LEBANESE AMERICAN UNIVERSITY

On Hughes' Model for Pedestrian Traffic: Interpreting Real-World Results

By

Fadi J Kaldawi

A thesis

Submitted in partial fulfillment of the requirements

for the degree of Master of Science in

Applied and Computational Mathematics

School of Arts and Science

July 2020

©2020

Fadi J Kaldawi

All Rights Reserved

LAU
الجامعة اللبنانية الأميركية
Lebanese American University

THESIS APPROVAL FORM

Student Name: Fadi J Kaldawi I.D. #: 201402403

Thesis Title: On Hughes' Model for Pedestrian Traffic: Interpreting Real-World Results

Program: Masters of Science in Applied and Computational Mathematics

Department: Computer Science and Mathematics

School: Arts and Science

The undersigned certify that they have examined the final electronic copy of this thesis and approved it in Partial Fulfillment of the requirements for the degree of:

Masters of Science in the major of Applied and Computational Mathematics

Thesis Advisor's Name: Dr. Nader El-Khatib

Signature [redacted] Date: 09 / 07 / 2020 (Day / Month / Year)

Committee Member's Name: Dr. Chadi Nour

Signature: [redacted] Date: 08 / 07 / 2020 (Day / Month / Year)

Committee Member's Name: Dr. Jean Takche

Signature: [redacted] Date: 08 / 07 / 2020 (Day / Month / Year)

LAU
الجامعة اللبنانية الأميركية
Lebanese American University

THESIS COPYRIGHT RELEASE FORM

LEBANESE AMERICAN UNIVERSITY NON-EXCLUSIVE DISTRIBUTION LICENSE

By signing and submitting this license, you (the author(s) or copyright owner) grants the Lebanese American University (LAU) the non-exclusive right to reproduce, translate (as defined below), and/or distribute your submission (including the abstract) worldwide in print and electronic formats and in any medium, including but not limited to audio or video. You agree that LAU may, without changing the content, translate the submission to any medium or format for the purpose of preservation. You also agree that LAU may keep more than one copy of this submission for purposes of security, backup and preservation. You represent that the submission is your original work, and that you have the right to grant the rights contained in this license. You also represent that your submission does not, to the best of your knowledge, infringe upon anyone's copyright. If the submission contains material for which you do not hold copyright, you represent that you have obtained the unrestricted permission of the copyright owner to grant LAU the rights required by this license, and that such third-party owned material is clearly identified and acknowledged within the text or content of the submission. IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED BY AN AGENCY OR ORGANIZATION OTHER THAN LAU, YOU REPRESENT THAT YOU HAVE FULFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH CONTRACT OR AGREEMENT. LAU will clearly identify your name(s) as the author(s) or owner(s) of the submission, and will not make any alteration, other than as allowed by this license, to your submission.

Name: Fadi J Kaldawi

Signature: [redacted] Date: 7 / 7 / 2020
Day Month Year

LAU
الجامعة اللبنانية الأميركية
Lebanese American University

PLAGIARISM POLICY COMPLIANCE STATEMENT

I certify that:

1. I have read and understood LAU's Plagiarism Policy.
2. I understand that failure to comply with this Policy can lead to academic and disciplinary actions against me.
3. This work is substantially my own, and to the extent that any part of this work is not my own I have indicated that by acknowledging its sources.

Name: Fadi J Kaldawi

Signature: [redacted]

Date: 7 / 7 / 2020
Day Month Year

DEDICATIONS

To my loving family and friends.

ACKNOWLEDGMENTS

This thesis would not have been possible without the guidance, support, and contributions of many people in my life. Firstly, I'd like to thank Dr. Nader El-Khatib, my research advisor, for not only introducing me to the topic I would end up so invested in, but also his ongoing support through the most trying of times. Through a life-changing revolution in the country and a global pandemic in the year of working on this research, words cannot describe Dr. El-Khatib's unparalleled support in times of uncertainty and confusion, and his support through the many steps it took to reach this moment can't be understated. I would also like to thank my other committee members, Dr. Chadi Nour and Dr. Jean Takche, for their support and contributions as well.

On a more personal level, I would like to thank my loving parents for the guidance on an emotional level. Once again, the times surrounding the build-up of this thesis have been as trying as ever, and in times where hope appears to be lost and motivation falls short, I could always rely on the comfort and love of my parents to remember to press on. Last but not least, I extend my thanks to my best friend, Omar, who has been there for me since we were in the 8th grade, and still has my back 11 years later. He's like the brother I wish I'd had, and his words of encouragement and inspiration help keep my head up and stay as dedicated as I am.

On Hughes' Model for Pedestrian Traffic: Interpreting Real-World Results

Fadi J Kaldawi

ABSTRACT

In this thesis, we explore the modeling and application of the pedestrian flow model proposed by Roger L. Hughes in 2002, a system of a hyperbolic conservation law to describe crowd densities (ρ) and an Eikonal equation to describe the path potential (ϕ) of this crowd:

$$\rho_t - \operatorname{div}(\rho f^2(\rho)\nabla\phi) = 0$$

$$||\nabla\phi|| = \frac{1}{f(\rho)}$$

Throughout this work, we explain this model in the context of a one dimensional walking facility, like a bridge or a hallway, and a two dimensional one, such as an open room with obstacles and obstructions. We revisit the motivation for the model, as well as properties and qualities of the resulting weak entropy solutions, some of which will aid in understanding numerical results. After this, we describe numerical methods to use in order to solve the eikonal equation for the path potential, then use this quantity to solve the conservation law for the density after a certain time step. With these methods established, we proceed to provide meaningful simulations in both the 1D and 2D cases, describing what our results mean from a mathematical perspective, then a real-life explanation, alongside a brief critique of the simulated situations with comments on how to improve the walking facility conditions from a design perspective.

Keywords: Pedestrian flow, Pedestrian traffic, Conservation law, Eikonal equation, Hughes model, Fast-sweeping method, Godunov scheme, Characteristics

TABLE OF CONTENTS

Chapter **Page**

1 Introduction **1**

2 Hughes' Model: Motivation and Existence Theory of Solutions **6**

2.1 The One-Dimensional Case 6

2.2 Analysis of the 1D Solutions 8

2.3 The Two-Dimensional Case 12

3 Numerical Solution Algorithms **17**

3.1 The One-Dimensional Case 17

3.1.1 The Fast-Sweeping Method 18

3.1.2 The Godunov Scheme 20

3.2 The Two-Dimensional Case 21

3.2.1 The Godunov Flux in 2D 22

3.2.2 The Fast-Sweeping Method 23

3.2.3 The third-order Total Variation Diminishing Runge-Kutta time discretization 24

4 Simulations and Interpretations **26**

4.1 The One-Dimensional Case 26

4.2 The Two-Dimensional Case 34

5 Conclusions and Moving Forward **42**

References **45**

Appendix **48**

LIST OF FIGURES

Figure Number **Page**

4.1 Snapshots of the change in ρ for $\rho_I(x) = 0.8$ and Dirichlet boundary condition $\rho_D = 0.5$ 27

4.2 Snapshots of the change in ρ for a one-sided initial condition . . . 29

4.3 Snapshots of the change in ρ for the new piecewise initial condition 31

4.4 Snapshots of a single-exit case where $\rho_I(x) = 0.75$ with Dirichlet boundary condition $\rho(1, t) = 0.5$. 33

4.5 Snapshots of our first 2D simulation where $\rho_I(x) = 0.4$ in the rectangle $[40, 60] \times [20, 30]$, showing the pedestrian flow in time at three different intervals. 36

4.6 Layout of the walking facility at $t = 0$ with details of the obstructions and initial crowd location. 38

4.7 Snapshots of our second 2D simulation (with obstacles and a smaller exit boundary) where $\rho_I(x) = 0.6$ in the rectangle $[0, 10] \times [10, 40]$, showing the pedestrian flow in time at four different times. 39

Chapter 1

Introduction

In a world where population growth is on the rise, and overcrowding in public places and expansive locales is becoming more and more of a prevalent problem, the study of pedestrian behavior proves to be important. Without proper care given to this topic, we not only ignore the potential for real-life pedestrian-related disasters that can happen in critical circumstances, but we also fail to learn from these mistakes that have happened throughout history. An example given in [1] is the overcrowding incident that occurred during the nineties in Saudi Arabia's Jamarat Bridge, due to high crowd densities and cramming issues during the time of the pilgrimage to Mecca. This was modeled mathematically at the time, see [2]. Another example from history, which has been studied from a mathematical crowd dynamics standpoint (see [3]), goes back to the Hundred Years War: the Battle of Agincourt, a march lead by King Henry V which directed a massive 11,000-man army from Southampton to Normandy. A more recent event of the past was the opening day of the London Millenium Footbridge in the year 2000, when the bridge was closed on day one due to lateral vibrations occurring from the crowds walking across.

Not only does studying pedestrian traffic using mathematical modeling help us understand past disasters, but it also helps us in the present to build a better understanding of how people behave in large crowds and, in turn, learn from his-

tory to prevent said disasters in the future by making the necessary changes to improve. Such studies are extremely important in events in which large crowds are gathered in one, oftentimes confined, location, such as social gatherings, political demonstrations, Black Friday sales every November, and many more. Similarly, the manner in which people behave or travel should be carefully studied in emergency situations like fires, earthquakes, hurricanes, and other disasters, especially when considering emergency exits and protocols designed to avoid further pedestrian complications. Even in more mundane instances, such as a simple grocery store or an airport terminal, it is interesting to examine the behavior of the average person or crowd of people in order to anticipate path choices and route strategies. For an even tamer example, look no further than computer games that try to emulate pedestrian actions and reactions to a dynamic world around them (example: Grand Theft Auto); in such examples, a solid mathematical model to work with can be a great help in perfecting a competent AI in these games.

When we want to model pedestrian behavior and flow, many factors come into play that will dictate and describe the movement of people within the bounds of a certain walking facility. Some of these attributes include the average walking speed, varying heights/sizes of people, local conditions such as nearby dense crowds or the conditions of the walking facility itself such as an unexpected divider or a dead end up ahead. Plenty of models reflecting the relationship between pedestrian speed, density, and flow, for instance, have been formulated and detailed in past literature, some of which including the Greenshields, Underwood, bell-shaped, and other models as seen in [4]. The walking facility itself, with its unique conditions and attributes, also factors in to the speed-density relationship; conditions will vary depending on the confines of the location, whether it's sidewalks, stairways, or a more open room, and each case would need a particular speed-density model to accompany it [5]. That being said, finding a concrete model to accurately fit a certain scenario is not a 'one size fits all' concept. Differing factors and features, like but not limited to the ones listed here, can greatly

impact how a particular model takes shape.

When studying pedestrian flow, it is also important to note the two different viewpoints to take into consideration, the first being the distinction between the microscopic and the macroscopic models. In the microscopic model, individual people are considered as their own entities and are treated as such in the calculation. This has its uses examples like queuing models [6], where the movement of the individual is more interesting to analyze than that of the masses. In contrast, the macroscopic model considers larger groups of pedestrians as one entity, neglecting the individual movement of each individual in these clusters. Macroscopic modeling is especially useful in studies of fluid and gas dynamics [7] and, as a topic closer to home in this paper, traffic models [8]. A recent study by [9] improves upon a previously-existing model developed by R.L. Hughes (which we will see later), and explores a one-dimensional case in, first, the macroscopic case, then the microscopic case. Their work on these models is a quintessential example of comparing and contrasting the same situation in both senses.

The second important viewpoint is the difference between a predictive and reactive user equilibrium model. This concept is related to how people devise a route strategy, in other words, how they plan to get from point A to point B. Sources like [10] describe a 'predictive' user equilibrium model, which assumes that pedestrians have perfect knowledge about their surroundings and environment to allow them to make a more calculated, idealistic decision for a route strategy. On the other hand, a more realistic situation is one where pedestrians don't have this perfect information, and may be unfamiliar with the walking facility or the potential circumstances surrounding it. It also takes into consideration sudden, unexpected changes such as overcrowding or an exit closing entirely. In such a case, they base their decision-making and route strategy on instantaneous information picked up on the spot and use that to dictate their decisions. This user equilibrium model is called a 'reactive' one, and is described further with a comparison with the predictive one found in [11].

With this in mind, we turn to a pedestrian flow model proposed by R. L. Hughes [12], a model that views crowds as a 'thinking fluid' (i.e., a macroscopic model) and assumes that people make decisions based on instantaneous information (i.e., a reactive user equilibrium model). The model is defined by Hughes as follows:

$$\rho_t - \operatorname{div}(\rho f^2(\rho)\nabla\phi) = 0 \tag{1.1}$$

$$|\nabla\phi| = \frac{1}{f(\rho)} \tag{1.2}$$

Here, x denotes the position in some region $\Omega \in \mathbb{R}^d$. $t \geq 0$ denotes the time. $\rho(x,t)$ is the crowd density at a given position and time. ϕ is known as the potential, and determines the 'common sense' of the crowd, which we will discuss in Chapter 2. $f(\rho) = 1 - \rho$, the right hand side of the Eikonal equation, models the existence of a maximum density for future numerical simulations, normalized at 1, and also impacts the maximum pedestrian walking speed, which we explore later in this paper.

For the rest of this paper, we revisit Hughes' model in both a one dimensional case - that of a hallway or corridor - and a two dimensional case - that of a more free, 2D single-floored room confined between four walls - and revisit previously established revisions of the model when we discuss methods of solving numerically. We establish motivation for both cases and discuss the existence of weak entropy solutions for the system in 1D. We explore existing numerical algorithms for finding solutions to the system of equations (1.1) and (1.2) in both one and two dimensions, and run different simulations to investigate how varying initial data sets (such as initial conditions for pedestrian densities, a change in the allowed pedestrian speed, and other spatial conditions like obstructions or obstacles in the 2D case) behave under these algorithms. Once we've performed these simulations, we then reflect upon each case, giving an explanation from a mathematical and real-life perspective, and follow up with a critique on the surrounding conditions (like initial densities, obstacles, choices of exits, and more) to see how the design

of the walking facility can be improved upon. Finally, as we wrap things up, we discuss how these results can contribute to a better society and improve on life as a whole, as well as mention future research directions and options for this field.

Chapter 2

Hughes' Model: Motivation and Existence Theory of Solutions

In this section, we discuss and provide details and motivation for Hughes' model in both a one-dimensional and two-dimensional case. Do note that we represent these cases slightly differently due to the added detail and complexity present in the two-dimensional case, though at its core, the two models branch from the same base of R.L. Hughes' work (which we also note in Chapter 4 when we perform simulations). Much of the motivation and work for both cases is inspired by Hughes' original work on the continuum model in his 2002 article (see [12]), which can be referred back to for the in-depth formulation of this work.

2.1 The One-Dimensional Case

We begin with an exposition of the one-dimensional Hughes' model, denoted by the pair of equations (1.1) and (1.2), restated below: a scalar conservation law and an eikonal equation respectively:

$$\rho_t - \operatorname{div}(\rho f^2(\rho)\nabla\phi) = 0$$

$$|\nabla\phi| = \frac{1}{f(\rho)}$$

We describe the derivation of Hughes' model as a motivation, namely following the likes of the work provided in [13]. To begin, we consider a one-dimensional domain $\Omega = [a, b]$ (we take this as $[-1, 1]$ in our future methods and computations) with two exits at $x = a$ and $x = b$. This walking facility can be visualized as a 1D structure or room such as a bridge or hallway. Our two variables are the crowd density $\rho(x, t)$ and the potential $\phi(x, t)$. First, we note that the $\rho(x, t)$ satisfies the continuity equation:

$$\rho_t(x, t) - \nabla \cdot (\rho(x, t)\vec{v}(x, t)) = 0 \tag{2.1}$$

Here, the velocity $\vec{v}(x, t)$ is represented by the following 'polar decomposition notation':

$$\vec{v}(x, t) = |\vec{v}(x, t)|\vec{z}(x, t),\ |\vec{z}(x, t)| = 1 \tag{2.2}$$

$\vec{z}$ is the directional unit vector, parallel to the gradient of the potential, ϕ, the variable that dictates the 'common sense of the task', represented in the model by the boundary of our domain $\partial\Omega$. Recall that the model is macroscopic; that being said, we consider crowds of pedestrians as a single homogeneous entity with a common goal and behavior. For this model, our assumptions of how people think or behave - in other words, what this 'task' dictates - is that they try to minimize travel time to get from point A to point B, and that they tend to account for locations of high densities and try to avoid them in an instantaneous, reactive manner. We represent this in the form of an Eikonal equation of the potential ϕ, along with an associated boundary condition, as follows:

$$|\nabla\phi| = \frac{1}{1 - \rho},\ \phi|_{\partial\Omega} = 0 \tag{2.3}$$

The arrangement of the level sets of ϕ is 'density driven', and it follows that:

$$\vec{z}(x,t) = \frac{\nabla\phi(x,t)}{|\nabla\phi(x,t)|} = (1-\rho)\nabla\phi \tag{2.4}$$

Replacing $\vec{z}$ in (2.2), we return $\vec{v}$ to the continuity equation (2.2) to find that it is, indeed, satisfied, and corresponds to Hughes' model given by equations (1.1) and (1.2).

2.2 Analysis of the 1D Solutions

Before we move to the motivation for the 2D case, we address some noteworthy properties about the system of equations (1.1) and (1.2) and its respective solutions. Some of these properties will help us further understand the numerical results that we will see in Chapter 4, while others are important for understanding the underlying proof for the existence of these solutions.

To start, we follow a common modification to the 1D model as done in sources like [13], [14], and [15]. This will ease our calculations when we perform numerical simulations later, and make for an easier study of possible initial conditions and the resulting behavior. To this end, we make an assumption to only take into consideration the 'viscosity solution' for the Eikonal equation (1.2) (see [16] for more about viscosity solutions to equations of this form). As a direct result, we witness a unique turning point in our solutions, call it $x_0(t)$ and it is given implicitly as follows:

$$\int_{-1}^{x_0(t)} \frac{1}{f(\rho)}dx = \int_{x_0(t)}^{1} \frac{1}{f(\rho)}dx \tag{2.5}$$

In terms of interpreting our results from a real-life perspective, the 'turning point' in 1D can be seen as the point where crowds split up into two different paths. We see this split very clearly in the figures in Chapter 4, and bring it up again in our in-depth interpretation of each simulation.

From this, we conclude that, using the conditions surrounding this turning

point, and the fact that $|\phi_x| = \phi_x \text{sgn}\phi_x$ (where sgn represents the sign function), the conservation law (1.1) becomes:

$$\rho_t - (\rho f(\rho)\text{sgn}\phi_x)_x = 0 \tag{2.6}$$

We will use equation (2.6) instead of (1.1) for our work on Hughes' model in 1D throughout this paper.

Next, we note some of the characteristic findings in order to better understand our results. Experiments performed in [13] compared different values of constant initial density data across the whole computational domain, namely values that were under, over, and exactly 1/2. We state the results from this source that we will use in our interpretations later:

- For $\rho(x, 0) = 1/2$, characteristics are vertical lines parallel to the vertical axis.

- For $\rho(x, 0) < 1/2$, characteristics point away from the turning point. In this case, there is no need for extra boundary conditions for the crowd density.

- For $\rho(x, 0) > 1/2$, characteristics point inward relative to the turning point. In this case, we need additional Dirichlet boundary conditions for the crowd densities to not only make up for missing information on the boundaries, but also to accurately visualize characteristic data in our results (as well as make sense of it realistically). We will implement this practice in our simulations later (Note: typically, these boundary conditions are less than or equal to 1/2 in this context).

Finally, we briefly address the existence of weak entropy solutions, and present some of the properties surrounding it. A thorough examination of these solutions and their properties, which this overview is based off of, can be found in [14]. While the existence of solutions for both conservation laws and eikonal equations (in this case, viscosity solutions, as mentioned previously) individually has been

well-established in a variety of past literature, determining such an existence proof for the coupled system (2.6) and (1.2), however, is not as straightforward. In such a case, what we desire is the existence of a class of solutions called 'weak entropy solutions'.

First, we set $F(x,t,\rho) = \rho f(\rho)\text{sgn}\phi_x$, and for simplicity, we deal with the following system:

$$\rho_t - F(x,t,\rho) = 0,\ x \in \Omega,\ t > 0 \tag{2.7}$$

$$|\phi_x| = \frac{1}{f(\rho)} \tag{2.8}$$

Equations (9) and (10) are accompanied by the following initial and boundary conditions:

$$\rho(x,0) = \rho_I,\ x \in [0,1] \tag{2.9}$$

$$\rho(\pm 1,t) = 0,\ \phi(\pm 1,t) = 0,\ t > 0 \tag{2.10}$$

We denote the system defined by equations (2.7) through (2.10) as system (*) for ease of notation. We now provide two definitions, as seen in [14]: one to give meaning to what a weak solution to equations (2.7) and (2.8) is, and one that adds in the entropy condition for conservation laws with discontinuous fluxes.

Definition 2.1: A function $\rho \in L^\infty(\mathbb{R}^+ \times \Omega)$ is a weak solution to problem (*) if, for all $\psi \in C_c^\infty(\mathbb{R} \times \Omega)$, the following equality holds:

$$\int_0^\infty \int_{-1}^1 (\rho\psi_t + F(x,t,\rho)\psi_x)dxdt + \int_{-1}^1 \rho_I(x)\psi(x,0)dx = 0$$

If we make another assumption that $\rho \in C^0(\mathbb{R}^+; L^1(\Omega)) \cap BV(\mathbb{R}^+ \times \Omega)$, the ρ is a weak solution to system (*) if and only if it satisfies the following:

$$\rho_t - (\rho f(\rho))_x = 0,\ t > 0,\ x \in]-1, x_0(t)[$$

$$\rho_t + (\rho f(\rho))_x = 0,\ t > 0,\ x \in]x_0(t), 1[$$

$$\rho(x,0) = \rho_I,\ x \in]-1, 1[$$

where $x_0(t)$ is the turning point of the system, defined implicitly as before in equation (2.5). This addition is useful when we add in the entropy condition, as we do for the next definition.

Definition 2.2: A function $\rho \in C^0(\mathbb{R}^+; L^1(\Omega)) \cap BV(\mathbb{R}^+ \times \Omega)$ is a weak entropy solution of system (*) if the Kruzkov entropy inequality, stated below, holds for all $k \in [0,1]$ and for all test functions $\psi \in C_c^\infty(\mathbb{R} \times \Omega; \mathbb{R}^+)$:

$$\begin{aligned} 0 \leq & \int_0^\infty \int_{-1}^1 (|\rho - k|\psi_t + \Phi(x,t,\rho,k)\psi_x)dxdt + \int_{-1}^1 |\rho_I - k|\psi(x,0)dx \\ & + \operatorname{sgn}(k) \int_0^\infty (f(\rho(1^-,t)) - f(k))\psi(1,t)dt \\ & + \operatorname{sgn}(k) \int_0^\infty (f(\rho(-1^+,t)) - f(k))\psi(-1,t)dt \\ & + 2\int_0^\infty f(k)\psi(x_0(t),t)dt \end{aligned} \tag{2.11}$$

Here, $\Phi(x,t,\rho,k) = \operatorname{sgn}(\rho - k)(F(x,t,\rho) - F(x,t,k))$ and the boundary conditions are in what is called 'weak form' (see [17]), where the traces of the solutions satisfy:

$$f(\rho(-1^+,t)) \geq f(k), \text{ for all } k \in [0, \rho(-1^+,t)]$$

$$f(\rho(1^-,t)) \geq f(k), \text{ for all } k \in [0, \rho(1^-,t)]$$

By [15], this implies that $\rho(-1^+,t) \leq 1/2$ and $\rho(1^-,t) \leq 1/2$. (Note that this is directly related to the study of whether or not we require Dirichlet boundary conditions for the densities, which we've seen earlier in this section.)

The important step towards proving that weak solutions (as defined in Definition 2.1) to system (*) also satisfy the entropy condition (2.11) is to recover the Rankine-Hugoniot (RH) condition, as defined below:

$$F(\rho^+) + F(\rho^-) = \dot{x}_0(t)(\rho^+ - \rho^-)$$

where $\rho^+ = \rho(x_0^+(t),t)$ and $\rho^- = \rho(x_0^-(t),t)$ are the left and right traces of ρ at the turning point $x_0(t)$. If we take a suitable test function ψ that vanishes along

the turning point, we can easily recover the RH condition. If $\rho(\cdot, t)$ is continuous at the turning point, then the RH condition simplifies to $f(\rho(x_0(t), t)) = 0$, and $\rho(x_0(t), t) \in \{0, 1\}$.

The work surrounding these weak entropy solutions laid out in [14] describes in detail the entropy condition at the jump discontinuity, the characteristic speeds before and after the turning point (depending on which side has the higher crowd density), the maximum principle that bounds the crowd density from above (prevents the value of ρ from exploding), and, like we've mentioned before, recovering the RH condition to arrive at our conclusion. The reader may refer back to this source for the in-depth proof of each property and step.

2.3 The Two-Dimensional Case

The approach for the two-dimensional case starts similarly to our work in one dimension; however, we make adjustments and additional considerations in our notation in order to better understand the sense of people's movement in 2D, as well as highlight behavioral attributes such as walking speed and pedestrian discomfort. This modeling and motivation follows similarly to the one performed by Ling Huang in reference [18], which can be referred back to for more details on this reformulation. Note that details like the existence of solutions and mathematical properties of the final system of equations are not mentioned here; for more details on those, the reader may refer to Hughes' original work [12].

We consider a domain Ω, this time, meant to describe an open room or space, like a railway, foyer, public park/attraction, or any other open walking facility of a similar nature. Its boundary, call it Γ, is split into three parts for ease of notation. We define Γ_o to be the entrance, where people will enter Ω (Reference [18] uses this entry boundary, but in Chapter 4, we focus more on evacuation than entry and passage). We then assign Γ_d to be the exit(s), or the destination that people will evacuate Ω from, and Γ_h to be the hard boundaries, such as walls or obstacles, that people navigate around or away from to travel from point A

(could be Γ_o, but for the purposes of our simulations in Chapter 4, we will take the case where point A is any interior point of Ω) to point B (the exit Γ_d).

Similar to the 1D case, our motivational work begins with the continuity equation, this time in two dimensions:

$$\rho_t(x, y, t) + \nabla \cdot (\rho(x, y, t)\vec{v}(x, y, t)) = 0 \tag{2.12}$$

Much of the variables we defined before return, but with a definition in 2D; $\rho(x, y, t)$ is the pedestrian density at a certain location and time, $(x, y) \in \Omega$ is a location in the domain, t is the time in seconds, and $\vec{v}(x, y, t) = (v_1(x, y, t), v_2(x, y, t))$ is the velocity vector at a certain location and time with an x and y component. We further define a vector $\vec{F} = \rho(x, y, t)\vec{v}(x, y, t) = (F_1(x, y, t), F_2(x, y, t))$ for ease of use later.

In a similar manner to the 1D case, we define the potential function $\phi(x, y, t)$, which determines the path that pedestrians will choose to traverse, which is used in an Eikonal equation later. Like before, the velocity vector is represented in a polar decomposition form:

$$\vec{v}(x, y, t) = ||\vec{v}(x, y, t)||\vec{z}(x, y, t),\ ||\vec{z}(x, y, t)|| = 1 \tag{2.13}$$

Like before, $\vec{z}$ is the directional unit vector, parallel to the gradient of the potential ϕ. In a similar sense to before, we can view this unit vector as:

$$\vec{z}(x, y, t) = \frac{\nabla\phi(x, y, t)}{||\nabla\phi(x, y, t)||} \tag{2.14}$$

We can now view the x and y velocities as follows:

$$v_1(x, y, t) = u(x, y, t)\frac{\phi_x(x, y, t)}{||\nabla\phi(x, y, t)||}, \text{ and } v_2(x, y, t) = u(x, y, t)\frac{\phi_y(x, y, t)}{||\nabla\phi(x, y, t)||} \tag{2.15}$$

Here, $u(x, y, t)$ is defined to be the 'isotropic pedestrian speed' (i.e, the same

speed regardless of direction), for a given location and time. Note that, according to the work done in [18], it is defined to be a function of density and location, i.e., $u(x, y, t) := U(\rho(x, y, t), x, y)$. We can use this function u to define a formula for the norm of our previously defined vector $\vec{F}$:

$$\vec{F}(x, y, t) = (\rho(x, y, t)v_1(x, y, t), \rho(x, y, t)v_2(x, y, t)) \tag{2.16}$$

$$||\vec{F}(x, y, t)|| = u(x, y, t)\rho(x, y, t) \tag{2.17}$$

We introduce a new attribute in this reformulation (mentioned by Hughes in his original 2002 work as well). We wish to account for the discomfort of pedestrians in each instance of the experiment time through a function $g(x, y, t) := G(\rho(x, y, t), x, y)$, which illustrates the behavior of pedestrians avoiding overly crowded areas. This discomfort function satisfies the following:

$$\frac{1}{||\nabla\phi(x, y, t)||} = g(x, y, t)||\vec{v}(x, y, t)|| = g(x, y, t)u(x, y, t) \tag{2.18}$$

We continue in line with [18] and define what is called the cost function. Its goal is to represent the local cost per unit distance of movement at a certain time t, i.e., how time-efficient (or, inversely, how time-expensive) will a certain path be in the immediate future. This cost value depends on the conditions of the walking area in question - location, time, and the local population density - i.e, $c(x, y, t) := C(\rho(x, y, t), x, y, t)$. This cost function ties in directly to defining the potential *phi* in an Eikonal equation.

In order to satisfy the conditions on the potential - minimizing travel time and avoiding overcrowded paths bearing in mind instantaneous information available at the current time - for some location $(x, y) \in \Omega$, Huang establishes another condition as follows:

$$c(x, y, t)\frac{\vec{F}(x, y, t)}{||\vec{F}(x, y, t)||} + \nabla\phi(x, y, t) = 0 \tag{2.19}$$

From this formula, we clearly see that $\vec{F}(x,y,t)$ and $-\nabla\phi(x,y,t)$ are parallel and pointing in the same direction. From this, the authors of [18] further show that

$$||\nabla\phi(x,y,t)|| = c(x,y,t) \tag{2.20}$$

The original source goes into further detail about how this formulation of equations (2.19) and (2.20) guarantee path efficiency; the full proof and path comparison is left for the reader to refer back to.

The Eikonal equation (2.20) is used to find the potential at a given location (x,y) and time t (much like we will do with equation (1.2) in the 1D case). As we've mentioned previously, it guarantees that pedestrians will take the most time-efficient, least cost-expending path to reach the destination (Γ_d). Accompanying it is a condition similar to the 1D case, in which, on the exit boundaries Γ_d of a given domain, we fix $\phi(x,y,t) = 0$ for $(x,y) \in \Gamma_d$. This condition further reassures that each path that pedestrians take is the most cost-efficient one.

Before formalizing a system of differential equations, the same authors of [18] give a specific definition for the cost function to be used later:

$$C(\rho(x,y,t),x,y,t) := \frac{1}{U(\rho(x,y,t),x,y)G(\rho(x,y,t),x,y)} \tag{2.21}$$

We show that this satisfies Hughes' model; first, use equation (2.19) to get:

$$\frac{\vec{F}(x,y,t)}{||\vec{F}(x,y,t)||} = -u(x,y,t)g(x,y,t)\nabla\phi(x,y,t) \tag{2.22}$$

We replace $\vec{F}$ and $||\vec{F}||$ by their values to get:

$$\vec{v}(x,y,t) = -u^2(x,y,t)g(x,y,t)\nabla\phi(x,y,t) \tag{2.23}$$

Take the norm on both sides, and we get:

$$||\nabla\phi(x,y,t)|| = \frac{1}{u(x,y,t)g(x,y,t)} \tag{2.24}$$

This result is exactly that of equation (2.18) and satisfies Hughes' Model.

Finally, we are left with the following set of differential equations that we are to work with and solve for:

$$\rho_t(x,y,t) + \nabla \cdot \vec{F}(x,y,t) = 0 \tag{2.25}$$

$$U(\rho(x,y,t),x,y)\rho(x,y,t) = ||\vec{F}(x,y,t)|| \tag{2.26}$$

$$C(\rho(x,y,t),x,y,t)\frac{\vec{F}(x,y,t)}{||\vec{F}(x,y,t)||} + \nabla\phi(x,y,t) = 0 \tag{2.27}$$

$$||\nabla\phi(x,y,t)|| = C(\rho(x,y,t)x,y,t) \tag{2.28}$$

Equations (2.25) through (2.28) are accompanied by the following initial and boundary conditions:

$$\rho(x,y,0) = \rho_I(x,y),\ \forall(x,y) \in \Omega \tag{2.29}$$

$$\phi(x,y,t) = 0,\ \forall(x,y) \in \Gamma_d \tag{2.30}$$

Here, ρ_I is simply the initial pedestrian density of the domain at time $t = 0$. It is good to note that in equation (2.30), we could've taken ϕ to be equal to any constant, but setting it equal to zero ensures that pedestrians will take the most cost efficient, time-saving route, while avoiding heavy crowds, to reach their destination, based on the information they observe on the spot.

In the next chapter, we describe numerical methods to approximate solutions for both the 1D and 2D models, using equations (2.6) and (1.2) for the 1D case, and (2.25) through (2.30) for the 2D one. After that, we run simulations under different initial conditions and explore how these algorithms may behave under these changes. We also provide a critique of these simulations, and what improvements can be made to the walking facility to facilitate the easier passage of pedestrians.

Chapter 3

Numerical Solution Algorithms

We now explain the numerical algorithms used for the numerical simulations shown in Chapter 4, as well as a set of steps on how to solve each set of PDE's. Section 3.1 goes over the two primary methods used in the 1D case, while Section 3.2 describes the three major methods/extensions used when we move to the 2D case.

3.1 The One-Dimensional Case

In this section, we go over a solution algorithm in order to solve the Hughes' model in one dimension. Some similarities in methods persist for both the one and two-dimensional cases, but each will be discussed individually and treated slightly differently from the other. For the one-dimemsional case, we solve the following coupled system of partial differential equations that we saw in the previous section, restated below:

$$\rho_t - (\rho f(\rho)\text{sgn}(\phi_x))_x = 0 \tag{3.1}$$

$$|\nabla\phi| = \frac{1}{f(\rho)} \tag{3.2}$$

Recall that $f(\rho) = 1 - \rho$. All of our simulations are done in the spatial interval $[-1, 1]$ with the initial and boundary conditions $\rho(x, 0) = \rho_I(x)$, $\rho(\pm 1, t) = \rho_D$, and $\phi(\pm 1, t) = 0$.

To solve the one-dimensional system, we will first use the fast sweeping method [19] to solve the eikonal equation (3.2), then we will use the Godunov scheme for scalar conservation laws to solve equation (3.1) [20].

Before discussing the methods and algorithms used, we set the space and time discretization. The walking facility in question - the hallway - is set to be the interval $[-1, 1] \subset \mathbb{R}$ in all work done in this paper in 1D. The interval is discretized by $N + 1$ gridpoints, indexed from 0 to N. This domain is then divided into cells of the form $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, each centered at the corresponding $x_i = i\Delta x$, with $i \in \mathbb{Z}$, and $h = \Delta x$ being the space step between x_i and x_{i+1} for all i (in our simulations, we take $h = 10^{-2}$). In a similar manner, the time domain is set to start at zero with increments of $k = \Delta t$ separating each point in time $t^n = n\Delta t$ (in our simulations, we take $k = 10^{-3}$). The result is time intervals of the form $I^n = [t^n, t^{n+1}]$.

3.1.1 The Fast-Sweeping Method

The fast-sweeping method used in our examples is based off of work done by Hongkai Zhao (see [19]) for solving Eikonal equations, like (3.2), of the general form:

$$|\nabla u(x)| = g(x),\ x \in \mathbb{R}^n$$

It is accompanied by a boundary condition $u(x) = \varphi(x)$ for $x \in \Gamma \subset \mathbb{R}^n$, where Γ is the boundary of the computational domain. In our model, $\varphi(x) = 0$ for all boundary points, corresponding to our boundary conditions on the potential ϕ, and $g(\rho) = \frac{1}{f(\rho)} = \frac{1}{1-\rho}$. Details about the approach and general procedure of the fast sweeping method, as well as other applications, are discussed by Zhao in [19]. In short, we apply upwind differencing to the interior points of Ω for the Eikonal

equation to get the following:

$$[(\phi_i - \min(\phi_{i+1}, \phi_{i-1}))^+]^2 = h^2 g^2(\rho_i),\ 2 \leq i \leq N-1$$

where

$$(x)^+ = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

Here, h is the space step between gridpoints, and the subscript i determines the ϕ at a certain gridpoint x_i, for $i = 2, ...N-1$. (Consequently, we have a total of N gridpoints ranging from index 1 to N.) Note that the above shown applies for a 'sweep' from the left side of the interval to the right. For the fast-sweeping method, we must also apply another 'sweep' in the opposite (right to left) direction, in other words, moving from $i = N-1, ...2$.

In a simpler manner for the one-dimensional case, we can describe the algorithm in the following four steps:

1. Set an initial guess for the potential, call it ϕ^{old}. For this guess, fix the boundary values as per the boundary condition given (in 1D where the endpoints are both considered exits, $\phi(\pm 1, t) = 0$), and set all other values very large (in the simulations in Chapter 4, we set them to 10^{12}). The boundary values will remain the same throughout our computations, while the interior points will change with each iteration.

2. For each interior gridpoint ϕ_i, for $i = 2, ...N-1$, we apply the following algorithm for a 'sweep' in the left-to-right direction. In 1D, this can simply reduce to the following:

$$\phi_i^{new} = \min(\min(\phi_{i-1}^{old}, \phi_{i+1}^{old}) + hg(\rho_i), \phi_i^{old})$$

The superscript 'old' represents the previous guess that we use to determine a new guess (the initial value), while 'new' represents the new approximation

that will replace the old.

3. Repeat step 2, but this time, we find ϕ_i for $i = N-1, ...2$, performing a second 'sweep' from right to left.

4. If the following convergence condition is satisfied:

$$||\phi^{new} - \phi^{old}|| < \delta$$

for some small positive value δ (in our simulations, we will take $\delta = 10^{-9}$), and for all $x \in \Omega \setminus \Gamma$ then we are done and proceed to solve the conservation law. If not, we assign the value of ϕ^{new} to ϕ^{old} and repeat steps 2 and 3, and continue in this loop until the above convergence inequality is satisfied.

Once we have our potential approximations, we proceed to the next step to find the pedestrian density at each gridpoint.

3.1.2 The Godunov Scheme

A hyperbolic conservation law of the form $u_t + F(u)_x = 0$ can be solved in a number of ways, with one of the most common being the Godunov scheme. In equation (3.1), our unknown variable is ρ, and $F(\rho) = -\rho f(\rho)\text{sgn}(\phi_x)$. The full derivation of the Godunov scheme can be seen in Randall LeVeque's work [20]. Here, we state the resulting scheme used in our simulations. At a certain gridpoint x_i at time t^{n+1}, given we know the pedestrian density at this same gridpoint at the previous time step t^n, we can find the new density using upwind differencing, and reach the following difference method:

$$\rho_i^{n+1} = \rho_i^n - \frac{k}{h}(\bar{F}^n_{i+\frac{1}{2}} - \bar{F}^n_{i-\frac{1}{2}})$$

where $\bar{F}^n_{i+\frac{1}{2}}$ is the Godunov flux, and, to avoid solving and Riemann solvers, can be approximated using:

$$\bar{F}^n_{i+\frac{1}{2}} = \begin{cases} \min\limits_{\rho^n_i \leq \theta \leq \rho^n_{i+1}} F(\theta) & \text{if } \rho^n_i \leq \rho^n_{i+1} \\ \max\limits_{\rho^n_{i+1} \leq \theta \leq \rho^n_i} F(\theta) & \text{if } \rho^n_{i+1} \leq \rho^n_i \end{cases}$$

Here, h represents the spatial step (like before), while k represents the given time step between t^n and t^{n+1}. F (without the bar) is the same one defined before to fit equation (3.1).

With this newfound ρ at the new time for each gridpoint, the process is repeated for the next time step, finding new potential approximations for the new density values with the fast-sweeping method, then finding the new density approximation using the Godunov scheme, and so on for the experiment's total duration.

3.2 The Two-Dimensional Case

For the two-dimensional case, we will follow the following process, loosely based off of that by [18], but using extensions of the methods we've seen in 1D, in order to arrive at the updated density values at each new time step:

1. For a given cost function, find $C(\rho(x, y, t), x, y, t)$. (We prescribe this in our simulations).

2. Using this cost function, solve the eikonal equation (2.28) using the fast sweeping method [19].

3. Find $||\vec{f}(x, y, t)||$ using equation (2.26).

4. Find $\vec{f}(x, y, t)$ using equation (2.27).

5. Find the 2D Godunov flux approximations in order to solve the scalar conservation law (2.25) [20].

These steps are performed over three forward Euler steps, and using the total-variation-diminishing Runge-Kutta method as a corrector step to ensure accuracy. We now discuss the details of each of the key algorithms mentioned above and how they change in two dimensions.

3.2.1 The Godunov Flux in 2D

We return to the well-known Godunov scheme that we've seen previously in the 1D case and extend its definition into 2D. While other high-order schemes do exist for solving 2D hyperbolic conservation laws, such as the WENO scheme (see [21], for instance), we opt for the easy-to-solve Godunov scheme with a numerical flux approximation. Like before, we begin with a 2D conservation law denoted by $\rho_t + F(\rho)_x = 0$, (here, $F(\rho)$ is defined the same way it was in Chapter 2, Section 3) and we perform the following upwind differencing to give us the 2D differencing method that we use later

$$\frac{d}{dt}\rho^n_{i,j} = -\frac{1}{h_x}(\bar{F}^n_{i+\frac{1}{2},j} - \bar{F}^n_{i-\frac{1}{2},j}) - \frac{1}{h_y}(\bar{F}^n_{i,j+\frac{1}{2}} - \bar{F}^n_{i,j-\frac{1}{2}}) \tag{3.3}$$

Here, h_x and h_y are the space steps in both the x and y directions respectively, and $\bar{F}^n_{i+\frac{1}{2},j}$ is the numerical Godunov flux in the x direction, which will be calculated similarly to before in the previous section:

$$\bar{F}^n_{i+\frac{1}{2},j} = \begin{cases} \min\limits_{\rho^n_{i,j} \le \theta \le \rho^n_{i+1,j}} F(\theta) & \text{if } \rho^n_{i,j} \le \rho^n_{i+1,j} \\ \max\limits_{\rho^n_{i+1,j} \le \theta \le \rho^n_{i,j}} F(\theta) & \text{if } \rho^n_{i+1,j} \le \rho^n_{i,j} \end{cases}$$

Similarly a definition for the numerical Godunov flux in the y direction, $\bar{F}^n_{i,j+\frac{1}{2}}$, is analogous. Using the Godunov flux as opposed to other alternatives is in its simplicity to implement (as opposed to the WENO scheme) and great accuracy (as opposed to schemes like the Lax-Friedrichs one). These fluxes will be used in equation (3.3), and we use the right hand side of this equation when we perform the three Euler steps for the TVD Runge-Kutta method. We see this implemen-

tation later in this chapter.

3.2.2 The Fast-Sweeping Method

Like we've seen in the 1D case, when solving for the cost potential, we are dealing with an Eikonal equation like the one seen in equation (2.28). We note this Eikonal equation is a special case of steady-state Hamilton-Jacobi equations. We would like to find an approximation to the viscosity solution, similar to the work found in [16]. The importance of this solution is physical relevance for a variety of applications, including our pedestrian model. Numerical methods for viscosity solution approximations for Hamilton-Jacobi equations bear similarities to those for solving conservation laws. To this end, it is possible extend a scheme like the third or fifth-order WENO scheme for conservation laws (for more details, see [22], [21], and [23]) to WENO schemes designed to solve Hamilton-Jacobi equations (see [24]). Instead, however, we will use the much simpler fast-sweeping method by Zhao [19], which is an extension of the method we've seen previously in the 1D case, seen first in the following discretization:

$$[(\phi_{i,j} - \min(\phi_{i+1,j}, \phi_{i-1,j}))^+]^2 + [(\phi_{i,j} - \min(\phi_{i,j+1}, \phi_{i,j-1}))^+]^2 = h^2 c_{i,j}^2$$

where $2 \leq i \leq N_x - 1$ and $2 \leq j \leq N_y - 1$. Here, (i, j) is the index pair for (x, y), and N_x and N_y are the number of gridpoints in the x and y directions respectively. We also note that $c_{i,j} = C(\rho(x_i, y_j, t)x_i, y_j, t)$ (which we prescribe in step 1) and h is the spatial step (which we are assuming to be the same in both the vertical and horizontal directions).

Much of the initialization here is very similar to before, where we fix the boundary condition on the exits to zero for all sweeps, and set all other values at the other gridpoints as high numbers that will be updated later (usually taken to be 10^{12}). One key difference is that in the 1D case, we only performed two sweeps in the left and right directions. In 2D, however, we now sweep in four directions,

defined as follows:

1) $i = 1 : N_x,\ j = 1 : N_y$

2) $i = N_x : 1,\ j = 1 : N_y$

3) $i = N_x : 1,\ j = N_y : 1$

4) $i = 1 : N_x,\ j = N_y : 1$

As we loop through each sweep, we update the solution as follows with this extension of the fast-sweeping method from the simple 1D case to 2D:

$$\phi_{i,j}^{new} = \min(\phi_{i,j}^{old}, \bar{\phi}_{i,j})$$

where,

$$\bar{\phi}_{i,j} = \begin{cases} \min(\phi_{i,j}^{xmin}, \phi_{i,j}^{ymin}) + c_{i,j}h & \text{if } |\phi_{i,j}^{xmin}, \phi_{i,j}^{ymin}| \geq c_{i,j}h \\ \phi_{i,j}^{xmin} + \phi_{i,j}^{ymin} + \sqrt{(2c_{i,j}^2h^2 - (\phi_{i,j}^{xmin} - \phi_{i,j}^{ymin})^2)} & \text{otherwise} \end{cases}$$

and

$$\begin{cases} \phi_{i,j}^{xmin} = \min(\phi_{i+1,j}^{old}, \phi_{i-1,j}^{old}) \\ \phi_{i,j}^{ymin} = \min(\phi_{i,j+1}^{old}, \phi_{i,j-1}^{old}) \end{cases}$$

We finally denote the convergence condition similarly to the 1D case by the following:

$$||\phi^{old} - \phi^{new}|| < \delta, \text{ for some convergence threshold } \delta > 0$$

The only difference is that the norm is performed in two dimensions instead of one. In our simulations, we will take $\delta = 10^{-9}$.

3.2.3 The third-order Total Variation Diminishing Runge-Kutta time discretization

Before we proceed to simulations, we address a corrector step that we perform to apply time discretization to equation (3.3). In order to maintain the stability

of spatial discretization, we will use the third-order total variation diminishing (TVD) Runge-Kutta method, shown in detail by Shu and Osher in [25]. This method, moving from ρ^n to ρ^{n+1}, is given by:

$$\rho^{(1)} = \rho^n + \Delta t L(\rho^n)$$

$$\rho^{(2)} = \frac{3}{4}\rho^n + \frac{1}{4}(\rho^{(1)} + \Delta t L(\rho^{(1)}))$$

$$\rho^{n+1} = \frac{1}{3}\rho^n + \frac{2}{3}(\rho^{(2)} + \Delta t L(\rho^{(2)}))$$

where $L(\rho)$ is the right-hand side of equation (3.3).

With our key methods before us, we now proceed to put these into practice with some numerical simulations. In the next chapter, we use the methods described here to craft meaningful real-world examples and see how Hughes' model behaves under these different initial conditions.

Chapter 4

Simulations and Interpretations

In this chapter, we implement the schemes and methods explained in Chapter 3 in both the 1D and 2D cases. We will describe each in terms of mathematics and in terms of real-life interpretation. In some cases, we also provide a brief critique of how the walking conditions can be improved for the easier evacuation of people from the domain.

4.1 The One-Dimensional Case

We begin with the one-dimensional case by setting the stage for what we will be simulating. Consider a one-dimensional hallway with no branching paths and two exits on both sides; that is, in our computational domain $\Omega = [-1, 1]$, the points 1 and -1 are the destinations which people in the hallway wish to reach (before we end this subsection, we will explore what happens when we block off one exit). The previously mentioned behavioral attributes, to minimize travel time and avoid high-density crowds, remain in play, which translates to the potential at the boundary points being zero at all times. This forms the boundary condition for the potential: $\phi(\pm 1, t) = 0$ for all $t \geq 0$. The initial densities across the hallway are given by $\rho_I(x) = \rho(x, 0)$ for all $x \in [-1, 1]$, and additional homogeneous Dirichlet boundary conditions whenever necessary, $\rho_D = \rho(\pm 1, t)$.

As mentioned in Section 3, we discretize the computational domain Ω with spatial steps $\Delta x = 10^{-2}$, and the timestep between each iteration is $\Delta t = 10^{-3}$.

Let's begin with a simple example of initial data where $\rho_I(x) = 0.8$ for all $x \in \Omega$ with the addition of a Dirichlet boundary condition $\rho_D = 0.5$ (see Chapter 2 for details). Figure 4.1 shows this simulation in action, showing the relationship between the density and position at a certain time, with notable time steps shown and labeled accordingly, over a 3-second total experiment time.

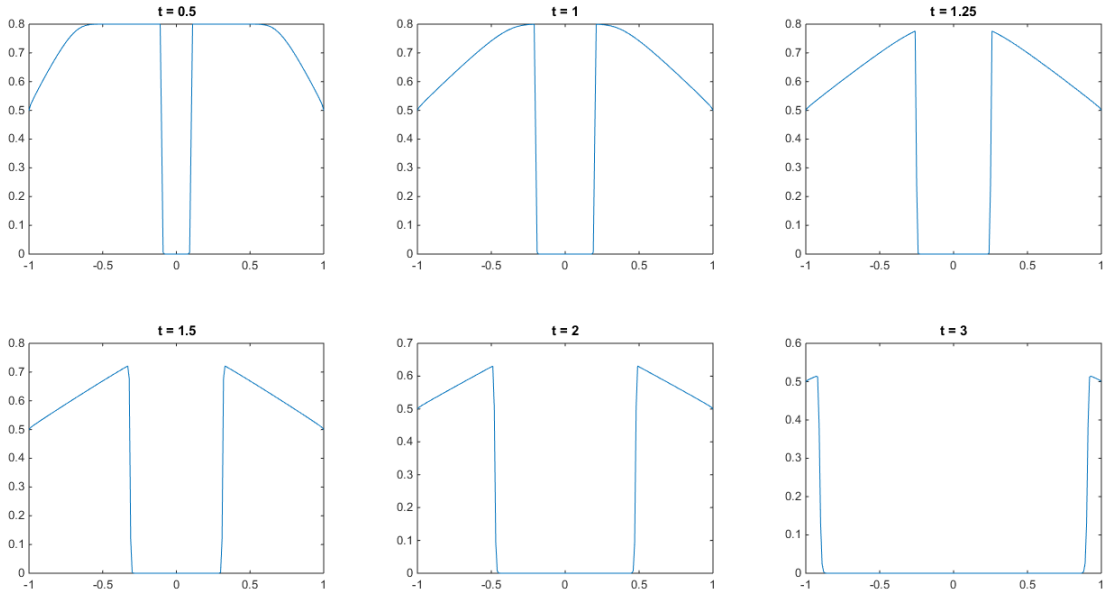

Figure 4.1: Snapshots of the change in ρ for $\rho_I(x) = 0.8$ and Dirichlet boundary condition $\rho_D = 0.5$

First, we note that with such a high density of people across the entire walking facility, it would make sense for it to take some time before the hallway is completely cleared out. If we were to take half the initial density of people (say, $\rho_I(x) = 0.4$ for all $x \in \Omega$), the experiment would take roughly half the time for all pedestrians to make their way through an exit.

The hallway starts off being entirely packed with a consistent density of people through the whole path. That being said, with densities towards both exits being relatively similar, the split that occurs between those who choose to exit from the left and the right is an even one straight down the middle, with the path that people choose depending mostly on their initial position; that is, those who start

before the point $x = 0$ (the halfway point from end to end) tend to the left, and those to the right of $x = 0$ tend to the right. This causes a drastic dip in the crowd density in the middle of the hallway, which is being cleared out first. This notable sudden change in densities represents the respective shock waves in both directions.

Unfortunately, those who started towards the middle face the dilemma of overcrowding, which stalls them from getting to their nearest exit sooner. Due to the high densities of people closer to the exit trying to make their way out, these incoming crowds are brought to a near halt as they wait for the crowd closer to the exit to ease up and leave the hallway, making more room for them to pass and depart as well. We can visually see this, especially in the first few snapshots in Figure 4.1.

What's also noteworthy is the density values close to the exits. Observe the first graph for $t = 0.5$. We notice a steady, smooth drop in pedestrian density at these exits as we approach $x = \pm 1$, in part due to the lower density of people we set to be at the boundaries (the Dirichlet boundary condition). Mathematically, these are the manifestation of the rarefaction waves in play. What this could represent realistically is that crowds eventually trickle down into a more manageable, lower-density ones to make for the easier transition of people as less total people are present in the hallway at once, gradually reducing the overcrowding issue.

Moreover, we notice that shortly before $t = 1.25$, the rarefaction and shock waves meet. Think of the constant densities between the shocks and rarefactions in previous graphs as people waiting, or progressing more slowly, until they can 'funnel through' and make their way to the exit more easily in a less-crammed space (the rarefaction parts). When the shock and rarefaction meet, then this funneling of people will occur more easily. Judging by how long it takes for the hallway to nearly clear out (more than 3 seconds), it will still take some time for those waiting to make their exit, since the overall densities in the hallway are still

rather high. Finally, we note that when the shock finally meets the boundary (just after 3 seconds), it indicates that the hallway is effectively cleared out.

This example showed a scenario where the entire hallway started with the same density levels all around. However, what if we had a situation in which these densities were not so evenly spread? What if one end of the hallway was more congested than another? We take this case as our second example, with initial condition:

$$\rho_I(x) = \begin{cases} 0.7 & \text{if } x \in [0.4, 1] \\ 0 & \text{elswhere} \end{cases}$$

It is accompanied by a Dirichlet boundary condition, like before, set to $\rho_D = 0.5$.

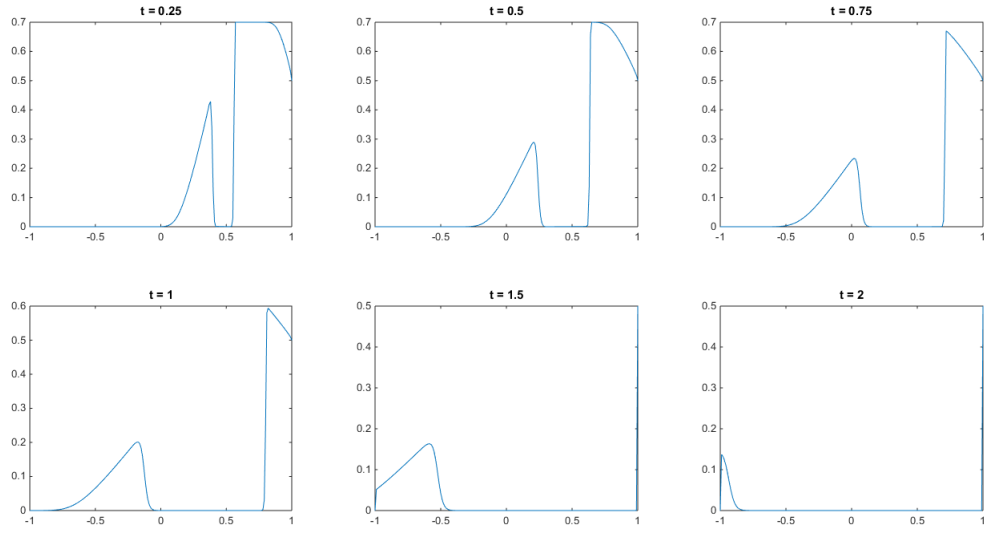

Figure 4.2: Snapshots of the change in ρ for a one-sided initial condition

Figure 4.2 shows this simulation in action across the timespan of 2 seconds. We set the initial position for all people in the hallway to be close to the rightmost exit. Those that are closer to the right exit will, obviously, make their way through with few issues. However, those that aren't so close are not as fortunate, and have two choices: stay nearby in the crammed environment until things clear up, or turn around and make their way to the left exit. For a fraction of these people, the latter option is more appealing, and we see this from the very first snapshot with an uneven split of people abandoning the heavy crowds and making their way through the empty side of the hallway in order to leave from the alternate

exit. This smaller crowd also manages to distribute itself more evenly than the dense, crammed group waiting at the right exit. In contrast, we notice that with the smaller group heading for the left, the curve representing their densities is flattening as they approach the left, making for less cramming. It is still worth noting, however, the fraction of people who remain in the crowd to the right and do not reconsider their path. To them, since they are near the exit, the existence of a heavy crowd doesn't outweigh the proximity to the way out, so they remain and wait.

Shortly after the 2-second mark, the hallway is cleared out. Surprisingly, the dense crowd to the right made their way out entirely before the ones who strayed away to the left. Examples like this are intriguing case studies in path efficiency: is it more time-efficient to wait in the crowd while close to the exit, or turn away and find an alternative? In this example, we witness both of these options in play. While in this case, it appeared that waiting paid off for those who did (as they evacuated sooner than the others who made their way to the left), but this result is certainly not an absolute. Different scenarios, and different pedestrian splits, will give way to different results.

Next, we explore a more intricate example with different densities across the hallway, and see how crowds behave under these conditions. Take the following piecewise example:

$$\rho_I(x) = \begin{cases} 0.65 & \text{if } x \in [-0.5, -0.25] \\ 0.3 & \text{if } x \in [-0.1, 0.3] \\ 0.85 & \text{if } x \in [0.5, 0.9] \end{cases}$$

We apply our algorithm with this initial data and view snapshots at every 1/4 of a second for this process in Figure 4.3. (Note: the source code for this simulation can be found in the Appendix.)

Let's explore this example present in the snapshots and interpret from both a mathematical and real-life perspective. For ease of notation, we will call the

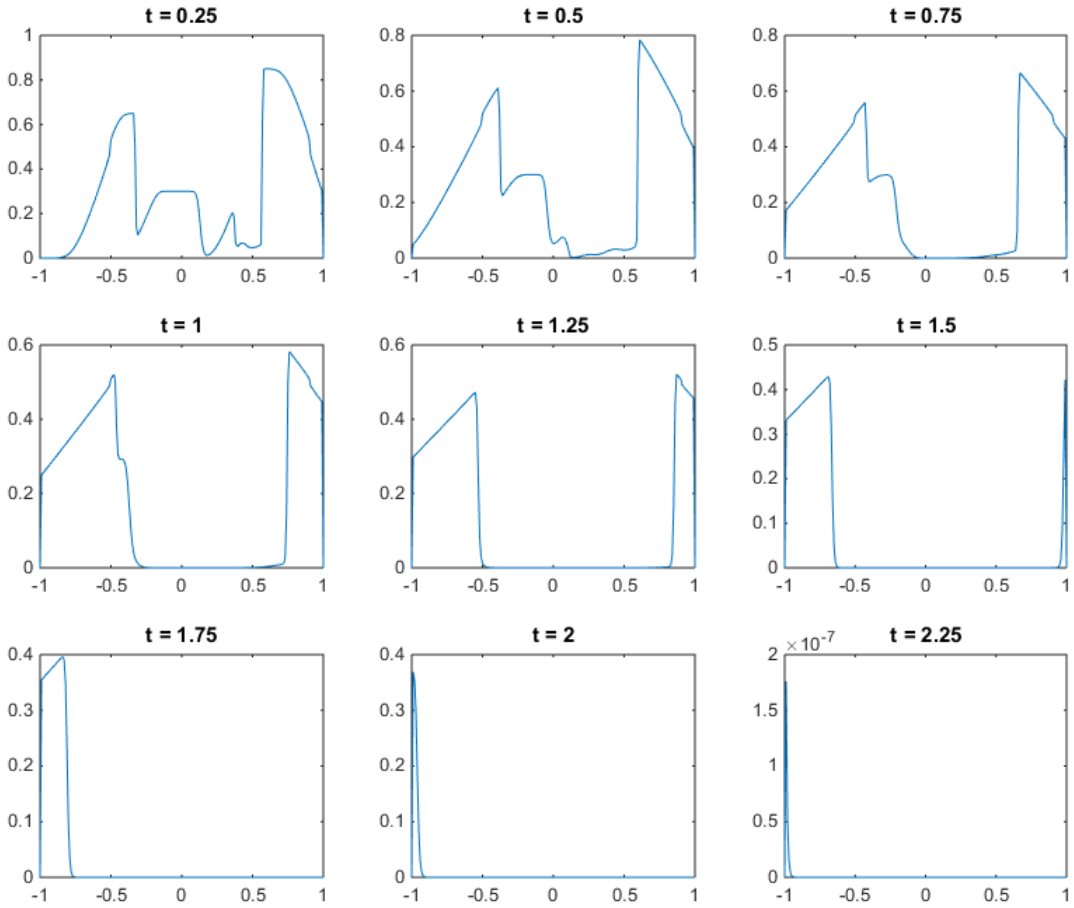

Figure 4.3: Snapshots of the change in ρ for the new piecewise initial condition

leftmost starting crowd Group A, the middle one Group B, and the rightmost one Group C. Starting from $t = 0.25$, we notice multiple jumps (shocks) that break up parts of the solution. The most notable ones appear in parts that split up Group A from B and Group B from C, and noticeable ones in Groups A and C on their way to their respective closer exit. Given the higher initial crowd densities of Groups A and C (compared to B), we see a situation comparable to what we saw in the second example with overcrowding at locations of higher densities, giving rise to these shocks. To put it into real-life perspective, members of Group B have to make a decision about the most travel-effective route. Considering that the left is slightly less crowded, joining Group A and navigating to the left exit is the more appealing option. Meanwhile, Groups A and C suffer from extreme overcrowding, and some members of Group C even try to turn away to try and join Group B and navigate to the left exit. Simply put, in the beginning, we witness

a great mishap of cramming, especially as we approach the exit boundaries.

A similar story can be said as we progress into $t = 0.5$ and $t = 0.75$. Most of group B is making their way to the left with Group A, though there are still a few in the crowds that haven't quite committed to a direction. As we progress, especially evident in Figure 4, we notice that a small amount of people from Group B try to change course to the farther exit (the right one) due to the overcrowding disaster at the closer one (this can be seen in the small, ever so slight oscillations in the snapshot for $t = 0.5$). This is a clear example of the reactive nature of the model in play, how people tamper with their travel route depending on the development of the immediate events before them. If a route appears too crowded or conditions change, they will make changes. Meanwhile, while some members of Group C remain committed to their route, despite the intense traffic (since they are so close to the exit anyway), a select few that are farther away, closer to the shock in the figure, try to make their way to the other side of the hallway for the alternative exit, since the wait in the heavy crowd is no long as cost efficient as changing course.

Moving forward to $t = 1$ and $t = 1.25$, Group B and those who strayed from Group C have completely joined Group A, making for the more appealing exit route in the eyes of the pedestrians in the immediate situation. The overcrowding of Group A and the remainder of Group C, however, has yet to completely diminish. We still see high densities of people in these groups in such a small interval, evidence that it is still far too crowded for a smooth transition out of the hallway.

Going forward, we note that Group C completely exits the hallway shortly after $t = 1.5$, while Group A continues to funnel out through the left exit. Time goes on, and Group A presses on until the hallway is nearly cleared out entirely around $t = 2.25$. This progression moved rather quickly now that the rest of the hallway was cleared, and only their group remained.All in all, this simulation serves as a solid showcase of how pedestrians can alter their travel route based

on immediate circumstances (mainly seen with Group B), and how intense overcrowding in one area, like we were given in the initial condition for Groups A and C, can hinder a smooth transition of crowds making their way to funnel into an exit, even if it's so nearby.

Finally, we address a modification to our model to illustrate a slightly different case. Most of the circumstances are the same, except for the boundary conditions. Previously, we worked with a hallway in which both endpoints were exits. Now, we consider a case in which only one of the boundaries is an exit. Without loss of generality, assume that, in the hallway defined by $\Omega = [-1, 1]$, $x = 1$ is an exit, while $x = -1$ is a dead end (a wall). In this case, the boundary condition on the potential changes so that $\phi(1, t) = 0$, while $\phi(-1, t)$ is treated like any other interior point, set to a high value, and not fixed with the fast-sweeping method. We turn to Figure 4.4 and observe the progression of pedestrians in such a scenario.

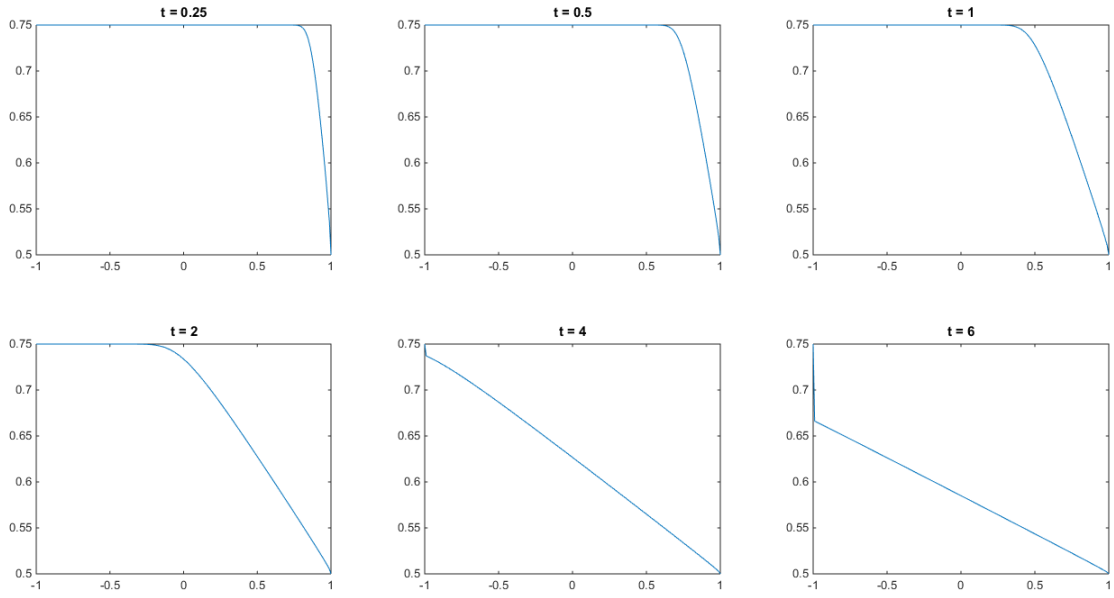

Figure 4.4: Snapshots of a single-exit case where $\rho_I(x) = 0.75$ with Dirichlet boundary condition $\rho(1, t) = 0.5$.

Figure 4 is based off of a simulation in which the right exit is open, while the left one is replaced by an impassable wall. The initial condition for density is given to be $\rho_I(x) = 0.75$ for all $x \in [-1, 1)$ and $\rho(1, t) = 0.5$ as the Dirichlet boundary

condition. The simulation abides by the same spatial and time discretization as before, and run for 6 seconds. The densities drop towards the right exit, as expected, but it takes a very long time for the hallway to become even close to being cleared, as there's still a good number of people who haven't reached the right exit by the end of the simulation. This is partly due to the relatively high crowd density across the whole walking facility and the lack of any other options or possible routes for people to take. This absence of an alternative forces pedestrians to abide by one route and march towards the only exit in the corridor, and in turn, results in an increase in time to clear the way.

4.2 The Two-Dimensional Case

We now move from a one-dimensional corridor to a two-dimensional walking space with much more in the way of possibilities and applications. In our examples, we focus namely on evacuation; that is, we will always start off with an initial condition across the entire domain to determine where people begin at time zero, and their goal is to make their way to the nearest exit. We consider a room of dimensions $\Omega = [0, 100] \times [0, 50]$. The exact nature of the boundaries (walls or exits) will be specified in each individual case. We take a uniform spatial discretization $\Delta x = \Delta y = 1$ and time step $\Delta t = 0.1$. Like before, we take into consideration the aforementioned assumptions about crowd behavior to determine the potential on the exit boundaries, that is $\phi(x, y, t) = 0$ for $(x, y) \in \Gamma_d$ for all $t \geq 0$.

We also prescribe some of the functions defined previously, with assumptions to keep some of the variables in line with the 1D case. We consider the following speed-density relationship (similar to Greenshields' traffic flow model, see [4] for details):

$$U(\rho) = 1 - \rho, \text{ for all } (x, y) \in \Omega$$

By this definition, the maximum walking speed is 1 (can be considered as meters

per second for a unit, if the reader chooses for visualization purposes), but when the crowd density equals its 1, movement is completely halted, much like in the 1D case, making it the maximum density for these simulations. We also consider no extra discomfort from external factors, i.e., $G(\rho) = 0$. Finally, we define the cost function, which determines our cost potential in the Eikonal equation (2.28):

$$C(\rho) = \frac{1}{1-\rho}$$

These factors remain consistent in both of our provided examples. Note that with these choices of U, C, and G, we observe that when we calculate $\vec{f}(x, y, t)$ and its norm using (2.26), then (2.27):

$$||\vec{f}(x, y, t)|| = \nabla\phi(1-\rho)\rho$$

$$\vec{f}(x, y, t) = \frac{(1-\rho)\rho}{(1-\rho)^{-1}} = \nabla\phi(1-\rho)^2\rho$$

This reduces back to the original Hughes' model that we saw in the beginning in equations (1.1) and (1.2).

Now, we give the details for our first simulation. Consider a walking facility with no obstacles in place and two exits along the left and right boundaries. The initial crowd composition is a group of density $\rho_I(x, y) = 0.4$ situated at the gridpoints (x, y) that are a part of the subset $[40, 60] \times [20, 30] \subset \Omega$. This placement choice is close to the center of the computational domain, and it serves to demonstrate in a simple manner how this small crowd splits up and chooses a path to the nearest exit, and how the crowd behaves on their way out. Despite its lack of complexity, the situation is applicable to real-life situations like a large sports stadium, where a crowd (in this case, it could be the sports teams) may be condensed in the center of the field at the end of a game, and make their way to one of the two respective exits on the far ends after the sporting event, such as soccer, basketball, or football, comes to a close. (Note: the source code for this simulation can be found in the Appendix.)

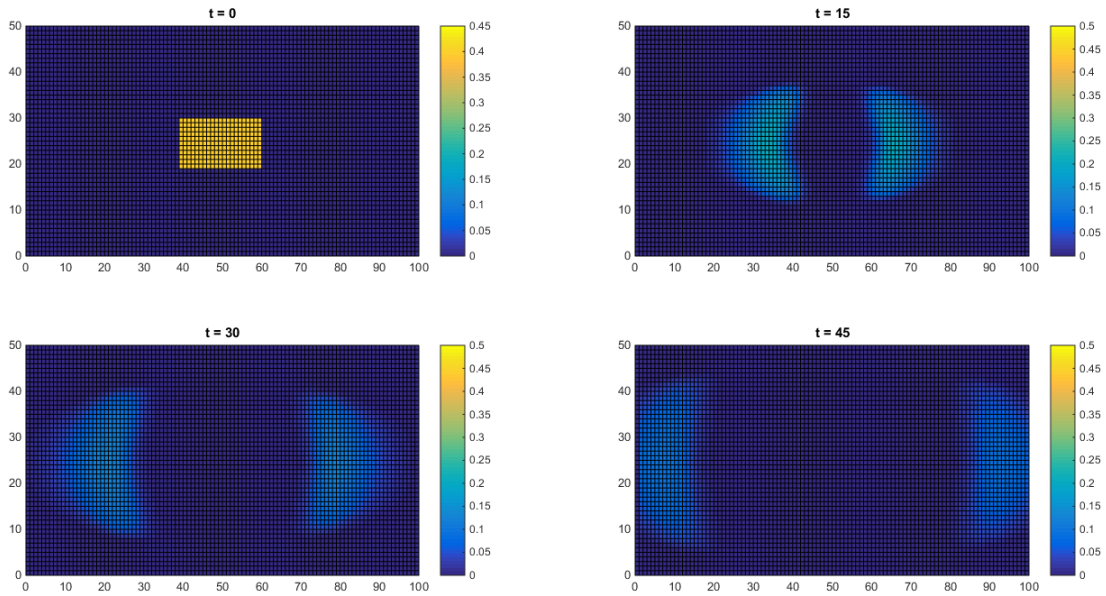

Figure 4.5: Snapshots of our first 2D simulation where $\rho_I(x) = 0.4$ in the rectangle $[40, 60] \times [20, 30]$, showing the pedestrian flow in time at three different intervals.

We turn to Figure 4.5 to view the propagation of the crowd from their starting point in the middle at time zero to the exits. The study lasts for 60 seconds, and we display the crowd densities at the initial time $t = 0$, then for $t = 15, 30$, and 45 seconds. We observe at $t = 15$ that the crowd evenly splits down the middle; those on the left side of the domain will march towards the left exit, and those on the right side will head for the right one, as expected. With no obstacles or obstructions before them, the pedestrians are free to move forward without any external limitations. Interestingly enough, with more elbow room, the crowd spreads out vertically, shown in the density spreads up and down on both the left and right groups, making for less dense crowding in the middle and more convenient passage. Similarly, we also notice some density spreading happening horizontally as well. With no external factors in the way, the crowds on the left and right sides can move at full speed without any need to stop. As a consequence, those who are behind also have more room to move forward and not be as crammed as initially.

This trend continues as time presses on, with the general maximum densities at one point decreasing progressively as people can spread out more in the open

room, traversing away from more crowded spaces by going around (explaining the curving and spreading in the vertical direction). Eventually, both crowds arrive at their exit boundaries with no overcrowding or bottlenecking. It takes just over 60 seconds for the domain to completely clear out. Overall, the walking facility was convenient for this crowd to split up and spread out and avoid any sort of pedestrian complications. This example was relatively easy to process due to the lack of obstacles or overcrowding, but not only can it be applied or visualized in real-life crowd-based situations, it also sets the standard for how a tame evacuation generally looks under these circumstances compared to what we will see next.

Next, we take a walking facility of the same dimensions with a single exit on only a part of the right boundary; that is, $\Gamma_d = \{(x, y) : x = 100, 15 \leq y \leq 35\}$. The rest of the boundaries are walls. We also add in three obstacles in the interior of Ω, defined as follows:

$$P_A = \{(x, y) : (x, y) \in [30, 40] \times [10, 20]\}$$

$$P_B = \{(x, y) : (x, y) \in [30, 40] \times [30, 40]\}$$

$$D = \{(x, y) : (x, y) \in [60, 80] \times [10, 40]\}$$

P_A and P_B can be visualized as pillars, being small structures placed side by side and in line with one another, while D can be seen as a divider, having great vertical length and standing in front of the designated exit. We account for these obstacles the same way that we would for walls; that is, we simply ignore them in the computation of the potential and the density. Our crowd of pedestrians will begin from the far left in the rectangle $[0, 10] \times [10, 40]$, with a starting density of $\rho_i(x, y) = 0.6$. While not as direct as our previous example, it is very possible for large crowds to be presented with a situation to travel from point A to point B like this one, navigating through narrowing corridors as such and re-convening on the other side. For a detailed visualization of the starting conditions, see Figure

4.6.

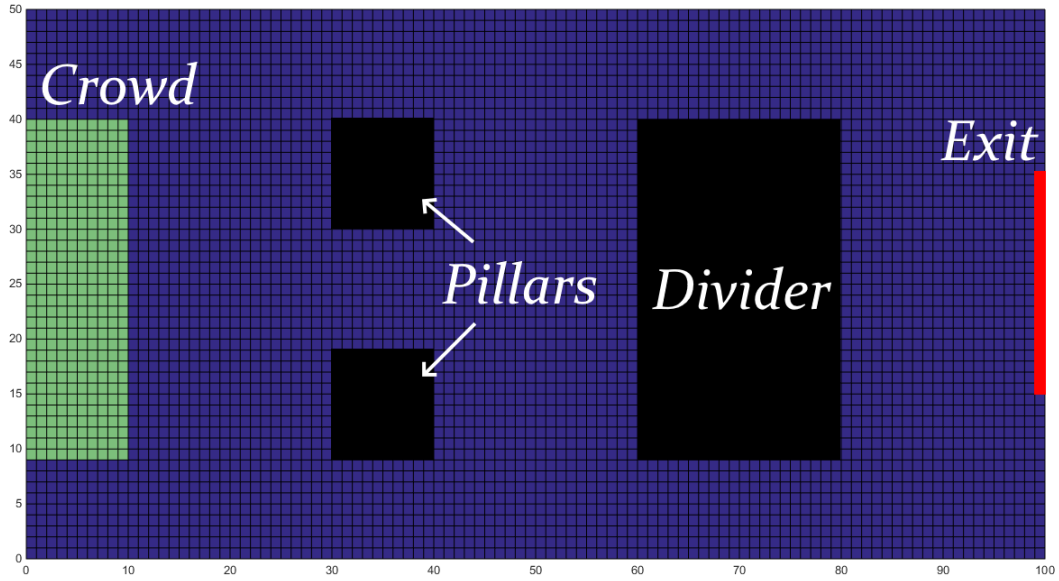

Figure 4.6: Layout of the walking facility at $t = 0$ with details of the obstructions and initial crowd location.

We are interested in seeing how crowds behave in their travels from the far left of the walking facility to the narrow exit on the far right, and how the crowds evolve while passing by various obstacles. Figure 4.7 shows overhead snapshots of the densities at time $t = 30, 60, 90$, and 120 seconds. We describe the crowd behavior as a whole from $t = 0$ until $t = 120$.

The crowd starts in their initial positions, as previously established, and pressed to the right in the direction of the exit. As seen before, those in the front move forward and make some room for those behind to spread out a bit, making for a less crammed crowd as a whole. Not long into their progress, however, they encounter their first obstacle: navigating around the two pillars P_A and P_B. There are three available paths to take to pass this barrier: from above, below, or through the middle. Depending on the initial vertical position of the crowds, they will funnel through one of these three options, whichever is closer. We notice the splitting begins before the crowds even reach the pillars; this is an example of pedestrians taking in instantaneous travel data and tampering with their routes accordingly.

One important remark to make about the first graph at $t = 30$ is the spike

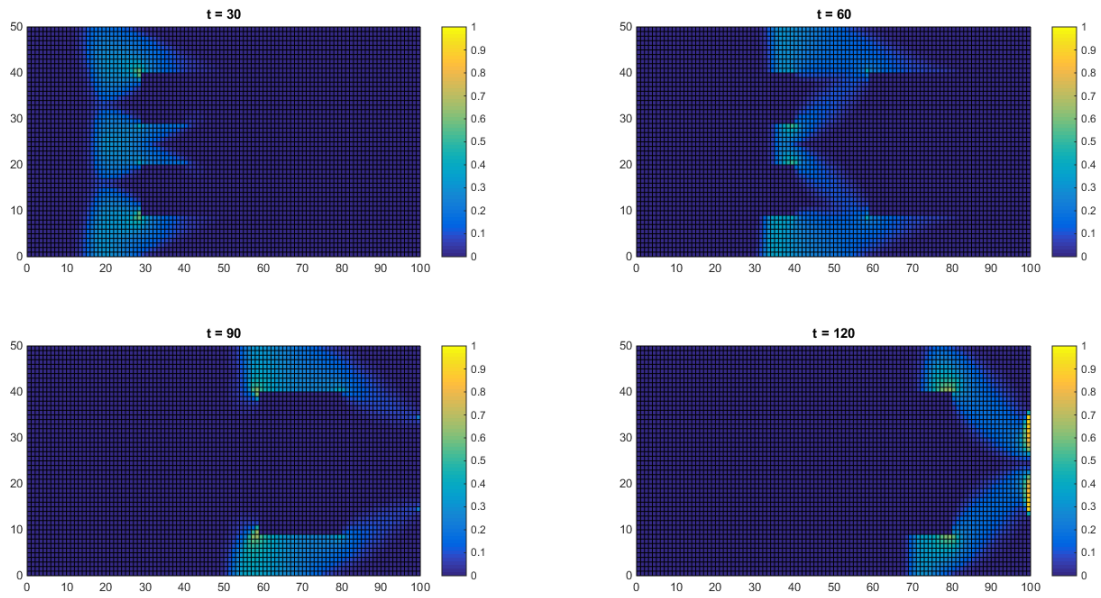

Figure 4.7: Snapshots of our second 2D simulation (with obstacles and a smaller exit boundary) where $\rho_I(x) = 0.6$ in the rectangle $[0, 10] \times [10, 40]$, showing the pedestrian flow in time at four different times.

in density that occurs on the top left edge of P_B and the bottom left edge of P_A. The densities at these corners is significantly higher than any other point, even with the narrower path that could give way to bottlenecking. This could be at the hands of people attempting to save time by moving close to this corner, but accumulating in the process. Such crowding has the potential to spread to the rest of an already-thin corridor and cause issues for those taking these respective routes. While they manage just fine in our simulations, this crowding issue is still noteworthy when considering possibilities for higher densities of people and even narrower paths that could present themselves in different situations. This can all add up to a less-than-ideal walking experience for those involved.

Regardless, the crowds carry on despite needing to travel through a smaller, thinner space than before, all without causing too much overcrowding besides at the two corners. We do expect, however, higher densities through these narrower corridors than the more evenly spread ones before. The snapshot at $t = 60$ shows these higher densities (in the three higher-density locations above, in between, and below the pillars).

Once the middle group gets through the pillars, we note that in the graph

for $t = 30$, and even moreso in that of $t = 60$, the middle group splits up in the upward and downward direction to join one of the two other groups after passing between the two pillars. Once again, they receive new information of a large divider D in front of them. Since they cannot pass through the middle, they must make an immediate decision and move to either the left or the right (or up or down from our perspective) in order to proceed. In doing so, they meet up with the upper or lower groups, whichever is closer.

Once past the pillars, the next hurdle that these crowds face is the divider D. It is rather large and, unlike before, only has two options for passage. As noted before, the middle group splits up to meet up with the upper and lower groups to pass through. The width of these narrow corridors between the upper and lower bounds of the divider and the walls of Ω is the same width of those seen between the pillars and the walls previously. While that might mean that we shouldn't see any other pedestrian issues due to similar hallway widths, we notice that this isn't necessarily the case. From the snapshot at $t = 60$, we notice similar spikes in density at the top left and bottom left corners of D, their existence justified by the same argument as that of the pillars. We notice that these densities grow and spread out through the hallway as more crowds pass through, evident in the more yellow-green marks in the snapshot for $t = 90$ (like we've seen in the pillar case). These heavy-density markers and be seen even in the fourth snapshot for $t = 120$.

Finally, once the divider D is maneuvered around, the crowds come together in the middle to pass through the exit, which is only a small part of the right boundary. With the crowds converging together and trying to make their way through a small exit, we notice that while they disperse into less dense crowds again, they eventually cluster together again and make for serious overcrowding at the exit mark, clear through the yellow lines along the exit (they are nearly maxed out at a density of 1). This phenomenon persists up until the entire walking facility is cleared with a slow progression of pedestrians out the exit.

Overall, we can extract some interesting conclusions from this experiment. First, and obviously, thinner corridors make for more overcrowding. A critique to improve the walking conditions of this room is to have wider hallways to get around obstructions, and more path options wherever applicable. Second, sharp corners that people must turn around are not ideal for an end goal to avoid overcrowding. A possible solution for this would be to try to make rounded edges along obstacles. This has the potential to facilitate for easier traversal of crowds, and make more room for more people to funnel through more smoothly. Third, the more empty space, the more opportunities for people to spread out and make for a less dense walking environment; this was seen in both examples and having such open spaces is encouraged to keep this spread. Finally, the smaller the exit, the more risk for overcrowding there. Wherever possible, either having more exit options or a wider exit would do wonders to improve the walking conditions of this case study.

Chapter 5

Conclusions and Moving Forward

Throughout this paper, we've had a look at Hughes' model for pedestrian traffic from both a theoretical and practical perspective. We started off with some motivation to familiarize ourselves with the derivation of the model in both one and two dimensions, as well as some brief analysis of the solutions to the systems. We then proceeded to go over a solution algorithm for each case, with efficient methods to solve the given Eikonal equations (using the fast-sweeping method) and the conservation laws (using the Godunov scheme). We described the process for all the relevant algorithms. With these in mind, we presented numerical results under different initial conditions, describing what we saw mathematically and gave a real-life interpretation and critique for the simulations performed.

This research has the potential to shed light on the development of walking facilities in the making. With a heavy focus on evacuation, as we've seen in Chapter 4's simulations, we are able to better visualize how people behave in a certain walking space with their primary objective being to exit. As mentioned in the introduction, it's important to study crowd behavior in emergency situations such as fires, earthquakes, hurricanes, and other natural disasters that can happen rather suddenly with constant changes. It can even apply to more specialized locations like laboratories or hospitals in case of the release of hazardous materials into the air. Efficient evacuation is critical in these worst-case scenarios, and being

able to properly design rooms and buildings with such escape routes in mind can be the difference between life and death. With an algorithm like the one we've used, we can test different initial conditions for crowd densities at the start of an evacuation, set other parameters like walking speed, a cost function (like we've seen in the 2D case) and obstacles that stand in the way, and interpret interesting results like when and where overcrowding might occur, where an alternative exit might be optimal, and how much time it takes for a particular crowd to completely evacuate. All of these factors should be handled with care in order to properly craft an ideal solution for bettering such protocols.

Furthermore, this research can certainly be expanded upon to emulate more crowd-related scenarios. First, this paper does have a heavy focus on evacuation - that is, the crowds are already in a certain area and need to leave through one of the available exit points. Research has already been (and is being) conducted on scenarios where people enter from a particular entry point and navigate a walking facility to reach the exit. The model presented in our research can certainly benefit from the addition of an inflow boundary in order to further expand our options and emulate different types of walking scenarios. Second, an interesting expansion to the model would be a jump into the third dimension, that is, describe a situation of a multi-floored building and determining the crowd behavior across facilities like stairs. However, at least in the case of humans, we cannot physically move freely in a 3D space, as opposed to 1D or 2D. Restrictions would have to apply in order to keep this sort of study grounded. Its application, however, to describe evacuation scenarios in buildings several stories high, like schools or business towers, to get from a certain part of the inside of the building, down a flight of stairs, and through an appropriate exit, is a fascinating prospect to explore and ponder over. Third, we can account for different definitions of the potential. We assume that people behave somewhat rationally and try to minimize their travel time and avoid dense crowds. While it would be complicated, it would be interesting to experiment with more extreme situations and different

goals that people may have in mind, especially because people behave erratically in emergency situations like fires and earthquakes. Another possibility for developments includes dynamic changes to the walking facility, such as if an exit opens or closes, or a new obstacle appears in the way of pedestrians. Incorporating these into the model would expand the scope of experimental possibilities. Finally, the study doesn't even have to stop at humans. Vehicle traffic in busy cities is just as great a problem and hindrance as pedestrian traffic. Modeling the movement of cars and trucks on the busy commute during peak hours to and from the overpopulated big city where most people work, for example, will help us better understand the sources of high vehicle congestion on the roads, as well as pave the way to adjusting for these problems, such as road expansion, building alternative path facilities like bridges, and other options. The options that such a model provides and inspires are plentiful, and give way to interesting future research projects.

Bibliography

[1] D. Helbing, I. J. Farkas, P. Molnar, and T. Vicsek, "Simulation of pedestrian crowds in normal and evacuation situations," *Pedestrian and Evacuation Dynamics*, vol. 21, no. 2, pp. 21–58, 2002.

[2] S. Z. Selim and A. H. Al-Rabeh, "On the modeling of pedestrian flow on the jamarat bridge," *Transportation Science*, vol. 25, no. 4, pp. 257–263, 1991.

[3] R. L. Hughes, "The flow of human crowds," *Annual Review of Fluid Mechanics*, vol. 35, no. 1, pp. 169–182, 2003.

[4] M. R. Virkler and S. Elayadath, *Pedestrian speed-flow-density relationships*, 1994, no. HS-042 012.

[5] W. H. Lam, J. F. Morrall, and H. Ho, "Pedestrian flow characteristics in hong kong," *Transportation Research Record*, no. 1487, 1995.

[6] S. J. Yuhaski and J. M. Smith, "Modeling circulation systems in buildings using state dependent queueing models," *Queueing Systems*, vol. 4, no. 4, pp. 319–338, 1989.

[7] L. Henderson, "The statistics of crowd fluids," *Nature*, vol. 229, no. 5284, pp. 381–383, 1971.

[8] A. Aw and M. Rascle, "Resurrection of "second order" models of traffic flow," *SIAM Journal on Applied Mathematics*, vol. 60, no. 3, pp. 916–938, 2000.

[9] J. A. Carrillo, S. Martin, and M.-T. Wolfram, "An improved version of the hughes model for pedestrian flow," *Mathematical Models and Methods in Applied Sciences*, vol. 26, no. 04, pp. 671–697, 2016.

[10] S. P. Hoogendoorn, P. H. Bovy, and W. Daamen, "Walking infrastructure design assessment by continuous space dynamic assignment modeling," *Journal of Advanced Transportation*, vol. 38, no. 1, pp. 69–92, 2004.

[11] C. Tong and S. Wong, "A predictive dynamic traffic assignment model in congested capacity-constrained road networks," *Transportation Research Part B: Methodological*, vol. 34, no. 8, pp. 625–644, 2000.

[12] R. L. Hughes, "A continuum theory for the flow of pedestrians," *Transportation Research Part B: Methodological*, vol. 36, no. 6, pp. 507–535, 2002.

[13] M. Di Francesco, P. A. Markowich, J.-F. Pietschmann, and M.-T. Wolfram, "On the hughes' model for pedestrian flow: The one-dimensional case," *Journal of Differential Equations*, vol. 250, no. 3, pp. 1334–1362, 2011.

[14] N. El-Khatib, P. Goatin, and M. D. Rosini, "On entropy weak solutions of hughes' model for pedestrian motion," *Zeitschrift für angewandte Mathematik und Physik*, vol. 64, no. 2, pp. 223–251, 2013.

[15] P. Goatin and M. Mimault, "The wave-front tracking algorithm for hughes' model of pedestrian motion," *SIAM Journal on Scientific Computing*, vol. 35, no. 3, pp. B606–B622, 2013.

[16] M. G. Crandall and P.-L. Lions, "Viscosity solutions of hamilton-jacobi equations," *Transactions of the American Mathematical Society*, vol. 277, no. 1, pp. 1–42, 1983.

[17] C. Bardos, A.-Y. LeRoux, and J.-C. Nédélec, "First order quasilinear equations with boundary conditions," *Communications in Partial Differential Equations*, vol. 4, no. 9, pp. 1017–1034, 1979.

[18] L. Huang, S. Wong, M. Zhang, C.-W. Shu, and W. H. Lam, "Revisiting hughes' dynamic continuum model for pedestrian flow and the development of an efficient solution algorithm," *Transportation Research Part B: Methodological*, vol. 43, no. 1, pp. 127–141, 2009.

[19] H. Zhao, "A fast sweeping method for eikonal equations," *Mathematics of Computation*, vol. 74, no. 250, pp. 603–627, 2005.

[20] R. J. LeVeque *et al.*, *Finite volume methods for hyperbolic problems*. Cambridge University Press, 2002, vol. 31.

[21] X. D. Liu, S. Osher, and T. Chan, "Weighted essentially nonoscillatory schemes," *Journal of Computational Physics*, vol. 115.

[22] Y.-T. Zhang, H.-K. Zhao, and J. Qian, "High order fast sweeping methods for static hamilton–jacobi equations," *Journal of Scientific Computing*, vol. 29, no. 1, pp. 25–56, 2006.

[23] G.-S. Jiang, "Shu ch.-w.: Efficient implementation of weighted eno schemes," *Journal of Computational Physics*, vol. 126, no. 1, pp. 202–228, 1996.

[24] G.-S. Jiang and D. Peng, "Weighted eno schemes for hamilton–jacobi equations," *SIAM Journal on Scientific Computing*, vol. 21, no. 6, pp. 2126–2143, 2000.

[25] C.-W. Shu and S. Osher, "Efficient implementation of essentially non-oscillatory shock-capturing schemes, ii," in *Upwind and High-Resolution Schemes*. Springer, 1989, pp. 328–374.

Appendix: Matlab Code

```
clear; %Code for our third example in Section 4.1
dx = 10^(-2); x = -1:dx:1; %space discretization
N = length(x);
ttime = 2.25;
dt = 10^(-3); T = ttime/dt; t = 0:dt:ttime; %time discretization and total time
    increments over 1.5s
%initial condition, set below
rho = zeros(1,N);
for i = 1:N,
if x(i) >= -0.5 && x(i) <= -0.25,
    rho(i) = 0.65;
end
if x(i) >= -0.1 && x(i) <= 0.3,
    rho(i) = 0.3;
end
if x(i) >= 0.5 && x(i) <= 0.9,
    rho(i) = 0.85;
end
end
writerObj = VideoWriter('Hughes1DHall2.avi');
writerObj.FrameRate = 50;
open(writerObj);
for i = 1:T,
    phiI = 10^12*ones(1,N); phiI(1) = 0; phiI(N) = 0; phi = zeros(1,N);
        phitester = phiI;
    %set the potential to zero at the exits, and high numbers elsewhere
    while norm(phitester - phi) > 10^(-9),
        phi = FastSweep(phiI,rho(i,:),dx); %fast sweeping method, see Zhao
            (2005)
        phitester = phiI; phiI = phi;
    end
    rhoNEW = Godunov(rho(i,:),phi,dx,dt);
    Time = t(i) %for me to keep track of what time we're at
    rho(i+1,:) = rhoNEW;
```

```
    plot(x,rho(i+1,:),'-'); ylim([0,0.9]);
    pause(0.001); %creates a mini-video of the pedestrian densities as time
        marches on.
    Frame = getframe(gcf);
    writeVideo(writerObj, Frame);
end
figure(2)
subplot(3,3,1)
plot(x,rho(0.25/dt,:),'-'); title('t = 0.25');
subplot(3,3,2)
plot(x,rho(0.5/dt,:),'-'); title('t = 0.5');
subplot(3,3,3)
plot(x,rho(0.75/dt,:),'-'); title('t = 0.75');
subplot(3,3,4)
plot(x,rho(1/dt,:),'-'); title('t = 1');
subplot(3,3,5)
plot(x,rho(1.25/dt,:),'-'); title('t = 1.25');
subplot(3,3,6)
plot(x,rho(1.5/dt,:),'-'); title('t = 1.5');
subplot(3,3,7)
plot(x,rho(1.75/dt,:),'-'); title('t = 1.75');
subplot(3,3,8)
plot(x,rho(2/dt,:),'-'); title('t = 2');
subplot(3,3,9)
plot(x,rho(2.25/dt,:),'-'); title('t = 2.25');
close(writerObj);
```

```
function y = FastSweep(phiI,rho,dx)
    N = length(rho);     f = ones(1,N)./func(rho); %setting up the function f on
        the rhs of the eikonal eq.
    phiF = zeros(1,N); phiFF = phiF; %allocating variables
    phiF(1) = phiI(1); phiF(N) = phiI(N); %boundary cond. for the potential
    for i = 2:N-1, %first sweep: left to right
        P = min(phiI(i-1),phiI(i+1));
        phiF(i) = min(P + dx*f(i),phiI(i));
    end
    phiFF(1) = phiI(1); phiFF(N) = phiI(N); %boundary cond. for the potential
    for i = N-1:-1:2, %second sweep: right to left
        P = min(phiF(i-1),phiF(i+1));
        phiFF(i) = min(P + dx*f(i),phiF(i));
    end
    y = phiFF;
end
```

```
function f = Godunov(rhoI,phiX,dx,dt)
```

```
    lam = dt/dx;
    N = length(rhoI); rhoNEW = rhoI;
    H = zeros(1,N-1); %variable allocation
    for i = 1:N-1, %setting up the Godunov flux at each node
        M = linspace(rhoI(i),rhoI(i+1),100);
        Gr = gradient(phiX(i:i+1));
        P = -1*F(M,Gr(1));
        if rhoI(i+1) >= rhoI(i),
            H(i) = min(P);
        else
            H(i) = max(P);  %H is the Godunov flux
        end
    end
    for j = 2:N-1,
        rhoNEW(j) = rhoI(j) - lam*(H(j) - H(j-1)); %calculating the new rho with
             the Godunov scheme
    end
    f = rhoNEW;
end
```

```
function f = func(x),
    N = length(x);
    f = ones(1,N) - x;
end
```

```
function f = F(rho,phi),
    f = sign(phi)*rho.*func(rho);
end
```

```
clear; format long; %Simulation of the first example in Section 4.2
dt = 0.1; dx = 1; dy = dx; Nx = 101; Ny = 51; ending = 60; T = dt:dt:ending;
%Nx is the number of points considered in the x direction.
%Same for Ny in the y direction. Reminder the grid is [0,100] x [0,50].
%dt: change step in time; dx and dy: change steps in x and y space.
%we will assume dx = dy for simplicity.
rho0 = zeros(Ny + 4,Nx + 4);
for i = 22:32,
    for j = 42:62,
        rho0(i,j) = 0.4;
    end
end
writerObj = VideoWriter('Hughes2DRoomSimple.avi');
writerObj.FrameRate = 15;
open(writerObj);
```

```
16 %Initial Condition is that people are situated in the center of the walking
       space
17 %Note that we take two extra increments to each side, to account for ghost
       points later.
18 rho = rho0; %New rho, just to allocate this matrix for later.
19 %Flux components, just to allocate for later.
20 HV = 10^12; %HV means high value for the FSM
21 %In this setup, F1 corresponds to the y flux, F2 for the x.
22 for t = 1:ending/dt, %Experiment runs for 3 seconds, taking into account the dt.
23     for k = 1:3,
24         cost = ones(Ny+4,Nx+4)./U(rho0); %cost function, taking to be as it was
               before in 1D.
25         phi0 = HV * ones(Ny + 4,Nx + 4);
26         %Here, we assume the initial potential at interior points is large.
27         phi0(3:Ny+2,1:3) = 0; phi0(3:Ny+2,Nx+2:Nx+4) = 0;
28         %Boundary condition for the 2 exits, we set phi to zero.
29         phi0(1:2,3:Nx+2) = HV; phi0(Ny+3:Ny+4,3:Nx+2) = HV; %HV means high value
30         phi = phi0; %New phi, just to allocate for later.
31         phitemp = zeros(Ny+4,Nx+4);
32         while norm(phi - phitemp) > 10^(-9)
33             phi = FSGuess(phi0,cost,dx); %Fast-Sweeping method in all four
                   sweeping directions. See Zhao (2005)
34             phitemp = phi0; phi0 = phi;
35         end
36         NewF1 = zeros(Ny+4,Nx+4); NewF2 = zeros(Ny+4,Nx+4);
37         for i = 3:Ny+2,
38             for j = 3:Nx+2,
39                 M = linspace(rho0(i,j),rho0(i+1,j),100); P = linspace(rho0(i,j),
                       rho0(i,j+1),100);
40                 if rho0(i+1,j) >= rho0(i,j),
41                     NewF1(i,j) = min(-1*(phi(i+1,j) - phi(i,j))*M.*(ones(1,100)-
                           M).^2); %Godunov Flux
42                 else
43                     NewF1(i,j) = max(-1*(phi(i+1,j) - phi(i,j))*M.*(ones(1,100)-
                           M).^2);
44                 end
45                 if rho0(i,j+1) >= rho0(i,j),
46                     NewF2(i,j) = min(-1*(phi(i,j+1) - phi(i,j))*P.*(ones(1,100)-
                           P).^2);
47                 else
48                     NewF2(i,j) = max(-1*(phi(i,j+1) - phi(i,j))*P.*(ones(1,100)-
                           P).^2);
49                 end
50             end
```

```
51          end
52          if k == 1, %first part of TVDRK
53              for i = 3:Ny+2,
54                  for j = 4:Nx+1,
55                          rho(i,j) = rho0(i,j) + dt*L(NewF2(i,j-1),NewF2(i,j),
                                NewF1(i-1,j),NewF1(i,j),dx,dy);
56                      %See L.m for details on how this function works.
57                  end
58              end
59              rhotemp = rho0; rho0 = rho;
60              else if k == 2, %second part of the TVDRK method.
61              for i = 3:Ny+2,
62                  for j = 4:Nx+1,
63                      rho(i,j) = 3/4*rhotemp(i,j) + (1/4)*(rho0(i,j) + dt*L(NewF2(
                            i,j-1),NewF2(i,j),NewF1(i-1,j),NewF1(i,j),dx,dy));
64                  end
65              end
66              rho0 = rho;
67              %Now repeat again with this third rho.
68              else
69                  for i = 3:Ny+2,
70                      for j = 4:Nx+1,
71                          rho(i,j) = 1/3*rhotemp(i,j) + (2/3)*(rho0(i,j) + dt*L(
                                NewF2(i,j-1),NewF2(i,j),NewF1(i-1,j),NewF1(i,j),dx,
                                dy));
72                      end
73                  end
74                  rho0 = rho;
75                  %new rho becomes the old, and we repeat everything for the next
                        time step.
76                  Time = T(t)
77                  %Outputting this for my tracking/debugging purposes.
78                  clf;
79                  X = 0:dx:Nx-1; Y = 0:dy:Ny-1;
80                  grid = rho(3:Ny+2,3:Nx+2);
81                  surf(X,Y,grid); colorbar; xlim([0,100]); ylim([0,50]); caxis
                        ([0,0.5]); view(0,90);
82                  pause(0.001);
83                  %We don't take the ghost points into consideration when plotting
                        ,
84                  %which is why we only took rho(3:Ny+2,3:Nx+2).
85                  %It also allows me to see the updated graph
86                  Frame = getframe(gcf);
87                  writeVideo(writerObj, Frame);
```

```
                end
            end
        end
end
% close the writer object
close(writerObj);
```

```
function y = FSGuess(phi,c,h),
    [Ny, Nx] = size(phi);
    phiNEW = phi; phiOLD = phi;
    for i = 3:Ny-2,
        for j = 3:Nx-2, %Left Sweep
            if phiOLD(i,j) ~= 0;
            a = min(phiOLD(i,j-1),phiOLD(i,j+1)); b = min(phiOLD(i-1,j),phiOLD(i
                +1,j));
            if abs(a-b) >= c(i,j)*h,
                phiNEW(i,j) = min(a,b) + c(i,j)*h;
            else
                phiNEW(i,j) = (a + b + (2*c(i,j)^2 * h^2 - (a -b)^2)^(0.5))/2;
            end
            phiNEW(i,j) = min(phiNEW(i,j),phiOLD(i,j));
            end
        end
    end
    phiOLD = phiNEW;
    for i = 3:Ny-2,
        for j = Nx-2:-1:3, %Right sweep
            if phiOLD(i,j) ~= 0;
            a = min(phiOLD(i,j-1),phiOLD(i,j+1)); b = min(phiOLD(i-1,j),phiOLD(i
                +1,j));
            if abs(a-b) >= c(i,j)*h,
                phiNEW(i,j) = min(b,a) + c(i,j)*h;
            else
                phiNEW(i,j) = (a + b + (2*c(i,j)^2 * h^2 - (a-b)^2)^(0.5))/2;
            end
            phiNEW(i,j) = min(phiNEW(i,j),phiOLD(i,j));
            end
        end
    end
    phiOLD = phiNEW;
    for i = Ny-2:-1:3,
        for j = Nx-2:-1:3,
            if phiOLD(i,j) ~= 0;
            a = min(phi(i,j-1),phiOLD(i,j+1)); b = min(phiOLD(i-1,j),phiOLD(i+1,
                j));
```

```
            if abs(a-b) >= c(i,j)*h,
                phiNEW(i,j) = min(a,b) + c(i,j)*h;
            else
                phiNEW(i,j) = (a + b + (2*c(i,j)^2 * h^2 - (a - b)^2)^(0.5))/2;
            end
            phiNEW(i,j) = min(phiNEW(i,j),phiOLD(i,j));
            end
        end
    end
    phiOLD = phiNEW;
    for i = Ny-2:-1:3,
        for j = 3:Nx-2,
            if phiOLD(i,j) ~= 0;
            a = min(phi(i,j-1),phiOLD(i,j+1)); b = min(phiOLD(i-1,j),phiOLD(i+1,
                j));
            if abs(a-b) >= c(i,j)*h,
                phiNEW(i,j) = min(a,b) + c(i,j)*h;
            else
                phiNEW(i,j) = (a + b + (2*c(i,j)^2 * h^2 - (a - b)^2)^(0.5))/2;
            end
            phiNEW(i,j) = min(phiNEW(i,j),phiOLD(i,j));
            end
        end
    end
    y = phiNEW;
end
```

```
function f = U(rho),
    [Ny,Nx] = size(rho);
    f = (ones(Ny,Nx) - rho);
    %speed-density relationship, can be changed for a new simulation
end
```

```
function f = L(f1i,f1f,f2i,f2f,dx,dy)
    f = (-1/dx).*(f1f - f1i) - (1/dy).*(f2f - f2i);
    %simple formula for the right hand side of the discretized equation (30) in
        the paper.
end
```