bed elevation at each node with the following value:

$$z(x) = \alpha(x_{max} - x) \tag{3.5}$$

Where x_{max} is the length of the channel bed and α is the slope value.

3.5 Hybrid Reverse Monte Carlo Algorithm

The algorithm used to produce synthetic city configurations follows the basic structure of the hybrid reverse Monte Carlo (HRMC) algorithm that was first introduced by Opletal et. al. [32] as a way to improve upon the modeling of amorphous carbon structures. The HRMC method was an adaptation of the Reverse Monte Carlo method introduced by McGreevy et. al. [26] in 1988 which produced models by implementing constraints to a random generation process. While the RMC algorithm is a robust and widely used tool, many studies since then have improved upon it by implementing additional constraints such as the constrained reverse Monte Carlo (CRMC) and HRMC algorithms [8]. The HRMC algorithm achieves this improvement with the incorporation of an interaction potential term that alleviates the problem of unrealistic or physically improbable configurations from RMC alone [32]. In this study, the HRMC algorithm is adopted and the constraint used is the user-defined Mermin spatial order value. Utilizing this method allows for efficient and accurate generation of synthetic cities.

Before the HRMC algorithm was applied to this study, the original RMC method was used. In this adaptation of the original reverse Monte Carlo algorithm, the desired porosity Φ' and spatial order χ_4' are defined, and the initial configuration starts with a two dimensional ordered lattice. It is important to note that the initial point configuration may also be generated at random [26], but for our study we begin with an ordered lattice. The number of points N in the initial configuration is determined by the user-defined porosity value, where each point will signify a building footprint covering a certain area, and does not change throughout the process. The initial spatial order $\chi_{4,old}$ is calculated for this configuration, which equals one because it is in a completely ordered state at the beginning of the simulation. One point is randomly chosen and moved in a random direction and distance, and the new spatial parameter $\chi_{4,new}$ is calculated. These are both compared to the desired value of χ_4' and an error is calculated as follows:

$$G_{old}^2 = (\chi_{4,old} - \chi_4')^2$$
$$G_{new}^2 = (\chi_{4,new} - \chi_4')^2 \tag{3.6}$$

If $G_{new}^2 < G_{old}^2$, then the new configuration is accepted, it becomes the initial configuration and the value of $\chi_{4,new}$ becomes the new $\chi_{4,old}$. If not, then the move is invalid and the old configuration and $\chi_{4,old}$ is retained. A new point is moved and the process is then repeated. This process is continued until the error value reaches the tolerance level in which the spatial order has reached close enough to the desired value. By following this algorithm, a final configuration is achieved with the desired spatial order and porosity values.

Although the reverse Monte Carlo method is a fairly robust and reliable algorithm, problems have been reported on its inability to avoid unrealistic configurations. In the production of atomic structures, highly strained, high energy configurations were an intrinsic consequence of the RMC algorithm [32]. In this study, building configurations created with the RMC method featured some areas in which buildings got extremely close to each other, causing issues that arose later on with the meshing process and overall efficiency of flood simulations. This unnecessarily complicated the domains without providing any valuable insight. To address these problems, Opletal et. al. [32] introduced the hybrid reverse Monte Carlo system which adds an energy constraint to the algorithm to help avoid unrealistic configurations. This energy term is included in the calculation of the configuration's error value whenever a move is made that leads to an unrealistic configuration. The energy term is accounted for as follows:

$$G^2 = E + (\chi_4 - \chi_4')^2 \tag{3.7}$$

and it automatically causes the algorithm to reject this configuration, since $G_{new}^2 > G_{old}^2$. Since the energy term is simply a binary to accept or deny a move, the value of the term is

arbitrary as long as it is enough to ensure $G^2_{new} > G^2_{old}$. In this study, the energy term is determined with respect to the one point j that has been moved and is defined as follows:

$$E = E_j = \sum_{k=1}^{N} E_{j,k} \tag{3.8}$$

$$E_{j,k} = \begin{cases} 1000 & d_{min} < d < d_{max} \\ 0 & otherwise \end{cases} \tag{3.9}$$

where d is the distance between the centroids of building j and each surrounding building k, while d_{min} and d_{max} are the conditions that would determine if two buildings are too close to each other. The minimum restriction is implemented to allow for buildings to get close enough so that they can merge together, while the maximum restriction is implemented to define the maximum distance that would require an energy penalty; this is illustrated in Figure 3.5.

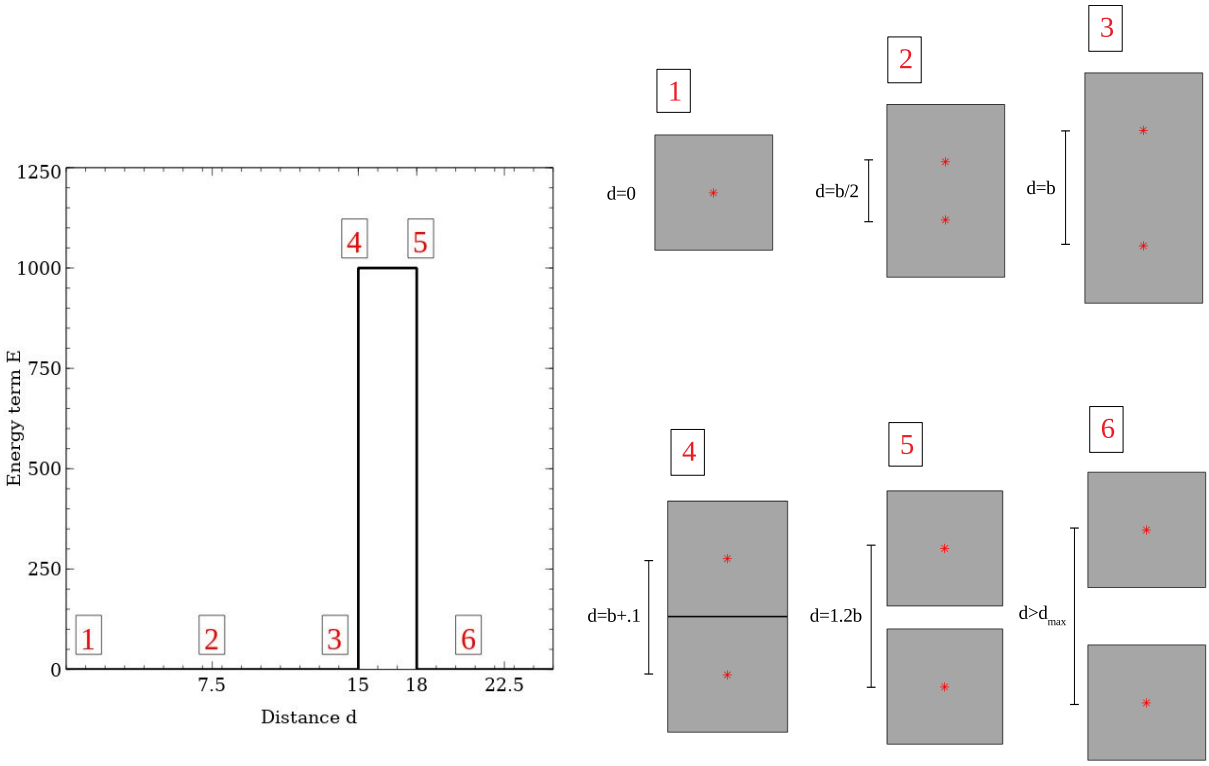


Figure 3.5: The energy graph (left) shows the value of the energy term E with respect to the distance between buildings d. Six different instances of building spacing are shown (right) and their corresponding energy values can be determined from the graph. Buildings that are close enough to merge can do so without an energy value. When the distance between buildings is $b < d < 1.2b$, the energy term jumps to 1,000. When $d > 1.2b$, the energy term returns to zero.

The value for d_{min} is the building side length b, since the maximum distance between two buildings at which they are still physically able to be combined is the building length b. The value for d_{max} is $1.2b$, since for this study the minimum acceptable distance between two buildings is 20 percent of the building length b. This measure was determined through an analysis of the meshing between buildings; it was discovered that if the distance between buildings was less than $1.2b$, the mesh would become very fine and it would increase the computation time for the flood simulations. Since our analysis does not focus on the smaller

scale behavior of fluid flow in these types of areas and instead observes behavior on the kilometer scale, we avoid these cases with our energy penalty.

The entire hybrid reverse Monte Carlo process for this study is represented as a workflow diagram in Figure 3.6. As the algorithm iterates through this process, the arrangement and corresponding Mermin order value converges towards the desired final configuration. The number of steps that it takes to reach the desired outcome is dependent on many factors: the number of total buildings in the domain, the desired Mermin order value, the tolerance for an acceptable final configuration, and also the inherit randomness of the algorithm. This convergence process can be seen in Figure 3.7.

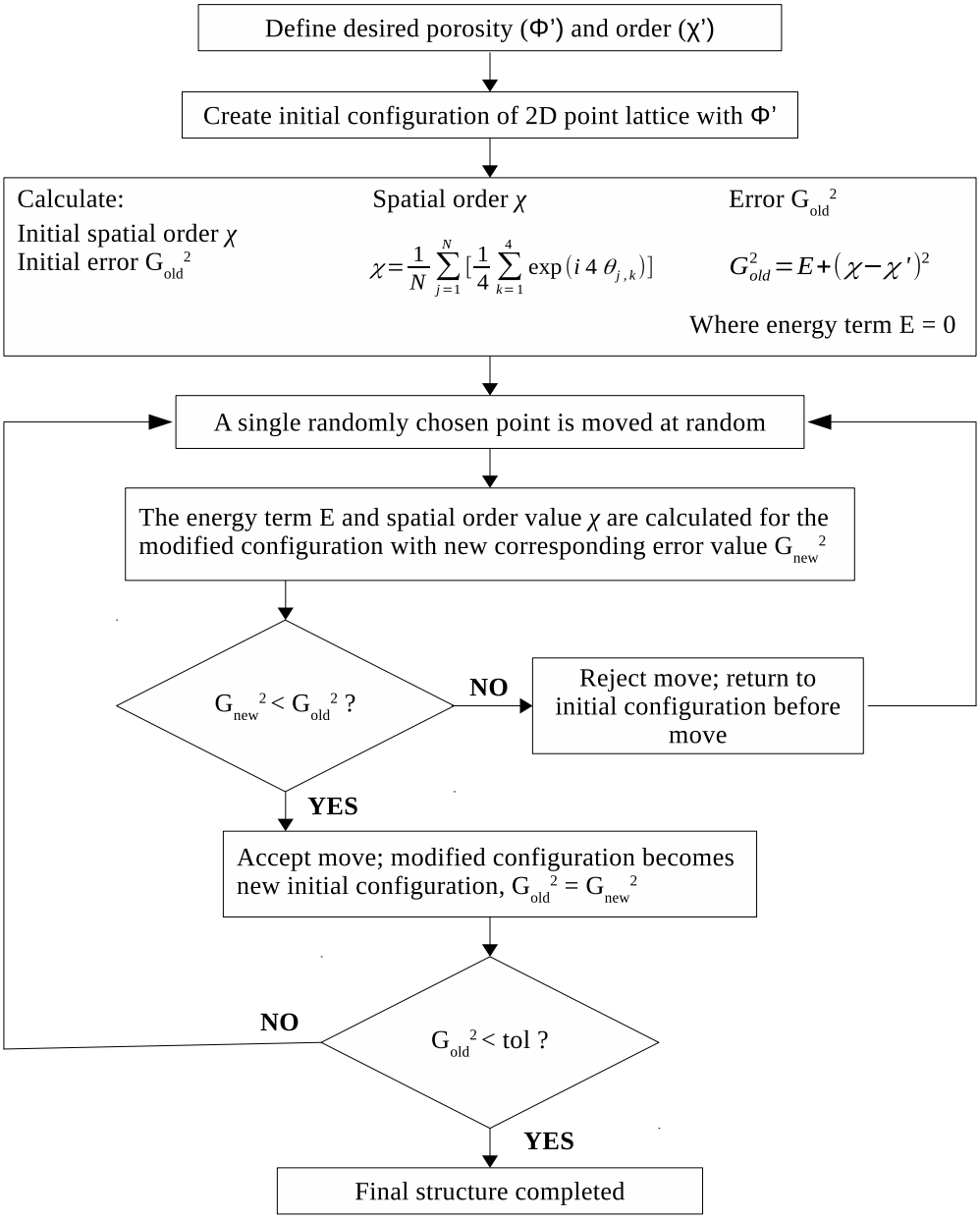


Figure 3.6: The workflow of the HRMC process as it progresses through each step. At the top, the algorithm begins with user-defined input parameters. An initial configuration is created and the initial spatial order and error values are calculated. The HRMC looping process then begins. A point is moved at random and the algorithm proceeds through steps to determine if the move will be accepted or rejected. If rejected, the move is undone and another point is moved. If accepted, the move becomes part of the configuration and another point is moved. This process continues until the error value drops below a certain tolerance.

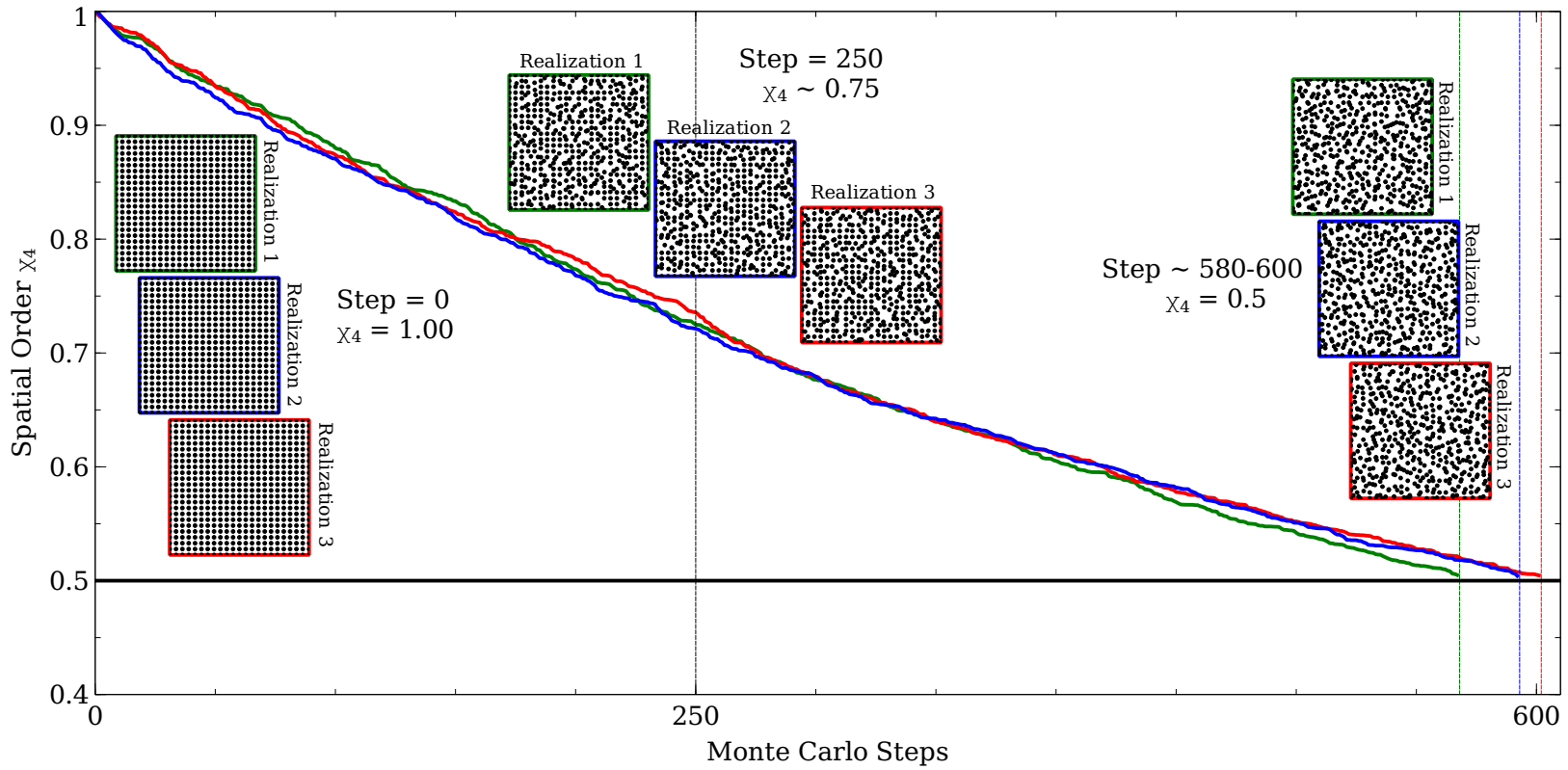


Figure 3.7: Three different realizations that converge from an initial 2D lattice of points to the final configuration with desired Mermin order value. Each realization maintains a porosity of $\Phi = 0.55$ and converges towards a desired Mermin order value of $\chi_4 = 0.5$. They all follow somewhat the same path and reach convergence at around the same step, but create three different realizations.

3.6 Model Refinement

3.6.1 Merging Overlapping Buildings

While the modeling process itself is robust and automated, some refinement must be done after the completion of the HRMC algorithm in order to improve upon the model. These improvements are generally done to simplify the geometries for efficiency in the mesh generation and simulation steps. The HRMC process allows for the points to come close enough that the resulting buildings overlap. In order to adjust this to more realistic geometries, the overlapping buildings are merged together into one as seen in Figure 3.8.

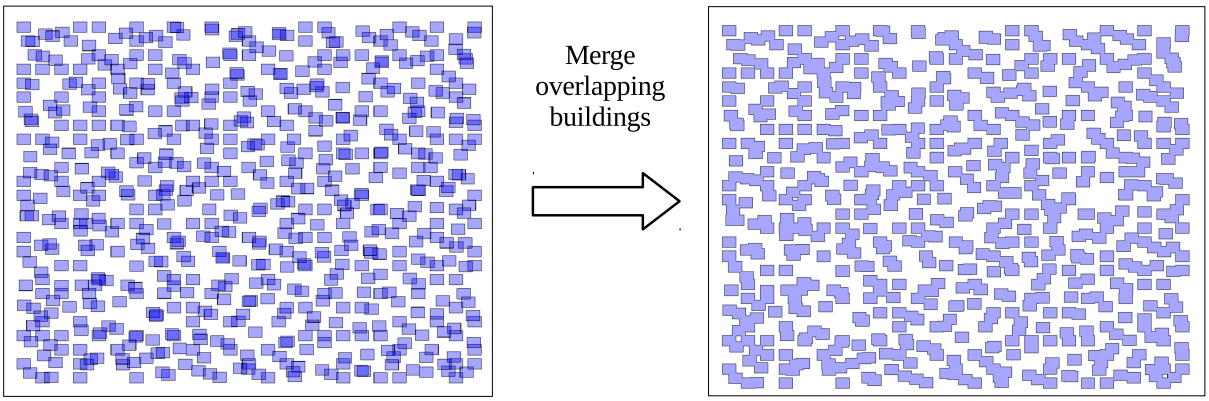


Figure 3.8: Overlapping buildings are merged together to improve the geometry.

These merged buildings are more common in configurations with lower Mermin order and porosity levels. After merging overlapping buildings, all holes are removed. Also, edges and verticies that over-complicate the geometries and create meaningless concavity, also known as outliers, are removed to smooth building edges. The removal of both holes and outliers can be seen in Figure 3.9.

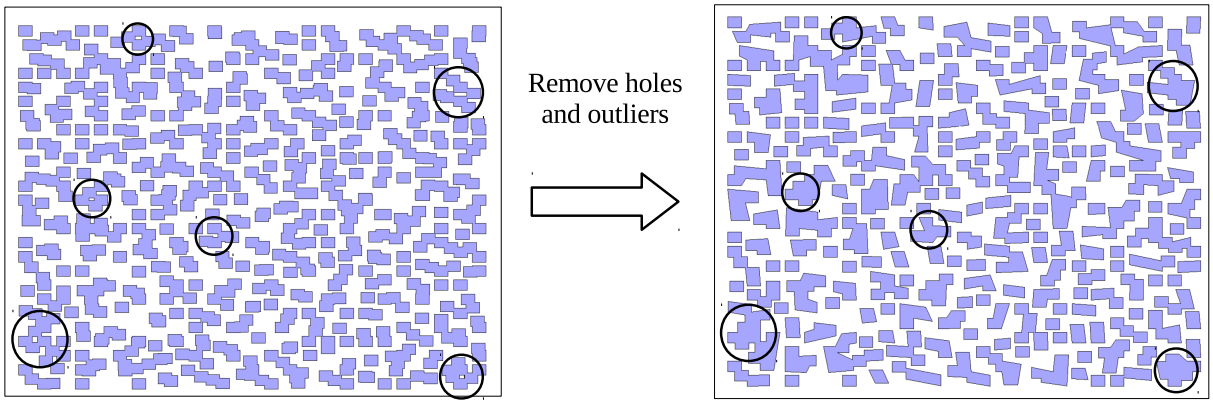


Figure 3.9: Holes and outliers are removed from the configuration. Holes are formed during the generation and merging processes and are removed because they are of no significance to the model. Outliers, which create meaningless concavity to the geometry, are also removed.

3.6.2 Expand Configuration to Channel Length

Due to the nature of the HRMC process, larger configurations with more points take exponentially longer to converge. This presents a problem when dealing with models on the scale of this study. As previously mentioned, the domain is extremely long in order to reach uniform flow and poses an issue with long inefficient run times. To bypass this issue and dramatically reduce convergence times for the HRMC process, a method of convergence and duplication can be used. McGreevy et. al. [26], when discussing calculation details of the reverse Monte Carlo method, explained that convergence for large configurations can be sped up by starting with a smaller set of points, such as a subset of the original set, and allowing it to converge before duplicating it to the intended size. This larger configuration of duplicated subsets in turn can be used as a starting point for the next HRMC process. This method is adopted in this study to increase the efficiency of the HRMC method. Since the size of the channel is 500 meters wide and 10,000 meters long, a subset of initial points that make up a 500 m x 500 m domain is extracted. The HRMC process is performed on this subset and when it is completed, it is duplicated along the x- dimension 20 times to fill the entire channel. Although the original HRMC algorithm then takes this configuration and

uses it as an initial configuration for another HRMC process, we determined that the spatial order and porosity of the newly generated configuration is close enough to the subset that another round would be trivial and would not improve the configuration by much. Figure 3.10 illustrates this duplication.

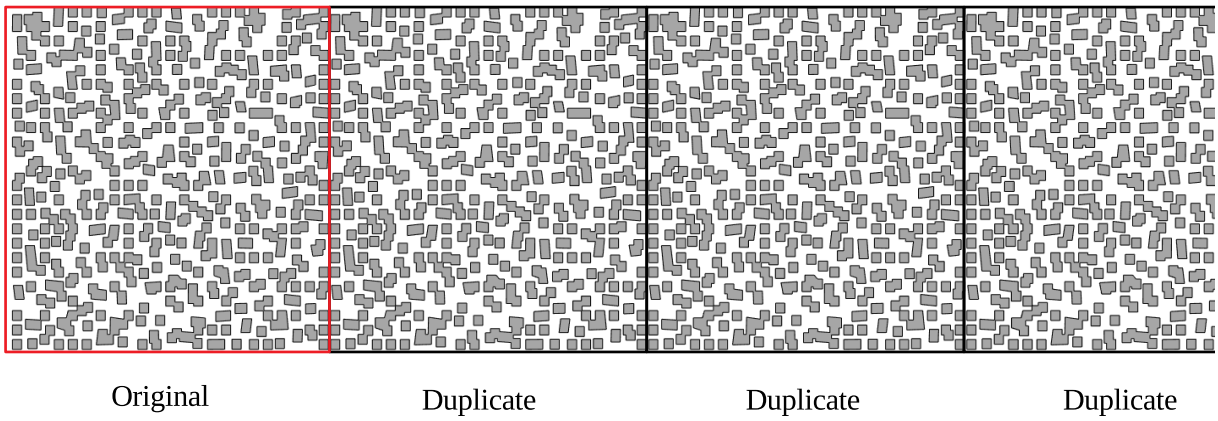


Figure 3.10: The final configuration which duplicates the 500m x 500m arrangement to fit the entirety of the 10,000 meter channel

3.7 Computational Grid/Mesh

A critical component to analyzing models with the finite volume method is producing a refined and high-quality two dimensional mesh that accurately represents the domain, since the governing shallow water equations are solved on this grid of triangular cells. By using the flood simulation algorithm BreZo, we are able to utilize an unstructured grid of cells which allows for the meshing of more complex geometries and irregular boundaries [6]. The algorithm is also able to perform localized grid refinement which ensures an accurate and

grid-independent solution. As described earlier, our algorithm for synthetic city generation includes many steps to simplify our domain for improved efficiency during the meshing and calculation stages. These adjustments include the rejection of buildings that are generated too close to each other, the merging of overlapping buildings, and the removal of holes and outliers in the final geometry.

To prepare the computational mesh, we employed the C program Triangle, created by Jonathan Richard Shewchuk at the University of California, Berkeley [41] for two dimensional mesh generation. This program creates exact Delaunay triangulations with guaranteed quality due to its robust refinement algorithm and small angle restrictions. One important feature to this program is the inclusion of user-defined holes, which is used in this study to represent buildings. The treatment of buildings in our flood inundation study is dictated by the building-hole (BH) method introduce by Schubert and Sanders [40] to effectively integrate geospatial data for unstructured mesh generation. With this method, building footprints define mesh holes, or interior wall boundaries, which analytically imitates the effects of buildings on flood hydrodynamics [40]. The edges of the holes, or walls of the building, are treated as no-slip boundaries [40]. This method was chosen above many of the other methods for representing spatial data in meshes, like the building block method, because it was found to better preserve the geometry of buildings and maintain better accuracy in relatively coarse meshes.

When generating a mesh, we need to strike a balance between computational accuracy and efficiency by defining mesh sizes and number of nodes. To ensure a high quality mesh, the maximum area of a cell is constrained to one tenth the size of a building for the first 500 meters of the domain. The remaining domain is meshed with a maximum cell area of one quarter the area of a building. This mesh quality difference is implemented because of the second important mesh attribute: number of nodes. Our main analysis in this study pertains to the flood heights near the inlet of the domain, and so the mesh is the finest

at this area. Although we need to mesh and analyze the remainder of the domain as well, the intricacies of the analysis are less important in this area. We also justify this decision by the fact that BreZo, which is the finite-volume flood modeling algorithm that will be described later, performs mesh refinement when needed. This method drastically reduces the number of nodes within our mesh by 40 percent, increasing the computational efficiency while maintaining a high level of accuracy. With this method the average number of nodes in our configurations of a domain size 500 m x 10,000 m ranges between 70,000 and 130,000. This mesh can be seen in Figure 3.11.

3.8 Summary

The generation of synthetic cities provides controlled environments in which flood inundation models can be tested and understood. By limiting the number of independent variables, direct links can be drawn between our parameters and the resulting flood behavior. Introducing the Mermin order parameter has allowed for control over the spatial order of these synthetic cities. Through the Monte Carlo method and subsequent model refinement, we were able to produce hundreds of synthetic cities that are ready for flood inundation tests.

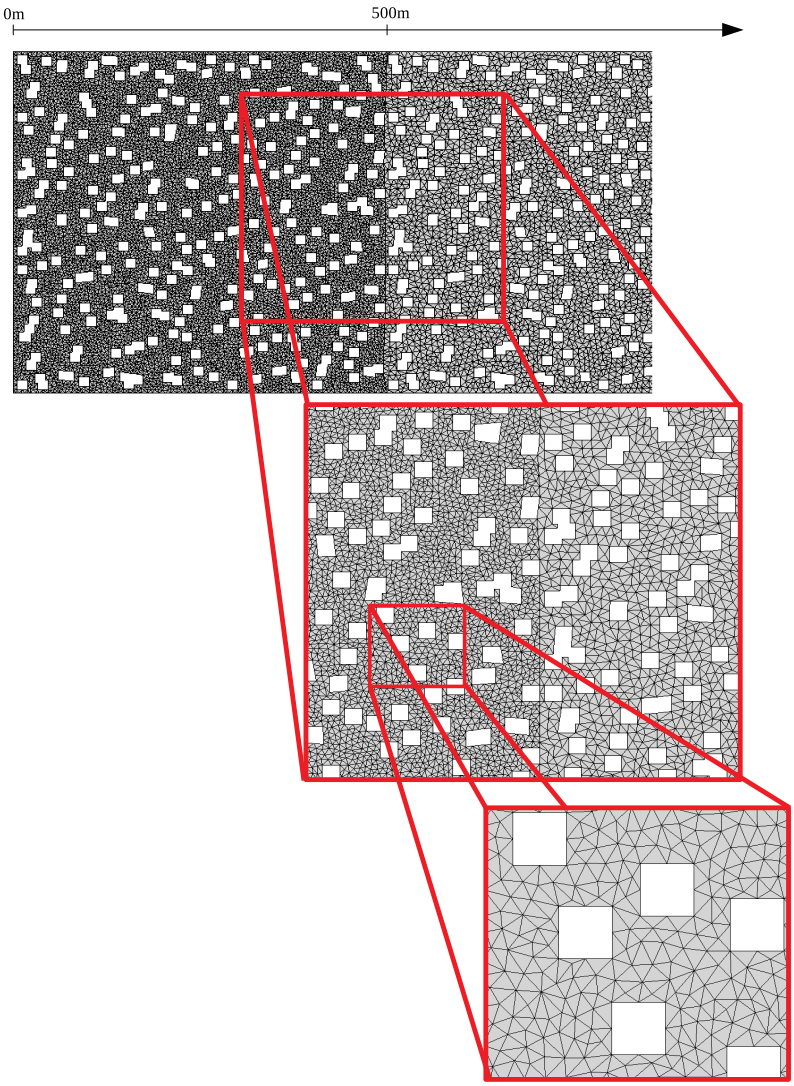


Figure 3.11: Computational mesh that is created from the synthetic cities. The mesh is finer at $x < 500m$ and is coarser at $x > 500m$. The holes in the mesh represent buildings, following the building hole method. The area of a mesh cell is 10% the size of the buildings in the first section, and 25% the size of the buildings in the second section.

Chapter 4

BreZo: An Adaptive Godunov-Based Model for Flood Simulation

4.1 Introduction

After the generation of the synthetic city models, shallow water flow analysis must be done to study the flow of presumed floods through the configurations. Extensive work has been done in the development of flood inundation models and its application in urban flood modeling. Teng et. al. [46] compared several 1D, 2D, and 3D hydrodynamic models for simulating flood inundation and presented many advantages and limitations for each. In most instances, two-dimensional shallow water approximations prove adequate when compared to more complex three-dimensional models [46], and are therefore utilized in this study. Of the available models, this study utilizes a Godunov-based finite volume model called BreZo [6, 7, 38], developed by Brett Sanders from the University of California Irvine and Lorenzo Begnudelli from the Univeristy of Ferrara. This model solves the shallow water

equations while presenting increased efficiency and minimized energy error compared to previous Godunov-based models, proving it to be the best option for this study.

4.2 Methodology

The BreZo model that is used here has seen many iterations throughout the years to improve accuracy and efficiency. The original iteration was developed by Bradford and Sanders [9] and has been improved upon through multiple optimization studies. At its core, a fixed grid finite volume Godunov-type scheme is utilized to solve the shallow water equations for unsteady, two-dimensional flow along a domain. Godunov-type schemes are very popular because of their ability to accurately model both subcritical and supercritical flow, capture hydraulic jumps, and conserve mass [7]. The shallow water flow equations that are directly discretized for a finite control volume are in integral form [5]. This scheme solves the SWEs in integral form in computational cells, ensuring the conservation of mass and momentum in each cell. The integral form of the equations are as follows:

$$\frac{\partial}{\partial t}\int_{\Omega}\mathbf{U}d\Omega + \oint_{\partial\Omega}(\mathbf{F}dx - \mathbf{G}dy) = \int_{\Omega}\mathbf{S}d\Omega \tag{4.1}$$

$$\mathbf{U} = \begin{pmatrix} h \\ uh \\ vh \end{pmatrix}; \qquad \mathbf{F} = \begin{pmatrix} uh \\ u^2h + \frac{1}{2}gh^2 \\ uvh \end{pmatrix}; \qquad \mathbf{G} = \begin{pmatrix} vh \\ uvh \\ v^2h + \frac{1}{2}gh^2 \end{pmatrix};$$
$$\mathbf{S} = \begin{pmatrix} 0 \\ -gh\frac{\partial z_b}{\partial x} - c_D u\sqrt{u^2+v^2} \\ -gh\frac{\partial z_b}{\partial y} - c_D v\sqrt{u^2+v^2} \end{pmatrix} \tag{4.2}$$

where Ω represents a 2D domain with a boundary of $\partial\Omega$; h = flow depth; u, v = depth-averaged velocities in the x- and y- direction, respectively; z = bed elevation; and c_D = bed drag coefficient which is calculated using the Manning coefficient n_m as follows: $c_D = gn_m^2 h^{-1/3}$.

Unstructured grids of triangular cells are utilized in this model's finite volume scheme because of its ability to mesh complex domains with irregular boundaries easily using Delaunay triangulation [6]. As mentioned before, the shallow water equations are solved at each finite cell as piecewise solutions that are discontinuous at the cell faces [7]. Then reconstruction is done to understand the fluxes at these interfaces. Due to the unstructured nature of the grid, organization of the neighborhood of cells, nodes, and faces at each point is crucial to the success of this model. A robust integer mapping system of arrays is implemented to define the neighborhood of each cell. Begnudelli and Sanders [6] explain this integer mapping system extensively and it is used for the advancement of the model in the next steps.

The latest version of BreZo uses an adaptive first-order accurate scheme that is paired with a second-order accurate terrain model which allows it to achieve close to second-order convergence with grid refinement [7, 38]. This is an improvement from a previous model which progresses with a second-order accurate time integration where a predictor is computed at the $n + 1/2$ time step and then a corrector is computed at a time step of $n + 1$. By adopting the new first-order accurate method, computational effort is minimized due to the avoidance of calculating a predictor solution and slope limiting gradients while maintaining high accuracy [7]. Also, a local time stepping scheme was adopted to improve run-time efficiency [37]. The updated equation for BreZo to advance the solution in time can be written as:

$$\mathbf{U}_j^{n+1} = \mathbf{U}_j^n - \frac{\Delta t}{A_j} \sum_{k=1}^{3} (\mathbf{F}_\perp)_k^n \Delta s_k - \Delta t (S_{o_j}^{\ n} + S_{f_j}^{\ n+1/2}) \tag{4.3}$$

where j is the cell index, n is the time level, k is the local edge index, $\mathbf{U}$ is the analytical solution to the shallow water equations $\mathbf{U} = (h\ uh\ vh)^T$, S_o and S_f are the source terms, and $\mathbf{F}$ is the flux term

$$\mathbf{F} = \begin{pmatrix} hu_\perp \\ huu_\perp + \frac{1}{2}gh^2 \cos\Phi \\ hvu_\perp + \frac{1}{2}gh^2 \sin\Phi \end{pmatrix} \tag{4.4}$$

where $u_\perp$ is the velocity perpendicular to the cell face and Φ is the angle between the face normal vector and the x- axis. Roe's approximate Riemann solver is used to compute the flux vector $\mathbf{F}$ in Equation 4.4 at the cell faces. Since the finite volume scheme produces discontinuities at the cell edges, reconstruction of flow variables to the left and right of these faces is required. This reconstruction is done using the monotone upstream scheme for conservation laws (MUSCL). Bradford and Sanders [7] found that the most efficient and accurate application of either primitive or conservative variables to this reconstruction depended on the circumstances. In order to minimize errors that are intrinsic to Godunov-type models due to this discontinuity reconstruction uncertainty, BreZo implements an adaptive scheme to the Godunov-based model. This adaptive scheme determines whether to select primitive or conservative variables for variable reconstruction depending on the local Froude number [7, 38]. When the flow is subcritical ($Fr < 0$), the depth h and velocities u and v are assumed cell-wise constant for reconstruction purposes. On the other hand, when the flow is supercritical ($Fr > 0$), the free surface height η and discharge per unit width $p = uh$ and $q = vh$, are instead considered cell-wise constant.

Another issue that the BreZo model overcomes is the problem that arises from partially submerged cells at the wet/dry boundary. A partially submerged cell is a cell in which at least one, but not all, of the cell nodes are submerged. The depth in this type of cell is badly represented because finite volume schemes relate the average depth in a cell with the cell centroid, filling the entire cell with this smaller averaged depth. This also leads to the

artificial spreading of water into adjacent cells, diffusing the wet/dry boundary [9]. To avoid this issue, a distinction is made between the free surface elevation η and the depth h at the centroid, and a relationship between them is determined. In completely wetted cells, this relationship is simply $\eta = h + z_c$, but it does not hold for partially wetted cells. Volume/free surface relationships (VFRs) are utilized in these cases to facilitate reconstruction of the free surface of partially submerged cells and determine the relationship between η and h.

4.3 Summary

The implementation of an accurate and computationally efficient flood modeling algorithm is crucial to the success of this study. By using the algorithm BreZo for this purpose, we are able to produce precise flood inundation models with ease using our previously generated synthetic cities. The results are then analyzed to study behaviors and trends that could present important information about the link between urban morphology and flood levels.

Chapter 5

Flood Inundation Results

5.1 Introduction

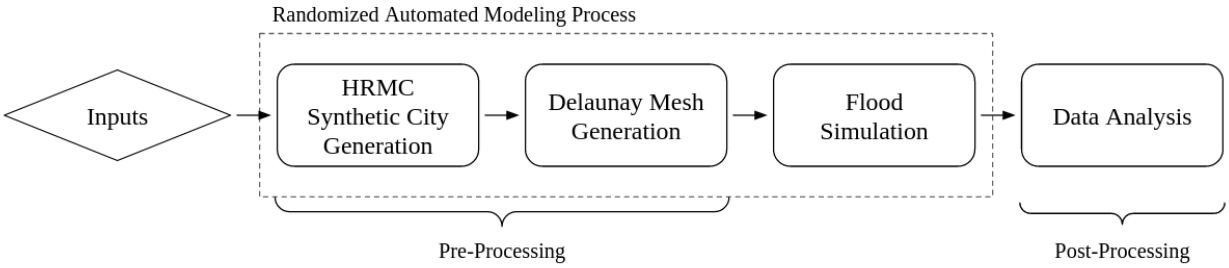


Figure 5.1: The basic workflow of this study progresses from synthetic city generation and meshing to the creation of inundated flood simulations and final data analysis. The three steps of synthetic city generation, mesh generation, and flood simulations are all automated processes that are done consecutively. These processes produce results that are analyzed to obtain trends.

In this chapter, we will present the results obtained from the synthetic city modeling and flood inundation processes. Convergence analysis is done to determine the optimal channel length for the model and the run time for the flood inundation. A subset of 20 configurations are presented here with their corresponding values of porosity and spatial order to illustrate

how these parameters affect the resulting model. Through dimensional analysis, we propose an equation that relates the parameters of this study to flood heights and we validate this equation with the Manning equation. By analyzing the resulting flood heights from our inundation models, we are able to quantify the relationship between configurations and their effects on flood behavior. This is used to further refine the proposed equation for flood heights. Through this process, we establish a compelling argument for the integration of porosity and spatial order parameters in the quantitative analysis of urban flooding.

5.2 Convergence Analysis

Applying assumed values of certain parameters to flood models can drastically affect the behavior of the models, as well as the accuracy of the results. Since this project introduces flood simulations that are unique due to the computationally generated synthetic cities, no previously assumed values for channel length and run time could be implemented in these tests. Due to the novelty of this project, optimal values for the length of the channel and the run time have to be established through convergence analysis. These two parameters are dependent on each other, since expanding the channel length will in turn lengthen the time needed to reach steady state, and so the analysis was done through iterations.

5.2.1 Channel Length

The length of the channel in which a flood is inundated is important because a long enough channel can achieve uniform flow. Uniform flow is a good reference case for studying external effects on flow behavior because the water surface is planar, the depth and velocity is the same at all sections along the flow, and the effect of gravity in shaping the flow can be ignored [44]. To determine the optimal channel length to achieve uniform flow, convergence

analysis was done by running multiple tests with a constant channel width of 500 meters and changing channel lengths. The optimal channel length varied depending on the porosity we were testing. Configurations with high porosity and high spatial order saw uniform flow at a shorter channel length than configurations with low porosity and order values. To keep the conditions constant among all simulations, we ran the analysis with a configuration of around 45 percent porosity and a Mermin order value of 0.5. The resulting water heights from these tests were then compared for each channel length in Figure 5.2. As we can see from the figure, shorter channels saw flood heights that were largely sensitive to the channel length. As we reach longer channels, these height variations begin to diminish. Although variations in flood heights were not completely diminished at a channel length of 10,000 meters, it was decreased enough to justify the application of this length to our studies.

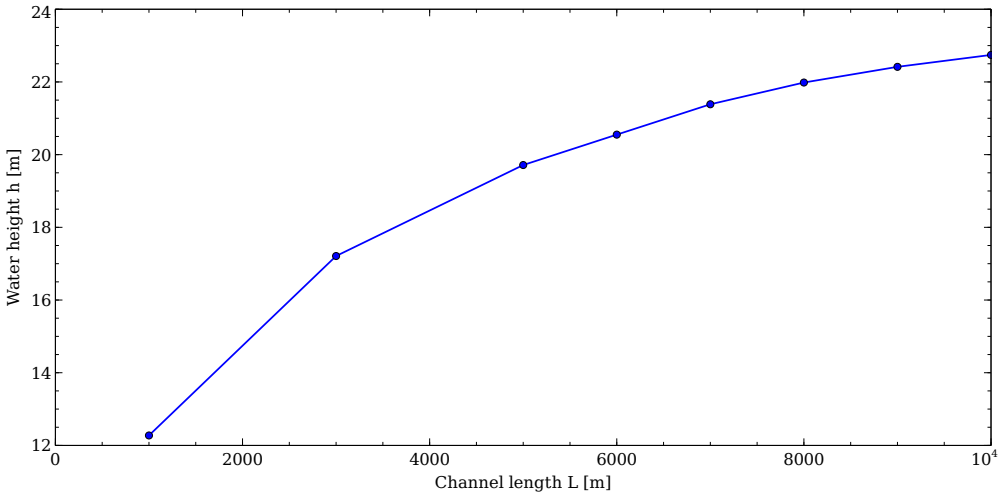


Figure 5.2: Resulting water heights from flood inundation models were determined with channels of different lengths, ranging between 1,000 and 10,000 meters long. These flood inundation models used a configuration with $\Phi = 0.45$ and $\chi_4 = 0.5$. We see that as the channel length increases, variations in water heights begin to diminish.

It should be noted that although extensive studies were done to achieve close to uniform flow for all simulations, some simulations did not achieve true uniform flow. Mostly, these were "worst case scenarios" of low porosity and spatial order. We also observed some inconsistencies and uncertainty during the analysis of our final results, which can be expected when dealing with a large number of simulations. For instance, plotting the flow profile that

resulted from an inundated configuration of $\Phi = 0.36$ and $\chi_4 = 0.55$ shows that this specific simulation did not reach completely uniform flow. From Figure 5.3, we can see that the free surface is not completely parallel to the channel bed, meaning that this model has not completely reached uniform flow conditions. Although this is the case for some of the more complex and low porosity cases, most other configurations reached closer to uniform flow with this channel length.

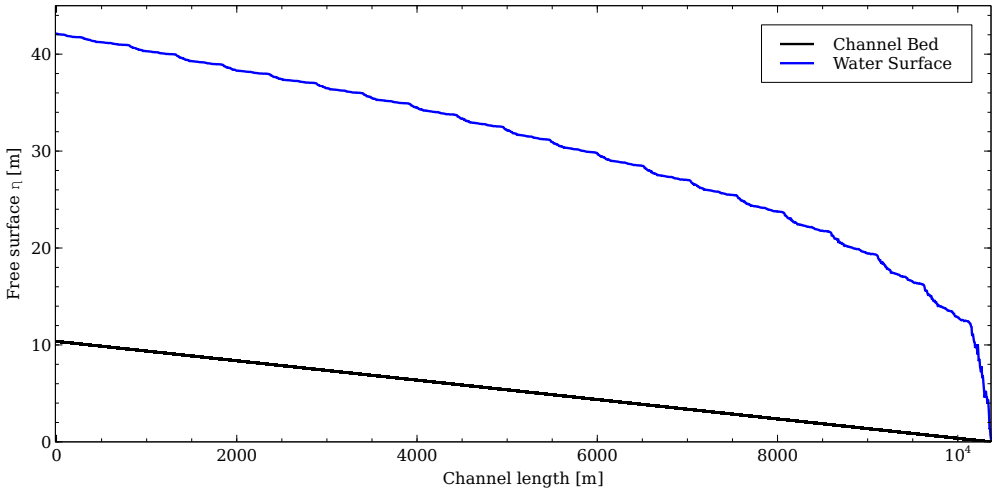


Figure 5.3: Flow profile of configuration with $\Phi = 0.36$ and $\chi_4 = 0.55$. This flow profile was created by averaging the water heights at each section of the channel length. The water heights seem to be inconsistent along the length of the channel, proving that true uniform flow was not achieved for this simulation.

5.2.2 Run Time

To ensure that the simulations had enough time to reach an equilibrium state, tests were done on the total time needed to run the flood simulations. This equilibrium state is desired because it ensures that the results taken at the last time step are not dependent on time. The total run time depends directly on the length of the channel because flows cannot reach equilibrium until they travel the entirety of the channel and reach the downstream boundary. Because of this, the tests for run time had to be done simultaneously with tests for channel

length. Each time the channel length was increased, the run time was also increased to ensure enough time was allocated for the flood to equilibrate. Similar to the channel lengths, the run time was also dependent on the configurations; lower porosity and spatial order values caused run times to become longer. Figure 5.4 shows the amount of time three different configurations took to reach equilibrium with a channel length of 10 km.

In order to increase efficiency in the flood modeling process, the run time of each simulation depended on the porosity value of the system. The three run time options are as follows:

$$t = \begin{cases} 10 & 0.7 < \Phi \\ 20 & 0.4 < \Phi < 0.7 \\ 30 & \Phi < 0.4 \end{cases} \tag{5.1}$$

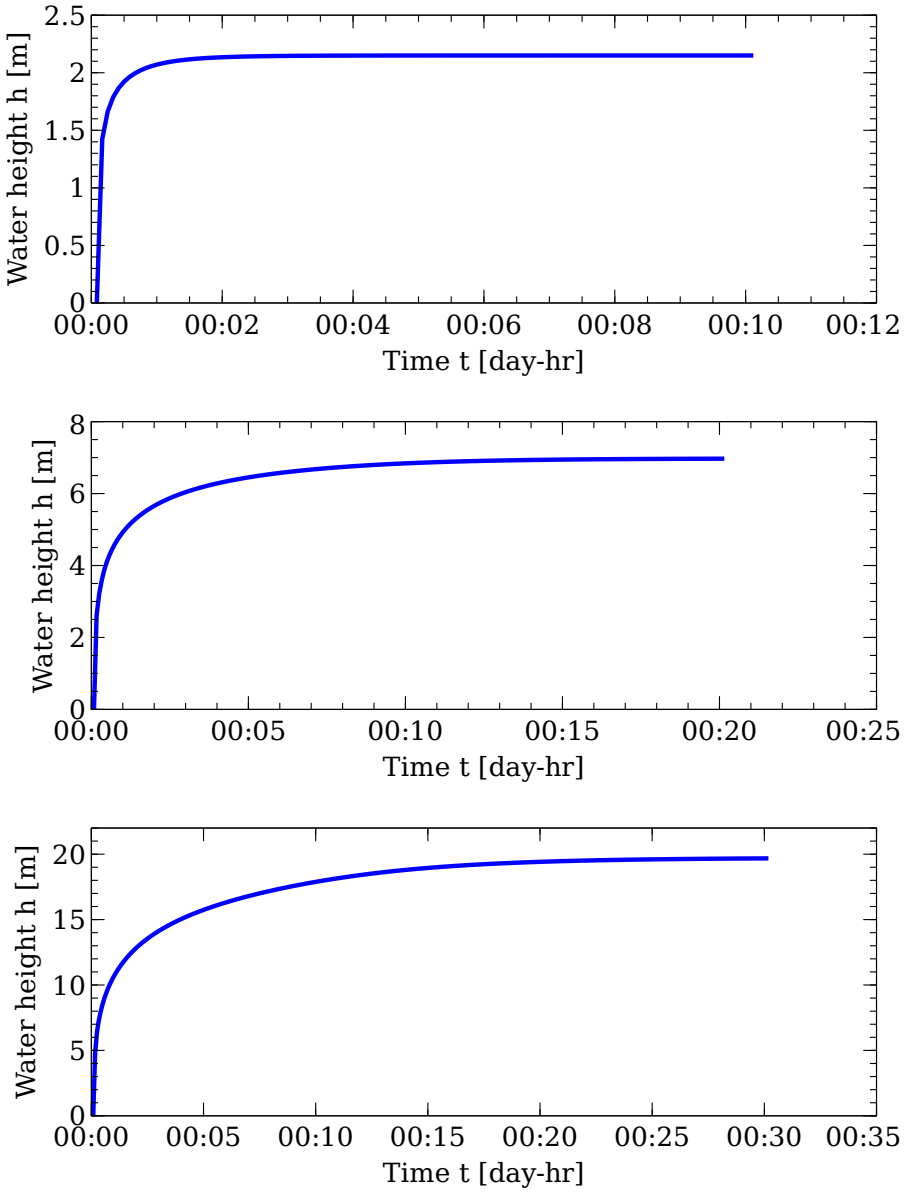


Figure 5.4: The total time for three different configurations to reach an equilibrium state. The first figure (top) is the total run time for a configuration with $\phi = 0.9$ and $\chi_4 = 0.5$. This configuration features high porosity and a moderate level of disorder, causing it to reach equilibrium fairly quickly at around 2-3 hours. The next configuration (middle) has a porosity $\phi = 0.7$ and spatial order $\chi_4 = 0.33$. It runs for a total of 20 hours and it reaches equilibrium a bit before that 20 hour mark. The final configuration (bottom) is the most complex with $\phi = 0.35$ and $\chi_4 = 0.7$. This configuration takes the longest to reach equilibrium with a total run time of around 30 hours.

5.3 Final Simulations

Using the synthetic cities that were generated from the previously described algorithm and implementing the necessary parameters that we determined through our convergence analysis, we were able to create inundation models that accurately determined the behavior of floods with respect to the changing city morphology. Because the entire process is automated and created to run only with the manual input of two parameters, we can easily generate hundreds of these simulations. For our study, a range of different porosity and spatial order values were used to create around 500 models. Of the 500 models that were created, there are around 160 different combinations of our two variables Φ and χ_4, and each combination was used to create three different realizations. Since the hybrid reverse Monte Carlo algorithm for synthetic city generation is probabilistic, each realization was created with the same values of porosity and Mermin spatial order, but observe a slightly different configuration. Figure 5.5 shows a subset of 20 different configurations of synthetic cities that were tested. Note that these figures show only the first 500 meters of the synthetic city, since this initial configuration is simply repeated as described in Section 3.6.2 to fill the entire channel. Table 5.1 lists the corresponding porosity and spatial order values for each configuration.

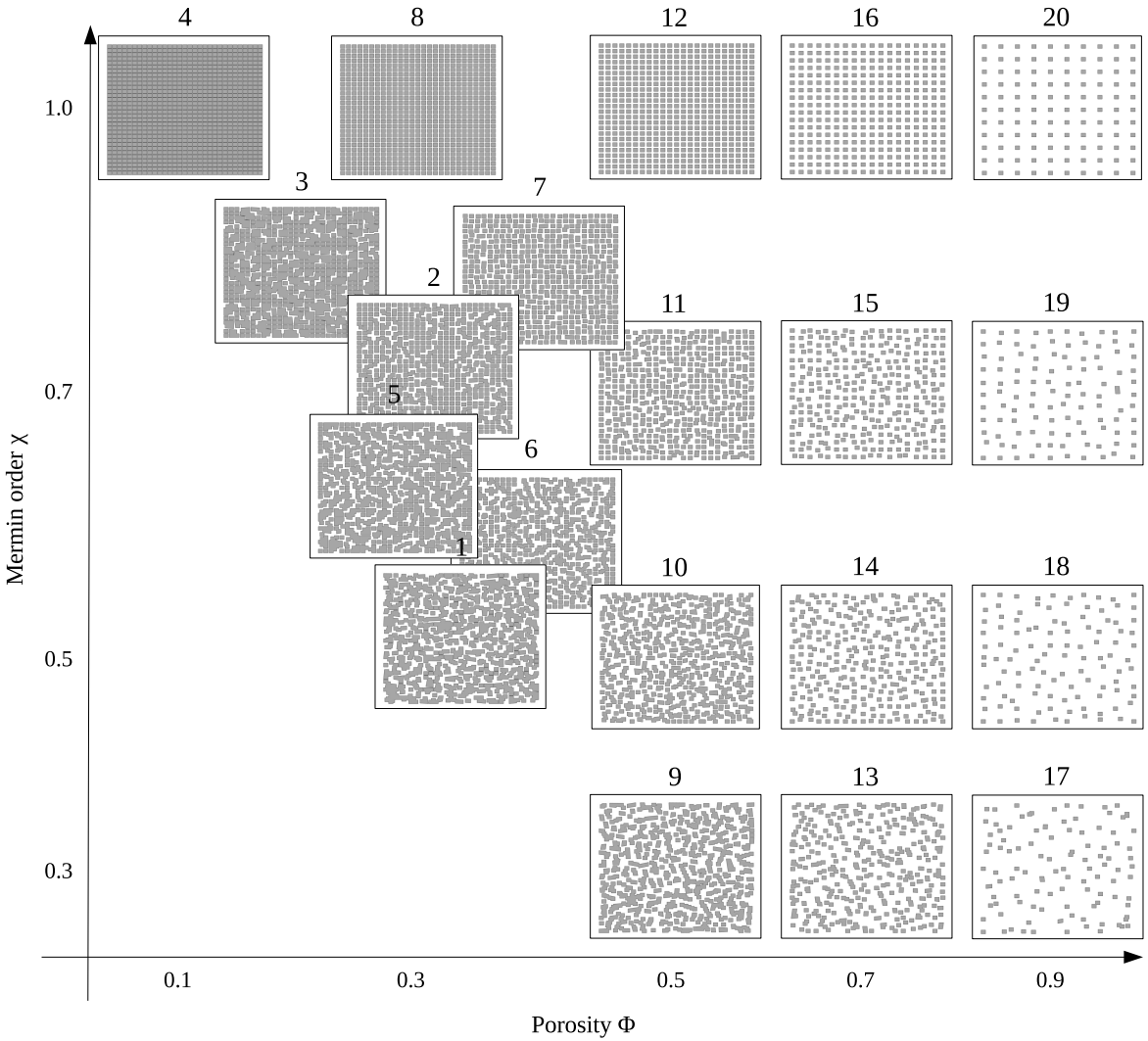


Figure 5.5: Images of 20 unique configurations that were created using the HRMC synthetic city generation algorithm with different input values of Φ and χ_4. The configurations are plotted according to their respective porosity and Mermin order values and Table 5.1 provides a full list of these values.

Table 5.1: Corresponding porosity and spatial order values for all configurations shown in Figure 5.5

Configuration	Porosity Φ	Spatial Order χ_4
1	0.36	0.55
2	0.34	0.77
3	0.21	0.86
4	0.11	1.00
5	0.29	0.69
6	0.40	0.64
7	0.42	0.86
8	0.32	1.00
9	0.50	0.33
10	0.49	0.51
11	0.52	0.73
12	0.50	1.00
13	0.69	0.33
14	0.72	0.51
15	0.71	0.72
16	0.71	1.00
17	0.90	0.33
18	0.90	0.50
19	0.90	0.70
20	0.90	1.00

All the configurations that were created in this study were then inundated with floods. The first case of inundation that we tested was a flood with an upstream volumetric flow rate of 1,000 m^3/s and a mild slope of 0.1%. The boundary condition at the upstream boundary is defined by the volumetric flow rate Q and the downstream boundary is defined by a dry boundary condition. The sides of the channel are considered no flow boundaries. Figure 5.6 illustrates these boundaries.

Since the buildings within the domain are represented as holes, the flood inundation is modeled through the meshed void space between buildings using the previously defined BreZo algorithm. Results of these simulations are analyzed with a focus on water heights. Since the flood modeling code BreZo solves the shallow water equations using the finite volume method, the resulting flood data can be presented as cell-centered values for each

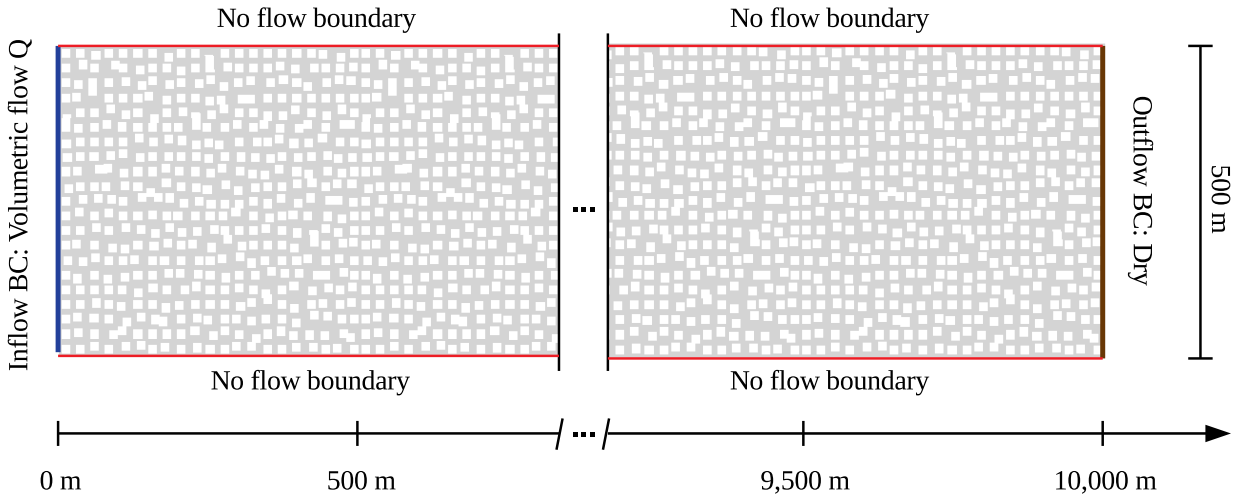


Figure 5.6: The boundary conditions for all simulations. The upstream B.C. is defined by a volumetric flow rate and the downstream boundary is defined by a dry B.C.

cell in the mesh which allows for the simple interpretation of results. Figure 5.7 shows the flow profiles of 60 configurations resulting from this inundation case. The 20 plots in the figure correspond to the 20 configurations from Figure 5.5 but each plot contains two more flow profiles; these are the second and third realizations of that specific combination of Φ and χ_4. By presenting all three realizations together, the variations in flood heights from slight changes in the configuration can be better understood.

As mentioned before in the discussion about convergence analysis, we were not able to reach uniform flow for some of the lower porosity models due to an insufficiently long channel. This can be seen in some of the flow profiles. Although this is the case, we believe that our analysis is still accurate enough to show some interesting trends regardless of the non-uniformity of the flow. We understand that our results contain a degree of uncertainty and error which should be considered throughout this discussion.

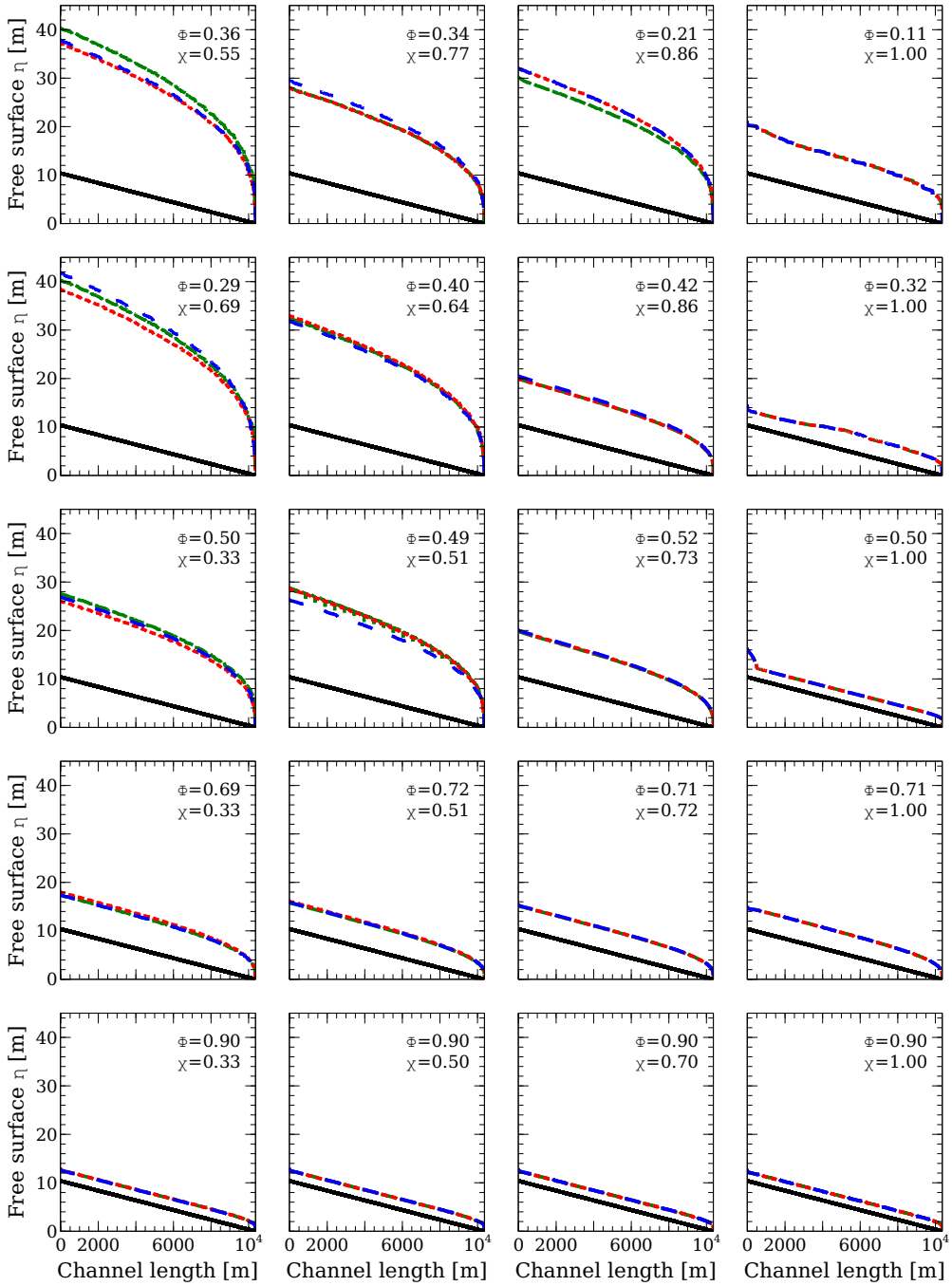


Figure 5.7: The flow profiles corresponding to the 20 configurations from Figure 5.5 along with their second and third realizations. Since many of the flow profiles overlap due to little to no height variations, the flows have been plotted with dashed lines. This allows for all profiles to show clearly, even if overlapping. The corresponding Φ and χ_4 values are displayed in the top right-hand corner of each plot.

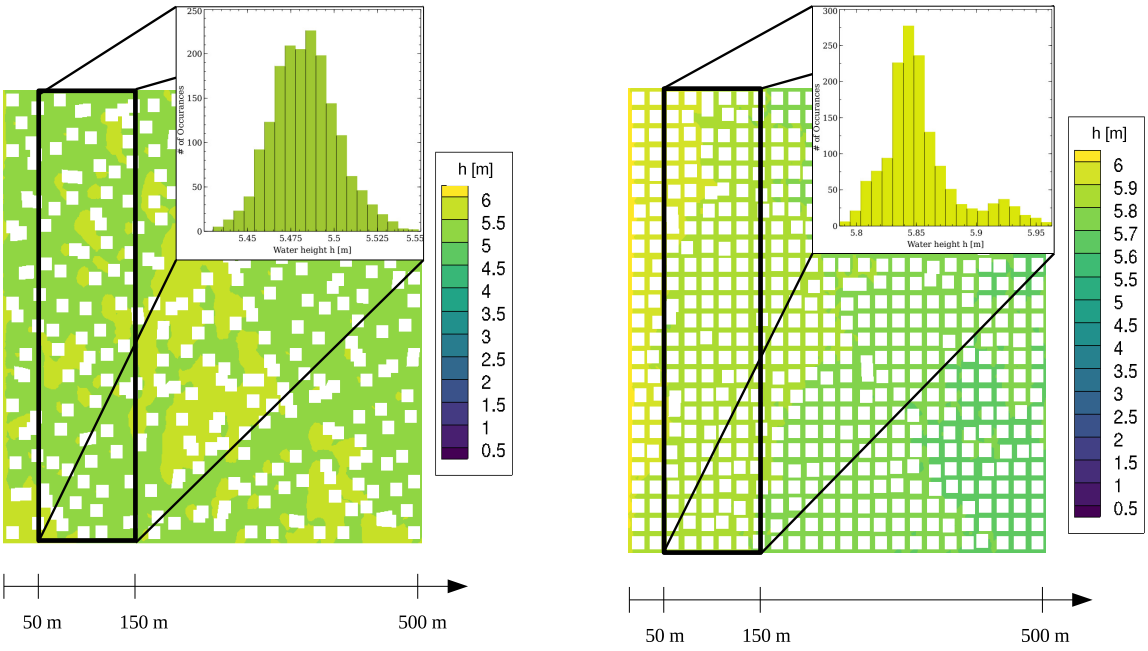
With simple back-of-envelope calculations, we determined that the flows for these simulations are all subcritical. This was done by calculating the critical depth of flow for a volumetric flow rate of 1,000 m^3/s with Equation 2.5. For instance, the critical depth for a configuration with porosity of 50% is calculated as follows:

$$q = \frac{Q}{w} = \frac{1000}{0.5 * 500} = 4$$
$$y_c = (\frac{q^2}{g})^{1/3} = (\frac{4^2}{9.81})^{1/3} = 1.177m$$

*Note: The width of the channel is taken as (0.5*500) because 50% of the domain is building coverage so the area of water flow is reduced by 50%.*

The resulting critical water depth is $y_c = 1.177m$, which is greater than the observed water depth for all configurations with $\Phi = 0.5$. Since $y > y_c$, this means that the flow is subcritical. With subcritical flow, a downstream boundary condition must be assigned. For all simulations in this study, we considered a dry downstream boundary.

The flood heights near the upstream boundary were averaged along the entire width of the domain and along 100 meters of the channel to obtain upstream water heights for each configuration. The specific area of interest for two different flood inundation models are shown in Figure 5.8. To avoid any irregularities at the upstream boundary, water height measurements were taken at a distance of 50 meters away from the boundary.



(a) $\Phi = 0.73, \chi_4 = 0.42$ (b) $\Phi = 0.46, \chi_4 = 0.95$

Figure 5.8: The flood heights are measured near the upstream boundary for two different configurations. The heights are taken from all points within the black bounding box which is 100 meters wide and spans the width of the channel. It is placed 50 meters away from the upstream boundary to avoid any irregularities.

5.4 Dimensional Analysis

Large sets of data can be better interpreted through dimensionless variables which simplify relationships and aid in the understanding of physical phenomena. By non-dimensionalizing variables, we can reduce a large range of results into a set of simplified curves to help make sense of findings. This can be done through dimensional analysis using the Buckingham Pi Theorem which determines how many non-dimensional variables are required for a given set of dimensional variables [33]. In our study, we are highlighting the variables that affect the water heights in our flood inundation models. The flood water heights (h) may be a function of volumetric flow rate (Q), channel bed slope (α), Manning's coefficient (n_M), building edge size (b), width of the channel (w), length of the channel (L), and gravity (g). Along with these parameters, the water heights that each configuration experiences is a direct result of the specific position $\mathbf{r}$ of each building in the domain. In a configuration with N buildings, the water heights can be represented as follows:

$$h = f(Q, \alpha, n_M, b, w, L, g, \mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N) \tag{5.2}$$

If the Buckingham Pi theorem were to be applied to this description of water heights, we would need $2N+5$ dimensionless variables. To reduce this, the two dimensionless parameters that are considered in this study, Φ and χ_4, are used to more simply represent the affects that building positions have on the water heights. Although this is a less exhaustive approach, it significantly streamlines the representation of building positions while maintaining the most crucial details. This application of dimensionless variables dramatically reduces the representation of water heights to

$$h = f(Q, \alpha, n_M, \Phi, \chi_4, b, w, L, g) \tag{5.3}$$

Now that the problem has been reduced to a manageable number of parameters, the Buckingham Pi Theorem can be applied to simplify parameter relationships. The units of every variable that is being analyzed is reduced down to the primary dimensions of mass (M), length (L), and time (T). This is shown in the dimensional matrix below.

Table 5.2: Dimensional Matrix

	h	Q	α	n_M	Φ	χ_4	b	w	L	g
M	0	0	0	0	0	0	0	0	0	0
L	1	3	0	$-\frac{1}{3}$	0	0	1	1	1	1
T	0	-1	0	1	0	0	0	0	0	-2

Applying the Buckingham Pi Theorem with 10 parameters and 2 dimensions, we know that the problem can be reduced to 8 dimensionless quantities. By performing dimensional analysis, we generated the following dimensionless parameters:

$$\Pi_0 = \frac{Q/hw}{\sqrt{hg}}$$

$$\Pi_1 = \alpha$$

$$\Pi_2 = \frac{n_M\sqrt{g}}{w^{1/6}}$$

$$\Pi_3 = \Phi$$

$$\Pi_4 = \chi_4$$

$$\Pi_5 = \frac{b}{w}$$

$$\Pi_6 = \frac{w}{L}$$

$$\Pi_7 = \frac{Q^2}{gw^5}$$

$$\frac{Q/hw}{\sqrt{hg}} = f(\alpha,\ \frac{n_M\sqrt{g}}{w^{1/6}},\ \Phi,\ \chi_4,\ \frac{b}{w},\ \frac{w}{L},\ \frac{Q^2}{gw^5}) \tag{5.4}$$

It is important to note that the dimensionless parameter Π_0 is similar to the Froude number which is an important dimensionless number discussed in Chapter 2. This reflects the intrinsic importance of the Froude number in any shallow water analysis.

In this analysis, Π_6 can be cancelled out because assuming that we have a long enough channel, $w << L$ and the term goes to zero. The Π_5 term can also be assumed as zero because $b << w$.

After this simplification, the relationship of dimensionless parameters can be expressed as follows:

$$\frac{Q/hw}{\sqrt{hg}} = f(\alpha,\ \frac{n_M\sqrt{g}}{w^{1/6}},\ \Phi,\ \chi_4,\ \frac{Q^2}{gw^5}) \tag{5.5}$$

5.5 Effects of Volumetric Flow Rate

After performing dimensional analysis, it becomes clear that the volumetric flow rate plays a significant role in affecting water heights. A large range of simulations were run for volumetric flow rates of 1,000 m^3/s and 500 m^3/s and while these simulations will be discussed in detail later on, the results showed that decreasing the upstream volumetric flow by 50% had the direct effect of also decreasing the upstream water height by 50%. To explore this trend more, a collection of configurations were used to test volumetric flow rates of 100 m^3/s, 250 m^3/s, 500 m^3/s, 750 m^3/s, and 1,000 m^3/s. The slopes were held constant at 0.1% for all these tests, and each configuration was run with these five different values. Figure 5.9 shows the results from this test. From this figure, we can confirm that there is a linear relationship between the volumetric flow rate and the water heights resulting from these simulations. By testing this trend with different configurations, we can also see that the rate at which the water heights change with respect to volumetric flow is dependent on the configuration geometry. The top two graphs represent changing porosity values while the bottom two represent changing spatial order.

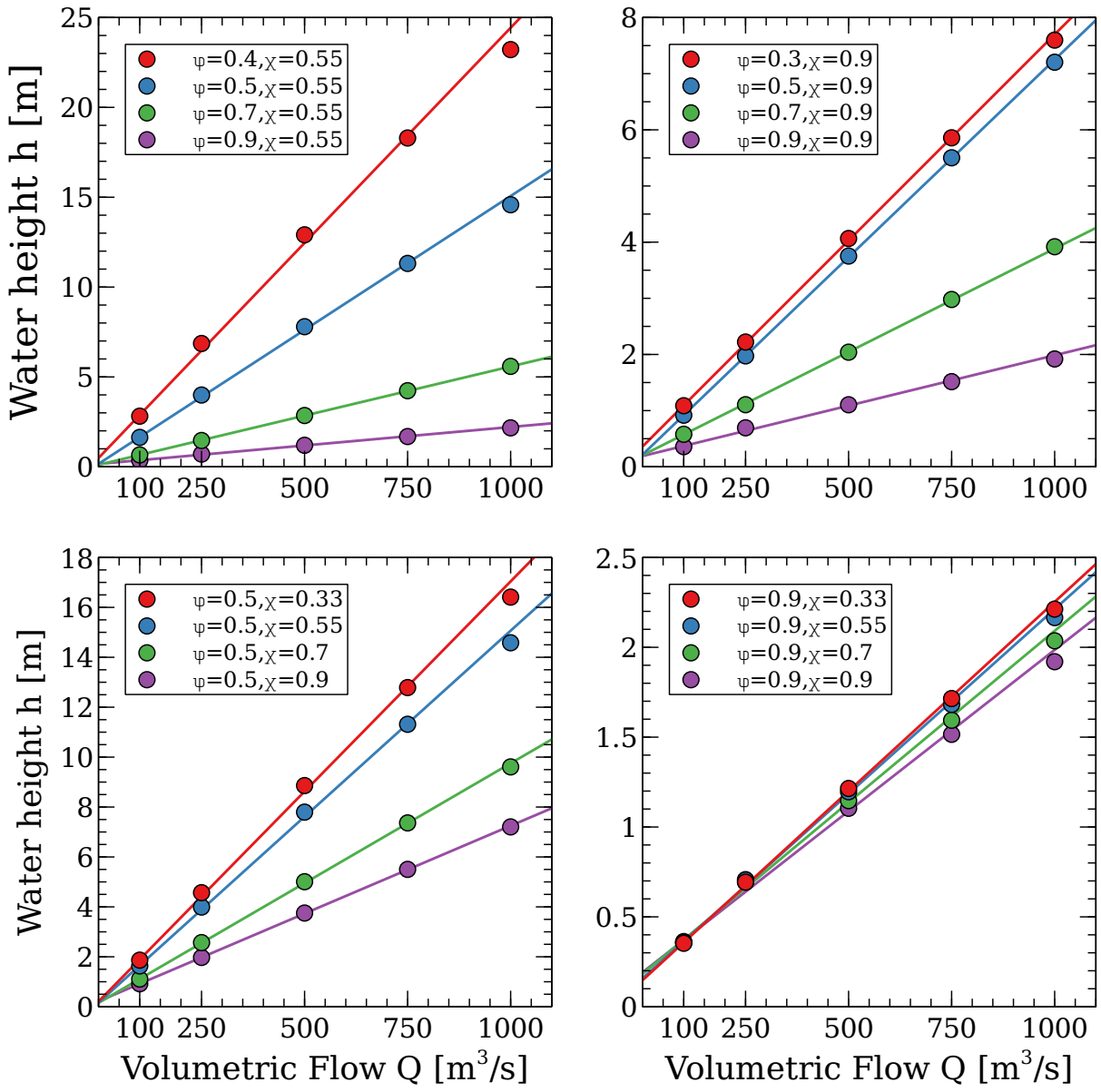


Figure 5.9: Linear relationship between flood heights and upstream volumetric flow rates. The top two figures feature data from configurations that keep the spatial order steady while changing the porosity values. Graph 1 (top left) features configurations with constant low spatial order and changing porosity. Graph 2 (top right) similarly features changing porosity, but with high spatial order. The bottom two figures inversely keep porosity values constant while changing the spatial order. Graph 3 (bottom left) features data from configurations with low porosity and altering spatial order. The final graph, graph 4 (bottom right), features flood heights from configurations with high porosity and changing spatial order.

After establishing through the modeling process that a linear relationship exists between volumetric flow rates and water heights, an expression for this relationship can begin to form. Through dimensional analysis, we had already established that the dimensionless variable $\frac{Q/hw}{\sqrt{hg}}$ is a function of a few different dimensionless parameters, including $\frac{Q^2}{gw^5}$. Now that this linear relationship has been discovered, Equation 5.5 can be reconstructed as

$$h = \frac{Q}{w\sqrt{wg}}\, f(\alpha, \Phi, \chi_4, \frac{n_M\sqrt{g}}{w^{1/6}}) \tag{5.6}$$

to reflect that the water height h is linearly related to volumetric flow rate Q and is still a function of α, Φ, χ_4, and n_M. We can consider a Taylor expansion around $\frac{n_M\sqrt{g}}{w^{1/6}}$, and neglecting second order effects, can refine the equation further as:

$$h = \frac{Qn_M}{w^{5/3}}\, f(\alpha, \Phi, \chi) \tag{5.7}$$

To understand this equation in relation to established hydrodynamic theory, it can be interpreted in light of the Manning equation for uniform open channel flow which is:

$$V = \frac{Q}{A} = \frac{1}{n_M} R^{2/3} S^{1/2} \tag{5.8}$$

where V is the velocity of the fluid, R is the hydraulic radius $R = \frac{wh}{w+2h}$, S is the bed slope, and n_M is the Manning coefficient [50]. This formula can be rewritten in terms of the variables in this study and solved for flood height h to produce:

$$h = \frac{Qn_M}{R^{5/3}\sqrt{\alpha}} \tag{5.9}$$

To make a simplification to this Manning equation, we can consider the limiting case of w/h approaching zero, which in turn makes the hydraulic radius $R = w\frac{1}{2+w/h} = \frac{w}{2}$, and

the limiting case of w/h approaching infinite, which makes the hydraulic radius $R = 0$. By considering these limiting cases, the hydraulic radius becomes dependent on only w and a constant term and our independently derived Equation 5.7 can directly be compared to the Manning equation (Eqn. 5.9).

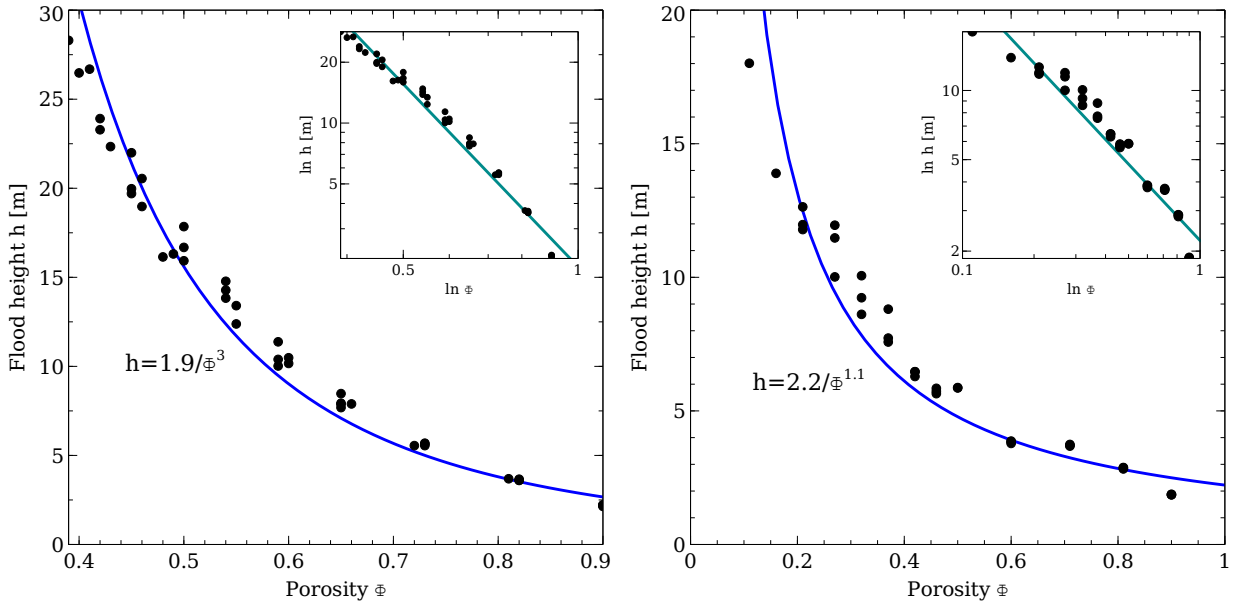
As we can see, Equation 5.7 that was independently derived in this study through dimensional analysis and simplifications is a comparable to the Manning equation (Eqn. 5.9), but it is now also dependent on the two dimensionless parameters Φ and χ which adds the effects of channel morphology to the determination of flood heights.

5.6 Effects of Porosity and Spatial Order

In order to further refine Equation 5.7, quantitative relationships between flood heights and the Φ and χ_4 parameters must be developed. First, an analysis on the effects of porosity is performed. Flood inundation results are presented in Figure 5.10.

The relationship between configuration porosity and flood behavior was assessed by comparing averaged data from multiple inundation models. In order to isolate the effects of porosity from any other variations that could influence flood heights, the data was taken from configurations with constant spatial order. Figure 5.10 shows the correlation between porosity and flood heights for disordered systems with $\chi_4 = 0.46$ in Figure 5.10a and ordered systems with $\chi_4 = 0.95$ in Figure 5.10b. From this figure, we determine that flood heights change inversely with respect to porosity; as porosity reaches zero, flood heights increase towards infinite. This establishes an inverse relationship between flood heights and porosity that follows the form:

$$h \propto f(\Phi) = \frac{a}{\Phi^n} \tag{5.10}$$



(a) Low spatial order $\chi_4 = 0.46$

(b) High spatial order $\chi_4 = 0.95$

Figure 5.10: Relationship between upstream flood heights and porosity values for disordered systems with $\chi_4 = 0.45$ (left) and ordered systems with $\chi_4 = 0.95$ (right). The results from flood inundation models are presented as data points and are fitted with an inverse relationship between porosity and flood height. The graph is also plotted on a logarithmic scale and presented as the inset figures.

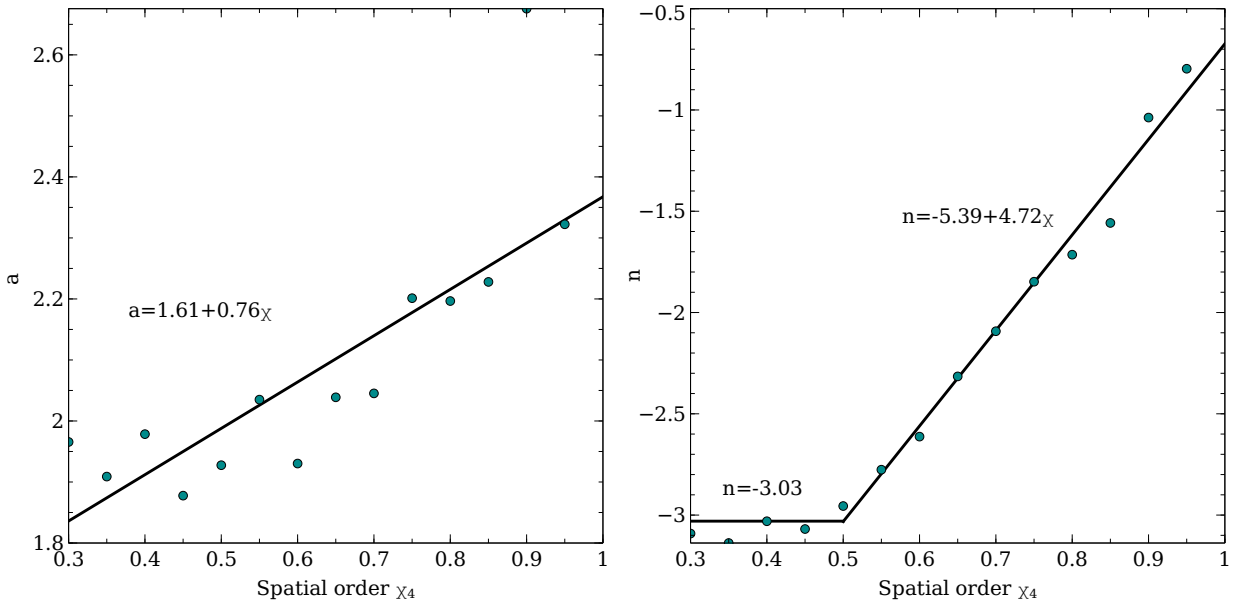
By fitting a curve to the trends obtained through our models, we can determine that the a and n terms in this relationship are dependent on the spatial order of the system. Studying this trend on a logrithmic scale for each case of spatial order, the value of a and n can be taken as the y-intercept and slope of the line for both systems. Figure 5.11 shows how the values of a and n are related to spatial order. The relationship between spatial order and a and n values is:

$$a = 1.61 + 0.76\chi_4 \tag{5.11}$$

$$n = \begin{cases} -3.03 & \chi_4 \leq 0.5 \\ -5.39 + 4.72\chi_4 & \chi_4 \geq 0.5 \end{cases} \tag{5.12}$$

While the a term varies linearly with respect to χ_4 along all values of χ_4, the n term sees more of a piece-wise linear relation. When χ_4 is greater than 0.5, the value for n increases

linearly. However, at a χ_4 less than 0.5, there is little to no variation in the value of n. This shows that when configurations reach $\chi_4 = 0.5$, the system is considered disordered and trying to decrease the spatial order below this point does not have any significant effect on the physical condition.



(a) Determination of a

(b) Determination of n

Figure 5.11: Relationship between spatial order and the values for terms a and n. 5.11a illustrates a linear relationship between the term n and χ_4. 5.11b shows a piece-wise linear relationship between the term n and χ_4, demonstrating that changes to spatial order below $\chi_4 = 0.5$ has little to no effect on flood heights.

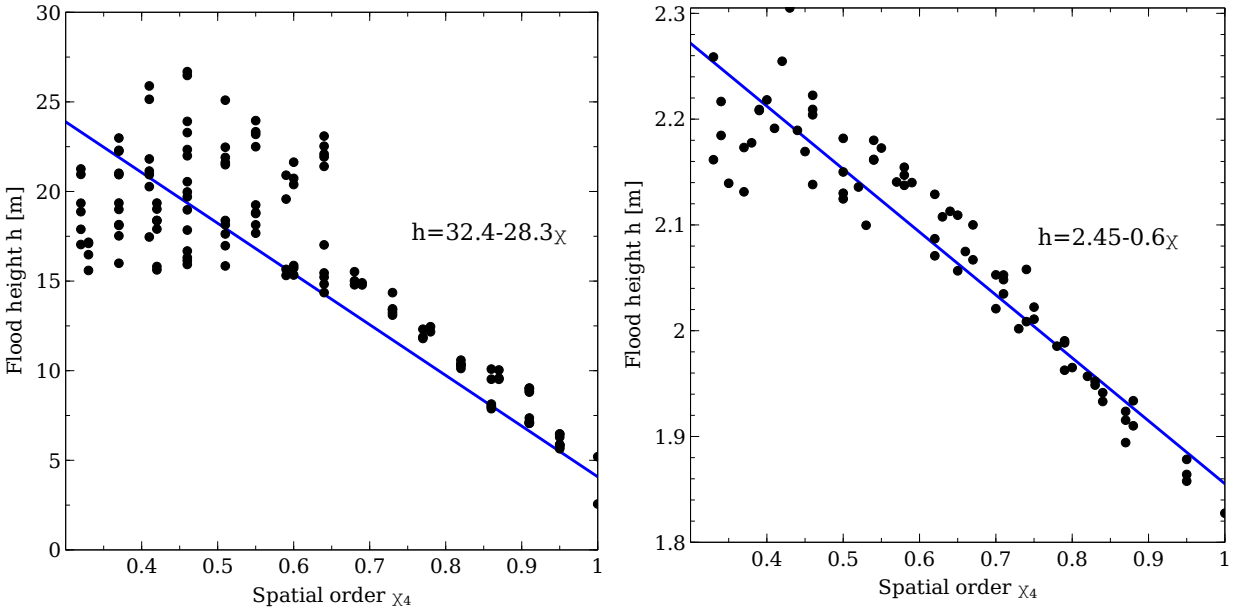
With this analysis, Equation 5.7 is further refined by the inclusion of this quantitative relation between Φ and h. The equation reads as follows:

$$h = \frac{Qn_M}{w^{5/3}} \frac{a}{\Phi^n} f(\alpha, \chi_4) \tag{5.13}$$

where a and n are dependent on χ_4.

The relationship between flood heights and spatial order is also subsequently determined through the analysis of inundation results. Studying systems with constant porosity allows for the isolation of the effects caused by spatial order alone. Figure 5.12 presents the rela-

tionship between spatial order and flood heights for systems of low porosity with $\Phi = 0.40$ (Figure 5.12a) and systems of high porosity with $\Phi = 0.90$ (Figure 5.12b).



(a) Low porosity $\Phi = 0.40$ (b) High porosity $\Phi = 0.90$

Figure 5.12: Relationship between upstream flood heights and Mermin order values for low porosity systems with $\Phi = 0.45$ (left) and high porosity systems with $\Phi = 0.9$ (right). The results from flood inundation models are presented as data points and are fitted with a linear relationship between spatial order and flood height.

Unlike its relationship to porosity, flood heights decrease linearly with respect to spatial order as:

$$h \propto f(\chi_4) = b + m\chi_4 \tag{5.14}$$

The values for m and b are also dependent on the porosity of the system, as systems of lower porosity have larger m and b values than those of higher porosity. By performing additional analysis on systems with a range of porosity values, this dependence can be quantified to produce an understanding on the determination of m and b. Figure 5.13 illustrates this dependence. The final relationship between porosity and the b and m values is:

$$b = \frac{21.17}{\Phi} - 20.17 \tag{5.15}$$

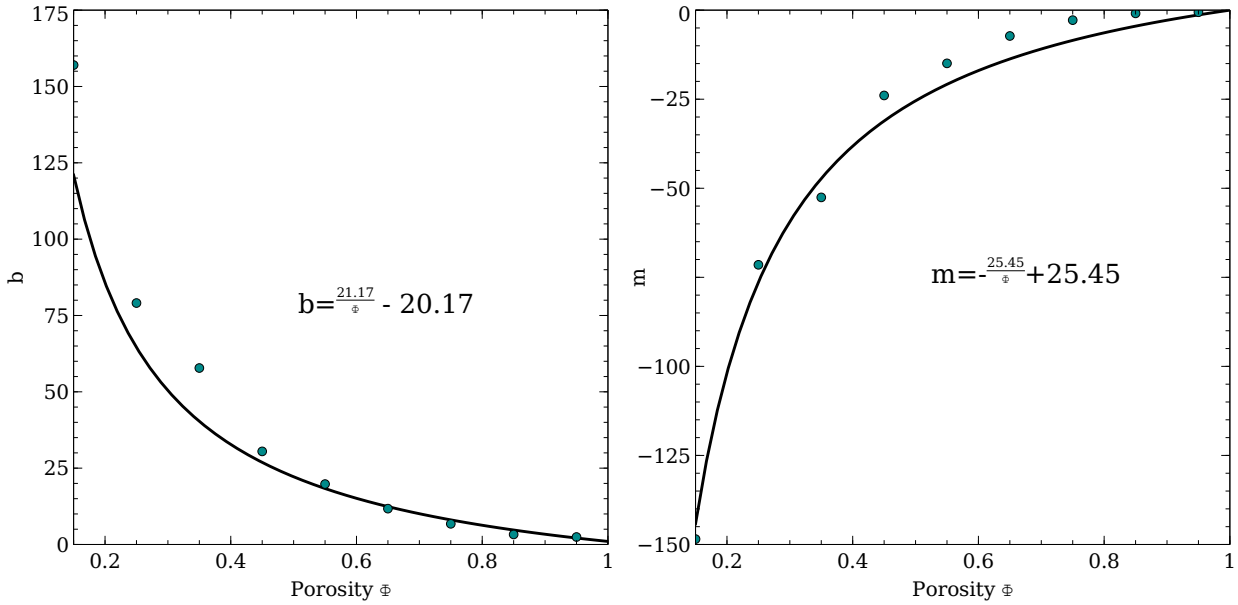
$$m = -\frac{25.45}{\Phi} + 25.45 \tag{5.16}$$

As porosity values increase and reach one, meaning that the buildings in the domain begin to disappear and we approach an empty channel, the b term approaches one and the m term approaches zero. When this happens, the weight that is given to $f(\chi_4)$ in the relation is reduced. These trends coincide with the idea that at low porosity, flood heights are highly dependent on spatial order. Any changes to spatial order is able to increase or decrease the flood heights when considering a fairly packed system. But, as porosity begins to increase, it subsequently causes the influence of spatial order on flood heights to diminish. High porosity values render the spatial order mostly inconsequential. This is due to the fact that high porosity allows for increased voids between buildings which in turn means that the flood is able to flow without concern for the placement of obstacles. While the curve proposed here is not a perfect fit to the data, it proves a quantitative relation for the purpose of this study. This fit, along with the fit for the a and n terms, may be subject to improvement with the generation of more data.

With this last analysis, the final form of the equation presented in this study is generated by the inclusion of the quantitative relationship between χ_4 and h as:

$$h = \frac{Qn_M}{w^{5/3}} \frac{a(b + m\chi_4)}{\Phi^n} f(\alpha) \tag{5.17}$$

where terms a and n are dependent on spatial order χ_4 and terms b and m are dependent on porosity Φ. If a configuration with porosity of $\Phi = 1.0$, meaning an empty channel with no buildings, is considered, this equation reduces to a version of the Manning equation (Eqn. 5.9). Further analysis can later be done to discover the effects of slope on the flood heights and its interaction with the other parameters.



(a) Determination of b

(b) Determination of m

Figure 5.13: Relationship between spatial order and the values for terms b and m. As Φ approaches one, b approaches one in Figure 5.13a and m aproaches zero in Figure 5.13b.

While the interactions between porosity Φ and spatial order χ_4 and their effects on flood heights were presented quantitatively here, it is also helpful to have a complete visualization of this relationship. Figure 5.14 presents a type of contour plot of the water heights for all configurations plotted with respect to both the Φ and χ_4 values.

Here, the interaction between porosity and spatial order in the resulting flood heights can be clearly understood. While lower porosity configurations are expected to observe high flood heights, Figure 5.14 shows that is not the case when the configuration also maintains a high measure of spatial order. Along with this, configurations with low spatial order also do not see high flood heights if the porosity is maintained at a high value. This shows the interaction between the two parameters and their influence on one another. The value of one parameter, say porosity, is able to determine the influence of the other parameter, which would be spatial order. It is only when both the porosity and spatial order are at minimum values with respect to each other that the highest flood heights are produced.

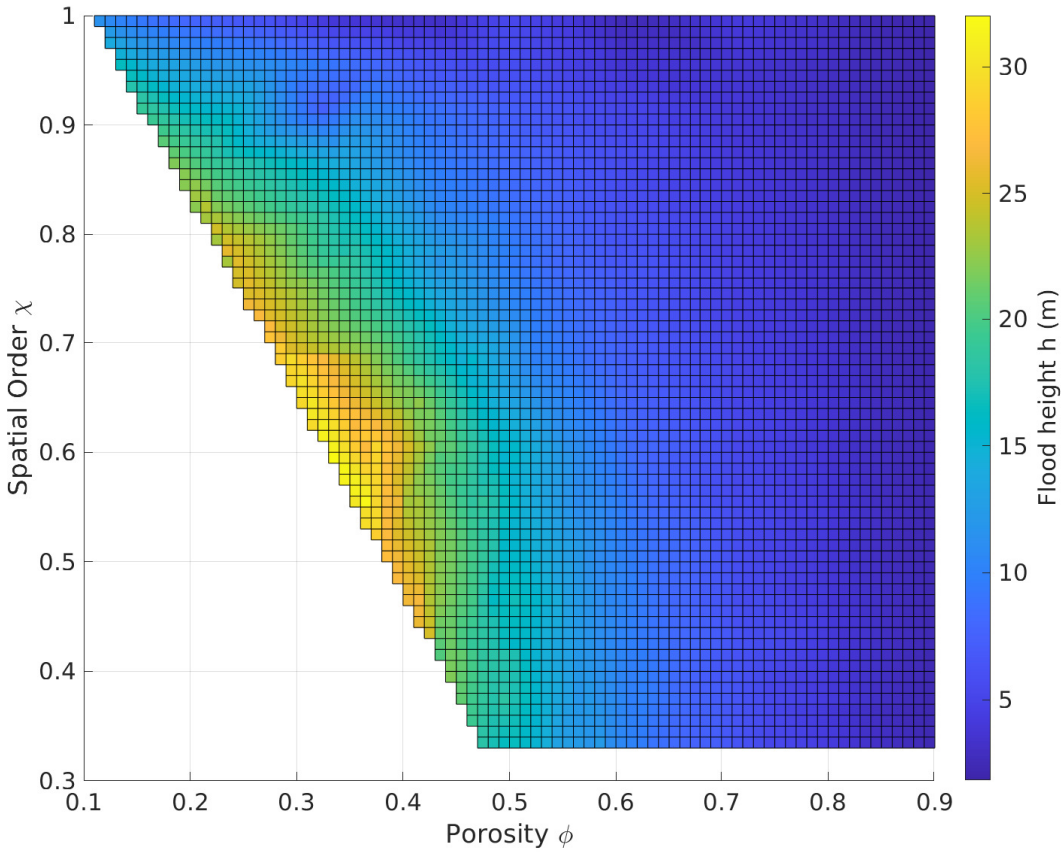


Figure 5.14: Contour plots of water heights for case one. The porosity Φ is plotted along the x-axis, the spatial order χ_4 is plotted along the y-axis, and the flood heights are plotted along the z-axis and shown as a colored contour plot. The colorbar shows the flood height values. The bottom left-hand corner is empty to avoid unrealistic or improbable configurations.

While the analysis of this study has been done mostly using simulations with a volumetric flow Q of $1,000 m^3/s$ and slope of 0.1%, tests were also done for all configurations with two more combinations of bed slopes α and upstream volumetric flow rates Q. The three cases that were tested are described in Table 5.3.

Table 5.3: The Three Flood Cases

	Case 1	Case 3	Case 3
Volumetric Flow Q $[m^3/s]$	1000	500	1000
Slope	0.001	0.001	0.01

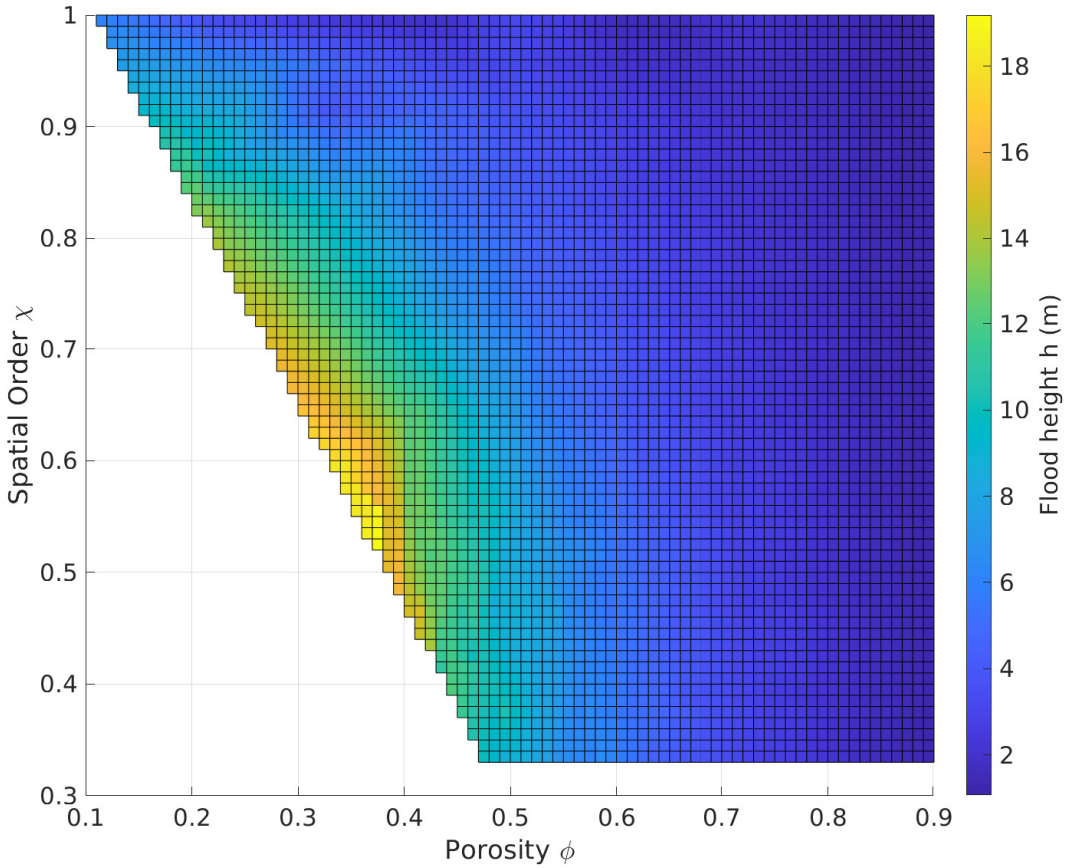


Figure 5.15: Contour plots of water heights for case two. The porosity Φ is plotted along the x-axis, the spatial order χ_4 is plotted along the y-axis, and the flood heights are plotted along the z-axis and shown as a colored contour plot. The colorbar shows the flood height values. The bottom left-hand corner is empty to avoid unrealistic or improbable configurations.

Case two, which can be seen in Figure 5.15, had been previously mentioned in the study for the effects of volumetric flow rates on flood heights. In this case, the volumetric flow rate was reduced by half from case one. Case three, which can be seen in Figure 5.16, was also tested to determine how changing the slope would effect the flood heights; the volumetric flow rate was held constant from case one but the slope for this case was increased by a factor of 10. While the values of flood heights are different for each case, the overall trends is consistent along all cases. Each case was run with the entire set of 500 configurations to provide a contour plot of flood heights relative to both porosity and Mermin order.

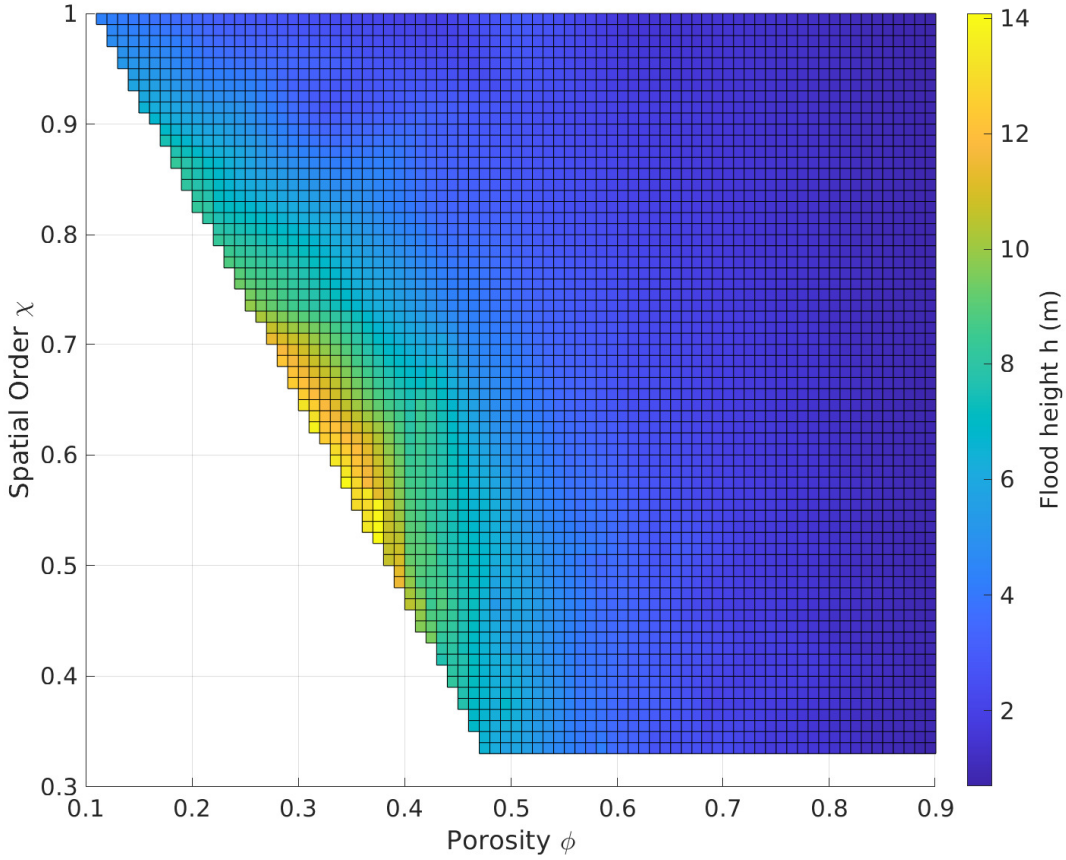


Figure 5.16: Contour plots of water heights for case three. The porosity Φ is plotted along the x-axis, the spatial order χ_4 is plotted along the y-axis, and the flood heights are plotted along the z-axis and shown as a colored contour plot. The colorbar shows the flood height values. The bottom left-hand corner is empty to avoid unrealistic or improbable configurations.

Figures 5.14, 5.15 and 5.16 show that changing the slope and upstream volumetric flow rate changes the flood heights but does not change the interaction between the two parameters Φ and χ_4. This subsequently confirms that the relationships and equations derived in this study hold true for many different cases.

5.7 Summary

Through the use of the computer-generated synthetic cities from this study, accurate measures of flood heights were produced for a wide range of model configurations. This allowed for the study of the effects of important urban morphology parameters Φ and χ_4. We independently derived an equation to explain flood heights with respect to a number of parameters, including Φ and χ_4, and proved the merit of this equation by successfully comparing it to the Manning equation. By implementing the relationships found through the study of urban morphology and its effects of flood heights, this equation was refined further. By testing three different cases of volumetric flow rate Q and slope α, we were able to confirm that the interactions between Φ and χ_4 hold true in different circumstances.

Chapter 6

Conclusion

In this work, we have established that the Mermin order parameter is a useful tool that can be applied to city science in order to better understand the spatial orientation of urban areas and infrastructure. Since this tool produces a single value that describes spatial orientation, it works to streamline many of the previously developed methods for quantifying urban form. Calculating this parameter for real-world city layouts can generate simple assessments of spatial order for massive complex urban regions, aiding in many different fields of urban physics. Through the introduction of this parameter into the realm of urban physics, it can be paired with many of the other previously established tools to help improve the accuracy and efficiency of modeling city systems.

Along with the introduction of the Mermin order parameter, we have also created an extremely useful automated probabilistic algorithm for generating hundreds, if not thousands, of simplistic city configurations using just two parameters: porosity and spatial order. While this algorithm was used to generate city layouts for the purpose of flood inundation studies, it could also be applied to many different situations. As previously discussed, there is a wide array of physical conditions that are affected by the form of urban cities, from urban

heat islands to city wind tunnels. Thanks to the automated, generic and randomized nature of the algorithm presented here, the configurations that are produced can be subsequently applied to study many different physical circumstances.

Beyond just the quantification and modeling of urban form that this study explored, we were also able to determine the effects that these urban forms had on flood behaviors. While this study reiterated the well-known fact that city porosity can increase or decrease flood levels [10], it also proved that the spatial order of the urban system has an effect on floods. Systems with a lower level of order saw much higher flood levels than systems with higher levels of order for the same porosity. This finding demonstrates that urban forms cannot be defined simply by porosity levels alone when studying their effects on urban floods because it lacks crucial information that aids in understanding trends. These studies must also consider urban spatial order and orientation in order to fully build connections between urban form and flood behavior. Future studies in the fields of urban physics and flood modeling can benefit from the adaptation of spatial order into the set of important parameters.

To quantitatively reflect the influence of urban form in the behavior of floods, this study proposes a new independently derived equation that determines flood heights from an array of important parameters. This equation is an improvement of the Manning equation, incorporating the established merit of the Manning equation with the studied effects of urban morphology. With more refinement, this equation can be finalized and presented as a tool for predicting flood heights in urban areas.

In the future, the proposed tools from this study can be used to quantify the urban form of major cities. It is important to understand though that cities are extremely complex systems that are ever-changing. They are the "characteristic physical and social unit of civilization" [25], and as such, are living communities that evolve over time based on the needs of the people. While the synthetic cities generated in this study accurately model a wide range of configurations using two parameters found to be crucial in understanding urban form, they

do not come close to reflecting the complexity of true cities. The discussion of urban form is one that will be studied for as long as cities exist, becoming even more important with trends of urban expansion and increased natural disasters.

Bibliography

[1] L. Adolphe. A simplified model of urban morphology: Application to an analysis of the environmental performance of cities. *Environment and Planning B: Planning and Design*, 28(2):183–200, 2001.

[2] I. Aguilar-Barajas, N. P. Sisto, A. I. Ramirez, and V. Magaña-Rueda. Building urban resilience and knowledge co-production in the face of weather hazards: flash floods in the monterrey metropolitan area (mexico). *Environmental Science & Policy*, 99:37–47, 2019.

[3] F. Ahmed, E. Moors, M. S. A. Khan, J. Warner, and C. Terwisscha van Scheltinga. Tipping points in adaptation to urban flooding under climate change and urban growth: The case of the dhaka megacity. *Land Use Policy*, 79:496–506, 2018.

[4] S. Al Wakeel, F. Ghanbari, and M. Hubler. Particle arrangement effects on the stress intensity in composite material. *Engineering Fracture Mechanics*, 202:33–46, 2018.

[5] J. D. Anderson. *Computational Fluid Dynamics: The Basics with Applications.* McGraw-Hill, Inc., United States of America, 1995.

[6] L. Begnudelli and B. F. Sanders. Unstructured grid finite-volume algorithm for shallow-water flow and scalar transport with wetting and drying. *Journal of Hydraulic Engineering*, 132(4):371–384, 2006.

[7] L. Begnudelli, B. F. Sanders, and S. F. Bradford. Adaptive godunov-based model for flood simulation. *Journal of Hydraulic Engineering*, 134(6):714–725, 2008.

[8] C. Bousige, A. Botan, F.-J. Ulm, R. Pellenq, and B. Coasne. Optimized molecular reconstruction procedure combining hybrid reverse monte carlo and molecular dynamics. *The Journal of Chemical Physics*, 142:114112, 03 2015.

[9] S. F. Bradford and B. F. Sanders. Finite-volume model for shallow-water flooding of arbitrary topography. *Journal of Hydraulic Engineering*, 128(3):289–298, 2002.

[10] M. Bruwier, C. Maravat, A. Mustafa, J. Teller, M. Pirotton, S. Erpicum, P. Archambeau, and B. Dewals. Influence of urban forms on surface flow in urban pluvial flooding. *Journal of Hydrology*, 582:124493, 12 2019.

[11] R. J. Burby and S. P. French. Coping with floods: The land use management paradox. *Journal of the American Planning Association*, 47(3):289–300, 1981.

[12] L. Cartwright. An examination of flood damage data trends in the united states. *Journal of Contemporary Water Research & Education*, 130(1):20–25, 2005.

[13] R. Charbeneau. Open channel hydraulics. class notes for Hydraulic Engineering Design.

[14] J. Church, P. Clark, A. Cazenave, J. Gregory, S. Jevrejeva, A. Levermann, M. Merrifield, G. Milne, R. Nerem, P. Nunn, A. Payne, W. Pfeffer, D. Stammer, and A. Unnikrishnan. 2013: Sea level change. In T. Stocker, D. Qin, G.-K. Plattner, M. Tignor, S. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex, and P. M. (eds.), editors, *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA., 2013.

[15] C. Dawson and C. M. Mirabito. The shallow water equations. Available at `https://users.oden.utexas.edu/~arbogast/cam397/dawson_v2.pdf`, September 2008.

[16] M. Gall, K. A. Borden, C. T. Emrich, and S. L. Cutter. The unsustainable trend of natural hazard losses in the united states. *Sustainability*, 3(11):2157–2181, 2011.

[17] S. Hallegatte, C. Green, R. J. Nicholls, and J. Corfee-Morlot. Future flood losses in major coastal cities. *Nature Climate Change*, 3(9):802–806, Sep 2013.

[18] S. Hanson, R. Nicholls, N. Ranger, S. Hallegatte, J. Corfee-Morlot, C. Herweijer, and J. Chateau. A global ranking of port cities with high exposure to climate extremes. *Climatic Change*, 104(1):89–111, 2011.

[19] M. E. Hauer, J. M. Evans, and D. R. Mishra. Millions projected to be at risk from sea-level rise in the continental united states. *Nature Climate Change*, 6(7):691–695, Jul 2016.

[20] J. Hinkel, D. Lincke, A. T. Vafeidis, M. Perrette, R. J. Nicholls, R. S. J. Tol, B. Marzeion, X. Fettweis, C. Ionescu, and A. Levermann. Coastal flood damage and adaptation costs under 21st century sea-level rise. *Proceedings of the National Academy of Sciences*, 111(9):3292–3297, 2014.

[21] C. Huang, M.-H. Hsu, W.-H. Teng, and Y.-H. Wang. The impact of building coverage in the metropolitan area on the flow calculation. *Water*, 6:2449–2466, 08 2014.

[22] J.-J. Kim and D.-Y. Kim. Effects of a building's density on flow in urban areas. *Advances in Atmospheric Sciences*, 26:45–56, 01 2009.

[23] R. E. Kopp, R. M. DeConto, D. A. Bader, C. C. Hay, R. M. Horton, S. Kulp, M. Oppenheimer, D. Pollard, and B. H. Strauss. Evolving understanding of antarctic ice-sheet physics and ambiguity in probabilistic sea-level projections. *Earth's Future*, 5(12):1217–1233, 2017.

[24] P. J. LaNasa and E. L. Upp. 2 - basic flow measurement laws. In P. J. LaNasa and E. L. Upp, editors, *Fluid Flow Measurement (Third Edition)*, pages 19–29. Butterworth-Heinemann, Oxford, third edition edition, 2014.

[25] K. Lynch. The form of cities. *Scientific American*, 190(4):54–63, 1954.

[26] R. L. McGreevy and L. Pusztai. Reverse monte carlo simulation: A new technique for the determination of disordered structures. *Molecular Simulation*, 1(6):359–367, 1988.

[27] N. D. Mermin. Crystalline order in two dimensions. *Phys. Rev.*, 176:250–254, Dec 1968.

[28] A. Mustafa, M. Bruwier, P. Archambeau, S. Erpicum, M. Pirotton, B. Dewals, and J. Teller. Effects of spatial planning on future flood risks in urban environments. *Journal of Environmental Management*, 225:193–204, 11 2018.

[29] A. Mustafa, X. Zhang, D. Aliaga, M. Bruwier, B. Dewals, and J. Teller. Investigating the impact of urban layout geometry on urban flooding. 02 2019.

[30] E. Ng, H.-H. Wong, and M. Han. Permeability, porosity and better ventilated design for high density cities. 01 2006.

[31] E. C. O'Donnell and C. R. Thorne. Drivers of future urban flood risk. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 378(2168):20190216, 2020.

[32] G. Opletal, T. Petersen, B. O'Malley, I. Snook, D. G. McCulloch, N. A. Marks, and I. Yarovsky. Hybrid approach for generating realistic amorphous carbon structure using metropolis and reverse monte carlo. *Molecular Simulation*, 28(10-11):927–938, 2002.

[33] R. L. Panton. *Incompressible Flow*. John Wiley & Sons, Inc., Hoboken, New Jersey, 4 edition, 2013.

[34] C. Pease. An overview of monte carlo methods. Towards Data Science Inc., Sept. 6 2018. [Online].

[35] R. Radhakrishnan and B. Trout. *Order Parameter Approach to Understanding and Quantifying the Physico-Chemical Behavior of Complex Systems*, pages 1613–1626. Springer Science+Business Media B.V., 01 2005.

[36] B. E. Rapp. Chapter 9 - fluids. In B. E. Rapp, editor, *Microfluidics: Modelling, Mechanics and Mathematics*, Micro and Nano Technologies, pages 243–263. Elsevier, Oxford, 2017.

[37] B. F. Sanders. Integration of a shallow water model with a local time step. *Journal of Hydraulic Research*, 46(4):466–475, 2008.

[38] 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*, 33(12):1456–1467, 2010.

[39] B. F. Sanders, J. E. Schubert, K. A. Goodrich, D. Houston, D. L. Feldman, V. Basolo, A. Luke, D. Boudreau, B. Karlin, W. Cheung, S. Contreras, A. Reyes, A. Eguiarte, K. Serrano, M. Allaire, H. Moftakhari, A. AghaKouchak, and R. A. Matthew. Collaborative modeling with fine-resolution data enhances flood awareness, minimizes differences in flood perception, and produces actionable flood maps. *Earth's Future*, 8(1):e2019EF001391, 2020. e2019EF001391 2019EF001391.

[40] J. E. Schubert and B. F. Sanders. Building treatments for urban flood inundation models and implications for predictive skill and modeling efficiency. *Advances in Water Resources*, 41:49–64, 2012.

[41] J. R. Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In M. C. Lin and D. Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry.

[42] J. Sobstyl, T. Emig, M. J. Abdolhosseini Qomi, F.-J. Ulm, and R. Pellenq. Role of city texture in urban heat islands at nighttime. *Physical Review Letters*, 120, 03 2018.

[43] J. M. Sobstyl. Urban physics: Molecular approach to city texture analysis for controlling urban heat island. Master's thesis, Massachusetts Institute of Technology, Cambridge MA, 2016.

[44] J. Southard. 12.090 introduction to fluid motions, sediment transport, and current-generated sedimentary structures. Available at `https://ocw.mit.edu` License: Creative Commons BY-NC-SA., Fall 2016.

[45] M. Tadi, C. Biraghi, M. Zadeh, and L. Brioschi. Urban porosity. a morphological key category for the optimization of the cas's environmental and energy performance. *Journal of Engineering Technology*, 4:138, 08 2017.

[46] J. Teng, A. Jakeman, J. Vaze, B. Croke, D. Dutta, and S. Kim. Flood inundation modelling: A review of methods, recent advances and uncertainty analysis. *Environmental Modelling & Software*, 90:201–216, 2017.

[47] E. Valette and O. Antibi. Free surface flow hands-on. Available at `http://hmf.enseeiht.fr/travaux/CD9899/travaux/optmfn/fsfho/98-99/mfn01/cr.htm#II%20%20%20A%20%20%20%20THE%20BOUNDARY%20CONDITIONS`(2021/02/12).

[48] T. N. Vartziotis. Investigation of urban networks by using computational fluid dynamics techniques and urban physics applications. Master's thesis, Massachusetts Institute of Technology, Cambridge MA, 2018.

[49] C. B. Vreugdenhil. *Numerical Methods for Shallow-Water Flow*. Kluwer Academic Publishers, P.O. Box 17, 3300 AA Dordrecht, The Netherlands., 1994.

[50] F. M. White. *Fluid Mechanics*. McGraw-Hill, 1221 Avenue of the Americas, New York, NY 10020, 7 edition, 2011.

[51] C. Yuan and E. Ng. Building porosity for better urban ventilation in high-density cities – a computational parametric study. *Building and Environment*, 50:176–189, 2012.

[52] C. Zevenbergen and B. Gersonius. *Challenges in Urban Flood Management*, pages 1–11. 02 2007.

[53] W. Zhou, G. Huang, and M. L. Cadenasso. Does spatial configuration matter? understanding the effects of land cover pattern on land surface temperature in urban landscapes. *Landscape and Urban Planning*, 102(1):54–63, 2011.

Appendix A

MATLAB Codes

Here we provide all the codes that were used to generate the synthetic cities and prepare files for BreZo flood inundation models. When used together, these codes create one large automated system which runs a hybrid reverse monte carlo algorithm to generate points, creates buildings, generates computational grids, and transforms data into textfiles which can be easily read by BreZo. The only manual input needed are the necessary values for the function MAIN.m.

A.1 MAIN Function

```
%%% MAIN.m %%%
%%%%% AUTOMATED ALGORITHM MAIN FUNCTION %%%%%
% All input values are applied to this function.
% This function can be run through computer's terminal.
```

```
function MAIN(width,L,phi,chi,iter,slope,Q,ts,out,sd)

% width = width of channel [m]
%     L = lengh of channel [m]
%   phi = porosity value
%   chi = Mermin order value
%  iter = iteration/realization number
% slope = slope of channel bed
%     Q = upstream volumetric flow rate [m/s^3]
%    ts = run time [hr]
%   out = Output option: 1 = Tecplot, 2 = Matlab
%    sd = seed for random number generator


%% Run buildingsnew to create configuration

[p,phi_r,chi_r] = Buildingsnew(width,L,phi,chi,sd);

% Outputs: p = matlab polyshape
%          phi_r = actual phi value of resulting configuration
%          chi_r = actual chi value of resulting configuration


%% Create unique name for configuration
fname = sprintf('w%dl%dp%dm%d_%d',width,L,round(phi_r*100), ...
            round(chi_r*100),iter);
```

```
%% Make directory for configuration and copy all necessary files to dir.

%         parbrezo: executable file to run BreZo
%     runbrezo.sub: SLURM job submission file
%  inputtimeseries: folder with input time series file

system("mkdir "+fname);
system("cp parbrezo "+fname);
system("cp runbrezo.sub "+fname);
system("cp -r inputtimeseries/ "+fname);
cd(fname);


%% Create subfolder for results 'fname.1' within 'fname' folder
system("mkdir "+fname+".1");
cd(fname+".1")


%% Transform polyshape to .poly file

% In order for BreZo to understand the domain geometry, the matlab
% polyshape must be transformed to a .poly text file.
polyshape2poly(p,fname);


%% Run Triangle and InputFiles
```

```
% While inside fname directory, run Triangle which is located in /home
% Triangle generates a computational mesh and saves text files inside fname
% directory that will be used by BreZo to understand mesh.
system("../../Triangle/triangle -p -q -a -n -e "+fname);

% Create multiple input files that will be needed by BreZo
InputFiles(fname,slope,Q,ts,out);



%% Prep buildings.txt
system("echo "+fname+".1"+" > buildings.txt");

%% Create .plt file

% makepltfile executable located in /home
!../../makepltfile

% Copy brezo.start file to fname directory
system("cp brezo.start ..");
```

A.2 Buildingsnew Function

```
%%% Buildingsnew.m %%%
%%%%% GENERATING CONFIGURATIONS ALGORITHM %%%%%
```

```
                        % Called by MAIN function
                        % No manual input needed


function [p,phi_r,chi_r] = Buildingsnew(width,L,phi,chi0,sd)

%% Establish constants

% Seed for random number generator
rng(sd);

% C_n for Mermin order parameter calculation
C_n = 4;

% Length of building side
b = 15;

% Minimum acceptable distance between buildings SF*b
SF = 1.2;


%% Domain

% Create initial domain square
x_dom = width;
y_dom = width;
```

```
% Domain boundary
outlinex = [0 x_dom x_dom 0 0];
outliney = [0 0 y_dom y_dom 0];


%% Determine number of buildings depending on porosity
% Each building is ~200 sqm (15m each side)

% Calculate area of building coverage
Ab = (1-phi)*(y_dom^2);

% Determine number of buildings and lattice dimensions
Nsqr = Ab/(b^2);
N = round((Nsqr*x_dom)/y_dom);
yy = round(sqrt(Nsqr)); xx = N/yy;


%% Create Original Mesh

% Generate initial lattice of points
[X,Y] = meshgrid(linspace(0,x_dom,xx),linspace(0,y_dom,yy));
x = X(:); y = Y(:);
pts = [x y];

% Calculate distance between buildings
dis = round(X(1,6)-X(1,5)); dis = dis*.7;
```

```
% Size of buffer zone for Mermin calculations
B = [dis x_dom-dis x_dom-dis dis dis;
    dis dis y_dom-dis y_dom-dis dis];


%% Hybrid Reverse Monte Carlo Algorithm

% Determine initial mermin and error value
chi = mermin_points(x',y',B,C_n);
g = (chi-chi0)^2; G0 = g;

step = 0; chi_t = [];

% Begin Monte Carlo loop
while g > 0.001*G0    % check that error is greater than tolerance
    g0 = g;

    % Select random point to move
    pt = randi(length(pts));

    % Move point randomly in x and y directions
    shftx = (x_dom*rand(1) - x_dom/2)*(1-chi0)*0.1;
    shfty = (y_dom*rand(1) - y_dom/2)*(1-chi0)*0.1;
    x(pt) = x(pt) + shftx; y(pt) = y(pt) + shfty;

    % Check that point is still inside domain
    if inpolygon(x(pt),y(pt),outlinex,outliney) == 0
```

```
    x(pt) = x(pt) - shftx; y(pt) = y(pt) - shfty;
    continue
end

% Calculate mermin of new configuration with moved point
chi = mermin_points(x',y',B,C_n);

% Initialize energy Penalty term
energy = 0;

% Mathematical determination of distances between "buildings"
for k = 1:length(pts)
    if k == pt
        continue
    else
        theta0 = atan2(y(k)-y(pt),x(k)-x(pt));
        theta = abs(theta0);
        d = pnt_dist([x(k),y(k)],[x(pt),y(pt)]);
        if theta < pi/4 || theta > 3*pi/4
            tol_min = b/cos(theta);
            tol_max = SF*tol_min;
        elseif theta > pi/4 && theta < 3*pi/4
            tol_min = b/sin(theta);
            tol_max = SF*tol_min;
        else
            tol_min = b*sqrt(2);
            tol_max = SF*tol_min;
```

```
            end

            % If distance between "buildings" not allowed, change energy
            % term to 1000.
            if (round(d,10) > round(abs(tol_min),10)) & ...
                    (round(d,10) <= round(abs(tol_max),10))
                energy = 1000;
            end
        end
    end

    % Calculate new error value with energy term and new chi
    g = energy + (chi-chi0)^2;

    % Determine whether move can be accepted
    if g > g0
        % Deny move
        x(pt) = x(pt) - shftx; y(pt) = y(pt) - shfty;
        g = g0;
    else
        % Accept move
        chi_t = [chi_t chi];
        step = step + 1;
    end

end
```

```
%% Model Post-Processing

X = []; Y = []; c = 0;

% Duplicate square configuration to fill entire length of channel L
for k = width:width:L
    X = [X x'+(k-width)+(c*(5*b/4))]; Y = [Y y'];
    c = c + 1;
end

% Generate buildings from points
p = Create_Buildings(X',Y',b,b);

% Determine porosity phi_r and mermin chi_r of entire system
phi_r = 1-(area(p)/(width*L));
chi_r = mermin_points(X,Y,B,C_n);

end
```

A.3 Create_Buildings Function

```
%%% Create_Buildings.m %%%
%%%%% GENERATING BUILDINGS ALGORITHM %%%%%
% Called by Buildingsnew function
```

```
                    % No manual input needed
  % This function takes the configuration points and converts them into
                    % buildings as a polyshape file


function P = Create_Buildings(x,y,w,l)
% x,y = x and y of center of building (must be column)
% w,l = width and length of building
%   P = output polyshape file

% Coordinates for 4 vertices of one building
xb = [-w/2 w/2 w/2 -w/2]';
yb = [-l/2 -l/2 l/2 l/2]';

% Create polyshape from one building
building = polyshape(xb,yb);
P = polyshape();

% Generate all buildings in configuration using initial building and point
% coordinates from input
for ii = 1:size(x,1)
    b = translate(building,[x(ii),y(ii)]);
    P(ii) = b;
end
P = union(P);

% Remove any outliers
```

```
P = rmslivers(P,3);

% Remove any holes
P = rmholes(P);

end
```

A.4 InputFiles Function

```
%%% InputFiles.m %%%
%%%%% GENERATE NECESSARY INPUT FILES %%%%%
% Called by MAIN function
% No manual input needed
  % This function creates five necessary textfiles, generated from "shell"
  % templates. These files are named with the specific fname of the current
  % working configuration. This is necessary for BreZo to recognize and
  % read the files.
  % This function also assignes a bed slope to the channel by changing
  % elevations of each node within the grid.


function InputFiles(filename,slope,Q,ts,out)

%% Create .bc file
% The .bc file is a textfile used by BreZo to assign boundary conditions to
```

```
% the simulation.
% Here we define a volumetric flow for the upstream boundary.

% Read shell textfile into cell A
fid = fopen('shell.bc','r');
i = 1;
tline = fgetl(fid);
A{i} = tline;
while ischar(tline)
    i = i+1;
    tline = fgetl(fid);
    A{i} = tline;
end
fclose(fid);

% Change inflow Q [m^3/s]
A{12}(32:36) = sprintf('%-5.f',Q);

% Recreate .bc file with new name that corresponds to working configuration
name = sprintf('%s.1.bc',filename);
fileID = fopen(name,'w');
for i = 1:length(A)-1
    fprintf(fileID,'%c',A{i});
    fprintf(fileID,'\n');
end
fclose(fileID);
```

```
%% Create Qinput.dat file
% The Qinput.dat file is a textfile used by BreZo to assign an input time
% series that defines the volumetric flow and run time of the simulation.
% This file is optional and can be ignored by making edits to the .input
% file to assign a constant bc instead of reading from this file.
cd ../inputtimeseries/

% Read shell textfile into cell C
fid = fopen('Qinput.dat','r');
i = 1;
tline = fgetl(fid);
C{i} = tline;
while ischar(tline)
    i = i+1;
    tline = fgetl(fid);
    C{i} = tline;
end
fclose(fid);

% Change inflow Q [m^3/s]
C{4}(57:60) = sprintf('%-4.f',Q);
C{5}(57:60) = sprintf('%-4.f',Q);

% Change run time [hr]
C{5}(33:34) = sprintf('%-2.f',ts);
```

```
% Recreate Qinput.dat file (no need to rename)
fileID = fopen('Qinput.dat','w');
for i = 1:length(C)-1
    fprintf(fileID,'%c',C{i});
    fprintf(fileID,'\n');
end
fclose(fileID);

cd ("../"+filename+".1")



%% Create .input file
% The .input file is a textfile used by BreZo to assign input values for a
% range of options.
% Here, we only change the values for run time and output format

% Read shell textfile into cell B
fid = fopen('shell.input','r');
i = 1;
tline = fgetl(fid);
B{i} = tline;
while ischar(tline)
    i = i+1;
    tline = fgetl(fid);
    B{i} = tline;
end
fclose(fid);
```

```
B{end} = '';

% Change run time [hr]
B{8}(20:21) = sprintf('%-2.f',ts);

% Change output format
B{18}(4) = sprintf('%d',out);

% Create .input file with new name that corresponds to working configuration
name = sprintf('%s.1.input',filename);
fileID = fopen(name,'w');
for i = 1:length(B)
    fprintf(fileID,'%c',B{i});
    fprintf(fileID,'\n');
end
fclose(fileID);


%% Create .wdlog file
% Simply creates .wdlog file with new configuration name

name = sprintf('%s.1.wdlog',filename);%(1:end-4));
fileID = fopen(name,'w');
fclose(fileID);


%% Create .start file
```

```
% The brezo.start file is a textfile used by BreZo to begin the program.
% BreZo reads the name of the configuration from this file to understand
% what filenames to look for.
% Here we simply copy the filename into a shell brezo.start file

% Read shell textfile into cell D
fid = fopen('brezo.start','r');
i = 1;
tline = fgetl(fid);
D{i} = tline;
while ischar(tline)
    i = i+1;
    tline = fgetl(fid);
    D{i} = tline;
end
fclose(fid);

% Change grid file prefix
D{4} = sprintf('%s.1',filename);%(1:end-4));

% Recreate brezo.start file (no need to rename)
fileID = fopen('brezo.start','w');
for i = 1:length(D)
    fprintf(fileID,'%c',D{i});
    fprintf(fileID,'\n');
end
fclose(fileID);
```

```
%% Edit .1.node file for changing elevation
% The .1.node file is a textfile used by BreZo to understand the location of
% all nodes within a computational mesh. This was originally created by the
% mesh generation tool Triangle.
% Here, we add the bed slope of the channel by assigning elevation values
% to all nodes.

% Read filename.1.node file from directory into cell nds
name = sprintf('%s.1.node',filename);%(1:end-4));
fid = fopen(name,'r');
nds_header = textscan(fid,'%f %f %f %f',1);
nds = textscan(fid,'%f %f %f %f %f','Headerlines',1,'Delimiter','\n');
fclose(fid);
nds = cell2mat(nds);

% Edit z data of all nodes to new elevation with bed slope
%             z = (x_max-x) * slope

nds(:,4) = (max(nds(:,2))-nds(:,2))*slope;

% Recreate filename.1.node file (no need to rename)
fileID = fopen(name,'w');
fprintf(fileID,'%.0f \t',nds_header{:});
fprintf(fileID,'\n');
fprintf(fileID, '%.0f \t %f \t %f \t %f \t %.0f \n',nds');
```

```
fclose(fileID);



end
```

A.5 Polyshape2poly Function

```
                    %%% polyshape2poly.m %%%
          %%%%% CONVERT MATLAB POLYSHAPE TO .POLY FILE %%%%%
                    % Called by MAIN function
                    % No manual input needed
% This function transforms the Matlab polyshape generated from
% Buildingsnew.m into a .poly textfile for the mesh generator to read. It
% specifies all verticies, segments, and holes of the configuration.



function polyshape2poly(p,filename)


%% Pre-Process Data from polyshape
% Assign X and Y values from polyshape
X = p.Vertices(:,1); Y = p.Vertices(:,2);


% Define bounding box for points
bb = [min(X) min(Y); max(X) max(Y)];
spacing = 5;
XX = [bb(1,1)-spacing 510 bb(2,1)+spacing ...
```

```
            bb(2,1)+spacing 510 bb(1,1)-spacing];
YY = [bb(1,2)-spacing bb(1,2)-spacing bb(1,2)-spacing ...
            bb(2,2)+spacing bb(2,2)+spacing bb(2,2)+spacing];

% Define number of regions
N = p.NumRegions;



%% Create new .poly file

% Create empty filename.poly file
name = sprintf('%s.poly',filename);
fileID = fopen(name,'w');

% Create arrays defining all verticies of geometry
Xnan = rmmissing(X);
Ynan = rmmissing(Y);
XX = [XX Xnan']; YY = [YY Ynan'];
XY = [XX' YY'];

% Create arrays defining segments and holes of geometry
segment = [1 2;2 3;3 4;4 5;5 6;6 1];
XYholes = zeros(N,2);
r = regions(p); cnt = 6;
for ii = 1:N
    x = r(ii).Vertices(:,1); y = r(ii).Vertices(:,2);
    nv = size(x,1);
```

```
    bnds = zeros(nv-1,2);
    for jj = 1:nv-1
        bnds(jj,:) = [jj+cnt jj+cnt+1];
    end
    segment = [segment; bnds; nv+cnt cnt+1];
    cnt = cnt + nv;
    Xcenter = min(x)+rand(20,1)*(max(x)-min(x));
    Ycenter = min(y)+rand(20,1)*(max(y)-min(y));
    IND = isinterior(r(ii),Xcenter,Ycenter);
    ind = find(IND,1);
    XYholes(ii,:) = [Xcenter(ind),Ycenter(ind)];
end

% Define line separating fine and coarse mesh
line_ind = find(XX>505 & XX<=510);
[~,I] = sort(YY(line_ind)); B = line_ind(I);
line_seg = [B(1:end-1)' B(2:end)'];
segment = [segment; line_seg];

% Compile all verticies, segments, and holes into textfile
XY = [(1:size(XY,1))' XY zeros(size(XY,1),1) ones(size(XY,1),1)];
XYholes = [(1:N)' XYholes];
segment = [(1:size(segment,1))' segment ...
                [ones(4,1); zeros(size(segment,1)-4,1)]];
segment(3,4) = 2; segment(6,4) = 3;
% Print to .poly file
txt = [size(XY,1) + "  2   1   1"
```

```
    string(num2str(XY))
    size(segment,1) + "  1"
    string(num2str(segment))
    string(N)
    string(num2str(XYholes))
    "2"
   "1  " + string(num2str([XY(1,2)+2,XY(1,3)+2]))+ "  1  20"
   "2  " + string(num2str([XY(3,2)-2,XY(3,3)+2]))+ "  1  50"];
fprintf(fileID,'%s\n',txt);
fclose(fileID);

end
```

A.6 Mermin_Points Function

```
                    %%% mermin_points.m %%%
       %%%%% CALCULATE MERMIN ORDER VALUE FOR SET OF POINTS %%%%%
                  % Called by Buildingsnew function
                     % No manual input needed
% This function calculates the mermin order value of a set of points.


function chi = mermin_points(X,Y,B,C_n)

X = X'; Y = Y';
```

```
coords = [X,Y];
chi_jsum = 0;

% Find points (idx) that are inside the inner boundary.
% These points are regular points and are the only points that
% are considered in the following loop
idx = inpolygon(X,Y,B(1,:),B(2,:)); idx = find(idx);
nn=size(idx,1);

% Begin looping through all regular points to find mermin value for each
% point.
for j = idx'
    sum_pj = 0;

    % Find distance between current point j and all points (including
    % points outside of the inner boundary).
    distJ = pnt_dist(coords,coords(j,:));

    % Find nearest neighbors to current point j
    [~,I] = mink(distJ,C_n+1);
    NN = I(2:end);

    % Find angles between current point and nearest neighbors, and use
    % angle theta to calculate value pj. Add value of pj to sum.
    % pj = exp ( i C_n theta_jk )
    for k = NN'
        theta = atan2(Y(k)-Y(j),X(k)-X(j));
```

```
        if theta < 0
            theta = (2*pi)+theta;
        end
        pj = abs(real(exp(i*C_n*theta)));
        sum_pj = sum_pj+pj;
    end

    % Average values of pj for nearest neighbors to find final mermin value
    % of specific regular point.
    chi_j = (1/C_n*sum_pj);
    chi_jsum = chi_jsum + chi_j;
end

% Average mermin values of all regular points to find mermin spatial order
% of entire system
chi = (1/nn)*chi_jsum;

end
```