# New Models for Crowd Dynamics and Control

Sadeq J. Al-nasur

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical and Computer Engineering

Pushkin Kachroo, Chair
William T. Baumann
Daniel J. Stilwell
A. Lynn Abbott
Saad A. Ragab

December 14, 2006
Blacksburg, Virginia

Keywords: Pedestrian Evacuation Models, Crowd Dynamics, one-dimension PDE Systems,
Two-dimension PDE Systems, Feedback Control, Finite volume methods

# New Models for Crowd Dynamics and Control

Sadeq J. Al-nasur

(ABSTRACT)

In recent years, there has been an increasing interest in modeling crowd and evacuation dynamics. Pedestrian models are based on macroscopic or microscopic behavior. In this work, we are interested in developing models that can be used for evacuation control strategies. Hence, we use macroscopic modeling approach, where pedestrians are treated in an aggregate way and detailed interactions are overlooked. In this dissertation, we developed two-dimensional space crowd dynamic models to allow bi-directional flow by modifying and enhancing various features of existing traffic and fluid dynamic models. In this work, four models based on continuum theory are developed, and conservation laws such as the continuity and momentum equations are used. The first model uses a single hyperbolic partial differential equation with a velocity-density relationship, while the other three models are systems of hyperbolic partial differential equations. For one of the system models presented, we show how it can be derived independently from a microscopic crowd model. The models are nonlinear, time-varying, hyperbolic partial differential equations, and the numerical simulation results given for the four macroscopic models were based on computational fluid dynamics schemes.

We also started an initial control design that synthesizes the feedback linearization method for the one-dimensional traffic flow problem applied directly on the distributed parameter system. In addition, we suggest and discuss the information technology requirements for an evacuation system.

# Dedication

*To my family and friends.*

# Acknowledgments

I would like to express my sincere thanks and gratitude to my advisor Dr. Pushkin Kachroo, who has enlightened me to knowledge and learning. Through his constructive discussion and superb analytical skills, he has helped me greatly in research.

I am also grateful to the committee members, Prof. William T. Baumann, Dr. Daniel J. Stilwell, Dr. A. Lynn Abbott, and Prof. Saad A. Ragab for their helpful remarks.

Special acknowledgment is also given to Kuwait University for sponsoring me during this research.

Finally yet importantly, I am greatly indebted to my wife and my family for their understanding, patience and support during the entire period of my study.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

As the population grows, the demand for pedestrian facilities increases and so does the potential for accidents. The resultant financial and human injuries present a massive burden on society. Thus scientists and engineers face an increasing challenge of finding solutions that greatly improve pedestrians' management and safety. One of the solutions is to design better pedestrian facilities. Therefore, pedestrian analysis studies need to be done. In this book we present an analysis study, which can be used to evaluate the effect of a proposed policy (e.g., evacuation plan, exit numbers and locations) on the facility design before adopting and implementing this policy on the actual facility. The implementation of a policy without pedestrian analysis studies might lead to very costly trial and error (i.e. user cost, construction, marketing etc.). On the other hand, by using good analysis tools, the trial and error of a policy could be done at the analysis level, where the design components can be tested in a simulation environment and then adopted for the final design.

The work presented in this dissertation is concerned with developing pedestrian dynamic models that can be used in the control and the investigation of crowd behavior in normal and panic conditions. The development of pedestrian dynamic models to implement evacuation system strategies is an ongoing research area. Since the early 1990s a strong interest in this topic has shown the importance of this issue (Smith and , Eds.). Nevertheless, as

stated in (Wigan, 1993), our knowledge of the flow of crowds is inadequate and behind that of other transportation modes. Studies like the ones in (Griffith, 1982; Southworth et al., 1989), are concerned with evacuation strategies for regional areas like cities and states. They are important in the decision making of an emergency evacuation such as in the case of a natural disaster. In smaller areas like airports, stadiums, theaters, buildings, and ships, an evacuation system is an important element in design safety (Lovas, 1998). A good evacuation system in the case of an emergency can prevent a catastrophic outcome as shown by (Helbing et al., 2002).

It has been suggested in (May, 1990), that pedestrian traffic flow can be treated similarly to vehicle traffic flow, where we could divide the problem into two categories: the microscopic level and the macroscopic level. The former involves individual units with characteristics such as individual speed and individual interaction. Microscopic pedestrian analysis studies were presented by (Helbing, 1992; Henderson, 1974), and followed by many researchers to improve on the *Car-Following* model. In (Helbing and Molnar, 1997), the importance of a detailed design and pedestrian interactions is shown by implementing several case studies. On the other hand, the drawback to mathematical microscopic models is their difficult and expensive simulation, which leads to another approach called pedestrian analysis by simulation and is being increasingly used instead of the mathematical models (Helbing, 1994). In such models, pedestrians are treated as discrete individuals moving in a computer simulated environment. In this approach there are mainly three types of models, cellular based, physical force based, and queuing network based. More information on these microscopic models can be found in (Blue and Adler, 1998; Gips and Marksjo, 1985; Helbing, 1991; Helbing and Molnar, 1995; Helbing and Vicsek, 1999; Lovas, 1994; Okazaki, 1979; Okazaki and Matsushita, 1993; Thompson and Marchant, 1995a,b; Watts, 1987).

Macroscopic models aimed at studying pedestrian behavior use a continuum approach, where the movement of large crowds exhibit many of the attributes of fluid motion. As a result, pedestrian dynamics are treated as a fluid. This idea provides flexibility since detailed interactions are overlooked, and the model's characteristics are shifted toward parameters

such as flow rate $f(\rho)$, concentration $\rho$ (also known as traffic density), and average speed $v$, all being functions of 2-D space $(x, y)$ and time $(t)$. This class of models is classified under the hyperbolic partial differential equations. This way of modeling the pedestrian behavior was first introduced by (Fruin, 1971a,b), and adopted by (Bord, 1985), where macroscopic models can be developed using (a) fluid flow theory, (b) a continuum responding to influences (local or non-local). The drawback to this way of modeling is due to the assumption that pedestrians behave similarly to fluids. Pedestrians tend to interact among themselves and with obstacles in their model area, which is not captured by macroscopic models.

To introduce feedback control as a strategy in evacuation plans, where the objective is to control each pedestrian in the model area is difficult and not practical for the microscopic modeling approach. The obvious reasons for the impracticality are the detailed design and pedestrian interactions which make it very difficult to control such models. Instead, the macroscopic modeling approach is well suited for understanding the rules governing the overall behavior of pedestrian flow for which individual differences are not that important. By monitoring panic situations like an evacuation, we see that (a) people tend to move as a group and, (b) behave in a rational manner, since they think the situation is fully visible to the tallest member of the group, who is supposed to convey information to the shorter members by his or her actions (Hughes, 2002, 2003).

The first step in this study is to start investigating potential mathematical models governing crowd dynamics. Two factors need to be considered in deriving the models: the accurate representation of the pedestrian flow in an emergency situation (evacuation), and the complexity of the selected model.

In this work, four macroscopic crowd dynamic models have been developed. Their theoretical approaches are adopted to represent crowd motion in two-dimensional space (2-D). Four distinct one-dimensional traffic flow models are extended and modified to represent crowd dynamics in a 2-D environment. The resulting models are divided into two types, mainly one-equation and two equation crowd models (or systems) of nonlinear hyperbolic partial

differential equations (PDE). These models start from conventional theory for ordinary fluids. All the models assume a conservation of continuity and in the last three, the second partial differential equation is a modified version of the momentum equation which differs in each model in order to distinguish them from fluid behavior.

The second step is to simulate the systems responses. For hyperbolic PDE's, computational fluid dynamics methods are used. These methods are aimed at finding the correct solution out of the many weak solutions available for the hyperbolic PDE. Here we applied first order accurate finite volume method schemes such as Lax-Friedrichs, FORCE, and Roe's schemes to find the models numerical solution. Then we designed a preliminary control to evacuate pedestrians based on feedback linearization applied directly on the PDE equation. We also discuss the issues and requirements of modeling, control, and communications for an intelligent evacuation system.

Chapter 2 will introduce the basic theory for the traffic flow problem in one-dimensional space along with the four macroscopic traffic flow models that will be used to develop the crowd models. In Chapter 3, new two-dimensional crowd models are presented, and the microscopic-to-macroscopic relationship for the last models is investigated, where a micro-to-macro link is established. Chapter 4 survey the state of the art numerical schemes based on computational fluid dynamic that will be used to fined the numerical solution for Chapter 5 problems using Matlab software environment. Feedback linearization control for the one-dimensional space one-equation model is given in Chapter 6. In Chapter 7, we suggest an intelligent evacuation system, and discuss its control and information technology issues. Finally, in Chapter 8 conclusions drown from this study, and propose areas for development and future work are presented

# Chapter 2

# Traffic Flow Theory For 1-D

## 2.1 Introduction

Interest in modeling traffic flow has been around since the appearance of traffic jams. Ideally, if you can correctly predict the behavior of vehicle flow given an initial set of data, then, in theory, adjusting the flow in crucial areas can maximize the overall throughput of traffic along a stretch of road. This is of particular interest in regions of high traffic density, which may be caused by high volume peak time traffic, accidents or closure of one or more lanes of the road.

The development of the pedestrian evacuation dynamic systems follows from the traffic flow theory in one-dimensional space (AlGadhi and Mahmassani, 1991; Helbing, 1992; Hughes, 2002). In many ways, the pedestrian evacuation system is similar to the vehicle traffic flow problem (May, 1990). The main conservation equations used in modeling the vehicle traffic flow and the pedestrian evacuation flow are the same, with the exception that vehicle traffic is a one-dimensional space problem and the evacuation system is a two-dimensional space problem. Other similarities exist from having escape routes and escape times in both problems. In most of the situations a vehicle or a pedestrian has more than one route to a

destination and each route has an associated cost, such as time.

In this chapter we will give the necessary background on traffic flow theory and survey the existing macroscopic mathematical models for single-lane, one-dimensional space traffic flow. These models will be used for crowd flow in one-dimension, and they will be modified in chapter 3 for two-dimensional flow. In Section 2.2, we start with the concept of macroscopic vs. microscopic ways of modeling the traffic flow problem, followed by Section 2.3, where a microscopic model is introduced. The derivation of the traffic flow theory based on conservation of mass law, and the relationships between velocity and density are given in Section 2.4. In Section 2.5, four macroscopic traffic flow models are presented, derived, and analyzed based on their mathematical characteristics. Finally, the exact and weak solutions to the scalar traffic flow PDE, and the concepts of shock wave, rarefaction wave, and the admissibility of a solution are considered.

## 2.2 Microscopic vs. Macroscopic

In the traffic flow problem, there are two classes of models: Macroscopic, which is concerned with average behavior, such as traffic density, average speed and module area, and a second class of models based on individual behavior referred to as microscopic models. The latter is classified into different types. The most famous one is the *Car-Following* models (Bando, 1995; Chandler et al., 1958; Kachroo and Ozbay, 1999), where the driver adjusts his or her acceleration according to the conditions in front. In these models the vehicle position is treated as a continuous function and each vehicle is governed by an ordinary differential equation (ODE) that depends on speed and distance of the car in front. Another type of microscopic models are the *Cellular Automata* or vehicle hopping which differs from *Car-Following* in that it is a fully discrete model. It considers the road as a string of cells which are either empty or occupied by one vehicle. One such model is the *Stochastic Traffic Cellular Automata,* given in (Nagel, 1996). However, microscopic approaches are computationally

expensive, as each car has an ODE to be solved at each time step, and as the number of cars increases, so does the size of the system to be solved. On the other hand, the macroscopic models are computationally less expensive because they have fewer design details in terms of interaction among vehicles and between vehicles and their environment. Therefore, it is desirable to use macroscopic models if a good model can be found satisfactorily to describe the traffic flow. In addition, this idea provides flexibility since detailed interactions are overlooked, and the model's characteristics are shifted toward parameters such as flow rate $f(\rho, v)$, concentration $\rho$ (also known as traffic density), and average speed $v$, all functions of 1-D or 2-D space $(x, y)$, and time $(t)$. This is also true for first-order fluid dynamic models of isothermal flow and gases through pipes.

Two main prototypes set the stage for macroscopic traffic flow: the first is called the LWR model which is a *non-linear*, first-order hyperbolic PDE based on law of conservation of mass. The second one is a *second-order* model known as the PW model, which is based on two coupled PDE's one given by the conservation of mass and a second equation that mimics traffic flow.

## 2.3 Car-Following Model

We present the well known *car-following* microscopic traffic flow model. In (Sugiyama et al., 2002), a two dimensional version of this model was used for pedestrian flow in two-dimensional space. To derive the one-dimensional model, first assume cars can not pass each other. Then the idea is that a car in one-dimension can move and accelerate forward based on two parameters; the headway distance between the current car and the one in front, and their speed difference. Hence, it is called *following*, where a car from behind follows the one in front, and this is the anisotropic property. This property is also desirable in macroscopic models, since it reflects the actual observed behavior of traffic flow (Daganzo, 1995).

Suppose the $n$th car location is $x_n(t)$, then the nonlinear model is given by

$$\ddot{x}_n(t) = c\,\frac{\dot{x}_n(t) - \dot{x}_{n-1}(t)}{x_n(t) - x_{n-1}(t)}, \tag{2.1}$$

The acceleration of the current car $\ddot{x}_n(t)$ depends on the front car speed and location, $c$ is the sensitivity parameter. Integrating the above yields

$$\dot{x}_n(t) = c\ln\left(x_n(t) - x_{n-1}(t)\right) + d_n. \tag{2.2}$$

Since by the definition of the density (number of cars per unit area)

$$\frac{1}{\rho(x,t)} = x_n(t) - x_{n-1}(t), \tag{2.3}$$

and the integration constant $d_n$ is chosen such that at jam density $\rho_m$, the velocity is zero. Then for steady-state we get

$$v = -c\,\ln\frac{\rho}{\rho_m}. \tag{2.4}$$

We see that for $\rho \to 0$ we get in trouble, but from observations in low traffic densities, car speed is the maximum allowed speed, hence we can assume $v = v_{max}$, which is the maximum allowed speed.

## 2.4 Traffic Flow Theory

In this section we will cover the vehicle traffic flow fundamentals for the macroscopic modeling approach. The relation between density, velocity and flow is presented for traffic flow. Then we derive the conservation of vehicles, which is the main governing equation for scalar macroscopic traffic models. Finally, the velocity-density functions that makes the conservation equation a function of only one variable (density) are given.

### 2.4.1 Flow

In this section, we will illustrate the close relationship between the three variables: density, velocity and traffic flow. Suppose there is a road with cars moving with constant velocity $v_0$,

and constant density $\rho_0$ such that the distance between the cars is also constant as shown in the Figure 2.1-a. Now let an observer measure the number of cars per unit time $\tau$ that pass him (i.e. traffic flow $f$). In $\tau$ time, each car has moved $v_0\tau$ distance, and hence the number of cars that pass the observer in $\tau$ time is the number of cars in $v_0\tau$ distance, see Figure 2.1-b.

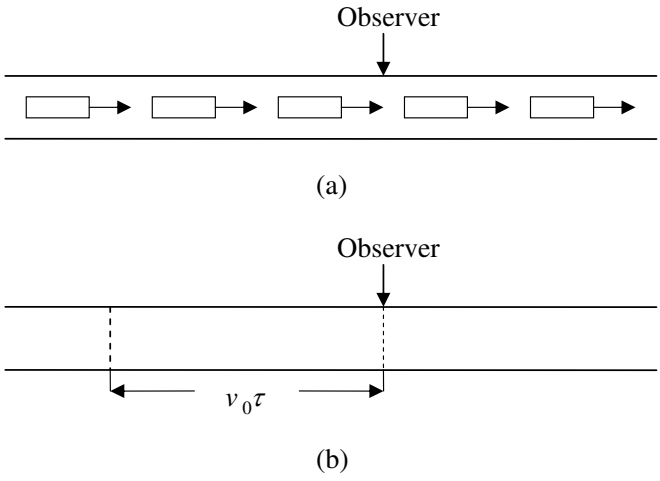


Figure 2.1: (a) Constant flow of cars, (b) Distance traveled in $\tau$ hours for a single car.

Since the density $\rho_0$ is the number of cars per unit area and there is $v_0\tau$ distance, then the traffic flow is given by

$$f = \rho_0 v_0 \tag{2.5}$$

This is the same equation as in the time varying case, i.e.,

$$f(\rho, v) = \rho(x,t)v(x,t). \tag{2.6}$$

To show this, consider the number of cars that pass point $x = x_0$ in a very small time $\Delta t$. In this period of time the cars have not moved far and hence $v(x,t)$, and $\rho(x,t)$ can be approximated by their constant values at $x = x_0$ and $t = t_0$. Then, the number of cars passing the observer occupy a short distance, and they are approximately equal to $\rho(x,t)v(x,t)\Delta t$, where the traffic flow is given by (2.6).

### 2.4.2 Conservation Law

The models for traffic, whether they are one-equation or system of equations, are based on the physical principle of *conservation*. When physical quantities remain the same during some process, these quantities are said to be conserved. Putting this principle into a mathematical representation will make it possible to predict the densities and velocities patterns at future time. In our case, the number of cars in a segment of a highway $[x_1, x_2]$ are our physical quantities, and the process is to keep them fixed (i.e., the number of cars coming in equals the number of cars going out of the segment). The derivation of the conservation law is given in (Farlow, 1982; Haberman, 1977), and it is presented here for completion. Consider a stretch of highway on which cars are moving from left to right as show in Figure 2.2. It is assumed here that there are no exit or entrance ramps. The number of cars within $[x_1, x_2]$ at a given time $t$ is the integral of the traffic density given by

$$N = \int_{x_1}^{x_2} \rho(x, t)\, dx. \tag{2.7}$$

In the above equation, it is implied that the number of people within $[x_1, x_2]$ is at maximum when traffic density is equal to jam density $\rho_m$ which is associated with the maximum number of cars that could possibly fit in a unit area.

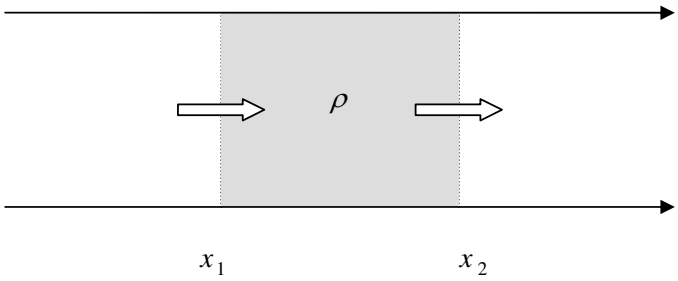


Figure 2.2: One-dimension Flow

The number of cars can still change (increase or decrease) in time due to cars crossing both

ends of the segment. Assuming no cars are crated or destroyed, then the change of the number of cars is due to the change at the boundaries only. Therefore, the rate of change of the number of cars is given by

$$\frac{dN}{dt} = f_{in}(\rho, v) - f_{out}(\rho, v), \tag{2.8}$$

since the number of cars per unit time is the flow $f(\rho, v)$. Combining (2.7), and (2.8), yields the **integral conservation law**

$$\frac{d}{dt}\int_{x_1}^{x_2} \rho(x,t)\,dx = f_{in}(\rho, v) - f_{out}(\rho, v). \tag{2.9}$$

This equation represents the fact that change in number of cars is due to the flows at the boundaries. Now let the end points be independent variables (not fixed with time), then the full derivative is replaced by partial derivative to get

$$\frac{\partial}{\partial t}\int_{x_1}^{x_2} \rho(x,t)\,dx = f_{in}(\rho, v) - f_{out}(\rho, v). \tag{2.10}$$

The change in the number of cars with respect to distance is given by

$$f_{in}(\rho, v) - f_{out}(\rho, v) = -\int_{x_1}^{x_2} \frac{\partial f}{\partial x}(\rho, v)\,dx, \tag{2.11}$$

and by setting the last two equations equal to each other, we get

$$\int_{x_1}^{x_2} \left[\frac{\partial \rho}{\partial t}(x,t) + \frac{\partial f}{\partial x}(\rho, v)\right] dx = 0. \tag{2.12}$$

This equation states that the definite integral of some quantity is always zero for all values of the independent varying limits of the integral. The only function with this feature is the zero function. Therefore, assuming $\rho(x,t)$, and $q(x,t)$ are both smooth, the **one-dimensional conservation law** is found to be

$$\rho_t + f_x(\rho, v) = 0. \tag{2.13}$$

We need to mention that this equation is valid for traffic and many more physical quantities. The idea here is conservation, and for vehicle traffic flow, the flow is given by (2.6).

### 2.4.3 Velocity-Density Relationship(s)

Traffic density and vehicle velocity are related by one equation, conservation of vehicles,

$$\rho_t(x,t) + (\rho(x,t)v(x,t))_x = 0, \tag{2.14}$$

where the notation $(.)_\phi = \frac{\partial(.)}{\partial\phi}$ will be used from here on. If the initial density and the velocity field are known, the above equation can be used to predict future traffic density. This leads us to choose the velocity function for the traffic flow model to be dependent on density and call it $V(\rho)$. The choice of such function depends on the behavior the model is trying to mimic. The following is a brief description of models that have been recognized and used by researchers (Kachroo and Ozbay, 1999), with emphasis on Greenshield model that will be used in several traffic (crowd) models throughout this work.

**Greenshield's Model (Greenshields, 1935)**

This model is simple and widely used. It is assumed here that the velocity is a linearly decreasing function of the traffic flow density, and it is given by

$$V(\rho) = v_f(1 - \frac{\rho}{\rho_m}) \tag{2.15}$$

where $v_f$ is the free flow speed and $\rho_m$ is the maximum density. Figure 2.3 shows the speed $V(\rho(x,t))$ as a monotonically decreasing function. For zero density the model allows free flow speed $v_f$, while for maximum density $\rho_m$ no car can move in or out.

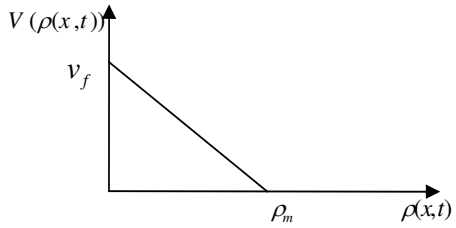


Figure 2.3: Greenshield's model for traffic flow speed

The flux-density relationship for Greenshield's model (2.15) is given in Figure (2.4), where it shows the flux increases to a maximum which occurs at some density $\hat{\rho}$ and then it goes back to zero. This kind of behavior is due to the fact that $f''(\rho) < 0$ (note that $f(\rho) = \rho V(\rho)$ is the flux flow).

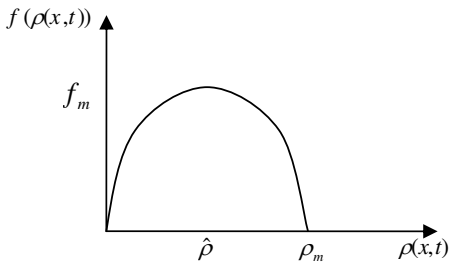


Figure 2.4: Traffic flow flux as a function of density

**Greenberg Model (Greenberg, 1959)**

In this model the speed-density function is given by

$$V(\rho) = v_f \, ln(\frac{\rho}{\rho_m}) \tag{2.16}$$

**Underwood Model (Greenberg, 1959)**

In the Underwood model the velocity-density function is represented by

$$V(\rho) = v_f \, \exp(\frac{-\rho}{\rho_m}) \tag{2.17}$$

**Diffusion Model (Burns and Kang, 1991; Musha and Higuchi, 1978)**

Diffusion is a good extension to the model given by equation (2.15), where the effect of gradual rather than instantaneous reduction of speed by the driver takes place in response to shock waves. This kind of reaction can be accomplished by adding an extra term such that the modified Greenshield model will become

$$V(\rho) = v_f(1 - \frac{\rho}{\rho_m}) - \frac{D}{\rho}(\frac{\partial \rho}{\partial x}) \tag{2.18}$$

where $D$ is a diffusion coefficient given by

$$D = \tau \, v_r^2$$

and $v_r$ is a random velocity, $\tau$ is a relaxation parameter.

## 2.5 Traffic Flow Model 1-D

In this section, we will present four different models for traffic flow in one-dimensional space. The first model is one equation model, and the rest are systems of two-equation models. All the models are described by partial differential equations and based on conservation of mass and a second equation that is intended to capture the complex interactions observed in traffic flow motion. In addition, this second equation provides another way to couple velocity and density. Hence the flow is not in equilibrium like the one equation model. This is one of the main differences between the scalar and the system models.

### 2.5.1 LWR Model

The first model used in describing the traffic flow problem known as the LWR model, named after the authors in (Lighthill and Whitham, 1955) and (Richards, 1956). The LWR model is a scalar, time-varying, non-linear, hyperbolic partial differential equation. The model governing equation is (2.14). In this equation, traffic density is the conserved quantity, and we rewrite the model as

$$\rho_t(x,t) + (\rho(x,t)V(\rho(x,t)))_x = 0 \tag{2.19}$$

with the flux being replaced by the velocity-density relationship

$$f(x,t) = \rho(x,t)V(\rho(x,t)) \tag{2.20}$$

and $V(\rho(x,t))$ is the velocity function given by (2.15).

One of the basic assumptions in the LWR model regarding the velocity is its dependence on density alone. Any changes to density will be reflected in the velocity. The drawback for this assumption, as pointed out in (Daganzo, 1995), is that traffic is in equilibrium when such velocity-density functions are used, i.e., given a particular density, especially for light traffic, the velocity will be fixed and the model does not recognize that there is a distribution of desired velocities across vehicles. Therefore the model is not able to describe observed behavior in light traffic, although one can argue that $v_f$ is an average speed which might take care of this issue. On the plus side, the model is anisotropic as the nature of the observed traffic flow, i.e. vehicle behavior is affected by mostly the car in front. This can be found from the model eigenvalue given by

$$f'(\rho(x,t)) = \frac{\partial f}{\partial \rho}(\rho(x,t)) = V(\rho(x,t)) + \rho V'(\rho(x,t)), \tag{2.21}$$

which means that the model allows information to travel as fast as the flow of traffic, and not more, since it satisfies $0 < f'(\rho) < V(\rho)$, because

$$V'(\rho) = -\frac{v_f}{\rho_m}. \tag{2.22}$$

The LWR model given by equations (2.19) and (2.15) is a simple model and it is unable to capture all of the complex interactions for a realistic traffic flow model. For this reason, modifications to the LWR model have been suggested. One way is by the various velocity-density functions we gave in subsection 2.4.3. The second way is by coupling the conservation of mass with a second equation that tries to mimic traffic motion instead of the velocity-density models as given next.

### 2.5.2 PW Model

The first system model to be presented is a two-equation model proposed in the 1970s independently in (Payne, 1971) and (Whitham, 1974). Their model was the first model to couple velocity dynamics as a second equation, and it is referred to as the PW model. The

first equation is the conservation of mass as discussed in the previous sections

$$\rho_t + (\rho v)_x = 0, \tag{2.23}$$

where the flux function $f(\rho, v) = \rho v$. In the LWR scalar equation model, a particular form of $v$ was assumed where velocity is a function of density, but in high order models, $v$ and $\rho$ are assumed to be independent and a second equation is formed to link them, as in fluid and gas models. The second equation is derived from the Navier-Stokes equation of motion for a one-dimensional compressible flow, but with the pressure term replaced by $P = C_0^2\rho$, where $C_0$ is the anticipation term that describes the response of macroscopic driver to traffic density, i.e. space concentration, and the pressure now is not "pressure" as such. The model also includes a traffic relaxation term that keeps speed concentration in equilibrium

$$\frac{V(\rho) - v}{\tau},$$

, where $\tau$ is a relaxation time, and the velocity $V(\rho)$ is the maximum out-of-danger velocity meant to mimic driver's behavior given earlier by equations (2.15) to (2.18). The second equation of the PW model in nonconservative form is then given by

$$v_t + v\ v_x = \frac{V(\rho) - v}{\tau} - \frac{C_0^2}{\rho}\ \rho_x. \tag{2.24}$$

To study this model, we have to find its eigenvalues by first rewriting the model in conservation form. The first step is to use the product rule

$$(\rho v)_t = \rho v_t + v\rho_t, \tag{2.25}$$

and by multiplying (2.23) by $v$, we get

$$v\rho_t + v(\rho v)_x = 0. \tag{2.26}$$

Then by substituting in for $v\rho_t$ from the product rule (2.25), we get

$$v(\rho v)_x + (\rho v)_t - \rho v_t = 0, \tag{2.27}$$

and by substituting (2.27) into (2.24), and multiply the result by $\rho$ we get

$$\rho v_t + \rho v\ v_x = \rho\left(\frac{V(\rho) - v}{\tau}\right) - C_0^2\rho_x.$$

Then by substituting for $\rho v_t$ from (2.27) we get

$$v(\rho v)_x + (\rho v)_t + \rho v\ v_x = \rho\left(\frac{V(\rho) - v}{\tau}\right) - C_0^2\rho_x. \tag{2.28}$$

Again using the product rule on $(\rho vv)_x$, i.e.,

$$(\rho vv)_x = (\rho v)_x v + (\rho v)v_x \tag{2.29}$$

and substituting in (2.28) we obtain

$$(\rho v^2)(\rho v)_t + (\rho v^2)_x = \rho\left(\frac{V(\rho) - v}{\tau}\right) - C_0^2\rho_x. \tag{2.30}$$

Hence we obtain the equation (2.24) with the lefthand side in conservation form

$$(\rho v)_t + (\rho v^2 + C_0^2\rho)_x = \rho\left(\frac{V(\rho) - v}{\tau}\right) \tag{2.31}$$

where now $\rho$ and $\rho v$ are the conserved variables. Equations (2.23), and (2.31) can be written in vector form as

$$Q_t + F(Q)_x = S \tag{2.32}$$

where,

$$Q = \begin{bmatrix} \rho \\ \rho v \end{bmatrix},\ F(Q) = \begin{bmatrix} \rho v \\ \rho v^2 + C_0^2\rho \end{bmatrix},\ S = \begin{bmatrix} 0 \\ \rho\left(\dfrac{V(\rho) - v}{\tau}\right) \end{bmatrix}. \tag{2.33}$$

Setting the source term $S = 0$, we can rewrite the system in quasi-linear form as

$$Q_t + A(Q)\,Q_x = 0, \tag{2.34}$$

where

$$A(Q) = \frac{\partial F}{\partial Q} = \begin{bmatrix} 0 & 1 \\ C_0^2 - \rho v^2 & 2\,v \end{bmatrix}. \tag{2.35}$$

Finally by solving for the eigenvalues from

$$|A(Q) - \lambda I| = 0 \tag{2.36}$$

we get two distinct and real eigenvalues

$$\lambda_{1,2} = v \pm C_0, \tag{2.37}$$

therefore, the system is strictly hyperbolic.

The model has a major drawback that researchers (see for example (Daganzo, 1995)) are concerned about, mainly, that the model strongly follows the fluid flow theory. In fluids, the behavior of a particle is affected by its surrounding particles. Thus the anisotropic nature of traffic is not preserved since the vehicles are allowed to move with negative velocity, i.e. against the flow. This is clear from the eigenvalue, where one of $\lambda_{1,2} = v \pm C_0$ is always grater than the vehicle speed $v$. So, information from behind affects the behavior of the driver, and this is not true for observed traffic flow. This is called the isotropic property.

### 2.5.3 AR Model

A new model in (Aw and Rascle, 2000) and improved in (Rascle, 2002) is argued to be an improvement on the PW model. The authors of this model say that other researchers have stuck too closely to fluid flow models and have not allowed for a significant difference between traffic and fluids, e.g., traffic is more concerned with the flow in front, rather than behind. Therefore in order to move away from fluids and toward the anisotropic property of traffic, they argue that replacing the "pressure" term with an anticipation term describing how the average driver behaves is not a sufficient fix for the differences between the two types of flow. They claim that the drawback in the PW model (letting information travel faster than the flow) is due to an incorrect anticipation factor involving the derivative of the pressure w.r.t. $x$. Therefore they suggest the correct dependence must involve the convective derivative (full derivative) of the pressure term. The convective derivative in its general form is given by

$$\frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + (\vec{v}.\nabla)\phi, \tag{2.38}$$

for $\phi(\vec{x},t)$, and $\vec{x} \in \Re^n$. They support their claim by the following example: "Assuming that in front of a driver traveling with speed $v$ the density is increasing with respect to $x$, but decreasing with respect to $(x-vt)$. Then the PW type models predict that this driver would slow down, since the density ahead is increasing with respect to $x$! On the contrary, any reasonable driver would accelerate, since this denser traffic travels faster than him."

We call the model AR for short, and the first PDE equation is the same conservation of cars given by (2.23). However, in the AR model the next lagrangian equation replaces the second PDE equation given in the PW model. This second equation is found by applying the full derivative (2.38) to describe traffic motion dynamics, and it is given by

$$(v+P(\rho))_t + v\,(v+P(\rho))_x = 0, \tag{2.39}$$

where $P(\rho)$ is an increasing function of density. This choice is to ensure that this model caries the anisotropic property, and it is given by

$$P(\rho) = C_0^2\,\rho^\gamma, \tag{2.40}$$

where $\gamma > 0$, and $C_0 = 1$. Next we will put the second equation in conservation form, and find the system eigenvalues. First multiply (2.39) by $\rho$, then by using the product rule

$$\begin{aligned} (\rho\,(v+P(\rho)))_t &= \rho_t(v+P(\rho)) + \rho\,(v+P(\rho))_t, & (2.41)\\ (\rho v\,(v+P(\rho)))_x &= (\rho v)_x(v+P(\rho)) + (\rho v)(v+P(\rho))_x, & (2.42) \end{aligned}$$

we obtain

$$(\rho\,(v+P(\rho))_t - \rho_t(v+P(\rho)) + (\rho v\,(v+P(\rho)))_x - (\rho v)_x(v+P(\rho)) = 0. \tag{2.43}$$

Now, using the conservation law (2.23) for $\rho_t$, we can simplify the above equation to

$$(\rho\,(v+P(\rho)))_t + (\rho v\,(v+P(\rho)))_x = 0, \tag{2.44}$$

which is the conservation form of (2.39). Then our conserved variables are $\rho$, and $\rho\,(v+P(\rho))$. We proceed now to find the system eigenvalues, let $X = \rho\,(v+P(\rho))$ for simplification, then

the AR model given by (2.23), and (2.44) can be rewritten as

$$\begin{cases} \rho_t + (X - \rho P(\rho))_x &= 0 \\ X_t + \left(\dfrac{X^2}{\rho} - XP(\rho)\right)_x &= 0 \end{cases} \tag{2.45}$$

and in vector form (2.32), the stats and the flux are given by

$$Q = \begin{bmatrix} \rho \\ X \end{bmatrix}, \; F(Q) = \begin{bmatrix} X - \rho P(\rho) \\ \dfrac{X^2}{\rho} - XP(\rho) \end{bmatrix}. \tag{2.46}$$

For the quasi-linear form (2.34), the Jacobian is given by

$$A(Q) = \frac{\partial F}{\partial Q} = \begin{bmatrix} -(\gamma+1) & 1 \\ -\left(\dfrac{X^2}{\rho^2} + \dfrac{\gamma X P(\rho)}{\rho}\right) & \left(\dfrac{2X}{\rho} - P(\rho)\right) \end{bmatrix}. \tag{2.47}$$

Finally, solving for the eigenvalues from $|A(Q) - \lambda I| = 0$, we find two distinct and real eigenvalues

$$\lambda_1 = v - \gamma P(\rho) \quad \& \quad \lambda_2 = v. \tag{2.48}$$

Therefore, the system is strictly hyperbolic and since the "pressure" is an increasing function, then it is guaranteed that $\lambda_1 < \lambda_2$ due to the fact that the maximum wave speed is equal to the velocity of the flow $v$. Hence, the anisotropic property of traffic is preserved.

### 2.5.4 Zhang Model

We present here a model that was proposed in (Zhang, 1998, 2002), and it claims not to be of fluid, or gas-like behavior. The model caries the anisotropic property, because the second equation is derived from the microscopic *car-following* model. Hence, a micro-to-macro link is established for this model. Again, as in the PW and AR models, the Zhang model is also a system consisting of the conservation of cars (2.23), and coupled with a second PDE that describe car motion given by

$$v_t + vv_x + \rho V'(\rho) v_x = 0 \tag{2.49}$$

We start the micro-to-macro derivation from the *homogeneous* microscopic *car-following* model (the relaxation term can be added for 2-D crowd flow given in the next chapter) given by

$$\tau(s_n(t))\,\ddot{x}_n(t) = \dot{x}_{n-1}(t) - \dot{x}_n(t), \tag{2.50}$$

where

$$s_n(t) = x_{n-1}(t) - x_n(t), \tag{2.51}$$

and $s_n(t)$ is a function of the local spacing between cars, $x_n(t)$ is the position of the $n$th car, $\ddot{x}_n(t)$ is the acceleration, $\dot{x}_n(t)$ is the velocity, and $\tau(s_n(t))$ is the average response time to the headway distance. Using the above notations, we rewrite (2.50), and define the velocity as $v(x,t) = \dot{x}(t)$ to obtain the following

$$\tau(s(x(t),t))\,\frac{dv(x,t)}{dt} = \frac{d(s(x(t),t))}{dt}, \tag{2.52}$$

and by using convective derivative $\partial_t + v\partial_x$ on the velocity component, we get

$$\tau(s)\,(v_t + vv_x) = (s_t + vs_x). \tag{2.53}$$

From the conservation law (2.23), let $\rho = 1/s$, and by using the following derivative form

$$D_x(\frac{a}{b}) = \frac{bD_xa - aD_xb}{b^2}, \tag{2.54}$$

for any $a$ and $b \neq 0$, we get

$$s_t + vs_x + us_y = sv_x + su_y, \tag{2.55}$$

and by direct substituting in the right hand side of equation (2.53), we obtain our desired equation in the following form

$$(v_t + vv_x) = \frac{s}{\tau(s)}v_x, \tag{2.56}$$

where

$$\frac{s}{\tau(s)} = -C(\rho) = -\rho V'(\rho) \geq 0 \tag{2.57}$$

is the sound wave speed. This completes the derivation of the macroscopic model (2.49) from its microscopic counterpart.

The conservative form of this model is derived next. First collect terms and rewrite (2.49) to get

$$v_t + (v + \rho V'(\rho))\, v_x = 0, \tag{2.58}$$

then expand the conservation of mass equation

$$\rho_t + \rho v_x + v\rho_x = 0. \tag{2.59}$$

We substitute for $\rho v_x$ from (2.59) into (2.58) to obtain

$$v_t + vv_x + V'(\rho)\,(-\rho_t - v\rho_x) = 0, \tag{2.60}$$

which can be rewritten as

$$v_t + vv_x - (V(\rho))_t - v\,(V(\rho))_x = 0, \tag{2.61}$$

or in lagrangian form as

$$(v - V(\rho))_t + v\,(v - V(\rho))_x = 0. \tag{2.62}$$

We now proceed to find the conservation form by multiplying (2.62) by $\rho$ and using the product rules

$$\begin{aligned} (\rho(v - V(\rho)))_t &= \rho_t(v - V(\rho)) + \rho(v - V(\rho))_t, & (2.63)\\ (\rho\, v\,(v - V(\rho)))_x &= (\rho v)_x(v - V(\rho)) + \rho v\,(v - V(\rho))_x, & (2.64) \end{aligned}$$

to get

$$(\rho(v - V(\rho)))_t - \rho_t(v - V(\rho)) + (\rho v\,(v - V(\rho)))_x - (v\rho)_x(v - V(\rho)) = 0. \tag{2.65}$$

From (2.59), we substitute for $\rho_t$ in the above equation to obtain our final conservation form given by

$$(\rho\,(v - V(\rho)))_t + (\rho v\,(v - V(\rho)))_x = 0, \tag{2.66}$$

where our states are given by $\rho$ and $\rho(v - V(\rho))$. For the quasi-linear form (2.34), let $X = \rho(v - V(\rho))$, and write the system in vector form (2.32), such that

$$Q = \begin{bmatrix} \rho \\ X \end{bmatrix},\ F(Q) = \begin{bmatrix} X + \rho V(\rho) \\ \dfrac{X^2}{\rho} - XV(\rho) \end{bmatrix}. \tag{2.67}$$

The Jacobian is then can be found to be

$$A(Q) = \frac{\partial F}{\partial Q} = \begin{bmatrix} V(\rho) + \rho V'(\rho) & 1 \\ -\left(\frac{X^2}{\rho^2} - XV'(\rho)\right) & \left(\frac{2X}{\rho} + V(\rho)\right) \end{bmatrix}. \tag{2.68}$$

Finally by solving for the eigenvalues from $|A(Q) - \lambda I| = 0$, we find two distinct and real eigenvalues

$$\lambda_1 = v + \rho V'(\rho) \quad \& \quad \lambda_2 = v \tag{2.69}$$

therefore the system is strictly hyperbolic. Since $V'(\rho)$ is negative and given by (2.22), then the maximum the information can travel is equal to the vehicle speed $v$.

### 2.5.5 Models Summary

The one-equation LWR model consists of a single wave whose velocity is given by the derivative of the flux function, and information travels forward at a maximum not faster than the speed of traffic. Therefore the model behavior is anisotropic, i.e. only reacts to conditions ahead. For the two-equation models, the PW has two waves traveling at speeds given by $v \pm C_0$, one of them will always be traveling faster than the current speed $v$. This is a major cause of criticism of this model. The AR model has wave speeds given by $v$ and $v - \gamma P(\rho)$. This seems reasonable since, as in the LWR model, the faster wave will move at the same speed as the traffic $v$. This is also true for the Zhang model, whose wave speeds are given by $v$ and $v + \rho V'(\rho)$ with $V'(\rho) \leq 0$. This demonstrates the desirable anisotropic nature of the LWR, AR and Zhang models, and the isotropic nature of the PW model, for which it is severely criticized. In addition, Zhang model has a microscopic counterpart, which is not true for the PW and AR models. Although, the AR model has an indirect micro-to-macro relationship (Aw and Rascle, 2000), where a numerical discretization of the AR model and a microscopic *car-following* model gave the same numerical formula. This suggest that the macroscopic AR model can be considered as an upper limit to the microscopic model.

## 2.6 Method of Characteristics

A typical problem in partial differential equation consists of finding the solution of a PDE subject to boundary conditions (BVP), initial conditions (IVP), or both (IBVP). In most cases it is difficult to find the exact (classical) solution of a hyperbolic PDE, but due to the simplicity of the LWR model and the fact it is a scalar one-dimensional space model we are able to find the exact solution by method of characteristics. The method of characteristics is a widely used technique to solve hyperbolic PDE's (Emmanuele, 1995; Kevorkian, 2000; Ockendon et al., 2003; Renardy and Rogers, 2004).

### 2.6.1 LWR Model Classification

The partial derivative scalar conservation law in (2.19) is classified as first-order quasi-linear partial differential equation. This is due to the fact that the derivative of the highest partial occurs linearly. We can rewrite equation (2.19) as

$$\rho_t(x,t) + f'(\rho(x,t))\ \rho_x = 0 \tag{2.70}$$

where $f'(\rho)$ is the vehicle speed, and it is called the characteristic slop or the eigenvalue of the PDE. By using Greenshield's model (2.15), we get

$$f'(\rho(x,t)) = \frac{df(\rho(x,t))}{dx} = v_f - \frac{2\ v_f\ \rho(x,t)}{\rho_m}, \tag{2.71}$$

where we see that this eigenvalue is real. Therefore, the LWR model is classified as strictly hyperbolic PDE. Figure 2.5 below shows the changes in speed $f'(\rho)$ with respect to the changes in density $\rho$. This relationship is important in finding the solution to the traffic flow model (2.70) by using method of characteristics discussed in the following subsection.

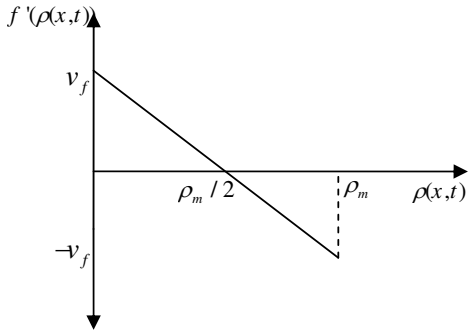


Figure 2.5: Characteristic slops vs. density

### 2.6.2 Exact Solution

Here we will use the method of characteristics to solve the initial value problem (IVP) and also called the Cauchy problem given by

$$\begin{cases} \rho_t(x,t) + f(\rho(x,t))_x \;=\; 0 \\ \qquad\qquad\qquad\qquad \rho(x,0) = \rho_0(x) \end{cases} \tag{2.72}$$

where $x \in \Re$ and time $t \in \Re^+$. Flow rate $f : \Re \mapsto \Re$ is assumed to be a smooth function, at least $C^2$ (i.e., twice differentiable) and the initial condition $\rho_0 : \Re \mapsto \Re$ is continuous. For the single conservation law, the eigenvalue of the PDE in (2.72) is given by the slope of the characteristic curve found from the quasi-linear form (2.70) as

$$\lambda(\rho) = f'(\rho) \tag{2.73}$$

**Theorem 2.6.1** *Any $C^1$ solution of the single conservation law in (2.72) is constant along its characteristics. Accordingly, characteristic curves for the partial derivative conservation law in (2.72) are straight lines.*

Proof. See (Renardy and Rogers, 2004). The above theorem implies that any curve of the form

$$x(t) = k\,t + x(0) \tag{2.74}$$

is a characteristic curve where $x(t)$ is the solution, $k = f'(\rho(x(t), t))$ is the constant slope of the characteristic rays and $x(0)$ is the initial position of the characteristics rays. To show that equation (2.74) is indeed a solution to our Cauchy problem, let us first define what is meant by a solution.

**Definition 2.6.1** *Let $f : \Re \mapsto \Re$ be smooth, and let $\rho_0 : \Re \mapsto \Re$ be continuous. We say that $\rho(x,t) : (\Re \times \Re^+) \mapsto \Re$ is a classical solution of the Cauchy problem if $\rho(x,t) \in C^1(\Re \times \Re^+) \cap C^0(\Re \times \Re^+)$ and (2.70) is satisfied.*

We can verify that it is indeed a solution by substituting for $x(0) = x - f't$ from (2.74) to get

$$\rho(x,t) = \rho_0(x - f'\,t), \tag{2.75}$$

then, by taking partial derivatives with respect to $t$ and $x$, respectively, we obtain

$$\rho_t = \rho_0'(x - f'\,t)(-f'), \quad and \quad \rho_x = \rho_0'(x - f'\,t).$$

Substituting the above in (2.70), we get

$$\rho_0'(x - f'\,t)(-f') + f'\rho_0'(x - f'\,t) = 0,$$

which shows (2.72) is satisfied. So, the exact solution is basically the initial data shifted by the slope of the characteristic as shown in Figure 2.6

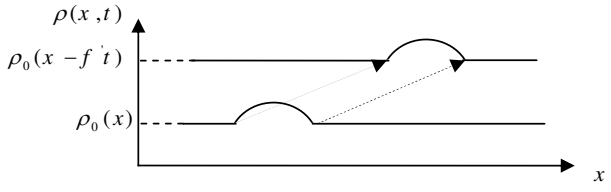


Figure 2.6: Exact solution is achieved by shifting the initial density profile

From the initial data we are able to generate the slopes of the characteristic rays originating from the x-axis. For some certain profiles of initial data, this gives us a method for solving

the Cauchy problem. To illustrate this method, let us consider the initial data in Figure 2.7, where we have the density profile of heavy traffic density at one end and light at the other end. Substituting for the initial density values in (2.71) will give the slopes of each of the characteristics. Then the solution follows from (2.74) along the rays of the characteristics as shown on the same figure.

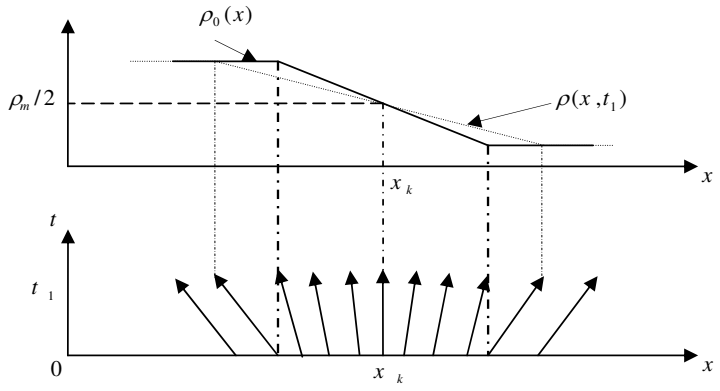


Figure 2.7: Density distribution in the upper part, and the corresponding characteristic rays in the lower part

### 2.6.3 Blowup of Smooth Solutions

Unfortunately, other examples with different initial data show how easily the procedure above fails. In Figure 2.8, we see a density profile that describes road condition when cars are approaching red traffic light. Although initial density is continuous, the characteristics overlap at some later point in time. Since our solution cannot be multi-valued, we must conclude (in light of Theorem 2.6.1) that the solution shown cannot be smooth. For this type of initial data, a theory of discontinuous solutions, or *shock wave* solution is used.

Moreover, in Figure 2.9 below, we face a different kind of problem. This time we have initial profile corresponding to heavy traffic at the beginning, then at some point $x_k$ it is lighter. This example can be related to conditions of a red light turns to green. As we all

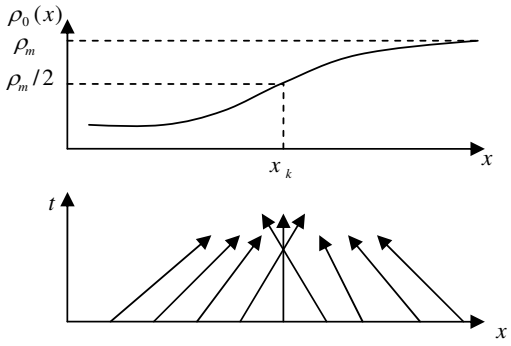


Figure 2.8: Overlapping characteristics from continuous initial data

observe in real situation, cars start to accelerate from high density (low speed) to low density (higher speed). The exact solution for this data shows that there is a region untouched by any characteristics from the given initial data. Thus, the method of characteristics did not identify a solution in this region. As we shall see next, for this case we will be able to identify a continuous solution called a *rarefaction or fan wave* solution to fill the wedge.

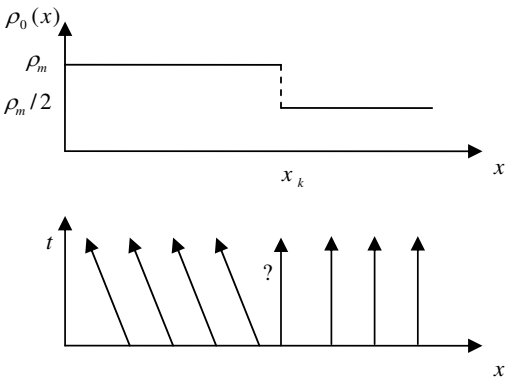


Figure 2.9: Characteristics do not specify solution in the wedge

### 2.6.4 Weak Solution

From the discussion above, smooth solutions of a single conservation laws can blow up (develop discontinuities or singularities) in finite time which fails to make any sense. Therefore,

one cannot follow the practice of accepting solutions to the hyperbolic partial differential equations as given directly by method of characteristics. In order to understand discontinuous solutions, one needs to extend the notion of solution itself. One of the main features of the quasi-linear theory for hyperbolic PDE's is the notion of *weak solutions*. For a given initial data,

**Definition 2.6.2** *Let $\rho_0 \in L^\infty$. Then $\rho$ is a weak solution or a solution in the distributional sense of (2.72) if and only if $\rho \in L^\infty(\Re \times \Re^+)$ and,*

$$\int_0^\infty \int_{-\infty}^\infty [\rho(x,t)\,\phi_t(x,t) \;+\; f(\rho(x,t))\,\phi_x(x,t)]\;dx\,dt + \int_{-\infty}^\infty \rho_0(x)\,\phi(x,0)\;dx = 0 \tag{2.76}$$

*is satisfied for every $\phi \in C_0^\infty(\Re \times [0,\infty[)$*

and $C_0^\infty(\Re\times[0,\infty[) := \{C_0^\infty(\Re\times[0,\infty[) \mid \exists\, r > 0$ s.t. support of $\phi \subset B_r(0,0)\cap(\Re\times[0,\infty[)\}$

Here $\phi(x,t)$ is a test function with compact support on the boundary (i.e., $\phi(x,t)$ is zero outside the boundary). In the weak solution (2.76), the partial derivative is moved to the test function that is guaranteed to be smooth. In addition, the definition above is an extension of the classical solution according to

**Theorem 2.6.2** *Suppose $\rho \in C^1(\Re \times [0,\infty[)$ is a classical solution of (2.72). Then $\rho$ is also a weak solution.*

Proof is given in (Renardy and Rogers, 2004).

We have to keep in mind that a *weak solution* might not be a classical solution. So we need necessary and sufficient conditions for the weak solutions to be the correct solution. We start by the necessary condition for a piecewise-smooth weak solution known as the Rankine-Hugoniot condition given by

$$s = \frac{[f(\rho)]}{[\rho]} = \frac{f(\rho_R) - f(\rho_L)}{\rho_R, -\rho_L} \tag{2.77}$$

where $s$ is the shock speed. Let's look at the earlier example in Figure 2.8, where the initial density was given by

$$\rho(x,0) = \begin{cases} \rho_m/2 & x < 0, \\ \rho_m & x \geq 0, \end{cases} \tag{2.78}$$

and we seek a solution to our Cauchy problem (2.72), using the method of characteristics. Since the characteristics do overlap at some point in time, the shock speed is calculated from (2.77)

$$\begin{aligned} s &= \frac{(v_f\rho_R - v_f\rho_R^2/\rho_m) - (v_f\rho_L - v_f\rho_L^2/\rho_m)}{\rho_R - \rho_L} \\ &= v_f - \frac{v_f}{\rho_m}(\rho_R + \rho_L) \\ &= v_f - \frac{v_f}{\rho_m}(\rho_m + \rho_m/2) \\ &= -0.5\, v_f \end{aligned}$$

and the solution is given by

$$\rho(x,t) = \begin{cases} \rho_m/2 & x < s\,t, \\ \rho_m & x \geq s\,t. \end{cases} \tag{2.79}$$

This solution is shown in Figure 2.10

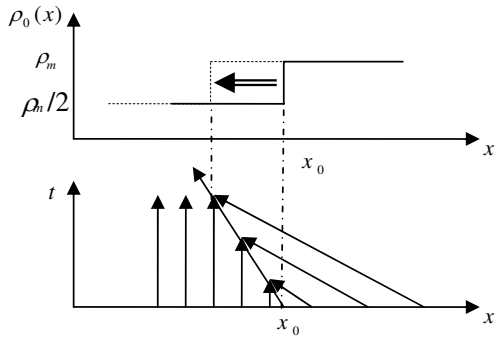


Figure 2.10: Shock solution

Let's look now at the example of Figure 2.9, and try to solve the traffic flow problem there. We will give two methods to find the solution and discuss which one must be used to get

the correct solution. Using the initial density values

$$\rho(x,0) = \begin{cases} \rho_m & x < 0, \\ \rho_m/2 & x \geq 0, \end{cases} \tag{2.80}$$

we get the first solution as a shock wave given by

$$s \;=\; v_f - \frac{v_f}{\rho_m}(\rho_R + \rho_L) \;=\; v_f - \frac{v_f}{\rho_m}(\rho_m/2 + \rho_m) \;=\; -0.5\, v_f$$

As we can see "$-0.5\ v_f$" is the same shock speed as in the previous example and it is plotted in the Figure 2.11-b. The second solution is continuous and it provides another way to fill the wedge. It is called the *rarefaction wave* solution. The general form of the solution for traffic flow is given by

$$\rho(x,t) = \begin{cases} \rho_L & x < f'(\rho_L)\, t, \\ f'(\frac{x}{t})^{-1} \qquad f'(\rho_L)\, t \leq & x < f'(\rho_R)\, t, \\ \rho_R & x \geq f'(\rho_R)\, t, \end{cases} \tag{2.81}$$

where the "-1" is an inverse mapping. For the traffic flow problem, $f'(\frac{x}{t})^{-1}$ can be found by letting

$$f'(\frac{x}{t})^{-1} \;=\; v_f - \frac{v_f \rho}{\rho_m} \;=\; \frac{x}{t}, \tag{2.82}$$

and solving for the density solution to get

$$\rho(x,t) = f'(\frac{x}{t})^{-1} = \frac{\rho_m}{2} - \frac{\rho_m\, x}{2 v_f t}. \tag{2.83}$$

Then, the continuous solution for the PDE is given by

$$\rho(x,t) = \begin{cases} \rho_L & x < -v_f\, t, \\ f'(\frac{x}{t})^{-1} \qquad -v_f\, t \leq & x < 0, \\ \rho_R & x \geq 0. \end{cases} \tag{2.84}$$

For the initial data given in Figure 2.11-a, the *rarefaction wave* solution is plotted in Figure 2.11-c.

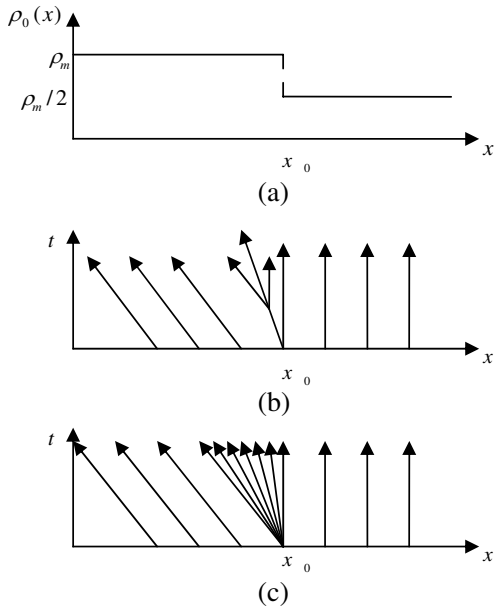


Figure 2.11: Initial density profile followed by two weak solutions, shock and fan respectively.

Such multiplicity of solutions is unacceptable. Thus we need a selection criterion that picks out the physically reasonable solution from among the possible weak solutions (2.81), and (2.84). Lax shock condition or *Lax entropy* condition (Lax, 1972) is a sufficient condition for scalar conservation laws. The condition for the traffic flow PDE states that the discontinuous solution for the traffic flow problem is admissible (i.e., a shock solution is selected with the direction of increased entropy) if

$$\rho_L < s < \rho_R, \tag{2.85}$$

otherwise, the rarefaction wave solution is the admissible one (see Figure 2.12 for easy interpretation). In the example of Figure 2.11-a, and according to the Lax condition, the rarefaction solution is the admissible one and the shock solution is not admissible. Finally, we summarize the solution for the LWR model by mentioning the following two points:

- The solution is piece-wise smooth as $t \mapsto \infty$ with jumps in density (shocks) separating the pieces.
- This means traffic is predicted to be stable with transition between stable regions

approximated by discontinuous shocks.


Figure 2.12: Lax shock condition for traffic flow problem: (a) not a shock, (b) shock

# Chapter 3

# Crowd Models for 2-D

## 3.1 Introduction

In this chapter we present four macroscopic crowd dynamic models that can be used to study crowd behavior. These models modify the one-dimensional traffic flow models so that bi-directional controlled flow is possible. The models have different characteristics; for instance, the first model is a simple conservation of pedestrians based on the one-equation partial deferential equation model given in subsection 2.5.1. The other three are based on a one-dimensional space systems of two-coupled partial deferential equations (two-equation models) given in subsections 2.5.2, 2.5.3, and 2.5.4 respectively.

The first system model which is called "the second" model in this chapter follows fluid flow closely, such that it carries the isotropic nature of fluid flow. The last two models move away from fluid flow and mimic traffic flow observed behavior by having the anisotropic property. In addition, the last model not only inherits the same anisotropic property, but it can also be derived from microscopic *car-following* model (Al-nasur and Kachroo, 2006).

This chapter is organized in the following matter. Section 3.2 gives the two-dimensional flow theory for hyperbolic PDE's. In section 3.3 we present the one-equation crowd model.

Section 3.4 shows the first system model derivation, conservation form, and eigenvalues followed by sections 3.5, and 3.6 for the second and third systems respectively. A comparison between the crowd models developed based on the analytical results is presented in section 3.7. Finally, we conclude this chapter with a linearized version of the four models.

## 3.2 Traffic Flow Theory in 2-D

In this section we extend the same ideas that is found in section 2.6. We consider the scalar equation in two-dimension:

$$\rho_t + f_x(\rho) + g_y(\rho) = 0 \quad in \ \Re^2 \times \Re^+, \tag{3.1}$$

$$\rho(.,0) = \rho_0 \quad in \ \Re^2, \tag{3.2}$$

for given functions $f$ and $g$ and $\rho$. We assume that $f$ and $g$ are sufficiently smooth. As we have seen the solution of the scalar one-dimension conservation law can become singular within finite time, the same can happen here. Therefore, similar to the definition given by 2.6.2, a weak solution of (3.1), and (3.2) need to be defined.

**Definition 3.2.1** *Let $\rho_0 \in L^\infty(\Re^2)$. Then $\rho$ is called a weak solution of (3.1), (3.2) if and only if $\rho \in L^\infty(\Re^2 \times \Re^+)$ and if,*

$$\int_{\Re^2} \int_{\Re^+} [\rho\, \phi_t + f(\rho)\, \phi_x + g(\rho)\, \phi_y] + \int_{\Re^2} \rho_0\, \phi(.,0) = 0$$

*is satisfied for every $\phi \in C_0^\infty(\Re^2 \times [0, \infty[)$*

For completion, we give the entropy condition for the two dimensional one-equation case.

**Definition 3.2.2** *A weak solution of (3.1), (3.2) is called an entropy solution if we have for all $\phi \in C_0^\infty(\Re^2 \times \Re^+)$, $\phi \geq 0$ and for all $k \in \Re$*

$$\begin{aligned} \int_{\Re^2} \int_{\Re^+} \{\partial_t \phi\, |\rho - k| &+ \partial_x \phi\, sign(\rho - k)\, [f(\rho) - f(k)] \\ &+ \partial_y \phi\, sign(\rho - k)\, [g(\rho) - g(k)]\} \geq 0 \end{aligned}$$

The following theorem gives the existence and uniqueness of an entropy solution for the scalar two-dimensional hyperbolic PDE in (3.1),

**Theorem 3.2.1** *Let $\rho_0 \in L^1(\Re^2) \cap L^\infty(\Re^2)$. Then there exist one and only one entropy solution $\rho$ of (3.1), (3.2) and $\rho \in C^0([0,T],\, L^1(\Re^2)) \cap L^\infty([0,T], \Re^2)$.*

The details of this proof can be found in (Kruzkov, 1970). For systems in two-dimension, Let $Q \in \Re^m$, then the scalar equation (3.1) and (3.2) become a system of PDE's for two-dimensions given by

$$Q_t + F_x(Q) + G_y(Q) = 0 \quad in \;\; \Re^2 \times \Re^+, \tag{3.3}$$

$$Q(.,0) = Q_0 \quad in \;\; \Re^m, \tag{3.4}$$

and

**Definition 3.2.3** *The system is called (strictly) hyperbolic system if all eigenvalues of $\alpha F^{'}(Q) + \xi G^{'}(Q)$ are real (and distinct) for all $\alpha,\, \xi \in \Re,\, Q \in \Re^m$.*

We also note that if the two matrices coefficients do not commute, i.e., $AB \neq BA$ and the matrices $A(Q)$ and $B(Q)$ do not have the same eigenvectors (Leveque, 2002), then they can be diagonalized separately

$$A = R^x \Lambda (R^x)^{-1}, \quad B = R^y \Lambda (R^y)^{-1}, \tag{3.5}$$

where the two matrices have different eigenvectors. This means that the PDE's are more intricately coupled.

For two-dimensional hyperbolic PDE's, the general existence of a solution does not exist for (3.3) globally in time (Bressan, 2003). This implies that there are no convergent results for numerical schemes. Therefore we can only use some schemes that have been successful in numerical test problems as we will see in the next chapter (Kröner, 1997).

## 3.3 One Equation Crowd Model

In this section, we present a 2-D scalar, nonlinear, time-varying, hyperbolic PDE crowd model. This is a simple extension to the LWR one-dimension conservation of continuity model given in subsection 2.4.3. The model consist of crowd concentration (density) and flow rate $f(\rho) = \rho V(\rho)$ in the *x-axis*, and $g(\rho) = \rho U(\rho)$ in the *y-axis*. We use the fundamental velocity-density relation given by (2.15) to describe $V(\rho)$ and $U(\rho)$. The free flow speed $v_{f1}$, $v_{f2}$ are the velocities in the *x-axis* and *y-axis*, and they are used as control parameters to direct crowd flow to any desired direction. Here, they are taken as constant, but in control design, they become a function of time and space. The model in conservation form is given by

$$\rho_t + (\rho V(\rho))_x + (\rho U(\rho))_y = 0, \tag{3.6}$$

and its quasi-linear form is

$$\rho_t + f'(\rho)\rho_x + g'(\rho)\rho_y = 0. \tag{3.7}$$

The characteristic slopes, or the eigenvalues of this scalar PDE are simply $f'(\rho)$, and $g'(\rho)$ and they are given by

$$\lambda^f = v_{f_1}(1 - 2\rho/\rho_m), \tag{3.8}$$

$$\lambda^g = v_{f_2}(1 - 2\rho/\rho_m), \tag{3.9}$$

The eigenvalues are real and distinct, therefore, the LWR model is classified as strictly hyperbolic since $f'(\rho)$, and $g'(\rho)$ are distinct and real respectively. The model carries the anisotropic property as the maximum characteristics are $v_{f1}$ and $v_{f2}$ respectively.

## 3.4 First System Crowd Dynamic Model

The first system model to be extended from 1-D to 2-D is based on the higher order model given in subsection 2.5.2. The model is classified as nonlinear, time-varying, hyperbolic

PDE traffic flow system. This model uses two-coupled PDE's to describe crowd flow; the conservation of continuity and a second equation that looks like the momentum equation in fluid flow with a modification to the "pressure" term to mimic crowd motion. This model is known for its isotropic nature that is preserved when we extend the model to two-dimensional space assuming pedestrian motion is influenced from all directions.

### 3.4.1 Model Description

Here we extend equations (2.23), and (2.24) from 1-D to 2-D, and add relaxation terms to allow bi-directional flow. The first equation is the two-dimensional conservation law given by

$$\rho_t + (\rho v)_x + (\rho u)_y = 0 \tag{3.10}$$

where the density $\rho(x, y, t)$ depends on the two spatial dimensions and time, $v(x, y, t)$ and $u(x, y, t)$ are the *x-axis* and *y-axis* components of the velocity. The flux flow rate in both directions is represented by $\rho v$ and $\rho u$. The above equation is defined on $x \in \Re$, $y \in \Re$ and time $t \in \Re^+$. We can derive the above equation from Figure 3.1, where the conserved density changes according to the changes in the flow at the boundary endpoints.

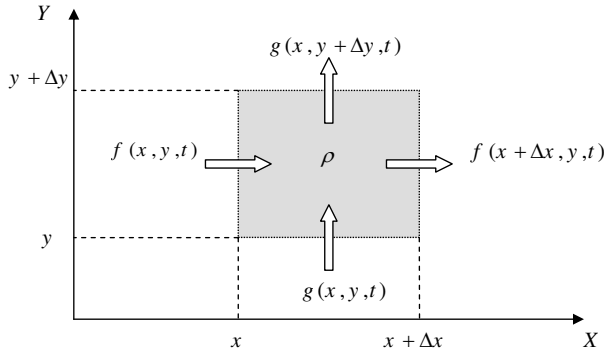


Figure 3.1: Two-dimensional Flow

The second equation is derived by taking a 2-D form of equation (2.24) and it is found to be

$$v_t + vv_x + uv_y + \frac{C(\rho)}{\rho}\rho_x = s_1, \tag{3.11}$$

$$u_t + vu_x + uu_y + \frac{C(\rho)}{\rho}\rho_y = s_2, \tag{3.12}$$

where the equations in non-conservative form represent the *x-axis*, and *y-axis* components. Initial condition are $\rho(x, y, 0) \geq 0$, and $\vec{v}(\vec{x}, 0) \leq |v_f|$ . The "pressure" term is the same as in the 1-D case, and it is equal to $P(\rho) = \rho C(\rho)$, where $C(\rho)$ is the anticipation factor that is equal to a constant squared $C_0^2$. The relaxation terms

$$s_1 = \frac{V(\rho) - v}{\tau} \tag{3.13}$$

$$s_2 = \frac{U(\rho) - u}{\tau} \tag{3.14}$$

rule is to capture how pedestrians modify their velocity according to the desired one $V(\rho)$, and $U(\rho)$ as given in equation (2.15) over a period of time denoted by the relaxation time $\tau$.

### 3.4.2 Conservation Form and Eigenvalues

The model developed is a nonlinear, time-varying hyperbolic PDE system. There is no analytical solution for this kind of systems, and we know for hyperbolic PDE's we have discontinuous solutions, i.e., more than one solution at some point in time and space (for more details on hyperbolic PDE's, see (Leveque, 2002)). Therefore, we need to put the model in a form that can be solved numerically and provide results that are consistent with the observed behavior. To do so, we will derive the conservation form of the model which will be used in finding the numerical solution (more on this will be discussed in the next chapter). Using this form we also prove the system hyperbolic property and its isotropic nature by finding the system roots according to definition 3.2.3.

After some manipulations to the equations in (3.11), and (3.12) we rewrite them in conser-

vation form as

$$(\rho v)_t + (\rho v^2 + \rho C_0^2)_x + (\rho v u)_y = \rho s_1, \tag{3.15}$$

$$(\rho u)_t + (\rho v u)_x + (\rho u^2 + \rho C_0^2)_y = \rho s_2. \tag{3.16}$$

In vector form, the model can be written as

$$Q_t + F(Q)_x + G(Q)_y = S \tag{3.17}$$

where $Q$ is the conservative variables (states), $F$ and $G$ are the fluxes in the two space dimension, and $S$ is considered as the source term. These are given by

$$Q = \begin{bmatrix} \rho \\ \rho v \\ \rho u \end{bmatrix}, \; F(Q) = \begin{bmatrix} \rho v \\ \rho v^2 + C_0^2 \rho \\ \rho v u \end{bmatrix}, \; G(Q) = \begin{bmatrix} \rho u \\ \rho v u \\ \rho u^2 + C_0^2 \rho \end{bmatrix},$$

$$S = \begin{bmatrix} 0 \\ \rho \left( \dfrac{V(\rho) - v}{\tau} \right) \\ \rho \left( \dfrac{U(\rho) - u}{\tau} \right) \end{bmatrix}.$$

Next we write the system in quasi-linear form, and by setting the relaxation terms to zero, our homogenous hyperbolic PDE in vector form is

$$Q_t + A(Q)\, Q_x + B(Q)\, Q_y = 0 \tag{3.18}$$

where the flux Jacobian matrices $A(Q)$ and $B(Q)$ can be found from the partial derivative of the fluxes given by

$$A = \frac{\partial F}{\partial Q} = \begin{bmatrix} 0 & 1 & 0 \\ C_0^2 - v^2 & 2v & 0 \\ -vu & u & v \end{bmatrix}, \; \textit{and} \; B = \frac{\partial G}{\partial Q} = \begin{bmatrix} 0 & 0 & 1 \\ -vu & u & v \\ C_0^2 - u^2 & 0 & 2u \end{bmatrix}.$$

The eigenvalues of the matrices A and B are found from the flux Jacobian matrices roots calculated from

$$|A(Q) - \lambda I| = 0. \tag{3.19}$$

The corresponding eigenvalues and eigenvectors for the $A$ matrix are given by

$$\lambda_1^A = v - C_0, \quad \lambda_2^A = v, \quad \lambda_3^A = v + C_0,$$

and

$$e_1^A = \begin{bmatrix} 1 \\ v - C_0 \\ u \end{bmatrix}, \; e_2^A = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \; e_3^A = \begin{bmatrix} 1 \\ v + C_0 \\ u \end{bmatrix}.$$

For the $B$ matrix they are given by

$$\lambda_1^B = u - C_0, \quad \lambda_2^B = u, \quad \lambda_3^B = u + C_0,$$

and

$$e_1^B = \begin{bmatrix} 1 \\ v \\ u - C_0 \end{bmatrix}, \; e_2^B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \; e_3^B = \begin{bmatrix} 1 \\ v \\ u + C_0 \end{bmatrix}$$

To check if the system is (strictly) hyperbolic or not, we need to do more than just getting the eigenvalues for A and B separately. Since this is a 2-D problem, we find the overall system eigenvalues from the roots of the combined Jacobian matrixes satisfying definition 3.2.3. Hence, the first crowd dynamic model eigenvalues are found to be

$$\check{\lambda}_1 = \check{v} - C_0, \quad \check{\lambda}_2 = \check{v}, \quad \check{\lambda}_3 = \check{v} + C_0, \tag{3.20}$$

where $\check{v} = \alpha v(x, y, t) + \xi u(x, y, t)$. The corresponding eigenvectors are given by

$$e_1^{\check{A}} = \begin{bmatrix} 1 \\ v - \alpha C_0 \\ u - \xi C_0 \end{bmatrix}, \quad e_2^{\check{A}} = \begin{bmatrix} 0 \\ -\xi \\ \alpha \end{bmatrix}, \quad e_3^{\check{A}} = \begin{bmatrix} 1 \\ v + \alpha C_0 \\ u + \xi C_0 \end{bmatrix}.$$

The system eigenvalues are real and distinct, and their corresponding eigenvectors are linearly independent, which confirms that the model is strictly hyperbolic. In addition, the system preserves its isotropic nature as seen from $\lambda_{1,3}$, where one of them is always grater than $\lambda_2$. This means that information from all directions affect pedestrians' motion (isotropic nature) and this is the most distinct characteristic of this model.

## 3.5 Second Crowd Dynamic Model

Here we present a crowd dynamic model in 2-D space based on a 1-D traffic flow model given in (Aw and Rascle, 2000; Rascle, 2002). This model is considered an improvement over the first system model since it does not closely follow fluid and gas-like models. In addition, it allows observed traffic conditions such as traffic flow with the flow in front, rather than with the flow that is upstream. Hence, the model carries the desired anisotropic nature of traffic flow, which is carried from the one-dimensional model by modifying the "pressure" terms.

### 3.5.1 Model Description

This model is also a nonlinear, time-varying, hyperbolic system of two PDE's. We develop the model by first extending (2.23), and (2.39) from a 1-D to a 2-D PDE system. The first PDE is the same conservation of continuity given in (3.10). The second equation is derived for the *x-axis* and *y-axis* by applying the convective derivative (2.38) on the pressure terms for the 2-D case. By doing so, the resulting second equation is given by

$$(v + P_1(\rho, v))_t + v(v + P_1(\rho, v))_x + u(v + P_1(\rho, v))_y = \rho s_1, \tag{3.21}$$

$$(u + P_2(\rho, u))_t + v(u + P_2(\rho, u))_x + u(u + P_2(\rho, u))_y = \rho s_2. \tag{3.22}$$

Initial condition are $\rho(x, y, 0) \geq 0$, and $v(x, 0) \leq |v_{f1}|$ and $u(y, 0) \leq |v_{f2}|$. The $P_1(\rho, v)$, and $P_2(\rho, u)$ functions proposed here are given by

$$P_1(\rho, v) = \frac{v\rho^{\gamma+1}}{\beta - \rho^{\gamma+1}}, \tag{3.23}$$

$$P_2(\rho, u) = \frac{u\rho^{\gamma+1}}{\beta - \rho^{\gamma+1}}, \tag{3.24}$$

and valid for $\gamma > 0$, and $\beta > \rho_m^{\gamma+1}$ such that we do not divide by zero or change the sign that could affect the system dynamics. We modified these functions from the one-dimension case to maintain the increasing property of the density that is required to achieve the anisotropic property of the system. Thus, the model preserve the anisotropic property and the modifica-

tion is justified by the system Eigenvalues found next. The second modification is adding the relaxation terms $s_1$, and $s_2$ that made possible to simulate bi-directional pedestrian motion.

### 3.5.2 Conservation Form and Eigenvalues

To show that the crowd dynamic model in (3.10), (3.21), and (3.22) is a hyperbolic PDE system, and numerically find its solution. We need to find the system conservation form and find its eigenvalues. We restrict our self to derive the *x-component* only since the *y-component* can be obtained by the same procedure. First, we start by multiplying (3.21) by $\rho$ to get

$$\rho(v + P_1(\rho, v))_t + \rho v\,(v + P_1(\rho, v))_x + \rho u\,(v + P_1(\rho, v))_y = \rho s_1, \tag{3.25}$$

and we drop $(\rho, v)$ from $P_1(\rho, v)$ for convenience for the rest of this subsection. We know from the product rule that

$$\begin{aligned}
(\rho\,(v + P_1))_t &= \rho_t(v + P_1) + \rho\,(v + P_1)_t, \\
(\rho v\,(v + P_1))_x &= (\rho v)_x(v + P_1) + (\rho v)(v + P_1)_x, \\
(\rho u\,(v + P_1))_y &= (\rho u)_y(v + P_1) + (\rho u)(v + P_1)_y,
\end{aligned}$$

and by substituting the above terms in (3.25) and using (3.10) to write the model in conservation form (follow same steps for the $Y$-component using $P_2$) we get

$$\rho_t + (\rho v)_x + (\rho u)_y = 0 \tag{3.26}$$

$$(\rho(v + P_1))_t + (\rho v\,(v + P_1))_x + (\rho u\,(v + P_1))_y = \rho s_1 \tag{3.27}$$

$$(\rho\,(u + P_2))_t + (\rho v\,(u + P_2))_x + (\rho u\,(u + P_2))_y = \rho s_2 \tag{3.28}$$

Next is to write the system in two-dimensional vector form (3.17) where $Q$ is the conservative variables (states), and $F$ & $G$ are the fluxes in the $x$ and $y - directions$ respectively. The model in vector form is given by

$$\begin{bmatrix} \rho \\ \rho(v + P_1) \\ \rho(u + P_2) \end{bmatrix}_t + \begin{bmatrix} \rho v \\ \rho v(v + P_1) \\ \rho v(u + P_2) \end{bmatrix}_x + \begin{bmatrix} \rho u \\ \rho u(v + P_1) \\ \rho u(u + P_2) \end{bmatrix}_y = \begin{bmatrix} 0 \\ \rho s_1 \\ \rho s_2 \end{bmatrix}. \tag{3.29}$$

From the quasi-linear form (3.18), we can obtain A and B eigenvalues by finding the flux Jacobian matrices $A = \partial F/\partial Q$ and $B = \partial G/\partial Q$. For this model the matrices and the corresponding eigenvalues and eigenvector are found by first simplifying the calculations. To do so, we change variables by setting the states to be

$$Q = \begin{bmatrix} \rho \\ w \\ z \end{bmatrix}, \; and \; \begin{cases} \rho(v + P_1) = w \Rightarrow v = \dfrac{w}{\rho} - P_1, \\ \rho(u + P_2) = z \Rightarrow u = \dfrac{w}{\rho} - P_2. \end{cases}$$

We substituted the new states in the fluxes to get

$$F(Q) = \begin{bmatrix} w - \rho P_1 \\ \dfrac{w^2}{\rho} - wP_1 \\ \dfrac{wz}{\rho} - zP_1 \end{bmatrix}, \quad G(Q) = \begin{bmatrix} z - \rho P_2 \\ \dfrac{wz}{\rho} - wP_2 \\ \dfrac{z^2}{\rho} - zP_2 \end{bmatrix},$$

and the Jacobian are

$$A(Q) = \begin{bmatrix} -P_1 - \rho P_{1\rho} & 1 - \rho P_{1w} & 0 \\ -\dfrac{w^2}{\rho^2} - wP_{1\rho} & 2\dfrac{w}{\rho} - P_1 - zP_{1w} & 0 \\ -\dfrac{wz}{\rho^2} - zP_{1\rho} & \dfrac{z}{\rho} - zP_{1w} & \dfrac{w}{\rho} - P_1 \end{bmatrix} \tag{3.30}$$

$$B(Q) = \begin{bmatrix} -P_2 - \rho P_{2\rho} & 0 & 1 - \rho P_{2z} \\ -\dfrac{wz}{\rho^2} - wP_{2\rho} & \dfrac{z}{\rho} - P_2 & \dfrac{w}{\rho} - wP_{2z} \\ -\dfrac{z^2}{\rho^2} - zP_{2\rho} & 0 & 2\dfrac{z}{\rho} - P_2 - zP_{2z} \end{bmatrix}. \tag{3.31}$$

We solve for the eigenvalues and eigenvectors for the $A(Q)$ matrix to get

$$\begin{aligned} \lambda_1^A &= \frac{w - (\rho P_1 + \rho^2 P_{1\rho} + w\rho P_{1w})}{\rho} = v - v\frac{(\gamma+1)\rho^{\gamma+1}}{\beta - \rho^{\gamma+1}} \\ &= v - (\gamma+1)P_1 \\ \lambda_{2,3}^A &= \frac{w - \rho P_1}{\rho} = v \end{aligned}$$

$$e_1^A = \begin{bmatrix} \dfrac{\beta-\rho^{\gamma+1}}{\beta u} \\ \dfrac{v}{u} \\ 1 \end{bmatrix},\ e_2^A = \begin{bmatrix} \dfrac{(\beta-\rho^{\gamma+1})^2}{\beta v(\beta+\gamma\rho^{\gamma+1})} \\ 1 \\ 0 \end{bmatrix},\ e_3^A = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

and for $B(Q)$ matrix we obtain

$$\begin{aligned} \lambda_1^B &= \frac{z-(\rho P_2+\rho^2 P_{2\rho}+z\rho P_{2z})}{\rho} = u - u\frac{(\gamma+1)\rho^{\gamma+1}}{\beta-\rho^{\gamma+1}} \\ &= u-(\gamma+1)P_2 \\ \lambda_{2,3}^B &= \frac{z-\rho P_2}{\rho} = u \end{aligned}$$

$$e_1^B = \begin{bmatrix} \dfrac{\beta-\rho^{\gamma+1}}{\beta v} \\ 1 \\ \dfrac{u}{v} \end{bmatrix},\ e_2^B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},\ e_3^B = \begin{bmatrix} \dfrac{(\beta-\rho^{\gamma+1})^2}{\beta u(\beta+\gamma\rho^{\gamma+1})} \\ 0 \\ 1 \end{bmatrix}.$$

Here we also find the eigenvalues by solving the combined Jacobian matrixes satisfying definition 3.2.3. For this system the eigenvalues are found to be

$$\check{\lambda}_1 = \check{v} - \check{v}\frac{(1+\gamma)\rho^{\gamma+1}}{\beta-\rho^{\gamma+1}} = \check{v}-(1+\gamma)\check{P}, \quad \check{\lambda}_{2,3} = \check{v}$$

and their corresponding eigenvectors are given by

$$e_1^{\check{A}} = \begin{bmatrix} \frac{\beta-\rho^{\gamma+1}}{\beta u} \\ \frac{v}{u} \\ 1 \end{bmatrix}, e_2^{\check{A}} = \begin{bmatrix} \alpha\frac{(\beta-\rho^{\gamma+1})^2}{\beta\check{v}(\beta+\gamma\rho^{\gamma+1})} \\ 1 \\ 0 \end{bmatrix}, e_3^{\check{A}} = \begin{bmatrix} \xi\frac{(\beta-\rho^{\gamma+1})^2}{\beta\check{v}(\beta+\gamma\rho^{\gamma+1})} \\ 0 \\ 1 \end{bmatrix}$$

Clearly, we have real eigenvalues with two of them repeated. Nevertheless, we are able to find linearly independent eigenvectors, and therefore the crowd dynamic model derived is a nonlinear hyperbolic system of PDE. In addition, the "pressure" term is an increasing function of density, and from the eigenvalue $\check{\lambda}_1$, we are sure to have the maximum wave speed to be $\check{v}$. Thus, the model has the desired anisotropic property (information that affect the current position depends on current $\check{\lambda}_{2,3}$, and ahead $\check{\lambda}_1$ information only).

## 3.6 Third Crowd Dynamic Model

The third crowd dynamic model presented here is a macroscopic model derived from a microscopic car-following model. The model is based a one-dimensional macroscopic traffic flow model given in subsection 2.5.4. This crowd two-dimension model is different from the other models because it has a direct microscopic-to-macroscopic link. The model is classified as a traffic flow nonlinear, time-varying, hyperbolic system of two partial differential equations. The anisotropic property is carried from the microscopic to the macroscopic model assuming that pedestrian motion is influenced mainly from current and front conditions.

Here we present the macroscopic model, and then we derive the macroscopic system from the microscopic model. Finally, we put the crowd model in conservation form and find its eigenvalues and eigenvectors.

### 3.6.1 Model Description

The first equation is the two-dimensional conservation of continuity that conserve mass (pedestrians) given by (3.10). The second equation is similar to the momentum equations in 2-D for comprisable flow with some manipulation to mimic crowd dynamics and it is given by

$$v_t + vv_x + uv_y + \rho V'(\rho)(v_x + u_y) = \frac{V(\rho) - v}{\tau} \tag{3.32}$$

$$u_t + vu_x + uu_y + \rho U'(\rho)(v_x + u_y) = \frac{U(\rho) - u}{\tau} \tag{3.33}$$

where $V(\rho)$ and $U(\rho)$ are the desired velocities functions meant to mimic pedestrian behavior given by the velocity-density relation (2.15), and $\rho V'(\rho)$ is the traffic sound speed at which small traffic disturbances are propagated relative to the moving crowd stream. The relaxation terms $(V(\rho)-v)/\tau$ and $(U(\rho)-u)/\tau$ are added to the model as a modification to keep speed concentration in equilibrium, where $\tau$ is this process relaxation time. They are important to the system because $v_{f1}$ and $v_{f2}$ can be used as control parameters for crowd speed and

direction. Initial conditions are $\rho(x, y, 0) \geq 0$, $v(x, 0) \leq |v_{f1}|$, and $u(y, 0) \leq |v_{f2}|$.

### 3.6.2 Derivation of A Macroscopic Model from A Microscopic Model in 2-D

In This section we will show how to derive the second equation of the model (3.32), (3.33) from the microscopic *car-following* model that is used to represent traffic flow in one-dimension. For 2-D we use the same idea given in chapter 2 by subsection 2.5.4 and start the derivation from a microscopic model in 2-D given by

$$\tau(\mathbf{s}_n(t))\,\ddot{\mathbf{x}}_n(t) = \dot{\mathbf{x}}_{n+1}(t) - \dot{\mathbf{x}}_n(t) + A\left[V\left(\frac{\Delta X}{s_n(t)}\right) - \dot{\mathbf{x}}_n(t)\right], \tag{3.34}$$

where

$$\mathbf{s}_n(t) = \mathbf{x}_{n+1}(t) - \mathbf{x}_n(t), \tag{3.35}$$

is a function of the local spacing between pedestrians, $\mathbf{x}_n(t)$ is the two dimension position of the $n$th pedestrian, $\Delta X$ is the width of a single pedestrian, $\ddot{\mathbf{x}}_n(t)$ is the acceleration, $\dot{\mathbf{x}}_n(t)$ is the velocity, and $\tau(\mathbf{s}_n(t))$ is the pedestrian response (relaxation) time to the headway distance and it depends on spacing. This response time is different than pedestrian reaction time which is on average a small constant number. For the constant $A > 0$ a relaxation term is added and for the *homogeneous* case we set $A = 0$. Using the above notations, we rewrite (3.34) for the $x$-component where the local spacing $s$ is in the *x-direction* only. We start by defining the velocity field $v(x, y, t) : \dot{x}(t) = v(x_n(t), t)$, and pedestrian spacing function $s(x, t) : s_n(t) = s(x_n(t), t)$. We also define local density by

$$\rho(x, y, t) : \rho_n(t) = \frac{\Delta X}{s_n(t)}, \tag{3.36}$$

which is the number of people per unit length. In our definition the density is normalized and therefor dimensionless, so that jam density (maximum capacity) is $\rho_m = 1$. We substitute the new variables in (3.34) for the *x-component* and get

$$\tau(s(x(t), t))\,\frac{dv(x, y, t)}{dt} = \frac{d(s(x(t), t))}{dt} + A\left[V(\rho(x, t)) - v(x, y, t)\right]. \tag{3.37}$$

Using the convective derivative $\partial_t + v\partial_x + u\partial_y$ on the velocity component of the *x-axis* (for *y-axis* use $u$ instead of $v$), we obtain

$$\tau(s)\left(v_t + vv_x + uv_y\right) = \left(s_t + vs_x + us_y\right) + A\left[V(\rho(x,y,t)) - v(x,y,t)\right]. \tag{3.38}$$

From the conservation law (3.10), let $\rho = 1/s$, and by using the full derivative

$$D_x(\frac{a}{b}) = \frac{bD_xa - aD_xb}{b^2}, \tag{3.39}$$

we get

$$s_t + vs_x + us_y = sv_x + su_y, \tag{3.40}$$

and by direct substituting in the right hand side of equation (3.38), we arrive to our desired equation

$$\left(v_t + vv_x + uv_y\right) = \frac{s}{\tau(s)}(v_x + u_y) + \frac{A}{\tau(s)}\left[V(\rho(x,y,t)) - v(x,y,t)\right], \tag{3.41}$$

where

$$\frac{s}{\tau(s)} = -C(\rho) = -\rho V'(\rho) \geq 0, \tag{3.42}$$

is the sound wave speed, and $A$ is a constant equals one. Finally, the process relaxation term $\tau$ replaces $\tau(s)$ for macroscopic behavior to get equation (3.32). This completes the derivation of the 2-D macroscopic model from its microscopic counterpart for the *x-axis*. We follow similar steps for the *y-axis*. The macroscopic conservative form of this model for the *x-axis* is derived next.

### 3.6.3 Conservation Form and Eigenvalues

To find the eigenvalues of the system and check if the system is hyperbolic (real eigenvalues), we write the model in conservation vector form. The same form will be used in the numerical simulation to obtain the system response. We start deriving the conservation form by expanding the derivatives in (3.10) to get

$$\rho(v_x + u_y) = -(\rho_t + \rho_x v + \rho_y u), \tag{3.43}$$

and we know that for $V(\rho)$ given by (2.15)

$$V'(\rho)\rho_t = V_t(\rho), \quad V'(\rho)\rho_x = V_x(\rho), \quad V'(\rho)\rho_y = V_y(\rho). \tag{3.44}$$

Then substitute the above in equation (3.32) and multiply by $\rho$ to get

$$\rho(v - V(\rho))_t + \rho v\,(v - V(\rho))_x + \rho u\,(v - V(\rho))_y = \rho\frac{V(\rho) - v}{\tau}, \tag{3.45}$$

and by using the product rules

$$\begin{aligned}
(\rho\,(v - V(\rho)))_t &= \rho_t(v - V(\rho)) + \rho\,(v - V(\rho))_t, &(3.46)\\
(\rho v\,(v - V(\rho)))_x &= (\rho v)_x(v - V(\rho)) + (\rho v)(v - V(\rho))_x, &(3.47)\\
(\rho u\,(v - V(\rho)))_y &= (\rho u)_y(v - V(\rho)) + (\rho u)(v - V(\rho))_y, &(3.48)
\end{aligned}$$

we substitute in (3.45) and use the conservation law (3.10) to write the model in conservation form (we follow similar steps for the $y$ component using $U(\rho)$ instead of $V(\rho)$). Finally, after manipulating (3.41) we get our equation in conservative form as

$$(\rho(v - V(\rho)))_t + (\rho v\,(v - V(\rho)))_x + (\rho u\,(v - V(\rho)))_y = \rho\frac{V(\rho) - v}{\tau}, \tag{3.49}$$

and for the *y-direction* it is given by

$$(\rho\,(u - U(\rho)))_t + (\rho v\,(u - U(\rho)))_x + (\rho u\,(u - U(\rho)))_y = \rho\frac{U(\rho) - u}{\tau}. \tag{3.50}$$

Next we write the system in two dimensional vector form (3.17), where $Q$ is the conservative variables (states), $F$ and $G$ are the fluxes in the $x$ and *y-directions* respectively, and $S$ can be considered as the source term. These are given by

$$\begin{bmatrix} \rho \\ \rho(v - V(\rho)) \\ \rho(u - U(\rho)) \end{bmatrix}_t + \begin{bmatrix} \rho v \\ \rho v(v - V(\rho)) \\ \rho v(u - U(\rho)) \end{bmatrix}_x + \begin{bmatrix} \rho u \\ \rho u(v - V(\rho)) \\ \rho u(u - U(\rho)) \end{bmatrix}_y = S \tag{3.51}$$

and $S = [0 : s_1 : s_2]^T$. Next we rewrite the system in the general quasi-linear form given by (3.18), where the source term is zero and the flux Jacobian matrices are found from

$A(Q) = \partial F(Q)/\partial Q$, and $B(Q) = \partial G(Q)/\partial Q$. For this system the matrices and their corresponding eigenvalues and eigenvectors are found by first setting the conservative values (states) as

$$Q = \begin{bmatrix} \rho \\ w \\ z \end{bmatrix}, \; where \; \begin{cases} \rho(v - V(\rho)) = w \Rightarrow v = \dfrac{w}{\rho} + V(\rho) \\ \rho(u - U(\rho)) = z \Rightarrow u = \dfrac{w}{\rho} + U(\rho) \end{cases}$$

We rewrite the fluxes $F(Q)$ and $G(Q)$ as

$$F(Q) = \begin{bmatrix} w + \rho V(\rho) \\ \dfrac{w^2}{\rho} + wV(\rho) \\ \dfrac{wz}{\rho} + zV(\rho) \end{bmatrix}, \quad G(Q) = \begin{bmatrix} z + \rho U(\rho) \\ \dfrac{wz}{\rho} + wU(\rho) \\ \dfrac{z^2}{\rho} + zU(\rho) \end{bmatrix}.$$

Their corresponding Jacobian matrices are found to be

$$A(Q) = \begin{bmatrix} V(\rho) + \rho V'(\rho) & 1 & 0 \\ -\dfrac{w^2}{\rho^2} + wV'(\rho) & 2\dfrac{w}{\rho} + V(\rho) & 0 \\ -\dfrac{wz}{\rho^2} + zV'(\rho) & \dfrac{z}{\rho} & \dfrac{w}{\rho} + V(\rho) \end{bmatrix}, \tag{3.52}$$

$$B(Q) = \begin{bmatrix} U(\rho) + \rho U'(\rho) & 0 & 1 \\ -\dfrac{wz}{\rho^2} + wU'(\rho) & \dfrac{z}{\rho} + U(\rho) & \dfrac{w}{\rho} \\ -\dfrac{z^2}{\rho^2} + zU'(\rho) & 0 & 2\dfrac{z}{\rho} + U(\rho) \end{bmatrix}. \tag{3.53}$$

Solving for the eigenvalues and eigenvectors of $A(Q)$ we get

$$\lambda_1^A = v + \rho V'(\rho), \;\; \& \;\; \lambda_{2,3}^A = v, \tag{3.54}$$

$$e_1^A = \begin{bmatrix} 1 \\ v - V \\ u - U \end{bmatrix}, \; e_2^A = \begin{bmatrix} 1 \\ v - V - \rho V' \\ 0 \end{bmatrix}, e_3^A = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

and for the $B(Q)$ matrix we obtain

$$\lambda_1^B = u + \rho U'(\rho), \;\; \& \;\; \lambda_{2,3}^B = u \tag{3.55}$$

$$e_1^B = \begin{bmatrix} 1 \\ v - V(\rho) \\ u - U(\rho) \end{bmatrix}, \; e_2^B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \; e_3^B = \begin{bmatrix} 1 \\ 0 \\ u - U(\rho) - \rho U'(\rho) \end{bmatrix}$$

Since the system is a two-dimensional problem, and in order to verify that the system is hyperbolic, we need to check that our eigenvalues found earlier are valid for any combination of the roots of the combined system as defined by 3.2.3. The eigenvalues are found to be

$$\lambda_1 = \check{v} + \rho \check{V}', \quad \lambda_{2,3} = \check{v} \tag{3.56}$$

where $\check{v} = \alpha v(x, y, t) + \xi u(x, y, t)$, and $\check{V}' = \alpha V' + \xi U'$. Since the eigenvalues above are real, we conclude that our model is hyperbolic, and the fact that we also have repeated eigenvalues our model is not strictly hyperbolic. The first eigenvalue is always less or equal than the second one due to $\rho V'$ effect ($V' \leq 0$). From this fact we acknowledge the anisotropic property of the system that shows information can not travel faster than the actual wave, i.e., pedestrian movement is influenced from current and front stimuli only. Secondly, despite repeated eigenvalues, each matrix has linearly independent eigenvectors.

For each eigenvalue the corresponding eigenvectors are given by

$$e_1^{\check{A}} = \begin{bmatrix} 1 \\ v - V(\rho) \\ u - U(\rho) \end{bmatrix}, \; e_2^{\check{A}} = \begin{bmatrix} 1 \\ \dfrac{\check{v} - \check{V}(\rho) - \rho \check{V}(\rho)}{\alpha} \\ 0 \end{bmatrix} or \begin{bmatrix} 0 \\ 1 \\ -\dfrac{\alpha}{\xi} \end{bmatrix}$$

$$e_3^{\check{A}} = \begin{bmatrix} 0 \\ -\dfrac{\xi}{\alpha} \\ 1 \end{bmatrix} or \begin{bmatrix} 1 \\ 0 \\ \dfrac{\check{v} - \check{V}(\rho) - \rho \check{V}(\rho)}{\xi} \end{bmatrix}.$$

For $e_2^{\check{A}}$ the"or" is to get back both $e_2^A$ and $e_2^B$, and the same can be said about $e_3^{\check{A}}$.

## 3.7 Comparison Between the Models

Four models that are aimed at studying crowd dynamics have been presented in this chapter. They are derived from the one-dimensional traffic flow theory, with the proper adjustment to account for the bi-directional pedestrian flow. Table 3.1 summarizes the main characteristics of the four models.

We divide the models into two types; crowd model that closely follow gas and fluid-like traffic flow, and models that do not. We further divided the models into two types; scalar and systems of PDE's as shown in Figure 3.2. The models are classified as nonlinear, hyperbolic, time-varying partial differential equation(s) with distinct characteristics separating them; for example, the first crowd model use Greenshield's equation (2.15) to describe the velocity as a function of density. This one-equation model is an extension to the scalar LWR traffic 1-D model. Although it is simple compared to the first model, it gives the desired anisotropic nature of traffic flow. In addition, the model is strictly hyperbolic. The second model is a

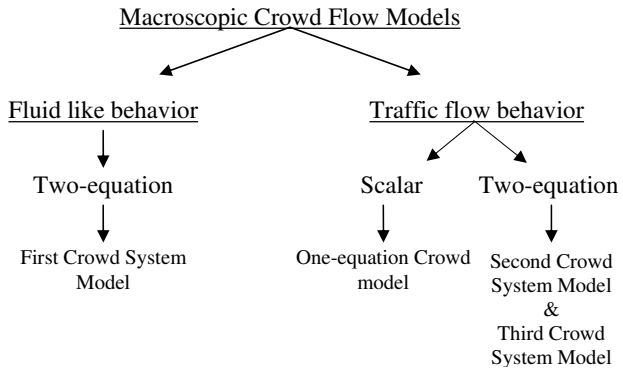


Figure 3.2: Crowd model types

system model that use two-coupled PDE's with an anticipation term $C(\rho)$ and the relaxation terms $\check{s} = \frac{\check{V}(\rho)-\check{v}}{\tau}$. These terms change the momentum equation to mimic crowd dynamic flow. The anticipation factor's role is to find the macroscopic response of pedestrian to traffic density, i.e., its response when interaction between pedestrians and obstacles are overlooked within their domain. On the other hand, the relaxation factor's role is to keep the velocity in

equilibrium. If a pedestrian velocity is greater than the preferred value, then this relaxation term will slow the pedestrian traffic down. From the system eigenvalues (real and distinct), we can conclude that this model has an isotropic nature since one of its eigenvalues is always moving faster than the velocity itself. This means that information can travel faster than the flow, which may contradicts the observed behavior for pedestrians in normal situations, where pedestrians movement are mostly influenced from current and front conditions and not from behind. On the other hand, in panic situations like in the case of an emergency situations such as evacuation, and at exits we know that when crowds/or pedestrians are moving close to each other there will be various influences from those surrounding them to some extent, including the ones from behind (Henderson, 1974). In the third model, we do have a relaxation term, but instead of the anticipation term that depends on space only in the first system model, a convective derivative on the "pressure" term is used. This change will enable the second PDE equation to predict the expected response of crowd behavior as time and space changes. The third system model has three real and two repeated Eigenvalues 3.1.; nevertheless, we found linearly independent eigenvectors that enabled us to solve the problem using numerical finite volume methods. The system has an anisotropic nature that is evident from its eigenvalues, where $\lambda_1 \leq \lambda_{2,3}$ for both velocity directions. This means that all information moves at a speed equal or less than the velocity of the corresponding state. This feature is preferred for traffic flow in general where pedestrian movement response is mostly due to conditions ahead like in normal conditions. In addition, we can relate a panic crowd situation to this from the fact that in most panic situations with high-density crowds, pedestrians tend to slowly form groups and move together without looking back. They just follow, assuming that the leader of this group knows or sees the way out.

The fourth model can be derived from a microscopic model, i.e., has a direct micro-to-macro link. In the model second PDE, the use of the convective derivative for the velocity terms leads to the sound wave speed $-C(\rho) = -\rho V'(\rho)$, which enables the model to predict the expected response of crowd behavior. That is they are used as anticipation terms to find the macroscopic response of pedestrians to traffic density. The terms $V(\rho(x, y, t))$, and

Table 3.1: Comparison Between the Four Models

| Characteristics \ Models | | One-Equation Model | First System | Second System | Third Model |
|---|---|---|---|---|---|
| Conservative Variables $Q$ | | $\rho$ | $\rho$ , $\rho v$ , $\rho u$ | $\rho$ , $\rho(v+P_1)$ , $\rho(u+P_2)$ | $\rho$ , $\rho(v-\rho V')$, $\rho(u-\rho U')$ |
| "Pressure" term $P$ | | No | $\rho\, C(\rho)=\rho\, C_0^2$ | $\frac{\vec{v}\rho^{\gamma+1}}{\beta-\rho^{\gamma+1}}, \gamma>0, \beta>\rho_m^{\gamma+1}$ | $-\rho V'(\rho)=\rho\frac{v_f}{\rho_m}$ |
| $X$ - Direction $F(Q)$ Eigenvalues | $\lambda_1$ | $f'(\rho)=v_{f1}(1-2\rho/\rho_m)$ | $v-C_0$ | $v-(\gamma+1)P_1(\rho)$ | $v+\rho V'$ |
| | $\lambda_2$ | No | $v$ | $v$ | $v$ |
| | $\lambda_3$ | No | $v+C_0$ | $v$ | $v$ |
| $Y$ - Direction $G(Q)$ Eigenvalues | $\lambda_1$ | $g'(\rho)=v_{f2}(1-2\rho/\rho_m)$ | $u-C_0$ | $u-(\gamma+1)P_2(\rho)$ | $u+\rho V'$ |
| | $\lambda_2$ | No | $u$ | $u$ | $u$ |
| | $\lambda_3$ | No | $u+C_0$ | $u$ | $u$ |
| PDE Type | | Strictly Hyperbolic | Strictly Hyperbolic | Hyperbolic | Hyperbolic |
| Isotropic | | No | Yes | No | No |
| Anisotropic | | Yes | No | Yes | Yes |

$U(\rho(x,y,t))$ change the momentum equation to mimic crowd dynamic flow. At the same time, these terms help to keep velocity in equilibrium. If a pedestrian velocity is greater than the preferred value, these terms will tend to reduce the velocity and vice versa. The source terms in this model are essential in providing controlled behavior. They are considered as relaxation terms, where the change in direction would not be possible if they are not present.

The model has six real eigenvalues; three for each direction and two of them are repeated pair. Although we have repeated eigenvalues, the corresponding eigenvectors are linearly independent. This enabled us to perform simulations using numerical finite volume methods. The system has an anisotropic nature that appears from its eigenvalues $\lambda_1 \leq \lambda_{2,3}$.

## 3.8 Linearization

In this section we will linearize the PDE's modeling the crowd flow dynamics. We use Taylor series expansion to find the linearized models around some solution that we assume it exist and a perturbation, and we ignore higher order terms. Taylor series is given by

$$f(x+\Delta x) = f(x) + \Delta x f'(x) + \frac{(\Delta x)^2}{2!} f''(x) + ... \tag{3.57}$$

and for two variables Taylor series it becomes

$$f(x+\Delta x, y+\Delta y) = f(x,y) + [f_x(x,y)\Delta x + f_y(x,y)\Delta y] + H.O.T. \tag{3.58}$$

### 3.8.1 One Equation Crowd Model

For the one equation model given by

$$\rho_t + (\rho V(\rho))_x + (\rho U(\rho))_y = 0 \tag{3.59}$$

where $V(\rho)$ and $U(\rho)$ are the velocity-density functions given in (2.15). To linearize this model, we first assume that we have a density solution $\rho_0$ and vary it with small perturbation

$\bar{\rho}$ to get

$$\rho = \rho_0 + \bar{\rho}.$$

We first need to find the velocity-density expansion to make our later substitutions easier. The expansion gives

$$\begin{aligned} V\left(\rho_0+\bar{\rho}\right) &= v_{f1}\left(1-\frac{\rho_0+\bar{\rho}}{\rho_m}\right) = v_{f1}\left(1-\frac{\rho_0}{\rho_m}\right) - \frac{v_{f1}}{\rho_m}\bar{\rho} \\ &= V\left(\rho_0\right) + V'\bar{\rho}, \end{aligned} \tag{3.60}$$

$$U\left(\rho_0+\bar{\rho}\right) = U\left(\rho_0\right) + U'\bar{\rho}, \tag{3.61}$$

where

$$V'(\rho) = \frac{\partial V(\rho)}{\partial \rho} = -\frac{v_{f1}}{\rho_m} = V', \tag{3.62}$$

$$U'(\rho) = \frac{\partial U(\rho)}{\partial \rho} = -\frac{v_{f2}}{\rho_m} = U'. \tag{3.63}$$

We consider $V(\rho_0)$, and $U(\rho_0)$ to be the velocity solutions, and in the subsequence sections when we have velocity dynamics it is equal to $v_0$, and $u_0$.

By substitution the above in our model and using Taylor series we get

$$\begin{aligned} (\rho_0+\bar{\rho})_t &+ \left((\rho_0+\bar{\rho})(V(\rho_0)+V'\bar{\rho})\right)_x \\ &+ \left((\rho_0+\bar{\rho})(U(\rho_0)+U'\bar{\rho})\right)_y = 0. \end{aligned} \tag{3.64}$$

Since $\rho_0$ is constant, all the derivatives associated with it are equal to zero. In addition, any high order terms are ignored (e.g. $\bar{\rho}\bar{\rho} = 0$) and the following linearized model is obtained

$$\bar{\rho}_t + V\left(\rho_0\right)\bar{\rho}_x + \rho_0 V'\bar{\rho}_x + U\left(\rho_0\right)\bar{\rho}_y + \rho_0 U'\bar{\rho}_y = 0. \tag{3.65}$$

If we collect terms and rearrange the above equation as shown

$$\bar{\rho}_t + \underbrace{\left(V\left(\rho_0\right)+\rho_0 V'\right)}_{f'(q)_{q=q_0}=v_{f1}\left(1-\frac{2\rho_0}{\rho_m}\right)}\bar{\rho}_x + \underbrace{\left(U\left(\rho_0\right)+\rho_0 U'\right)}_{g'(q)_{q=q_0}=v_{f2}\left(1-\frac{2\rho_0}{\rho_m}\right)}\bar{\rho}_y = 0, \tag{3.66}$$

where $q = \rho V(\rho)$ is the crowd flow, $f'(q) = \frac{\partial f(q)}{\partial \rho}$, and $g'(q) = \frac{\partial g(q)}{\partial \rho}$ are the characteristics speeds (also known as the eigenvalues). Thus, we end up with the same quasi-linear form that was introduced in (3.7), but here the eigenvalues are constant and the final form of the linearized model is given by

$$\bar{\rho}_t + f'(q)_{q=q_0}\bar{\rho}_x + g'(q)_{q=q_0}\bar{\rho}_y = 0. \tag{3.67}$$

### 3.8.2 First System Model

In this section we linearize the first system crowd dynamic model given by equations (3.10), (3.11), and (3.12). As we have done in the previous subsection, we use Taylor series expansion around an assumed solution trajectory. The difference here is because we linearize around the solutions of the density, and velocity in both directions as given by

$$\rho = \rho_0 + \bar{\rho}, \quad v = v_0 + \bar{v}, \quad u = u_0 + \bar{u},$$

where $v_0 = V(\rho_0)$, and $u_0 = U(\rho_0)$ are the assumed velocity solutions. We substitute the above into our model to get

$$(\rho_0 + \bar{\rho})_t + ((\rho_0 + \bar{\rho})(v_0 + \bar{v}))_x + ((\rho_0 + \bar{\rho})(u_0 + \bar{u}))_y = 0 \tag{3.68}$$

$$(v_0 + \bar{v})_t + (v_0 + \bar{v})(v_0 + \bar{v})_x + (u_0 + \bar{u})(v_0 + \bar{v})_y + \frac{C_0^2}{(\rho_0 + \bar{\rho})}(\rho_0 + \bar{\rho})_x = s_1 \tag{3.69}$$

$$(u_0 + \bar{u})_t + (u_0 + \bar{u})(u_0 + \bar{u})_x + (v_0 + \bar{v})(u_0 + \bar{u})_y + \frac{C_0^2}{(\rho_0 + \bar{\rho})}(\rho_0 + \bar{\rho})_y = s_2 \tag{3.70}$$

where

$$s_1 = \frac{(V(\rho_0 + \bar{\rho}) - (v_0 + \bar{v}))}{\tau}, \qquad s_2 = \frac{(U(\rho_0 + \bar{\rho}) - (u_0 + \bar{u}))}{\tau}.$$

After terms cancelation we get

$$\begin{aligned}
\bar{\rho}_t + v_0\bar{\rho}_x + u_0\bar{\rho}_y + \rho_0\left(\bar{v}_x + \bar{u}_y\right) &= 0, && (3.71)\\
\bar{v}_t + v_0\bar{v}_x + u_0\bar{v}_y + \frac{C_0^2}{\rho_0}\bar{\rho}_x &= \frac{V'\bar{\rho} - \bar{v}}{\tau}, && (3.72)\\
\bar{u}_t + v_0\bar{u}_x + u_0\bar{u}_y + \frac{C_0^2}{\rho_0}\bar{\rho}_y &= \frac{U'\bar{\rho} - \bar{u}}{\tau}, && (3.73)
\end{aligned}$$

which is the final form of the linearized model, and in vector form it is

$$Q_t + AQ_x + BQ_y = S(Q). \tag{3.74}$$

The matrices $A$ and $B$ are constant, and the vector $S(Q)$ contains the relaxation terms which are functions of the states. The vector form of the model is given by

$$\begin{bmatrix} \bar{\rho} \\ \bar{v} \\ \bar{u} \end{bmatrix}_t + \begin{bmatrix} v_0 & \rho_0 & 0 \\ C_0^2/\rho_0 & v_0 & 0 \\ 0 & 0 & v_0 \end{bmatrix} \begin{bmatrix} \bar{\rho} \\ \bar{v} \\ \bar{u} \end{bmatrix}_x + \begin{bmatrix} u_0 & 0 & \rho_0 \\ 0 & u_0 & 0 \\ C_0^2/\rho_0 & 0 & u_0 \end{bmatrix} \begin{bmatrix} \bar{\rho} \\ \bar{v} \\ \bar{u} \end{bmatrix}_y = S(Q), \tag{3.75}$$

where

$$S(Q) = \begin{bmatrix} 0 \\ (V'\bar{\rho} - \bar{v})/\tau \\ (U'\bar{\rho} - \bar{u})/\tau \end{bmatrix}$$

The eigenvalues of the system above are the same ones found in section 3.4.2 for the nonlinear model, and they are given by

$$\lambda_1^A = v_0, \quad \lambda_{2,3}^A = v_0 \pm C_0, \qquad \textit{and} \qquad \lambda_1^B = u_0, \quad \lambda_{2,3}^B = u_0 \pm C_0$$

We need to rewrite $S(Q)$ to separate the control variables for control design by the following manner

$$S(Q) = \underbrace{N(Q)}_{EQ} + M(Q)\eta$$

where $\eta$ denote the controlled variables and $E$ is a constant matrix. The above equation is given by

$$S(Q) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1/\tau & 0 \\ 0 & 0 & -1/\tau \end{bmatrix} \begin{bmatrix} \bar{\rho} \\ \bar{v} \\ \bar{u} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\bar{\rho}/(\rho_m \tau) & 0 \\ 0 & 0 & -\bar{\rho}/(\rho_m \tau) \end{bmatrix} \begin{bmatrix} 0 \\ v_{f1} \\ v_{f2} \end{bmatrix}. \tag{3.76}$$

### 3.8.3 Second System Model

Here we linearize the second system model given by equations (3.10), (3.21), and (3.22). To start, we first linearize the "pressure" terms using Taylor series expansion given by (3.58) on the *x-axis* velocity component (use $u$ for *y-axis* given by

$$P(\rho, v) = \left( \frac{v\rho^{\gamma+1}}{\beta - \rho^{\gamma+1}} \right),$$

and we get

$$\begin{aligned} P(\rho_0 + \bar{\rho}, v_0 + \bar{v}) &= P(\rho_0, v_0) + [P_{\rho_0}(\rho_0, v_0)\bar{\rho} + P_{v_0}(\rho_0, v_0)\bar{v}] \\ &= \left( \frac{v_0 \rho_0^{\gamma+1}}{\beta - \rho_0^{\gamma+1}} \right) + \left[ \frac{\beta(\gamma+1) v_0 \rho_0^{\gamma+1}}{\left(\beta - \rho_0^{\gamma+1}\right)^2} \bar{\rho} + \frac{\rho_0^{\gamma+1}}{\beta - \rho_0^{\gamma+1}} \bar{v} \right] \\ &= P(\rho_0, v_0) + \underbrace{P_\rho(\rho, v)_{\rho_0, v_0}}_{\text{call it } P_\rho(\rho_0, v_0)} \bar{\rho} + \underbrace{P_v(\rho, v)_{\rho_0, v_0}}_{\text{call it } P_v(\rho_0, v_0)} \bar{v}. \end{aligned}$$

The linearized conservation of mass is given by equation (3.71). The linearized form for equations (3.21) and (3.22) is given by

$$\begin{aligned} \bar{v}_t &+ (P_\rho(\rho_0, v_0)\bar{\rho} + P_v(\rho_0, v_0)\bar{v})_t + v_0 \bar{v}_x + v_0 (P_\rho(\rho_0, v_0)\bar{\rho} + P_v(\rho_0, v_0)\bar{v})_x + u_0 \bar{v}_y \\ &+ u_0 (P_\rho(\rho_0, v_0)\bar{\rho} + P_v(\rho_0, v_0)\bar{v})_y = \frac{V'\bar{\rho} - \bar{v}}{\tau} \end{aligned} \tag{3.77}$$

$$\begin{aligned} \bar{u}_t &+ (P_\rho(\rho_0, u_0)\bar{\rho} + P_u(\rho_0, u_0)\bar{u})_t + v_0 \bar{u}_x + v_0 (P_\rho(\rho_0, u_0)\bar{\rho} + P_u(\rho_0, u_0)\bar{u})_x + u_0 \bar{u}_y \\ &+ u_0 (P_\rho(\rho_0, u_0)\bar{\rho} + P_u(\rho_0, u_0)\bar{u})_y = \frac{U'\bar{\rho} - \bar{u}}{\tau} \end{aligned} \tag{3.78}$$

In vector form $DQ_t + AQ_x + BQ_y = S$, we obtain

$$
\begin{aligned}
Q &= \begin{bmatrix} \bar{\rho} \\ \bar{v} \\ \bar{u} \end{bmatrix} \\
D &= \begin{bmatrix} 1 & 0 & 0 \\ P_\rho(\rho_0, v_0) & 1 + P_v(\rho_0, v_0) & 0 \\ P_\rho(\rho_0, u_0) & 0 & 1 + P_u(\rho_0, v_0) \end{bmatrix} \\
A &= \begin{bmatrix} v_0 & \rho_0 & 0 \\ v_0 P_\rho(\rho_0, v_0) & v_0\left(1 + P_v(\rho_0, v_0)\right) & 0 \\ v_0 P_\rho(\rho_0, u_0) & 0 & v_0\left(1 + P_u(\rho_0, v_0)\right) \end{bmatrix} \\
B &= \begin{bmatrix} u_0 & 0 & \rho_0 \\ u_0 P_\rho(\rho_0, v_0) & u_0\left(1 + P_u(\rho_0, u_0)\right) & 0 \\ u_0 P_\rho(\rho_0, v_0) & 0 & u_0\left(1 + P_u(\rho_0, u_0)\right) \end{bmatrix} \\
S &= \begin{bmatrix} 0 \\ (V'\bar{\rho} - \bar{v})/\tau \\ (U'\bar{\rho} - \bar{u})/\tau \end{bmatrix}.
\end{aligned}
$$

To put the system in the general vector form (3.74), we multiply the equation by the inverse of the nonsingular matrix $D$ to get

$$Q_t + \left(D^{-1}A\right) Q_x + \left(D^{-1}B\right) Q_y = \left(D^{-1}S\right) \tag{3.79}$$

The eigenvalues found from (3.79) are the same as the ones found in section 3.5.2, but here they are time-invariant given by

$$\lambda^A_{1,2} = v_0, \quad \lambda^A_{2,3} = v_0\left(1 - \frac{(\gamma+1)\,\rho^{\gamma+1}}{\beta - \rho^{\gamma+1}}\right)$$

and

$$\lambda^B_{1,2} = u_0, \quad \lambda^B_{2,3} = u_0\left(1 - \frac{(\gamma+1)\,\rho^{\gamma+1}}{\beta - \rho^{\gamma+1}}\right)$$

### 3.8.4 Third System Model

For the third nonlinear system model given by equations (3.10), (3.32), and (3.33) we linearize the dynamics around some solution with a small perturbation. The result is given by equation (3.71), and the following two equations

$$\bar{v}_t + v_0\bar{v}_x + u_0\bar{v}_y + \rho_0(V')^2\bar{\rho}_x + \rho_0(V')^2\bar{\rho}_y = \frac{V'\bar{\rho} - \bar{v}}{\tau} \tag{3.80}$$

$$\bar{u}_t + v_0\bar{u}_x + u_0\bar{u}_y + \rho_0(U')^2\bar{\rho}_x + \rho_0(U')^2\bar{\rho}_y = \frac{U'\bar{\rho} - \bar{u}}{\tau}. \tag{3.81}$$

In vector form (3.74), the linearized model is given by

$$\begin{bmatrix} \bar{\rho} \\ \bar{v} \\ \bar{u} \end{bmatrix}_t + \begin{bmatrix} v_0 & \rho_0 & 0 \\ \rho_0 (V')^2 & v_0 & 0 \\ \rho_0 (U')^2 & 0 & v_0 \end{bmatrix} \begin{bmatrix} \bar{\rho} \\ \bar{v} \\ \bar{u} \end{bmatrix}_x + \begin{bmatrix} u_0 & 0 & \rho_0 \\ \rho_0 (V')^2 & u_0 & 0 \\ \rho_0 (U')^2 & 0 & u_0 \end{bmatrix} \begin{bmatrix} \bar{\rho} \\ \bar{v} \\ \bar{u} \end{bmatrix}_y = \begin{bmatrix} 0 \\ (V'\bar{\rho} - \bar{v})/\tau \\ (U'\bar{\rho} - \bar{u})/\tau \end{bmatrix}.$$

The system eigenvalues are fount to be

$$\lambda_1 = v_0, \lambda_{2,3} = v_0 \pm \rho_0 \mathrm{V}', \quad \textit{and} \quad \lambda_1 = u_0, \lambda_{2,3} = u_0 \pm \rho_0 U',$$

which are the the time-invariant version of the ones found in subsection 3.6.3. This concludes the linearization section

# Chapter 4

# Numerical Methods

## 4.1 Introduction

This chapter presents numerical schemes that approximate the solution of the hyperbolic, non-linear, time-varying, partial differential equations that represent crowd models developed in chapter 3. Due to the hyperbolic nature of PDEs, discontinuous solutions like shocks and rarefaction waves can occur. Therefore, we use *shock capturing finite difference schemes* called Finite Volume Methods (FVM) (Kröner, 1997; Leveque, 2002; Toro, 1999). These kind of schemes are capable of automatically choosing the correct weak solution, including shocks. This can be achieved because these methods are based on the integral form of the conservation laws which allow discontinuous solution, and not on the differential form where discontinuous solutions are not defined.

For the purpose of finding the numerical solution to the crowd models we use three first order accurate methods. The methods used are mainly, the Lax-Friedrichs scheme, First-Order Centered scheme (FORCE), and Roe's scheme. Higher order methods can be applied to obtain more accurate results but they are not considered here. The numerical solution can also be obtained by composite schemes, using first order Lax-Friedrichs method to smooth

shocks in conjunction with a second order Lax-Wendroff method to increase accuracy (Liska and Wendroff, 1997), by using second or higher order schemes with flux limiters to smooth the shock waves instability (Roe's scheme can be modified to be a second order accurate). One family of schemes can not be applied on the models of chapter 3, they are the flux vector splitting methods. This is due to the homogeneity property $F(Q) = A(Q)Q$ which is not satisfied in our crowd models. The organization of this chapter is as follows, section 4.2 presents the theory behind the FVM, followed by the description of three schemes.

## 4.2 Fundamentals of FVM

A finite volume method is based on dividing the space into grid cells or finite volumes and approximating the integral of the state $Q$ over each volume. In each time step we update these values using approximations to the flux through the endpoints $[x_{i-1/2}, x_{i+1/2}]$ of the intervals as given by

$$Q_i^n \approx \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} q(x, t_n)\, dx \equiv \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \tag{4.1}$$

where $\Delta x = x_{i+1/2} - x_{i-1/2}$ is the length of each cell on a uniform grid. By using a conservative numerical method we will guarantee that $\sum_{i=1}^{N} Q_i^n \Delta x$ over the entire space will change only due to fluxes at the boundaries. We will show this by recalling the integral conservation form (2.9) and rewrite it as

$$\frac{d}{dt} \int_{x_{i-1/2}}^{x_{i+1/2}} q(x, t)\, dx = f_{in}(q(x_{i-1/2}, t)) - f_{out}(q(x_{i+1/2}, t)). \tag{4.2}$$

For a given $Q_i^n$ we would like to approximate $Q_i^{n+1}$, that is the volume average after $\Delta t = t_{n+1} - t_n$. To do so, we integrate (4.2) over time to obtain

$$\int_{x_{i-1/2}}^{x_{i+1/2}} q(x, t_{n+1})\, dx - \int_{x_{i-1/2}}^{x_{i+1/2}} q(x, t_n)\, dx = \int_{t_n}^{t_{n+1}} f_{in}(q(x_{i-1/2}, t)) - \int_{t_n}^{t_{n+1}} f_{out}(q(x_{i+1/2}, t)).$$

Dividing the above by $\Delta x$ and rearranging we get

$$\begin{aligned}\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} q(x,t_{n+1})\,dx &= \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} q(x,t_n)\,dx \\ &- \frac{1}{\Delta x}\left[\int_{t_n}^{t_{n+1}} f_{out}(q(x_{i+1/2},t))\,dt - \int_{t_n}^{t_{n+1}} f_{in}(q(x_{i-1/2},t))\,dt\right],\end{aligned}$$

and by comparing with (4.2), this translate to numerical method of the following form

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}[F_{i+1/2}^n - F_{i-1/2}^n], \tag{4.3}$$

where

$$F_{i+1/2}^n \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f(q(x_{i+1/2},t))\,dt.$$

The difference in finding the fluxes $F$ at the boundary points of each cell is how these schemes vary (examples are in next section). Lets go back to the point of conservation, if we take $\sum_{i=I}^{J} Q_i^{n+1}\Delta x$ we will get

$$\sum_{i=I}^{J} Q_i^{n+1}\Delta x = \sum_{i=I}^{J} Q_i^n \Delta x - \frac{\Delta t}{\Delta x}[F_{J+1/2}^n - F_{I-1/2}^n]. \tag{4.4}$$

that shows very clearly the conservative nature of this method. Here the changes to the hyperbolic PDE if any are due to the flow at the extreme boundary points only.

### 4.2.1 Formulation of Two-dimensional Numerical Schemes

We are trying to solve the following problem

$$\left.\begin{array}{lcl} PDEs & : & Q_t + F(Q)_x + G(Q)_y = 0 \\ ICs & : & Q(x,y,0) = Q^0 \\ B.Cs & : & non-slip\ walls,\ and/or\ one\ exit \end{array}\right\} \tag{4.5}$$

We start by identifying the notations involved in this process. The solution space $(x, y, t)$ is split up into a uniform computational grid, where the grid spaces in the $x$ and $y$ directions are given by $\Delta x$, and $\Delta y$ respectively as shown in Figure 4.1, and in time direction by $\Delta t$.

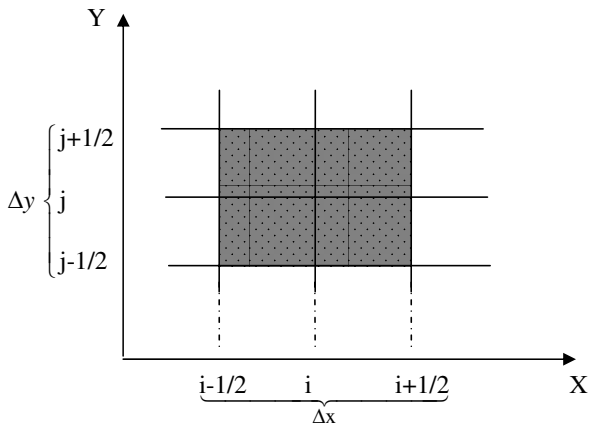


Figure 4.1: Two-dimensional Grid

The positions of the $i$th and $j$th nodes in the $x$ and $y$ directions, and $n$th node in time direction $(x_i, y_j, t^n)$ are given by $(i\Delta x, j\Delta y, n\Delta t)$.

There are various ways of approximating the spatial and time derivatives. Here we will concentrate on schemes that approximate the time derivative $Q_t$ by a one-sided approximation given by

$$Q_t \approx \frac{Q^{n+1} - Q^n}{\Delta t}.$$

By doing so, the schemes for two-dimensional space can be written in a general form as

$$Q_{i,j}^{n+1} = Q_{i,j}^n - \frac{\Delta t}{\Delta x}[F_{i+1/2,j}^n - F_{i-1/2,j}^n] - \frac{\Delta t}{\Delta x}[G_{i,j+1/2}^n - G_{i,j-1/2}^n]. \tag{4.6}$$

We use two approaches to solve the two-dimensional space problem. One is *fully discrete flux difference method* like Lax-Friedrichs scheme and obtain a solution according to (4.6), and another is *dimensional splitting* scheme like FORCE and Roe's. In the later method, we split the two-dimensional problem into a sequence of two one-dimension problems as follows

$$\left.\begin{array}{lcl} PDEs & : & Q_t + F(Q)_x = 0 \\ ICs & : & Q(x,y,0) = Q^0 \\ B.Cs & : & non-slip\ walls,\ and\ one\ exit \end{array}\right\} \xrightarrow{\Delta t} Q^{n+\frac{1}{2}} \tag{4.7}$$

$$\left.\begin{array}{lcl} PDEs & : & Q_t + G(Q)_y = 0 \\ ICs & : & Q^{n+\frac{1}{2}} \\ B.Cs & : & non-slip\ walls,\ and\ one\ exit \end{array}\right\} \xrightarrow{\Delta t} Q^{n+1} \tag{4.8}$$

and use

$$Q_{i,j}^{n+\frac{1}{2}} = Q_{i,j}^{n} - \frac{\Delta t}{\Delta x}[F_{i+1/2,j}^{n} - F_{i-1/2,j}^{n}] \tag{4.9}$$

$$Q_{i,j}^{n+1} = Q_{i,j}^{n+\frac{1}{2}} - \frac{\Delta t}{\Delta x}[G_{i,j+1/2}^{n+\frac{1}{2}} - G_{i,j-1/2}^{n+\frac{1}{2}}]. \tag{4.10}$$

## 4.3 Numerical Schemes

### 4.3.1 Lax-Friedrichs Scheme

Lax-Friedrichs is a classical scheme with first order accuracy that solves (4.5) using (4.6) with numerical fluxes calculated at the interface. For one-dimension and two-dimensions the scheme is given by

$$Q_i^{n+1} = \frac{1}{2}[Q_{i-1}^{n} + Q_{i+1}^{n}] - \frac{\Delta t}{2\Delta x}[F(Q_{i+1}^{n}) - F(Q_{i-1}^{n})] \tag{4.11}$$

$$\begin{aligned} Q_{i,j}^{n+1} = {} & \frac{1}{4}[Q_{i-1,j^n}^{n} + Q_{i+1,j}^{n} + Q_{i,j-1}^{n} + Q_{i,j+1}^{n}] - \frac{\Delta t}{2\Delta x}[F(Q_{i+1,j}^{n}) - F(Q_{i-1,j}^{n})] \\ & - \frac{\Delta t}{2\Delta x}[G(Q_{i,j+1}^{n}) - G(Q_{i,j-1}^{n})]. \end{aligned} \tag{4.12}$$

The scheme was selected because it is less expensive (takes less time computationally), and gives a good idea of the expected solution behavior. This method has numerical diffusion that damps the instabilities arising when shock solution is approximated. The method in one-dimension is stable for $\nu = \frac{\Delta t}{\Delta x} \max_p |\lambda^p| \leq 1$ with $\lambda$ as one of the system eigenvalues. For two-dimension, we use *von Neumann Stability Analysis* (Leveque, 2002) to get $\nu$.

### 4.3.2 FORCE Scheme

The First Order Centered (FORCE) method is a first order accurate *total variation diminishing* (TVD) scheme that comes from the mean fluxes of the Lax-Friedrichs (4.11) and the Richtmyer second order accurate scheme (Toro, 1999). The FORCE scheme solves numerically (4.5) by (**??**), and (4.8). The scheme computes a numerical flux by first defining an intermediate state $Q_{i-1}^{n+\frac{1}{2}}$, then the flux is computed based on this new intermediate state. Mathematically for one-dimension problem this translates to

$$Q_{i+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}[Q_i^n + Q_{i+1}^n] - \frac{\Delta t}{2\Delta x}[F_{i+1}^n - F_i^n] \tag{4.13}$$

$$Q_{i-\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}[Q_{i-1}^n + Q_i^n] - \frac{\Delta t}{2\Delta x}[F_i^n - F_{i-1}^n]. \tag{4.14}$$

then,

$$Q_i^{n+1} = \frac{1}{2}[Q_{i+\frac{1}{2}}^{n+\frac{1}{2}} + Q_{i-\frac{1}{2}}^{n+\frac{1}{2}}] - \frac{\Delta t}{2\Delta x}[F_{i+\frac{1}{2}}^{n+\frac{1}{2}} - F_{i-\frac{1}{2}}^{n+\frac{1}{2}}] \tag{4.15}$$

where

$$F_{i+\frac{1}{2}}^{n+\frac{1}{2}} = F(Q_{i+\frac{1}{2}}^{n+\frac{1}{2}}). \tag{4.16}$$

For the two-dimensional problem we use $i, j$ instead of $i$ as in (4.9) and (4.9) and use *dimensional splitting* method.

### 4.3.3 Roe's Scheme

The idea behind Roe's scheme is to take a non-linear PDE system in quasi-linear form

$$Q_t + A(Q)Q_x = 0. \tag{4.17}$$

and linearize locally by approximating the Jacobian matrix $A(Q)$ on an interval using Roe averages and in every time step repeat the process. The resulting system can then approximate speed found from the eigenvalues of the averaged Jacobian matrix $A(\tilde{Q})$ (see below). For the purpose of this subsection we define $Q_R$ and $Q_L$ that represent the states on the

right and left in any direction from the flow boundary points $i \pm \frac{1}{2}$. This will make the two-dimensional problem easier. First step in this process is to linearize locally, given by

$$F(Q) = F(Q_L) + A(\tilde{Q})[Q - Q_L] \tag{4.18}$$

$$F(Q) = F(Q_R) + A(\tilde{Q})[Q - Q_R] \tag{4.19}$$

then add the above two equations to get

$$F(Q_R) - F(Q_L) = A(\tilde{Q})[Q_R - Q_L] \tag{4.20}$$

where here we solve for the averages in $A(\tilde{Q})$, and substitute them in

$$F_{i+\frac{1}{2}}(Q_L, Q_R) = \frac{1}{2}[F(Q_L) + F(Q_R)] - \frac{1}{2}\tilde{R}|\tilde{\Lambda}|\tilde{R}^{-1}[Q_R - Q_L] \tag{4.21}$$

to obtain our approximated solution $A(\tilde{Q}) = \tilde{R}|\tilde{\Lambda}|\tilde{R}^{-1}$, where $R$ is an eigenvector and $\Lambda$ is diagonal matrix of the eigenvalues. The same thing is done for $G(Q)$.

For the first system crowd model the averages are found to be

$$\tilde{\rho} = \sqrt{\rho_L \rho_R} \tag{4.22}$$

$$\tilde{v} = \frac{v_L\sqrt{\rho_L} + v_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \tag{4.23}$$

$$\tilde{u} = \frac{u_L\sqrt{\rho_L} + u_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}. \tag{4.24}$$

# Chapter 5

# Simulation

This chapter presents the numerical solutions for the four models developed in Chapter 3 simulated by the numerical methods discussed in Chapter 4. We show the Roe scheme result for the first crowd system given by (3.18), while the Lax-Friedrichs and FORCE methods results are applied on the four two-dimensional models. Test problem were conducted on each model to produce a behavior close to what we would physically expect from real observation. The tests fall under two main types:

1. Gaussian density distribution with different initial velocities and changing the magnitude and sign of $v_{f1}$ and $v_{f2}$.

2. Gaussian density distribution with one exit at a boundary.

We analysis and compare the results from all the models. First, we start with initial and boundary conditions.

## 5.1 Initial and Boundary Conditions

In the test problems we use gaussian distribution in two-dimensional space given by

$$\rho(x, y, 0) = C \exp^{(-(x-a)^2 - (y-b)^2)} . \tag{5.1}$$

where C is the maximum density value, and a & b determines the center of the density distribution. The free flow average speed (preferred speed) is assumed to be $v_{f1} = v_{f2} = 1.36\, ms^{-1}$ as many studies have used this value (it depends on service level concept), while density values vary from one study to another due to weather conditions, and location at the time of the study (winter, summer, men, children, Virginia, Moscow, etc.). Density is the number of pedestrians per unit area, and for the purpose of this simulation we set the maximum density $\rho_m = 1$.

Simulation is done on a square area with no obstacles that can divert the crowd flow, therefore, pedestrian are assumed to move freely within the boundaries. At the boundary non-slip conditions are enforced, except when we have an exit at some point $(x_e, y_e)$ (i.e., large room, one exit, and without any obstacles). By non-slip we mean closed walls where no pedestrian (density) can pass through, but they can move tangent to the walls. We use ghost cells, which is basically an imaginary extra cells that works as an extra boundary layer on the outer sides of the grid to simulate the boundary effect as described in (Leveque, 2002). The exit is a free flow point, where we assume that once a pedestrian reach the exit cell, his exiting velocity is equal to the free flow velocity $v_{f1}$ or $v_{f2}$ (sucking point).

For the crowd to move toward the exit, we simply point them to it by forcing them to follow the desired direction using the velocity-density function $V(\rho)$, and this is done by

$$V = V(\rho)\cos\theta = V(\rho)\frac{x_e - x_i}{\sqrt{(x_e - x_i)^2 + (y_e - y_i)^2}} \tag{5.2}$$

$$U = U(\rho)\sin\theta = U(\rho)\frac{y_e - y_i}{\sqrt{(x_e - x_i)^2 + (y_e - y_i)^2}}. \tag{5.3}$$

## 5.2 Simulation Results

In all the results, part (a) is the density initial profile.

**First model** (one-equation model): We have two simulation results, the first show the model dynamic response to direction change as the velocity sign (direction) change to obtain the contours in Fig. 5.1. This show the model ability to have a controlled bi-directional crowd flow and the response is consistent with the actual crowd behavior. When the space is not a constraint, pedestrians will move with free flow speed and diffuse in their given direction as time progresses. The scale of the diffuse in this simulation is bigger than the later figures due to simulation space and computational time. To distinguish this model dynamics from the others. We compare the result given in Fig. 5.2 part (d) with the Fourth model result in Fig. 5.16 part (d). We notice that the inner contour in both plots are different, the one corresponding to the one-equation model is smaller. In addition, due to the lake of the relaxation term the direction change occur instantaneously by $\pm v_{f_{1,2}}$ for one-equation model. Therefor, pedestrians are moving faster as seen by reaching the bottom left corner in part (d).

**Second model** (or first system model): Here we did several test to show the anticipation factor effect on the model. In Fig. 5.3, and Fig. 5.4 we changed the anticipation factor $C_0$ from 0.5 in the first to 1.1 in the second. The effect is clear from the figures part (d) plots. The first exceeded the maximum jam density, while the second its on target. In Fig. 5.5, and it corresponding contours in Fig. 5.6 show the effect of the shock wave formation as crowd move toward a corner in a closed area. In Fig. 5.7, we introduced an exit and the results show a) how crowd move toward the exit, and b) traffic jam at the exit point. In Fig. 5.8, and Fig. 5.9 we compare Roe's and FORCE schemes. Roe's scheme results are more concentrated than the FORCE scheme as seen from the center contour ring in part (d) in both figures.

**Third model** (or Second system model): To compare this model with the previous one, we

did the same last test as shown in Fig. 5.10. Although FORCE scheme is used, we got a response that is almost similar to the one in Fig. 5.8 for Roe's scheme. This shows the effect of the numerical schemes and their accuracy. In Fig. 5.11, we introduced an exit to simulate the second test response. We can see from the results how crowd flow toward the exit, and traffic jam is obvious at the exit point.

**Fourth model** (or the $3^{ed}$ system model): The density contours shown in Fig. 5.16 correspond to the density response when the initial free flow velocities are positive and equal (i.e., $v_{f1} = v_{f2}$). After some time the sign is changed to negative as seen from the system simulation in part (c). The results show that as time progresses, crowd (density) moves as expected toward the right upper corner, and then they change direction. At the same time the variation in the density concentration is changing as seen by plots (b) and (c) (diffusing in the direction of the flow). Fig. 5.12, and its corresponding contours in Fig. 5.13 show a clear picture of the density flow. Here $y - axis$ velocity is positive and $x - axis$ is negative. The last test is to show the model response at the exit (see Fig. 5.14, and Fig. 5.15). The simulation predicts jam density at the exit which is verified by the observed behavior in an emergency evacuation near bottleneck areas.

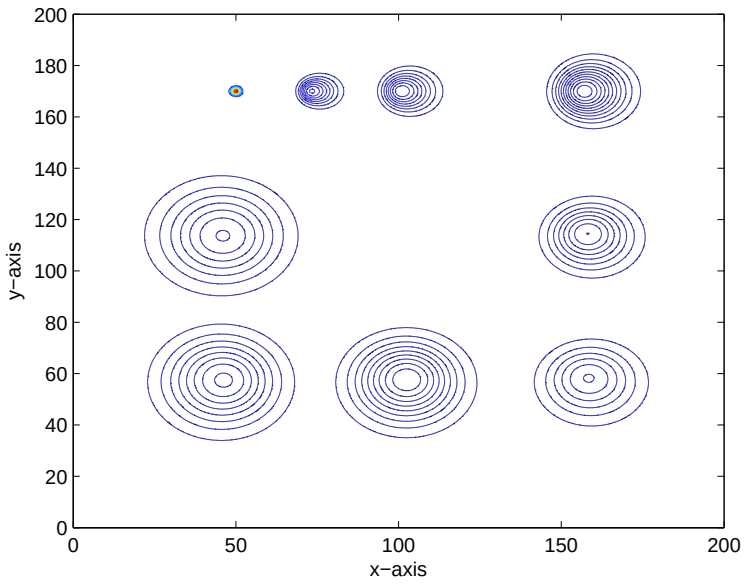


Figure 5.1: Test 1 Contours of the density response for the one-equation model using Lax-Friedrichs method

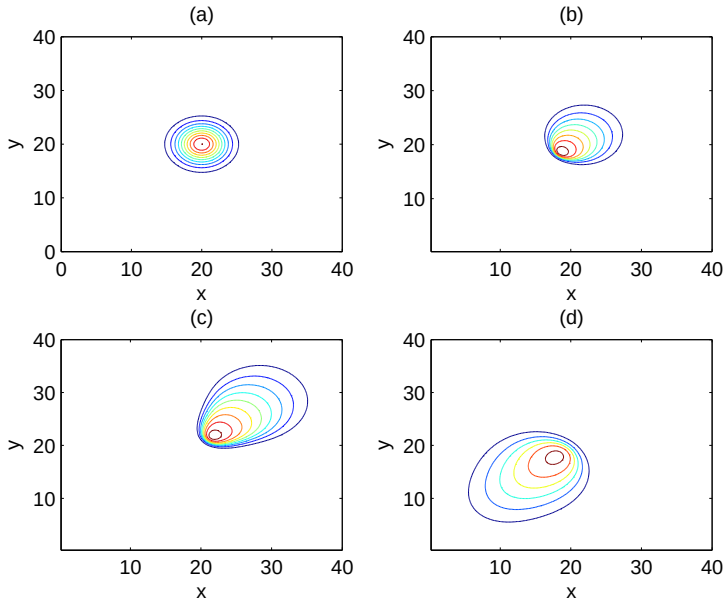


Figure 5.2: Test 1 Contours of the density response to direction change at different time frames for the one-equation model using Lax-Friedrichs scheme

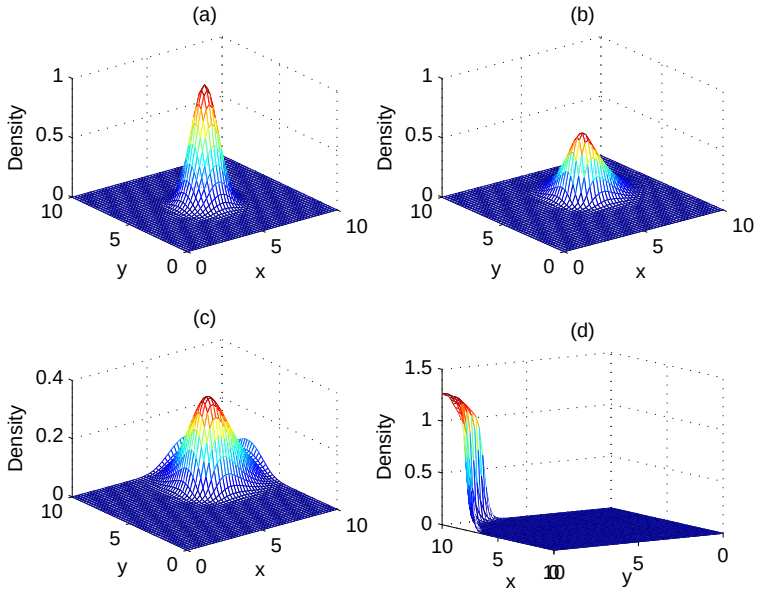


Figure 5.3: Test 1 Density response at different time frame for the first system model with $C_0 = 0.5$ using Roe scheme

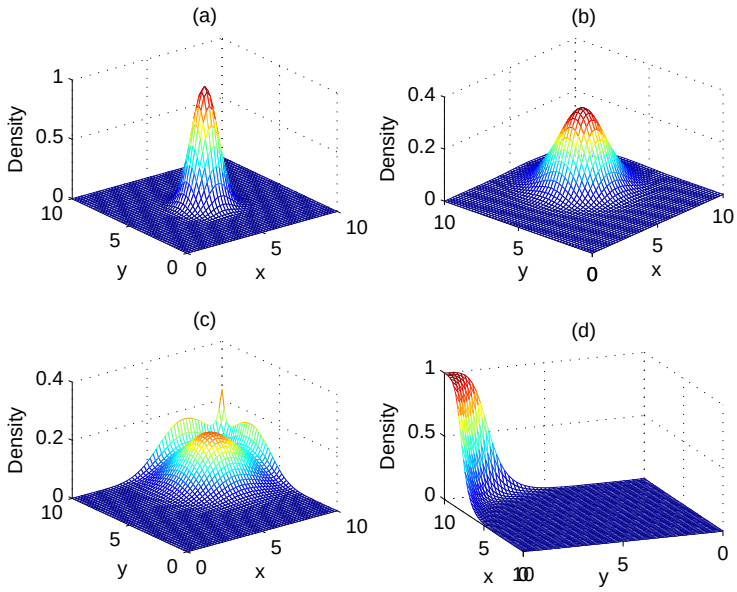


Figure 5.4: Test 1 Density response at different time frame for the first system model with $C_0 = 1.1$ using Roe scheme

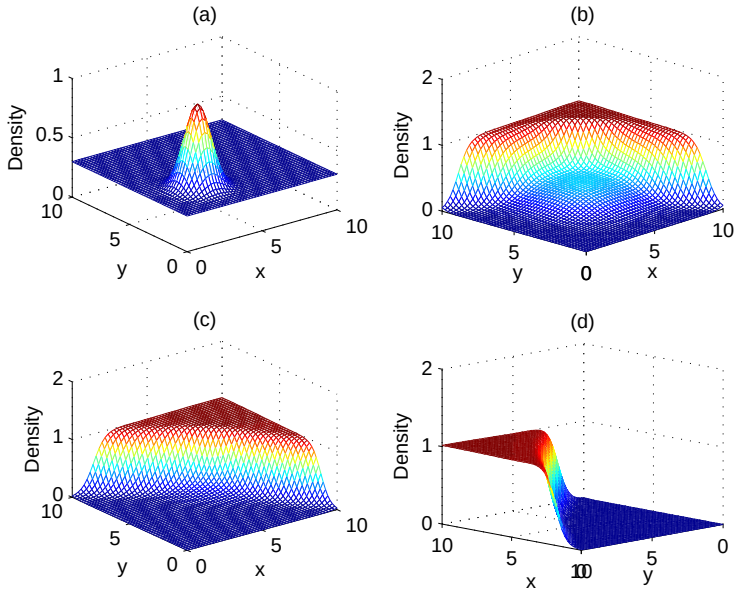


Figure 5.5: Test 1 Density response at different time frame for the first system model with $C_0 = 1.1$ using Roe scheme

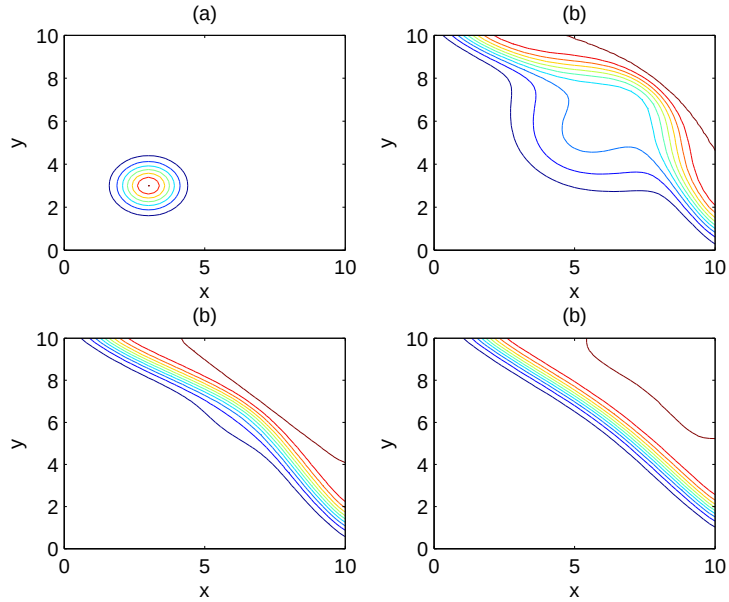


Figure 5.6: Test 1 Contours of the density response at different time frame for the first system model with $C_0 = 1.1$ using Roe scheme

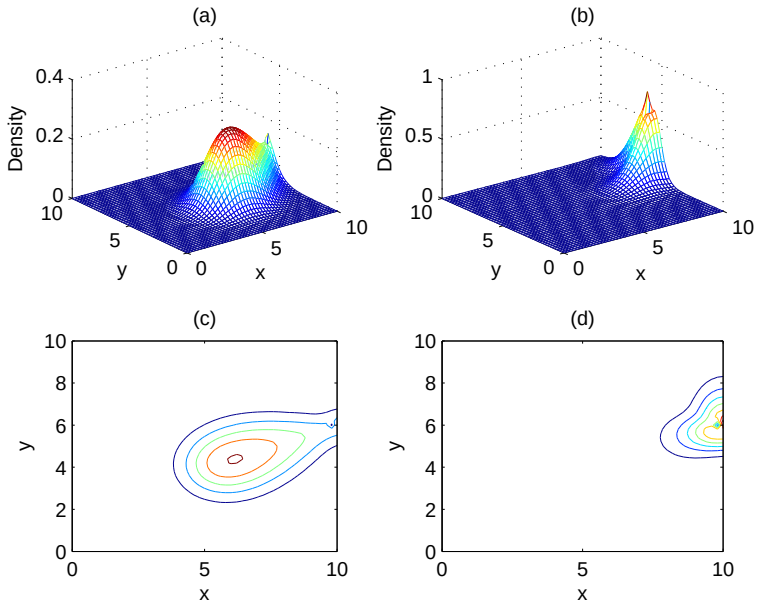


Figure 5.7: Test 2 Density response at different time frame for the first system model with an exit. Simulated using Roe Scheme and $C_0 = 0.8$

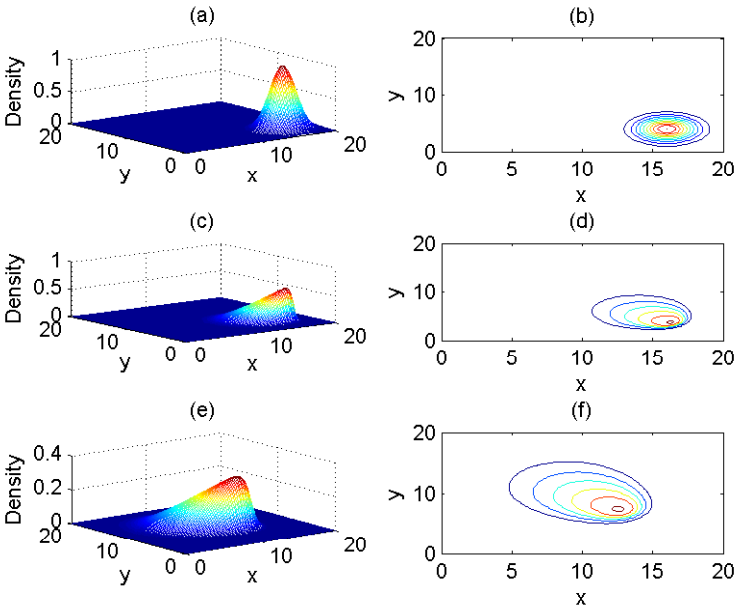


Figure 5.8: First system model using Roe scheme, response at different time frame

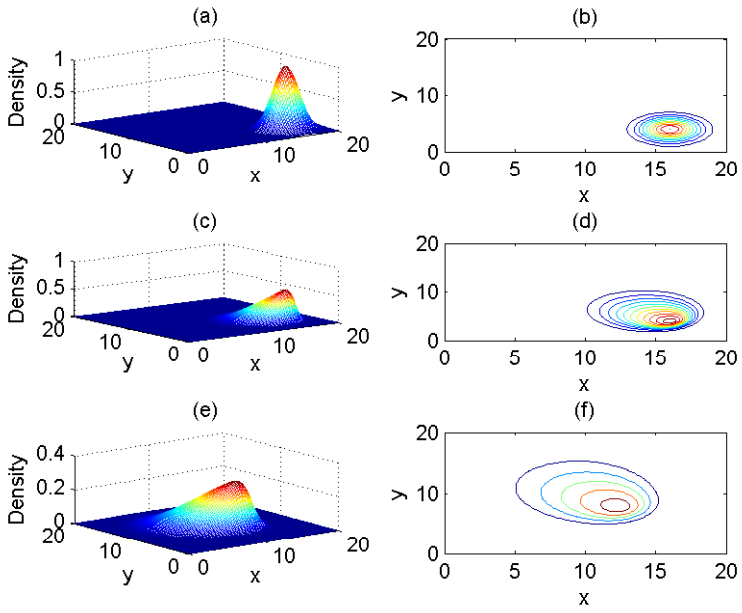


Figure 5.9: First system model using FORCE scheme, response at different time frame

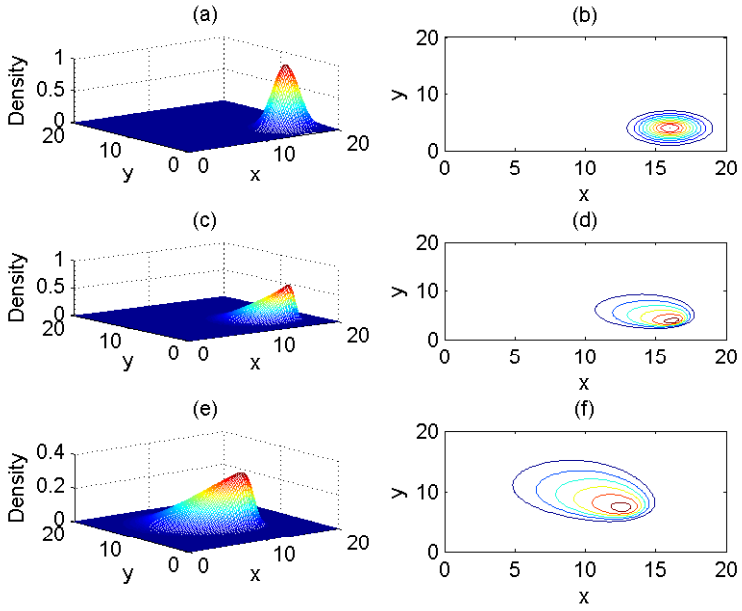


Figure 5.10: Second system model using FORCE scheme, response at different time frame

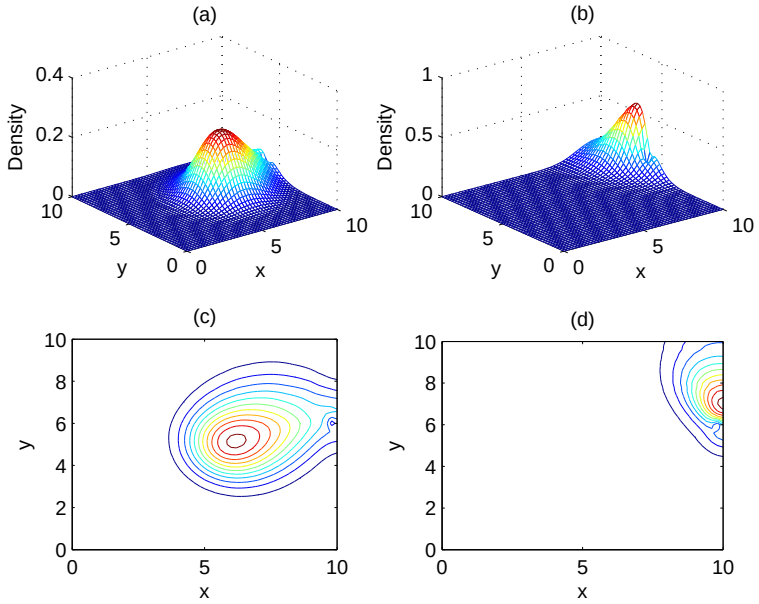


Figure 5.11: Test-2 Density response at different time frame for the second system model with an exit. Simulated using Roe Scheme and $\gamma = 5$

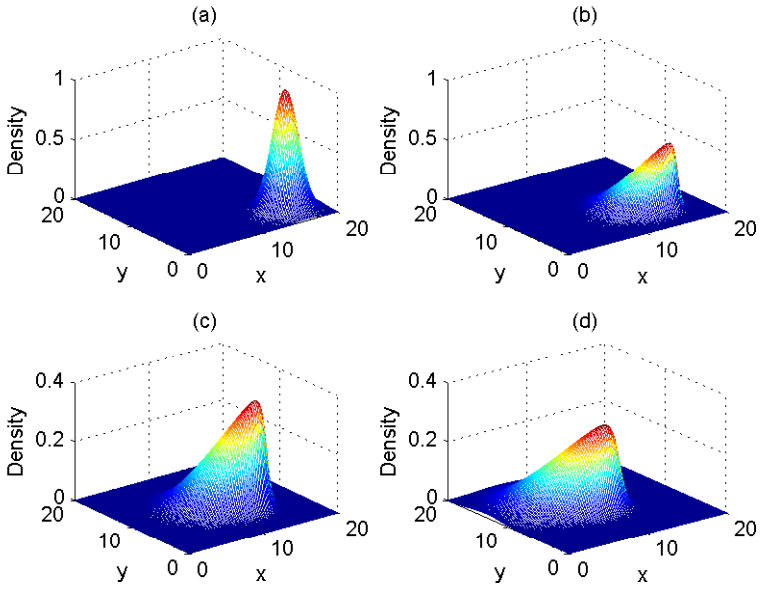


Figure 5.12: Test 1 Density response to direction change at different time frames for the $3^{ed}$ system using FORCE scheme

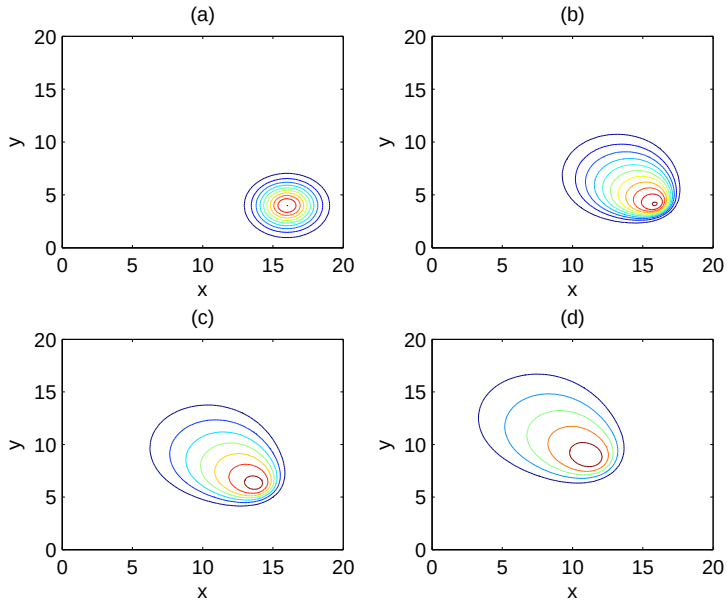


Figure 5.13: Test 1 Contours of the density response to direction change at different time frames for the $3^{ed}$ system using FORCE scheme

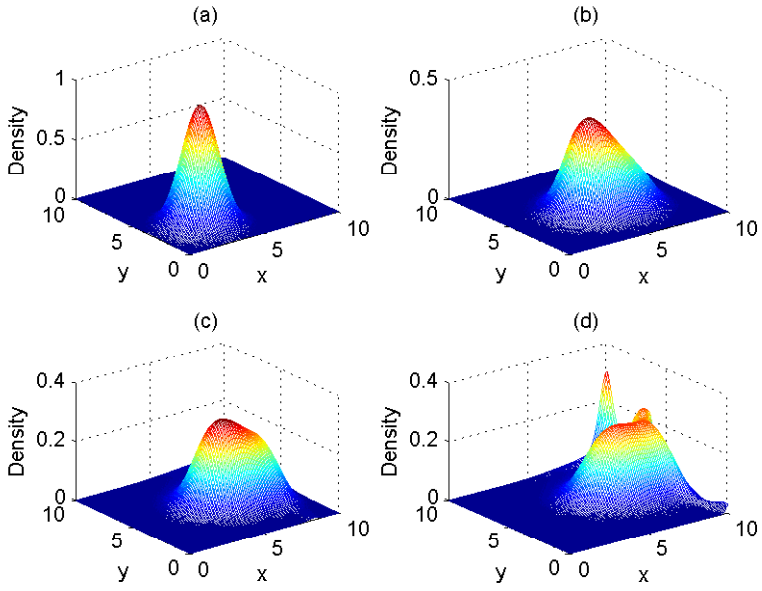


Figure 5.14: Test 2 Density response directed toward an exit at different time frames for the $3^{ed}$ model using FORCE scheme

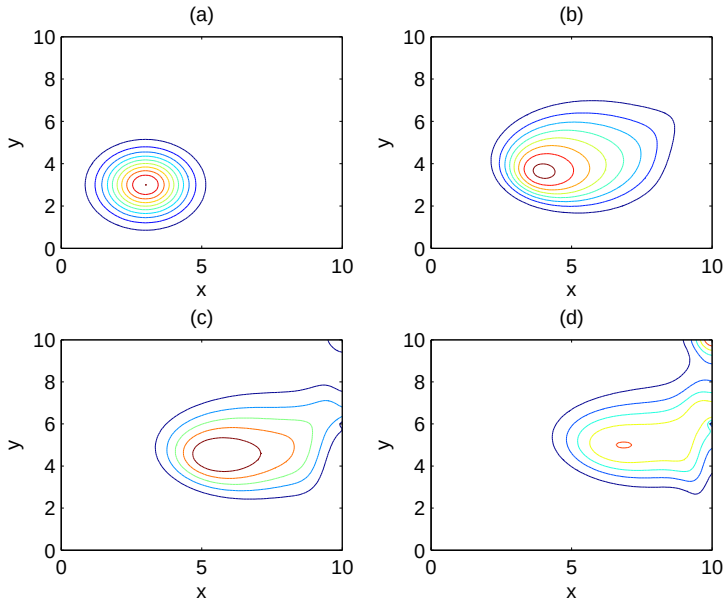


Figure 5.15: Test 2 Contours of the density distribution directed toward an exit at different time frames for the $3^{ed}$ system using FORCE scheme

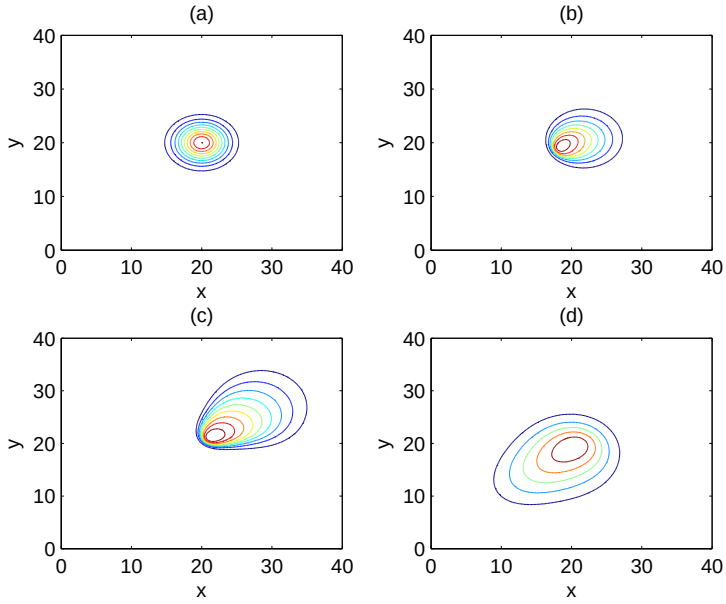


Figure 5.16: Test 1 Contours of the density response to direction change at different time frames for the $3^{ed}$ system using Lax-Friedrichs scheme

# Chapter 6

# Feedback Linearization Control

## 6.1 Introduction

In this chapter we present a controller design for the LWR one-equation model given in chapter 2 using feedback linearization method. The usual approach for the control of PDEs is by controlling the linear or nonlinear ODEs that result from spatial discretization of the original PDEs. Known difficulties and disadvantages associated with this approach are, for example controllability and observability that should depends only on the locations of sensors and actuators, may also depend on the discretization points number, location, and method (Ray, 1981). In addition, moving from infinite dimensional to finite dimensional systems may lead to conclusions about the stability of the open-loop and/or closed-loop system that are not generally correct. Therefore, significant amount of interest is aimed toward the development of a control design based on the distributed parameter systems which accounts for the nature of the PDEs (Christofides, 2000) and references therein.

Here we are interested in designing a feedback controller to evacuate pedestrians from a one-dimensional area (e.g., corridor). The method of feedback linearization works in a way to cancel the non-linearities in the system and it is well-know in nonlinear control for ODEs

(Khalil, 1996), and it was introduced to quasi-linear hyperbolic PDEs in (Christofides, 2000). This chapter presents the feedback linearization control design for the LWR model.

This chapter is organized in the following manner: In section 6.2, we give a mathematical background on the control design using state feedback linearization. Section 6.3 presents an application of the controller and discusses its closed-loop stability and simulation results. In 6.4, we extend this approach by dividing the one-dimension area into n number of sections (also called patches).

## 6.2 Theory

### 6.2.1 Control Problem

The general quasi-linear one-dimensional PDE system can be written as

$$\begin{aligned} \frac{\partial Q}{\partial t} &= A(Q)\frac{\partial Q}{\partial x} + f(Q) + g(Q)\bar{u}, \\ y &= h(Q), \\ B.C &: C_1\rho(a,t) + C_2\rho(b,t) = R(t) \end{aligned} \tag{6.1}$$

where $Q(x,t)$ is the vector of state variables defined on $Q(x,t) \in H^2[(a,b),\Re]$. Using the patch idea (see Figure 6.1) we set

$$\bar{u}(x,t) = \sum_{i=1}^{n} b^i(x)u^i(t) \tag{6.2}$$

$$y^i(t) = \mathbf{C}^i\bar{y}(x,t) = \mathbf{C}^i h(Q) = \int_{x_i}^{x_{i+1}} \mathbf{c}^i(x)h(Q(x,t))\,dx \tag{6.3}$$

where $\mathbf{C}^i$ is an operator that depends on the desired performance given by an integral same as in the majority of the applications in Christofides (2000), $\mathbf{c}^i(x)$ and $b^i(x)$ are known smooth functions on $x$ assumed to be normalized and are given by

$$\sum_{i=1}^{n}\int_{x_i}^{x_{i+1}} b^i(x)\,dx = \sum_{i=1}^{n}\int_{x_i}^{x_{i+1}} \mathbf{c}^i(x)\,dx = 1 \tag{6.4}$$

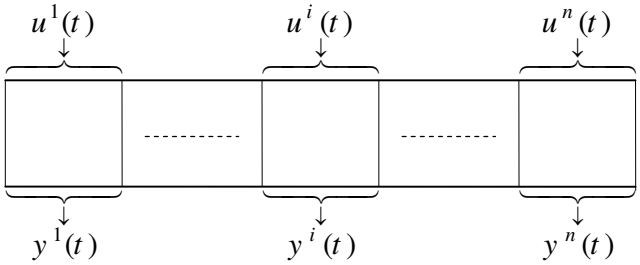


Figure 6.1: Control specification in case of a one-dimensional problem

Using the above notation we rewrite the quasi-linear PDE to get

$$\begin{aligned} \frac{\partial Q}{\partial t} &= A(Q)\frac{\partial Q}{\partial x} + f(Q) + g(Q)b(x)u, \\ y &= \mathbf{C}h(Q), \\ B.C &: C_1\rho(a,t) + C_2\rho(b,t) = R(t) \end{aligned} \tag{6.5}$$

where $u = [u^1 \ ... \ u^l]^T$, and $y = [y^1 \ ... \ y^l]^T$.

### 6.2.2 Characteristic Index

The concept of characteristic index $\sigma$ is simply the smallest number of Lie derivatives (the derivative of a scalar field in the direction of a vector field) that establishes a direct relationship between the output $y$ and the input $u$. The characteristic index can be found from

$$\begin{aligned} y &= \mathbf{C}h(Q) \\ \frac{dy}{dt} &= \mathbf{C}\left(\sum_{j=1}^{l}\frac{\partial Q_j}{\partial x}L_{a_j} + L_f\right)h(Q) \\ &\vdots \\ \frac{d^\sigma y}{dt^\sigma} &= \mathbf{C}\left(\sum_{j=1}^{l}\frac{\partial Q_j}{\partial x}L_{a_j} + L_f\right)^\sigma + \mathbf{C}L_g\left(\sum_{j=1}^{l}\frac{\partial Q_j}{\partial x}L_{a_j} + L_f\right)^{\sigma-1} h(Q)b(x)u. \end{aligned}$$

Once the term that includes the input $u$ is nonzero we stop and proceed to the following step in control design. If $\sigma = \infty$, then $u$ does not show up and this design method will not work.

### 6.2.3 State Feedback Control

Assuming all states are available for measurement, the controller we seek can be found from

$$\gamma_\sigma \frac{d^\sigma y}{dt^\sigma} + \ldots + \gamma_1 \frac{dy}{dt} + y = v \tag{6.6}$$

and for our system (6.5) it is given by

$$\begin{aligned} u &= \left[ \gamma_\sigma \mathbf{C} L_g \left( \sum_{j=1}^{n} \frac{\partial Q_j}{\partial x} L_{a_j} + L_f \right)^{\sigma-1} h(Q) b(x) \right]^{-1} \\ &\quad \times \left\{ v - \mathbf{C} h(Q) - \sum_{\upsilon=1}^{\sigma} \gamma_\upsilon \mathbf{C} \left( \sum_{j=1}^{n} \frac{\partial Q_j}{\partial x} L_{a_j} + L_f \right)^{\upsilon} h(Q) \right\} \end{aligned} \tag{6.7}$$

### 6.2.4 Closed-Loop Stability

Here we use the definition of closed-loop stability given in (Christofides, 2000) for the system (6.5) under the controller (6.7). The controlled system is exponentially stable (i.e., the differential operator of the linearized closed-loop system generates an exponentially stable semigroup) if the following conditions are satisfied:

1. The roots of the equation

$$1 + \gamma_1 s + \ldots + \gamma_\sigma s^\sigma = 0$$

   lie in the open left hand of the complex plain.

2. The zeros dynamics of the system (6.5) are locally exponentially stable.

The first condition can be designed easily by pole placement, and the second condition is checked by setting the output $y = \mathbf{C}h(Q) = 0$. For more details see Christofides (2000); Ray (1981).

## 6.3 Application to the LWR Model (One patch)

Our aim is to control the LWR model using one patch only. The one-equation PDE is given by

$$\begin{aligned}\frac{\partial \rho}{\partial t} &= -\frac{\partial q}{\partial x} \\ &= -\frac{\partial \left(\rho v_f \left(1-\frac{\rho}{\rho_m}\right)\right)}{\partial x} = -v_f \frac{\partial \left(1-\frac{2\rho}{\rho_m}\right)}{\partial x} \\ &= -v_f \frac{\partial q(\tilde{\rho})}{\partial x} \end{aligned} \tag{6.8}$$

where $u = v_f$ is the control variable, and the density $\rho$ is the variable we want to control (for pedestrian evacuation, final $\rho = 0$). The area is $[0, 1] \in \Re$ and the boundary condition is closed from the left and open at the right end. Comparing this PDE to the general form (6.1) we find that $f(Q) = 0$, and $g(Q) = 0$. If we take $a = \frac{\partial \tilde{q}}{\partial \rho}$ the system can be represented by

$$\frac{\partial \rho}{\partial t} = -v_f \; a \; \frac{\partial \rho}{\partial x} \tag{6.9}$$

where $a = \left(1 - \frac{2\rho}{\rho_m}\right)$. The choice of $\mathbf{c}^i(x) = \frac{1}{x_{i+1}-x_i}$ agrees with (6.4) and gives

$$y = \mathbf{C}h(\rho) = \int_0^1 \frac{1}{1-0}\rho(x,t)\; dx = \int_0^1 \rho(x,t)\; dx \tag{6.10}$$

where the output operator $\mathbf{C}$ is one patch only, and $b(x) = 1$. Next we proceed with the control design as follows:

1. For the characteristic index if we just take the derivative of the output one time $v_f$ will show as:

$$\begin{aligned} y &= \mathbf{C}h(Q) = \mathbf{C}\rho(x,t) \\ \dot{y} &= \mathbf{C}\frac{\partial \rho}{\partial \rho}\,\dot{\rho} = \mathbf{C}\frac{\partial \rho}{\partial \rho}\,\frac{\partial \rho}{\partial t} \\ &= -\mathbf{C}v_f\, a\, \frac{\partial \rho}{\partial x} = \mathbf{C} - v_f \frac{\partial \tilde{q}}{\partial x} \\ &= -v_f \int_0^1 \frac{\partial \tilde{q}}{\partial x}\; dx \\ &= -v_f(\tilde{q}(1) - \tilde{q}(0)). \end{aligned}$$

Hence, the characteristic index is equal to one. The manipulated input $v_f$ is outside of the integral because it depends on time only. The integral and the partial derivative operator cancel out and we are left with the $\tilde{q}$ evaluated at the boundary points.

2. The state feedback controller comes from (6.6) which in our case reduces to $\dot{y}+K\ y = 0$, where $v = 0$ and $K = \frac{1}{\gamma_1}$. Thus the controller is given by

$$v_f = u = \frac{K \int_0^1 \rho(x,t)dx}{(\tilde{q}(1) - \tilde{q}(0))}. \tag{6.11}$$

3. The closed-loop system can be found by substituting (6.11) in (6.8) to get

$$\frac{\partial \rho(x,t)}{\partial t} = -\left\{ \frac{K \int_0^1 \rho(x,t)dx}{(\tilde{q}(1) - \tilde{q}(0))} \right\} \times \frac{\partial \tilde{q(\rho)}}{\partial x}. \tag{6.12}$$

For the stability of the closed-loop system, and according to the conditions given in 6.2.4 we first choose $\gamma > 0$ or $K > 0$ because $1 + \gamma_1 s = 0 \Longrightarrow s = -\frac{1}{\gamma_1}$ is in the left half of the complex plain. For the zero dynamics we set $y = 0$, and this can happen if and only if $\rho = 0$ because the density is a non-negative value ($(y = 0) \Longleftrightarrow (\rho = 0)$) and we get

$$\begin{aligned} \frac{\partial \rho(x,t)}{\partial t} &= -\frac{K \overbrace{\int_0^1 \rho(x,t)\ dx}^{=0}}{(\tilde{q}(1) - \tilde{q}(0))} \times \frac{\partial \tilde{q}}{\partial x}. \\ \frac{\partial \rho}{\partial t} &= 0 \end{aligned}$$

Hence, zero dynamics are stable. Therefore the closed-loop system is stable.

The control action response for a corridor with exit at the right hand side is shown in Fig. 6.2. After some finite time, evacuation is completed and the density is zero. This controller attempt to control the flow around $y = \int_0^1 \rho(x,t)\ dx$ every time which is shown in the figure by the different time responses.

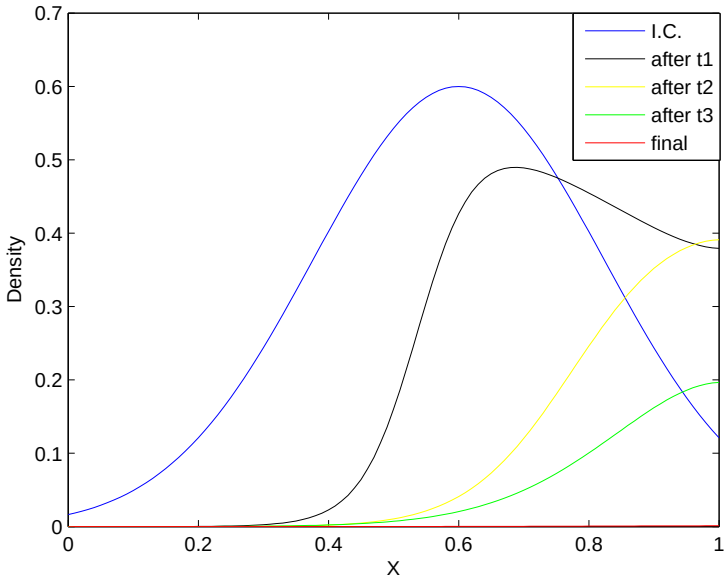


Figure 6.2: Density response for one patch controller. Initial density maximum value is 0.6, $\Delta t = 0.002sec$ and $\Delta x = 0.01$, gain K=1, and total time$\approx$ 7 sec. to reach zero density

## 6.4 Application to the LWR Model (n=5 patches)

Here we repeat the design steps 1 to 4 shown in the previous section with five patches instead of one. Therefore, we have i=1,...5 and $b^i u^i = u^i$, where $b^i$ are taken as unity distribution functions and

$$u(t) = \begin{cases} u^1(t), \ [0.0, 0.2] \\ u^2(t), \ [0.2, 0.4] \\ u^3(t), \ [0.4, 0.6] \\ u^4(t), \ [0.6, 0.8] \\ u^5(t), \ [0.8, 1.0] \end{cases} \tag{6.13}$$

The desired performance is to control the outputs

$$y(t) = \begin{cases} y^1(t) = \int_{0.0}^{0.2} \rho(x,t)dx \\ y^2(t) = \int_{0.2}^{0.4} \rho(x,t)dx \\ y^3(t) = \int_{0.4}^{0.6} \rho(x,t)dx \\ y^4(t) = \int_{0.6}^{0.8} \rho(x,t)dx \\ y^5(t) = \int_{0.8}^{1} \rho(x,t)dx \end{cases} \tag{6.14}$$

So the output is

$$y^i(t) = \int_{x_i}^{x^{i+1}} \mathbf{c}^i(x)\rho(x,t)\ dx \tag{6.15}$$

where $\mathbf{c}^i(x) = \frac{1}{x_{i+1}-x_i}$.

1. The characteristic index is equal to one as shown by item 1 in section 6.3.

2. The state feedback controller is given by

$$u^i = \frac{K \displaystyle\int_{x_i}^{x^{i+1}} \frac{\rho(x,t)}{x_{i+1}-x_i}\ dx}{(\tilde{q}(x_{i+1}) - \tilde{q}(x_i))}. \tag{6.16}$$

3. Closed-loop system is given by

$$\frac{\partial \rho(x,t)}{\partial t} = -\left[\sum_{i=1}^{5} \left(H(x-x_i) - H(x-x_{i+1})\right) u^i\right] \times \frac{\partial \tilde{q(\rho)}}{\partial x}. \tag{6.17}$$

4. For closed-loop stability, condition one is satisfied if $K > 0$ as in the one patch case. For the zero dynamics we set $y = 0$, and this happens if and only if in each patch $\rho = 0$. Substitute $y = 0$ in the closed-loop in 3 we obtain $\frac{\partial \rho(x,t)}{\partial t} = 0$. Hence the closed-loop system is stable.

The control action response for a corridor with exit at the right hand side is shown in Fig. 6.3. After some finite time, evacuation is completed and the density is zero. The response here is different than the one for the single patch as seen by the responses approaching zero.

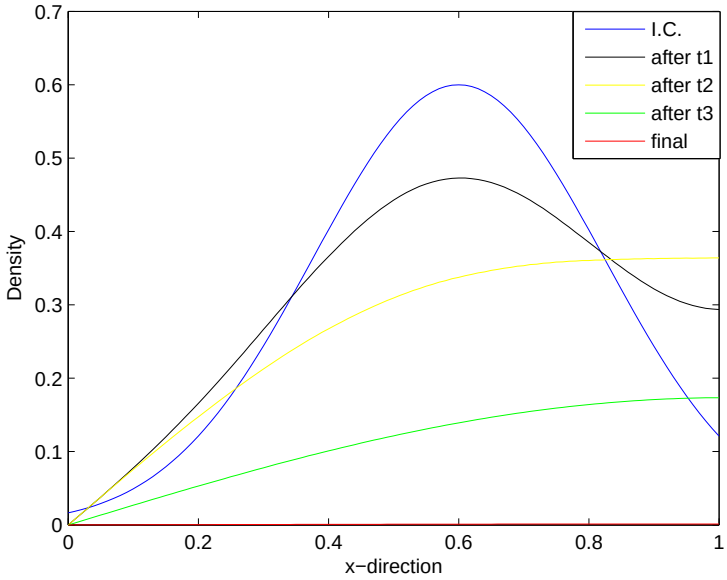


Figure 6.3: Density response for five patch controller. Initial density maximum value is 0.6, $\Delta t = 0.000005 sec$ and $\Delta x = 0.01$, gain K=1, and total time$\approx$ 0.2 sec. to reach zero density

# Chapter 7

# Suggested Intelligent Evacuation Systems (IES)

In this chapter we discuss the new concept of intelligent evacuation system or IES. This new technology will make a major change in evacuation strategies. Control, communication and computing technologies will be combined into IES system that can increase crowd safety and management without changing the physical structure of the facility. This chapter outlines the key features of medium area IES and shows how core crowd decisions may improve. We also propose a basic IES control system architecture based on the one given by (Varaiya, 1993), and we discuss the information technology issues related to its implementation.

## 7.1 Introduction

Congestion conditions during an emergency evacuation poses a great challenge for any evacuation management system. Congestion occur when the demand for travel exceeds the runway (corridor, stairs, highways), and bottleneck (exits, intersections) capacity. Hence a sound approach for reducing congestions will involve a mix of policies affecting demand and

capacity depending on local circumstances and priorities. These policies include buildings, parking lots, stadiums, subway stations, airports, highways, and reducing demand by mass transit, carpooling on highways. This chapter discusses the potential of another strategy option called Intelligent Evacuation System (IES). In this strategy, we claim that a proper combination of control, communication and computing technologies, placed on corridors, exits, TVs , internet, cars, and highways can assist people's decisions in ways that will increase evacuation safety and management without changing the physical structure of a facility or roads.

The intelligent evacuation system we propose here should take the following aspects into consideration

- Function: The range of functions associated with the evacuation system to be automated, and the degree of automations.
- Architecture: The breakdown of these functions into various control tasks, and the delegation of these tasks.
- Design: The division of intelligence between evacuees and facility, and how the use of technologies can realize this architecture.
- Effectiveness: The cost effectiveness and benefits of different evacuation systems.

Other aspects of this strategy relate to the timing of the system development, which concerns the accommodation of new functions in new design. This chapter is concerned with the above four points only.

Section 7.2 presents a framework for describing IES functions and their relation to key evacuee decision for different evacuation scenarios. Different IES proposals can be compared according to the degree of influence they exert on those decisions. These proposals may range from systems that are relatively simple from control viewpoint, since they merely provide information or advice to the evacuee, to more automated systems using open loop and

feedback control strategies. In an open loop scheme the traffic will flow in a predetermined way which would be assessed based on the initial condition of the flow. Since the evacuation of people is a very dynamic situation hence a feedback solution is obviously better suited. We argue that an evacuation system that use feedback control strategy can achieve much better evacuation time reduction and management. The evacuation strategies will be optimally calculated on a real-time basis using advanced concepts of the feedback control theory.

Section 7.3 outlines two evacuation scenarios of evacuating people, train, and cars from a train substation, and an airport. For city evacuation we refer the reader to (Varaiya, 1993). We discuss these scenarios in terms of the evacuee decisions listed in table 7.1. Designing such systems poses a challenging problem for control.

Section 7.4 proposes a four-layer hierarchical control architecture which splits this problem into more manageable units. Starting at the top, the layers are called the network, link, planning and regulation layers. The regulation layer's task is to execute feedback laws for acceleration, direction change, and stop. The planning layer's task is to coordinate the movement of neighboring evacuee/vehicles and supervise the regulation layer during the execution of feedback laws. The link layer is responsible for the control of aggregated traffic flow over some area of the facility. Finally, the network layer assigns routes to evacuees/vehicles and controls admission into exits.

Section 7.5 discusses the control architecture for evacuation scenarios described in section 7.3. Section 7.6 discusses the information technology issues.

Section 7.7 gives a brief discussion on feedback control and modeling of the crowd dynamics.

## 7.2 IPES Functions

Here we present five decisions made by an evacuee in the course of an emergency evacuation for small size area (includes floor, building, and parking lot), mid size area ( subway station,

Table 7.1: Evacuee decisions and IES functions

| Stage | Evacuee decision | IES aim | IES job | Approach |
|---|---|---|---|---|
| Pre-evac | Evacuation plan and methodology | Efficient use of resource | Demand shift | 1, 2, 4 |
| In-evac | Route choice | Optimize travel time | Route guidance and flow control | 1, 2, 3 |
| | Path planning | Smooth flow | Congestion control | 1, 2, 3 |
| | Regulation | Increase flow, safety | Proper spacing, steering, etc. | 1, 3 |
| Post-evac | Parking,Shelters etc. | Add safety | Efficient use | 1, 2, 3, 4 |

airport), and large size area (includes city, county, and state). Table 7.1 gives these decisions and points out how they can be modified to increase safety and management of an evacuee. We divide the decisions into three stages, pre-evac, in-evac, and post-evac. Corresponding to these decisions, IES objectives and tasks are given, which if carried out accordingly, would realize the desired objectives.

We can add more decisions to this decision list, but in our view, the five decisions given in the table are the most important to IES plan and the rest do not affect it significantly. Also including more decisions to the list will increase its size considerably. In addition, people will be more comfortable with a system that can take care of these non-core issues. For instant, an automated exit doors can eliminate the need to stop, find the keys, and open these doors. Doors here could be building exit doors, subway or train doors, parking lot gates, highway toll, etc.

The last column in the table gives possible approaches to carry out the IES tasks, and these are

1. Supplying information (road condition, exit locations, etc.)

2. Giving advice.

3. Controlling the decisions.

4. Changing choices based on cost related issues (the use of a car, carpooling, toll, etc.)

These approaches increase the evacuation system difficulty, as we move from providing information to giving advice to applying control (can be either open loop or feedback) as represented by

$$\text{Information} \rightarrow \text{Advice} \rightarrow \text{Control}$$

However, at the same time evacuee decisions are becoming more automated as we go from providing information to applying control.

In the next section we see how the functions listed in table 7.1 work for two different evacuation scenarios. Generally speaking, scenarios can be divided into three broad categories: small area ,mid-size area, and large area. The small area scenarios cover the cases of a floor, building, and a parking lot. The mid-size area scenarios which we discuss next covers a subway station and an airport. The large area scenarios deal with evacuating people/cars from towns, cities etc.

## 7.3 IES functions for evacuation scenarios

Here we will argue that a controlled evacuation system will provide better safety and management results than the information or advice evacuation strategies. To do that, we take a close look at two scenarios and discuss the effect of table 7.1 on improving the evacuation process.

### 7.3.1 Subway Station

First scenario is a one level subway station given in 7.1. In a pre-evac phase, we will discuses how information, advice and cost can affect the evacuation generation and model choice, and how the IES system makes them more efficient. An information based evacuation system can consist of sensors connected to an alarm, and lights signs showing train doors and exit locations. An alarm will start and evacuees' decision will be to take the train, walk or run to safety (exit the station), or maybe ask for information. In an advice system, trained human resources can provide the decision by asking the evacuees to move slowly, run toward an exit, or take a shelter and so on. To make this phase more efficient a controlled process can provide the decision based on open loop or feedback controller. Possible methods are voice commands, automatic doors, or light matrix that shows slow light changing for walking and fast for running. These methods are different in open loop from feedback based on sensing other information like jam at one of the exits, possibility of trains colliding (from network information), biological threat (Okumura et al., 1996), etc.

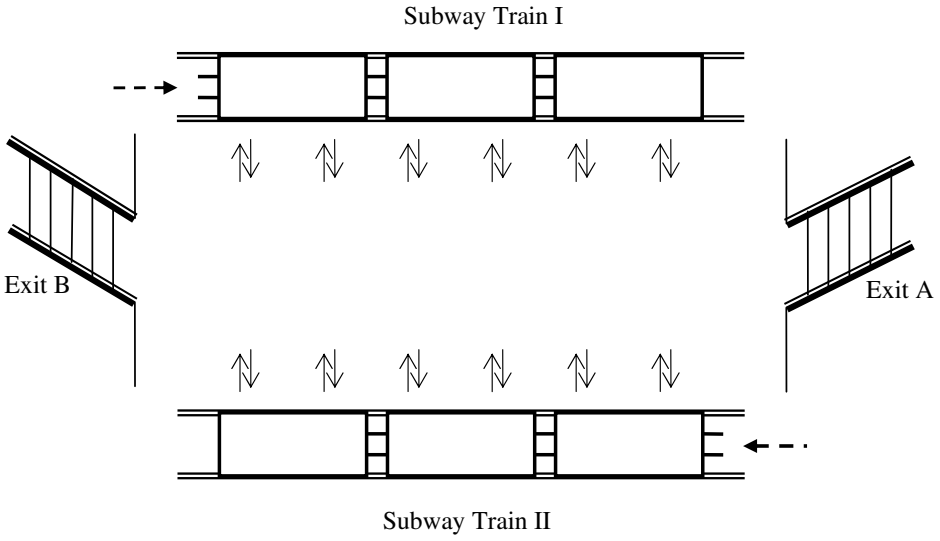


Figure 7.1: Subway station

The in-evac phase, the route choice decision for an information based system can be following

signs or other evacuees toward what is supposed to be safety. However, this could guide to congestions at bottleneck locations (exit doors, and stairs) and increase evacuation time (Hoogendoorn, 2001; Hoogendoorn and Bovy, 2004). Information that evacuees obtain from visual, and interactions with other evacuees/obstacles that is categorized under *behavior observations* of the traffic flow will influence their maneuver, and regulation decisions. For the advice system, route choice and path planning can be provided. Nevertheless, since advice is given based on visual area conditions, same bottleneck problem could still occur, even if advisors (trained human resources) are communication with each other. For example, they have no control on exit doors, changes to trains route, etc. To improve this phase, control design can choose the route with minimum time to reach an exit or safety based on current location and flow control. It also provides the evacuee with a plan to reach the exit by preventing congestion near bottleneck locations. In addition, to increase safety and management of evacuees, maneuver and regulation decisions can be used to coordinate exits (open and close), train doors, etc., and flow control (e.g., variation in moving velocities and direction). Finally, the post-evacuation phase is concerned with providing help by notifying emergency services, guide them to the disaster area and guiding injured evacuee to their parking area to get the fastest help possible.

### 7.3.2 Airport

An airport is considered a mid size area. Example of an airport schematic is given in Fig. 7.2. It consists of a terminal, airplanes, parking lots, building services, and vehicle routs. The aim here is to discuss this scenario and how the functions in Table 7.1 improve the evacuation process. Before we do so, we note that the plan is to evacuate all pedestrians in the affected area. If this area is a floor, building, or a parking lot we can create scenarios for them as well, but this is not considered here.

In pre-evac, a simple evacuation system can consist of sensors in a building connected to an alarm, and lights signs showing exit locations. When an alarm starts, evacuee's decision

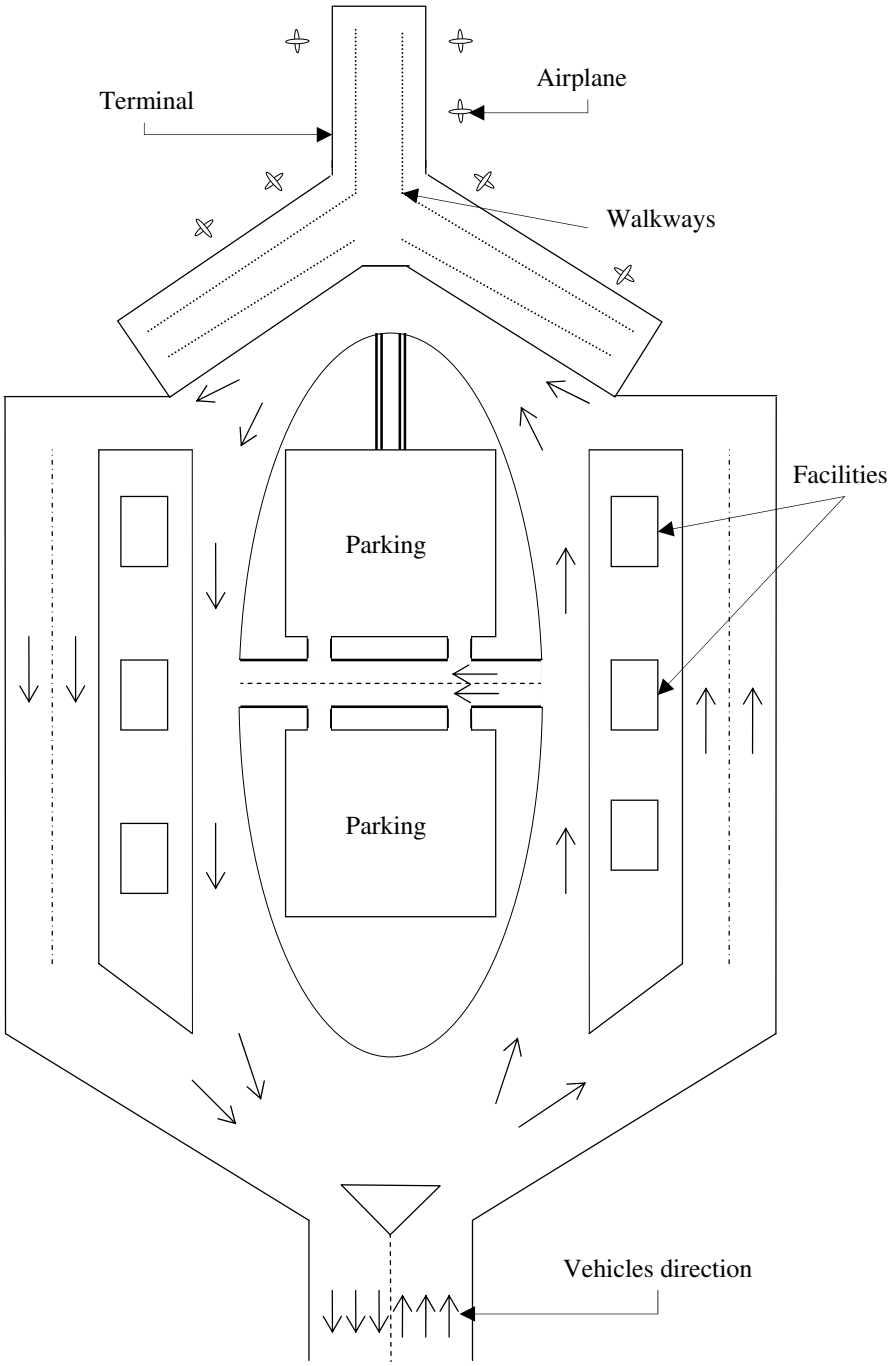


Figure 7.2: Airport traffic flow

will be to ask for information, walk or run to safety following the exit signs that will lead them to exit the building. Next they go to the parking lot to their vehicles, or any other transportation available to drive them out of the airport area (disaster area). In an advice system, trained human resources can guide the evacuees to use certain routs, like taking refuge in a shelter and so on. But both systems can not deal with mass evacuation where congestion and jam will occur in buildings, parking lots, and on the roads connecting these facilities. To improve this process, an open loop or feedback controller can be used to assign an evacuation rout, and synchronize the number of evacuees and vehicles flow into the roads. By doing so, smooth flow could be achieved. Possible methods are commands asking the controller of each node (facility) to evacuate a certain number of evacuees per unit time, automatic road and air traffic control, etc. These methods are different in open-loop from feedback based on sensing information like congestions, road blocks, number of airplanes landing and leaving from the airport, etc.

The in-evac phase, the route choice decision for a simple system can be self or following other evacuees that will eventually lead to congestions at bottleneck locations and increase evacuation time. For the advice system, route choice and path planning is based on visual and limited communication between system advisors (security personals, police). In addition, their control on roads, and airplanes traffic is insufficient (many airplane accidents happen in normal situation). To improve this phase, control design can choose the route with minimum time to reach an exit or safety based on current location and flow control of the different system nodes. Maneuver and regulation decisions can increase safety and management of evacuees by controlling velocity and direction of evacuees/vehicles. Finally, the post-evac phase is concerned with identifying an appropriate shelter, notifying emergency services and lead them to the disaster area, and guide injured evacuee to safe areas to get the fastest help possible.

## 7.4 Four-Layer System Architecture

With the above mentioned scenarios as background we now discuss how to formulate the problem of designing the control system which carries the tasks of IES. In this section we propose a control system architecture similar to one proposed in (Varaiya, 1993). Here we will represent the functions given in Table 7.1 by a control system architecture whose block diagram representation is given by Fig. 7.3. From top, the layers are called network, link, planning and regulation.

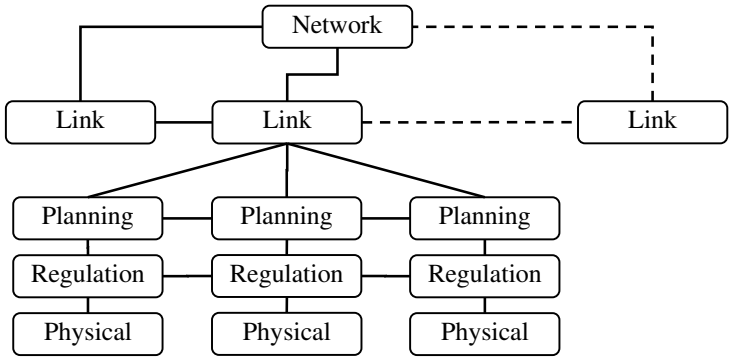


Figure 7.3: Four layer architecture

There is a single network layer. The controller task in this layer is to assign an escape route for the evacuees in the system, and an entrance route for the emergency services. From abstract point of view, this controller must determine a routing table with entries of the form

$$(\text{ Current evacuee location, exit}) \rightarrow (\text{Evacuee route})$$

This table is updated based on the area affected in a way to optimize evacuee flow. The updates would be based on the information about the aggregate state of the flow of people. To find the entries for this table, we make use of controllers that identify the route associated with optimum time. This can be done either by feedback or open-loop controller. A feedback

assignment of routes would correspond to assessing the current traffic and incident situation, then deciding dynamically about the route assignment. In an open-loop scheme the traffic will flow in a predetermined way which would be assessed based on the initial condition of the flow. Since the evacuation of people is a very dynamic situation, hence a feedback solution is obviously better suited. We suggest the use of central computer for the calculations of the entries, since on board calculation is difficult, because it requires equipments such as computing devices, sensing information, and communication to be available with every evacuee.

In the IES, the disaster area is divided into various sections with a link layer controller assigned to each section. Each link layer assigns target speed and path to all the evacuees in such a way that a) evacuee follow the assigned rout toward the exit, b) congestion in and between the links is avoided. Thus, if there is a jam condition at a front link, the control link layer must adopt and change its speed command (slow down or stop). The calculations are updated based on the response in density distribution and speed changes in each section. The updates depends on the link area size and they are made more frequently in this layer to assist the situation. The link layer controller design is based on the macroscopic models for crowd dynamics given in chapter 3.

There is a planning layer and a regulation layer controller associated with each evacuee. The planning layer has a task of path planning. It generates a plan which closely follows the path or route assigned to evacuees by the link layer. Once a plan is generated the regulation layer's task is to execute a trajectory for an evacuee target speed which conforms with the plan.

The planning layer is assumed to have information at all times of the evacuees position, its path and target speed as assigned by link layer. It then decides what actions to take in order to evolve a plan which should stay as close as possible to the proposed path. In addition, it exchanges this information with the planning layers of neighboring evacuees to coordinate the action. Then, requests the regulation layer to implement the feedback control law to

accomplish that action.

The regulation layer implements the necessary feedback law pertaining to the request from its planning layer. It informs the planning layer of completion of the action which then plans the next action. The feedback laws compute the value for the velocity vector actuators based on the sensor inputs of the evacuees present state. The physical layer is a macroscopic model of the evacuee density represented by partial differential equations. This layer has actuator inputs and sensor outputs.

The layers we have presented are in a general form. They can be applied on different physical scenarios and different choices of control authority based on the size of the problem.

## 7.5 Four-Layer System: Scenarios

In this section we discuss the control system architecture described in previous section for our scenarios.

### 7.5.1 Subway Station

Here we describe the four layer control system architecture for the subway station:

1. **Network layer:** This layer assigns route to the evacuees in the station that could be toward an exit or taking the train to the next or final stop. Feedback control is used to identify the route associated with optimum time. A feedback assignment of routes would correspond to assessing the current state of the situation (danger, evacuee density and location).

2. **Link layer:** Here links are section of the waiting area, train cabins, stairs, or a train in a subway network. The controller in this layer assigns a path to each evacuee such that the flow is smooth from all links. It also assigns a target speed for each link to

ensure smooth flow and avoid congestion.

3. **Planning layer:** This layer decides the actions to take by evacuee in different links in order to stay as close as possible to the assigned path. It should coordinates the maneuvers of neighboring evacuees or trains such that there is no conflict between the evacuees or collision between trains as well as to ensure the safety of motion.

4. **Regulation layer:** Here the various control laws based on the control inputs given by the planning layer are implemented using the hardware in the physical layer

### 7.5.2 Airport

In the airport scenario, the four layer control system architecture are given by the following:

1. **Network layer:** In this layer the feedback controller assigns routes to the evacuees that achieve optimum time based on their location, which might be airplane door, terminal exits, parking lot exit, on road or air traffic. A feedback assignment of routes would correspond to assessing the current conditions of the system (danger, road conditions, evacuee density and location).

2. **Link layer:** In this scenario links are roads, buildings, parking lots, airplanes. The controller in this layer assigns a path to each evacuee such that the flow is smooth from all links. It also assigns a target speed for each link to ensure smooth flow and avoid congestion.

3. **Planning layer:** This layer trays to keep evacuees as close as possible to their assigned path. Coordinates the maneuvers of neighboring evacuees, airplanes, and cars such that there is no conflict between the evacuees and to ensure the safety of motion.

4. **Regulation layer:** Here the various control laws based on the control inputs given by the planning layer are implemented using the hardware in the physical layer.

## 7.6 IT Issues and Requirements

From recent disasters (e.g., Hurricane Katrina), communication systems used by early responders agencies, such as, police, firefighters, costal guard, homeland security, and others lead to a slower and uncoordinated response. This problem is due to the lack of common standards (interoperability) between the different communications systems. When the lack of compatibility coupled with the loss of infrastructure, the needed resources are not dispatched where they are most wanted. For the control architecture given by 7.3 we suggest the use of a communication architecture that deals with the different communication issues in case of an evacuation.

Network topology deals with how the network is physically arranged, physically constructed, and electrically connected. It also have a set of rules that control communications (protocols). These issues need to be address for the IES system proposed. We require that information collecting and sharing within each layer and with other layers should be robust, and the different standards must be able to communicate with each other correctly.

To physically construct a network (i.e., the way a medium connect the network nodes) there are two ways. The wired media and its technologies, and second is the wireless topology. For the IES system the wireless topology can be used as a backup medium due to bad weather conditions associated with evacuations (like hurricanes, storms, etc.). Electrical topology would be the communication path between nodes and how they can share information and resources, i.e., the way the wires or cables are actually connected. One solution could be the use of a hybrid network, which is a combination of the three basic star, bus, and ring networks. This form provides redundancy of paths that will enable other path choices in case of some link failure. To have a valuable network, it should be

**reliable** error free, error detection system

**available** 24*7, from any point, continuous stable power source

**perform** provide stable bandwidth under varying conditions, such as fixed, or dynamic bandwidth, bandwidth on demand, and brusty traffic

**secure** physical, Virtual, data

## 7.7 Feedback Control and Dynamic Modeling

The control laws executed by the regulation layers of IES will be based on the feedback control theory. It will interpret and synthesize the data collected from sensors (hazard detection sensors and video cameras) to formulate feedback control command to be delivered to the entrapped people through actuators (speakers, TV monitors, and lighted matrix displays). The system will continuously monitor, on a real-time basis, how the evacuation evolves including the development of congestion at bottleneck points, to re-direct people to other less congested locations if it is deemed necessary.

The optimal evacuation strategy will be established based on the most advanced mathematical concepts of the feedback control theory of distributed systems. To do control design in a microscopic model environment, we need to keep following each evacuee's position, its interactions with other evacuees, and give a targeted control input to each one. Obviously this increases the complexity of the system as the number of evacuees increase and it is practically impossible to control each evacuee. Therefore, the IES uses macroscopic dynamic models where the evacuees are treated as a continuum and the affected area can be divided to sections and the control commands are locally implemented.

The development of the preliminary theoretical framework for the feedback control design has been already done with controllers being designed for a simple evacuation scheme. This preliminary work is expected to provide a sound theoretical framework for the solution of the more complex evacuation problem. The feedback control design for distributed dynamics can be done either in distributed setting (pde control) or in lumped setting (ode control). The distributed feedback controllers have been developed in chapter 6 for one-dimensional space

and in (Wadoo and Kachroo, 2006b,a) for one and two-dimensional spaces respectively.

# Chapter 8

# Conclusions and Future Work

## 8.1 Conclusions

The evacuation problem is a multi-dimensional problem and this work is an attempt to model crowd dynamic flows in autonomous and controlled scenarios. A macroscopic approach is used to model this problem based on observed collective behavior in emergency conditions, where detailed design of interactions is over shadowed by group behavior.

We have presented four different two-dimensional crowd dynamic nonlinear hyperbolic systems of partial differential equations, which are derived from traffic flow one-dimensional models. Those models enable us to capture crowd dynamic behavior in normal and bottleneck situations as shown by the simulation results. The dynamics of the one-equation model are different than the other models due to its inability to capture complex wave interactions that appear when flow directions change. The second equation in the three system models introduced velocity dynamics that enabled the models to capture complex wave interactions.

The anticipation constant $C_0$ in the first system model turned out to be a very significant factor. At some values it could lead to unrealistic results, while for other values it does predict crowd flow to some extent. On the other hand, the second and third system models

weren't that sensitive to changes which may be a better choice to be used in crowd evacuation simulation.

We also presented preliminary feedback control design in the distributed setting applied on the one-equation model in one-dimensional space. Finally, we introduced an intelligent evacuation system frame work that utilizes the concepts of macroscopic modeling, feedback control, and information technology requirements.

## 8.2 Future Work

We would like to have a better understanding of the parameters involved like the anticipation factor in the first system model and how much it affects the stability of the model by conducting experimental tests; same can be said for the other models. A microscopic-to-macroscopic relation for the second model needs to be explored.

Analyzing and applying feedback control designs for the various models (linearized models included) is a challenging subject that needs to be researched more. Finally, intelligent evacuation system on a micro scale is a very challenging subject that we are interested in.

# Bibliography


Al-nasur, S. and Kachroo, P. (2006). A microscopic-to-macroscopic crowd dynamic model. In *9th International IEEE Conference on ITSC*, pages 606–611.

AlGadhi, S. and Mahmassani, H. (1991). Simulation of crowd behavior and movement: Fundamental relations and applications. *Transportation Research Record*, (1320):260–268.

Aw, A. and Rascle, M. (2000). Reconstruction of 'second order' models of traffic flow. *SIMA J. Appl. Math.*, 60:916–938.

Bando, M. (1995). Dynamical model of traffic congestion and numerical simulation. *Phy. Rev. E*, 51:1035–1042.

Blue, V. and Adler, J. (1998). Emergent fundamental pedestrian flows from cellular automata microsimulation. *Trasportation Research Record*, 1644:29–36.

Bord, T. R. (1985). Highway capacity manual. Special Report 204 TRB.

Bressan, A. (2003). An ill posed Cauchy problem for a hyperbolic system in two space dimensions. Rend. Sem. Mat. Univ. Pasova.

Burns, J. A. and Kang, S. (1991). A control problem for Burgers equation with bounded input/output. *Nonlinear Dynamics*, (2):235–262.

Chandler, R. E., Herman, R., and Montroll, E. W. (1958). Traffic dynamics; studies in car following. *Operations Research*, (6):165–184.

Christofides, P. D. (2000). *Nonlinear and Robust Control of PDE Systems*. Birkhauser.

Daganzo, C. (1995). Requiem for second-order fluid approximation to traffic flow. *Transp. Res. B*, 29B(4):277–286.

Emmanuele, D. (1995). *Partial differential equation*. Birkhauser, Boston.

Farlow, S. J. (1982). *Partial Differential Equations for Scientists and Engineers*. John Wiley, Canada.

Fruin, J. J. (1971a). Designing for pedestrians: A level of service concept. *Highway research record*, (355):1–15.

Fruin, J. J. (1971b). *Pedestrian Planning and Design*. Metropolitan Association of Urban Designers and Enviromental Planners, Inc, New York.

Gips, P. G. and Marksjo, B. (1985). A micro-simulation model for pedestrian flows. *Mathmatics and Computer in Simulation*, 27:95–105.

Greenberg, H. (1959). An analysis of traffic flow. *Operationa Research*, 7:78–85.

Greenshields, B. D. (1935). A study in highway capacity. *Highway Research Board*, 14:458.

Griffith, D. A. (1982). The critical problem of hurricane evacuation and alternative solutions. In *CAITR Paper 15th Conf. of Australian Institutes of Trasport Research*, volume 12, pages 990–994.

Haberman, R. (1977). *Mathematical Models*. Prentice-Hall, New Jersey.

Helbing, D. (1991). A mathmatical model for the behavior of pedestrians. *Behavioral Science*, 36:298–310.

Helbing, D. (1992). A fluid-dynamic model for the movement of pedestrians. *Complex Systems*, 6:391–415.

Helbing, D. (1994). Computer simulation of pedestrian dynamics and trail formation. *Evolution of Natural Science*, 230:229–234.

Helbing, D., Farkas, I. J., Molnor, P., and Vicsek, T. (2002). Simulation of pedestrian crowd in normal and evacuation situations. *Pedestrian and Evacuation Dynamics Journal*, pages 21–58.

Helbing, D. and Molnar, P. (1995). Social force model forpedestrian dynamics. *Physical Review E*, 51:4282–4286.

Helbing, D. and Molnar, P. (1997). Self-Organization Phenomena in Pedestrian Crowds. *ArXiv Condensed Matter e-prints*, pages 569–577.

Helbing, D. and Vicsek, T. (1999). Optimal self-organization. *New Journal of Physics*, (1):13.1–13.17.

Henderson, L. F. (1974). On the fluid mechanic of human crowd motion. *Transportation research*, 8:509–515.

Hoogendoorn, S. P. (2001). Pedestrian flow behavior: Theory and applications. Technical report, Transportation and Traffic Engineering Section, Delft University of Technology.

Hoogendoorn, S. P. and Bovy, P. H. L. (2004). Pedestrian route-choice and activity scheduling theory and models. *Transpn. Res. B.*, 38:169–190.

Hughes, R. L. (2002). A continuum theory for the flow of pedestrians. *Tranportation Research Part B*, 36:507–535.

Hughes, R. L. (2003). The flow of human crowds. *Annu. Rev. Fluid*, 35:169–182.

Kachroo, P. and Ozbay, K. (1999). *Feedback Control Theory For Dynamic Traffic Assignment*. Springer-Verlag.

Kevorkian, J. (2000). *Partial differential equation*. Springer, NY., 2ed edition edition.

Khalil, H. K. (1996). *Nonlinear Systems*. Prentice Hall.

Kröner, D. (1997). *Numerical Schemes for Conservation Laws*. Wiley, Teubner.

Kruzkov, S. N. (1970). First order quasi-linear equations in several independent variables. *Math. USSR Sbornik*, 10:217–243.

Lax, P. (1972). Hyperbolic systems of conservation laws and mathmatical theory of shock waves. In *SIMA Regional Conf. Series in Appl. Math*, number 11.

Leveque, R. J. (2002). *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, UK.

Lighthill, M. J. and Whitham, G. B. (1955). On kinematic waves. i:flow movement in long rivers. ii:a theory of traffic on long crowded roods. In *Proc. Royal Soc.*, number A229, pages 281–345.

Liska, R. and Wendroff, B. (1997). 2d shallow water equations by composite schemes. *International Journal for Numerical Methods in Fluids*, 30(4):461–479.

Lovas, G. G. (1994). Modeling and simulation of pedestrian traffic flow. *Tranportation Research*, 28B:429–443.

Lovas, G. G. (1998). On the importance of building evacuation system components. *IEEE Trans. On Eng. Management*, 45(2):181–191.

May, A. C. (1990). *Traffic Flow Fundemental*. Prentice Hall, New Jersey.

Musha, T. and Higuchi, H. (1978). Traffic current fluction and the burgers. *Japanese Jounal of Applied Physics*, 17:811–816.

Nagel, K. (1996). Partical hopping models and traffic flow theory. *Phy. Rev. E,*, (53):4655–4672.

Ockendon, J., Howison, S., Lacey, A., and Movchan, A. (2003). *Applied partial differential equations*. Oxford, NY., revised edition edition.

Okazaki, S. (1979). A study of pedestrian movement in architectural space, part 1: Pedestrian movement by the application on magnetic models. *Trans. of A.I.J.*, (283):111–119.

Okazaki, S. and Matsushita, S. (1993). A study of simulation model for pedestrian movement with evacuation and queuing. In *Proceeding of the International Confference on Engineering for Crowd Safety*, volume 12, pages 271–280.

Okumura, T., Takasu, N., Ishimatsu, S., Miyanoki, S., Mitsuhashi, A., Kumada, K., Tanaka, K., and Hinohara, S. (1996). Report on 640 Victims of the Tokyo Subway Sarin Attack. *Annals of Emergency Medicine*, 28(2):129–135.

Payne, H. J. (1971). Models of freeway traffic and control. In *Math. Models Publ. Sys. Simul. Council Proc.*, number 28, pages 51–61.

Rascle, M. (2002). An improved macroscopic model of traffic flow: Derivation and lonks with the lighthill-whitham model. *Math. and Computer Modelling*, (35):581–590.

Ray, W. H. (1981). *Advanced Process Control*. McGraw-Hill, New York.

Renardy, M. and Rogers, R. (2004). *Introduction to Partial differential equation*. Springer, NY., 2ed edition edition.

Richards, P. I. (1956). Shockwaves on the highway. *Operationa Research*, 4:42–51.

Smith, R. A. and (Eds.), J. F. D. (1993). Engineering for crowd safety. In *Elsevier, Amsterdam*, page 442.

Southworth, F., Chin, S.-M., and Cheng, P. D. (1989). A telemetric monitoring and analysis system for use during large scale population evacuations. In *Road Traffic Monitoring, Second International Conference*, pages 99–103.

Sugiyama, Y., Nakayama, A., and Hasebe, K. (2002). 2-dimensional optimal velocity methods for granular flow and pedestrian dynamics. *Pedestrian and Evacuation Dynamics*, pages 155–160.

Thompson, P. and Marchant, E. (1995a). A computer model the evacuation of large building populations. *Fire and Safety Journal*, (24):131–148.

Thompson, P. and Marchant, E. (1995b). Testing and application of the computer model 'simulex'. *Fire and Safty Journal*, (24):149–166.

Toro, E. (1999). *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, Germany, 2nd edition edition.

Varaiya, P. (1993). Smart cars on smart roads: Problems of control. *IEEE Transactions On Automatic Control*, 38(2).

Wadoo, S. and Kachroo, P. (2006a). Feedback control design and stability analysis of one dimensional evacuation system. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 618–623, Canada.

Wadoo, S. and Kachroo, P. (2006b). Feedback control design and stability analysis of two dimensional evacuation system. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 1108–1113, Canada.

Watts, J. (1987). Computer models for evacuation analysis. *Fire and Safety Journal*, 12:1237–245.

Whitham, G. (1974). *Linear and Nonlinear Waves*. John Wiley, NY.

Wigan, M. (1993). Why should we worry about pedestrians? In *CAITR Paper 15th Conf. of Australian Institutes of Trasport Research*, page 12.

Zhang, H. (1998). A theory of nonequilibrium traffic flow. *Transportation Research B*, 32:485–498.

Zhang, H. (2002). A non-equilibrium traffic model deviod of gas-like behaviour. *Transpn. Res.B*, 36B:275–290.

# Chapter 9

# Appendix A

## 9.1 Chapter 3 Matlab Program Code

### 9.1.1 One-equation Model

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% One-equation Model, FORCE scheme %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
D_t=0.04;  D_x=0.2;   D_y=0.2;  vf=1.36/2; vff=0*1.36; max=80;
Rm=1;max_t=2400; x=[0:D_x:max];y=[0:D_y:max]; N=length(x);
%   To find the I.C
y1=0;
for yy=0:D_y:max x1=0;y1=y1+1;
for xx=0:D_x:max x1=x1+1;
R_new(y1,x1)=1*exp(-1/2*((xx-10)^2+(yy-65)^2));
end
end
Rin=R_new;                                          %I.C

for t=1:max_t         % time loop
% velocity direction
    if t==max_t/4; vf=0*1.36;vff=-1.36/2;end
    if t==2*max_t/4; vf=-1.36/2;vff=0*1.36;end
    if t==3*max_t/4; vf=0*1.36;vff=1.36/2;end
% x and y loops
    i=2:N+1;
```

```
    j=2:N+1;
    R(j,i)=R_new(1:N,1:N);
    % Goust Cell B.C Down, Top, Right, Left
    R(1,i)=R(2,i); R(N+2,i)=R(N+1,i); R(j,1)=R(j,2); R(j,N+2)=R(j,N+1);
    for j=2:N+1
        for i=2:N+1
            [FL]=FORCE(R(j,i),R(j,i-1),vf,D_t,D_x,Rm);
            [FR]=FORCE(R(j,i+1),R(j,i),vf,D_t,D_x,Rm);
            R_04(j,i)=R(j,i)-D_t/(D_x)*[FR-FL];
        end
    end
    i=2:N+1;
    j=2:N+1;
    R_04(1,i)=R_04(2,i);R_04(N+2,i)=R_04(N+1,i);R_04(j,1)=R_04(j,2);...
    R_04(j,N+2)=R_04(j,N+1);
    for j=2:N+1
        for i=2:N+1
            [GL]=FORCE(R_04(j,i),R_04(j-1,i),vff,D_t,D_y,Rm);
            [GR]=FORCE(R_04(j+1,i),R_04(j,i),vff,D_t,D_y,Rm);
            R_new(j-1,i-1)=R_04(j,i)-D_t/(D_y)*[GR-GL];
        end
    end
end

function [F]=FORCE(Rr,Rl,Vf,Dt,Dx,Rm);
    F_R=Rr*Vf*(1-Rr/Rm);
    F_L=Rl*Vf*(1-Rl/Rm);
    q_r=Rr;
    q_l=Rl;
    Q_RI=0.5*(q_l+q_r)+0.5*Dt/Dx*(F_L-F_R);
    F_LF=0.5*(F_L+F_R)+0.5*Dx/Dt*(q_l-q_r);
    F_RI=Q_RI*Vf*(1-Q_RI/Rm);
    F=0.5*(F_LF+F_RI);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% One-equation Model, FORCE scheme %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 D_t=0.04;  D_x=0.2;   D_y=0.2; max=40;
 time=450; row_m=1; Vm=1.36; Vmm=1.36;
 x=[0:D_x:max];y=[0:D_y:max];
 N=length(x);
y1=0; for yy=0:D_y:max
```

```
    x1=0;y1=y1+1;
    for xx=0:D_x:max
        x1=x1+1;
        row_ic(y1,x1)=1*exp(-1/12*((xx-20)^2+(yy-20)^2));
    end
end row=row_ic;
%%%%%%%%%%%%%%%
Nx=length(x); Ny=length(y); for t=1:time
    if t==180 ; Vm=-1.36;Vmm=-1.36;end
    for i=2:Ny
        for j=2:Nx
%Setting B.C, ghost cells and speed=0 at the boundary
% and calculating the fluxes
            if j==2;   %left B.C
                row(i,j-1)=row(i,j);
                f(i,j-1)=row(i,j-1)*-Vm*(1-row(i,j-1)./row_m);
            else
                f(i,j-1)=row(i,j-1)*Vm*(1-row(i,j-1)./row_m);
            end

            if j==Nx;  % Right B.C
                row(i,j+1)=row(i,j);
                f(i,j+1)=row(i,j+1)*-Vm*(1-row(i,j+1)./row_m);
            else
                f(i,j+1)=row(i,j+1)*Vm*(1-row(i,j+1)./row_m);
            end

            if i==2;  %bottom
                row(i-1,j)=row(i,j);
                g(i-1,j)=row(i-1,j)*-Vmm*(1-row(i-1,j)./row_m);
            else
                g(i-1,j)=row(i-1,j)*Vmm*(1-row(i-1,j)./row_m);
            end

            if i==Ny;  % Top
                row(i+1,j)=row(i,j);
                g(i+1,j)=row(i+1,j)*-Vmm*(1-row(i+1,j)./row_m);
            else
                g(i+1,j)=row(i+1,j)*Vmm*(1-row(i+1,j)./row_m);
            end
 %   Updating the density "row"
            row_updated(i,j)=(row(i-1,j)+row(i+1,j)+row(i,j-1)+...
```

```
            row(i,j+1))/4-(D_t/(2*D_x)*(f(i,j+1)-f(i,j-1))+...
            D_t/(2*D_y)*(g(i+1,j)-g(i-1,j)));
        end
    end
    row(2:Ny,2:Nx)=row_updated(2:Ny,2:Nx);
    r(1:Ny-1,1:Nx-1)=row(2:Ny,2:Nx);
end
```

### 9.1.2 First System Model

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% First System Model by Roe's scheme %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Roe clear all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     Initial condition
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 D_t=0.04;  D_x=0.2;   D_y=0.2; tow=0.2; vf=1.36; max=20;Rm=1;
x=[0:D_x:max];y=[0:D_y:max]; N=length(x); y1=0; for yy=0:D_y:max
x1=0;y1=y1+1; for xx=0:D_x:max x1=x1+1;
R_new(y1,x1)=1*exp(-1/4*((xx-16)^2+(yy-4)^2));
U_new(y1,x1)=-vf*(1-R_new(y1,x1)/Rm);
V_new(y1,x1)=vf*(1-R_new(y1,x1)/Rm); end end
Rin=R_new;Uin=U_new;Vin=V_new;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for t=1:150
    i=2:N+1;
    j=2:N+1;
    R(j,i)=R_new(1:N,1:N);
    U(j,i)=U_new(1:N,1:N);
    V(j,i)=V_new(1:N,1:N);
    clear R_new U_new V_new Qold f_s FL FR GL GR g_s Qold_04 Qold_34...
        Q_new Q_new_04 Q_new_34 R_04 U_04 V_04 R_34 U_34 V_34
    % Goust Cell B.C Down, Top, Right, Left
    R(1,i)=R(2,i);R(N+2,i)=R(N+1,i);   R(j,1)=R(j,2); R(j,N+2)=R(j,N+1);
    U(1,i)=U(2,i);U(N+2,i)=U(N+1,i);   U(j,1)=-U(j,2);U(j,N+2)=-U(j,N+1);
    V(1,i)=-V(2,i);V(N+2,i)=-V(N+1,i);   V(j,1)=V(j,2);V(j,N+2)=V(j,N+1);

    for j=2:N+1
        for i=2:N+1
```

```
[FL]=Approx_Roe_PW(R(j,i),R(j,i-1),U(j,i),U(j,i-1),V(j,i),V(j,i-1),0);
[FR]=Approx_Roe_PW(R(j,i+1),R(j,i),U(j,i+1),U(j,i),V(j,i+1),V(j,i),0);

        Qold_04=[R(j,i);R(j,i)*U(j,i);R(j,i)*V(j,i)];
        q_new_04=Qold_04-D_t/(D_x)*[FR-FL];
        %UR=vf*(1-R(j,i)/1)*sign(U(j,i));
        r=q_new_04(1);
        u=q_new_04(2)/q_new_04(1);
        v=q_new_04(3)/q_new_04(1);
        UR=vf*(1-r/1)*sign(u);
        f_s=[0;r*(UR-u)/tow;0];%u
        Q_new_04=q_new_04+D_t*f_s;

        R_04(j,i)=Q_new_04(1);
        U_04(j,i)=Q_new_04(2)/Q_new_04(1);
        V_04(j,i)=Q_new_04(3)/Q_new_04(1);
    end
end
i=2:N+1;
j=2:N+1;
R_04(1,i)=R_04(2,i); R_04(N+2,i)=R_04(N+1,i); R_04(j,1)=R_04(j,2);
R_04(j,N+2)=R_04(j,N+1);U_04(1,i)=U_04(2,i);U_04(N+2,i)=U_04(N+1,i);
U_04(j,1)=-U_04(j,2);U_04(j,N+2)=-U_04(j,N+1); V_04(1,i)=-V_04(2,i);
V_04(N+2,i)=-V_04(N+1,i);V_04(j,1)=V_04(j,2);V_04(j,N+2)=V_04(j,N+1);
for j=2:N+1
    for i=2:N+1
        [GL]=Approx_Roe_PW(R_04(j,i),R_04(j-1,i),V_04(j,i),...
            V_04(j-1,i),U_04(j,i),U_04(j-1,i),1);
        [GR]=Approx_Roe_PW(R_04(j+1,i),R_04(j,i),V_04(j+1,i),...
            V_04(j,i),U_04(j+1,i),U_04(j,i),1);

        Qold_34=[R_04(j,i);R_04(j,i)*U_04(j,i);R_04(j,i)*V_04(j,i)];
        q_new_34=Qold_34-D_t/(D_x)*[GR-GL];
       % VR=vf*(1-R_04(j,i)/1)*sign(V_04(j,i));
        r=q_new_34(1);
        u=q_new_34(2)/q_new_04(1);
        v=q_new_34(3)/q_new_34(1);
        VR=vf*(1-r/1)*sign(v);
        g_s=[0;0;r*(VR-v)/tow]; %v
        Q_new_34=q_new_34+D_t*g_s;

        R_new(j-1,i-1)=Q_new_34(1);
```

```
            U_new(j-1,i-1)=Q_new_34(2)/Q_new_34(1);
            V_new(j-1,i-1)=Q_new_34(3)/Q_new_34(1);
        end
    end
end

function [F]=Approx_Roe_PW(Rr,Rl,Ur,Ul,Vr,Vl,M); c0=0.8;tow=0.2;
u=(Ur*(Rr)^(1/2)+Ul*(Rl)^(1/2))/((Rr)^(1/2)+(Rl)^(1/2));
v=(Vr*(Rr)^(1/2)+Vl*(Rl)^(1/2))/((Rr)^(1/2)+(Rl)^(1/2));

eig=[-c0+u  0  0; 0   u    0; 0  0 c0+u]; if M==0
    RR=[1   0   1; c0+u   0 -c0+u; v  1   v];
    F_R=[Rr*Ur; Rr*Ur^2+c0^2*Rr; Rr*Ur*Vr];
    F_L=[Rl*Ul; Rl*Ul^2+c0^2*Rl; Rl*Ul*Vl];
    q_r=[Rr; Rr*Ur; Rr*Vr];
    q_l=[Rl; Rl*Ul; Rl*Vl];
else
    RR=[1   0   1; v  1   v; c0+u   0 -c0+u];
    F_R=[Rr*Ur; Rr*Ur*Vr; Rr*Ur^2+c0^2*Rr];
    F_L=[Rl*Ul; Rl*Ul*Vl; Rl*Ul^2+c0^2*Rl];
    q_r=[Rr; Rr*Vr; Rr*Ur];
    q_l=[Rl; Rl*Vl; Rl*Ul];
end

eps1=max([0, eig(1,1)-(c0+Ul), (c0+Ur)-eig(1,1)]);
Lamda_1=max(eps1,abs(eig(1,1)));
eps2=max([0,eig(2,2)-(Ul),(Ur)-eig(2,2)]);
Lamda_2=max(eps2,abs(eig(2,2)));
eps3=max([0,eig(3,3)-(-c0+Ul),(-c0+Ur)-eig(3,3)]);
Lamda_3=max(eps3,abs(eig(3,3))); eig_f=[Lamda_1  0  0; 0   Lamda_2
0; 0  0 Lamda_3]; Qa=RR*abs(eig_f)*inv(RR); F=0.5*[F_R +
F_L]-0.5*Qa*[q_r-q_l];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% First System Model by FORCE scheme %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ForcedPW clear all
%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Initial condition   %
%%%%%%%%%%%%%%%%%%%%%%%%%%
D_t=0.04;  D_x=0.2;   D_y=0.2; tow=0.2; vf=-1.36; vff=1.36;
max=20;Rm=1; c=0.8; x=[0:D_x:max];y=[0:D_y:max]; N=length(x); y1=0;
```

```
for yy=0:D_y:max x1=0;y1=y1+1; for xx=0:D_x:max x1=x1+1;
R_new(y1,x1)=1*exp(-1/4*((xx-16)^2+(yy-4)^2));
U_new(y1,x1)=-vf*(1-R_new(y1,x1)/Rm);
V_new(y1,x1)=vff*(1-R_new(y1,x1)/Rm); end end
Rin=R_new;Uin=U_new;Vin=V_new;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for t=1:150
    i=2:N+1;
    j=2:N+1;
    R(j,i)=R_new(1:N,1:N);
    U(j,i)=U_new(1:N,1:N);
    V(j,i)=V_new(1:N,1:N);
    clear R_new U_new V_new Qold f_s FL FR GL GR g_s Qold_04 Qold_34...
        Q_new Q_new_04 Q_new_34 R_04 U_04 V_04 R_34 U_34 V_34
    % Goust Cell B.C Down, Top, Right, Left
    R(1,i)=R(2,i); R(N+2,i)=R(N+1,i);   R(j,1)=R(j,2); R(j,N+2)=R(j,N+1);
    U(1,i)=U(2,i);U(N+2,i)=U(N+1,i);   U(j,1)=-U(j,2); U(j,N+2)=-U(j,N+1);
    V(1,i)=-V(2,i);V(N+2,i)=-V(N+1,i);   V(j,1)=V(j,2); V(j,N+2)=V(j,N+1);

    for j=2:N+1
        for i=2:N+1
            [FL]=FORCE(R(j,i),R(j,i-1),U(j,i),U(j,i-1),V(j,i),...
                V(j,i-1),D_t,D_x,c,0);
            [FR]=FORCE(R(j,i+1),R(j,i),U(j,i+1),U(j,i),V(j,i+1),...
                V(j,i),D_t,D_x,c,0);

            Qold_04=[R(j,i);R(j,i)*U(j,i);R(j,i)*V(j,i)];
            q_new_04=Qold_04-D_t/(D_x)*[FR-FL];

            Rr=q_new_04(1);
            Uu=q_new_04(2)/(q_new_04(1));

            UR=vf*(1-Rr/Rm);%*sign(Uu);

            f_s=[0;Rr*(UR-Uu)/tow;0];
            Q_new_04=q_new_04+D_t*f_s;

            R_04(j,i)=Q_new_04(1);
            U_04(j,i)=Q_new_04(2)/(Q_new_04(1));
            V_04(j,i)=Q_new_04(3)/(Q_new_04(1));

        end
```

```
    end
    i=2:N+1;
    j=2:N+1;
    R_04(1,i)=R_04(2,i);   R_04(N+2,i)=R_04(N+1,i);
    R_04(j,1)=R_04(j,2);   R_04(j,N+2)=R_04(j,N+1);
    U_04(1,i)=U_04(2,i);   U_04(N+2,i)=U_04(N+1,i);
    U_04(j,1)=-U_04(j,2);   U_04(j,N+2)=-U_04(j,N+1);
    V_04(1,i)=-V_04(2,i);   V_04(N+2,i)=-V_04(N+1,i);
    V_04(j,1)=V_04(j,2);   V_04(j,N+2)=V_04(j,N+1);
    for j=2:N+1
        for i=2:N+1

            [GL]=FORCE(R_04(j,i),R_04(j-1,i),U_04(j,i),U_04(j-1,i),...
                V_04(j,i),V_04(j-1,i),D_t,D_x,c,1);
            [GR]=FORCE(R_04(j+1,i),R_04(j,i),U_04(j+1,i),U_04(j,i),...
                V_04(j+1,i),V_04(j,i),D_t,D_x,c,1);

            Qold_34=[R_04(j,i);R_04(j,i)*U_04(j,i);R_04(j,i)*V_04(j,i)];
            q_new_34=Qold_34-D_t/(D_x)*[GR-GL];

            Rn=q_new_34(1);
            Vn=q_new_34(3)/(q_new_34(1));

            VR=vff*(1-Rn/Rm);%*sign(Vn);

            g_s=[0;0;Rn*(VR-Vn)/tow];
            Q_new_34=q_new_34+D_t*g_s;

            R_new(j-1,i-1)=Q_new_34(1);
            U_new(j-1,i-1)=Q_new_34(2)/(Q_new_34(1));
            V_new(j-1,i-1)=Q_new_34(3)/(Q_new_34(1));
        end
    end
end

function [F]=FORCE(Rr,Rl,Ur,Ul,Vr,Vl,Dt,Dx,c0,M);

if M==0
    F_R=[Rr*Ur; Rr*Ur^2+c0^2*Rr; Rr*Ur*Vr];
    F_L=[Rl*Ul; Rl*Ul^2+c0^2*Rl; Rl*Ul*Vl];
    q_r=[Rr; Rr*Ur; Rr*Vr];
    q_l=[Rl; Rl*Ul; Rl*Vl];
```

```
    F_LF=0.5*(F_L+F_R)+0.5*Dx/Dt*(q_l-q_r);
    Q_RI=0.5*(q_l+q_r)+0.5*Dt/Dx*(F_L-F_R);
    r=Q_RI(1);
    u=Q_RI(2)/(r);
    v=Q_RI(3)/(r);
    F_RI=[r*u; r*u^2+c0^2*r; r*u*v];
    F=0.5*(F_LF+F_RI);
else
    G_R=[Rr*Vr; Rr*Ur*Vr; Rr*Vr^2+c0^2*Rr];
    G_L=[Rl*Vl; Rl*Ul*Vl; Rl*Vl^2+c0^2*Rl];
    q_r=[Rr; Rr*Ur; Rr*Vr];
    q_l=[Rl; Rl*Ul; Rl*Vl];

    G_LF=0.5*(G_L+G_R)+0.5*Dx/Dt*(q_l-q_r);   %Lax
    Q_RI=0.5*(q_l+q_r)+0.5*Dt/Dx*(G_L-G_R);
    r=Q_RI(1);
    u=Q_RI(2)/(r);
    v=Q_RI(3)/(r);
    G_RI=[r*v; r*u*v; r*v^2+c0^2*r];   % Richm...
    F=0.5*(G_LF+G_RI);                   % Force
end
```

### 9.1.3 Second System Model

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Second System Model by FORCE scheme %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ForcedAW clear all
%%%%%%%%%%%%%%%%%%%%%%%%%%
%     Initial condition %
%%%%%%%%%%%%%%%%%%%%%%%%%%
D_t=0.04;  D_x=0.2;   D_y=0.2; tow=0.2; vf=1.36; vff=1.36;
max=10;b=13; g=1;Rm=1; x=[0:D_x:max];y=[0:D_y:max]; N=length(x);
y1=0; for yy=0:D_y:max x1=0;y1=y1+1; for xx=0:D_x:max x1=x1+1;
R_new(y1,x1)=1*exp(-1*((xx-5)^2+(yy-5)^2));
U_new(y1,x1)=-vf*(1-R_new(y1,x1)/Rm);
V_new(y1,x1)=vff*(1-R_new(y1,x1)/Rm); end end
Rin=R_new;Uin=U_new;Vin=V_new;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for t=1:100
```

```
i=2:N+1;
j=2:N+1;
R(j,i)=R_new(1:N,1:N);
U(j,i)=U_new(1:N,1:N);
V(j,i)=V_new(1:N,1:N);
clear R_new U_new V_new Qold f_s FL FR GL GR g_s Qold_04 Qold_34...
    Q_new Q_new_04 Q_new_34 R_04 U_04 V_04 R_34 U_34 V_34
% Goust Cell B.C Down, Top, Right, Left
R(1,i)=R(2,i);R(N+2,i)=R(N+1,i);   R(j,1)=R(j,2);R(j,N+2)=R(j,N+1);
U(1,i)=U(2,i); U(N+2,i)=U(N+1,i);   U(j,1)=-U(j,2);U(j,N+2)=-U(j,N+1);
V(1,i)=-V(2,i);V(N+2,i)=-V(N+1,i);   V(j,1)=V(j,2);V(j,N+2)=V(j,N+1);

for j=2:N+1
    for i=2:N+1
        [FL]=FORCE(R(j,i),R(j,i-1),U(j,i),U(j,i-1),V(j,i),...
            V(j,i-1),D_t,D_x,g,b,0);
        [FR]=FORCE(R(j,i+1),R(j,i),U(j,i+1),U(j,i),V(j,i+1),...
            V(j,i),D_t,D_x,g,b,0);

        P1=U(j,i)*R(j,i)^(g+1)/(b-R(j,i)^(g+1));
        P2=V(j,i)*R(j,i)^(g+1)/(b-R(j,i)^(g+1));
        Qold_04=[R(j,i);R(j,i)*(U(j,i)+P1);R(j,i)*(V(j,i)+P2)];
        q_new_04=Qold_04-D_t/(D_x)*[FR-FL];

        R_4(j,i)=q_new_04(1);
        U_4(j,i)=q_new_04(2)*(b-q_new_04(1)^(g+1))/...
            (b*q_new_04(1)+0.000001);
        UR=vf*(1-R_4(j,i)/Rm);
        f_s=[0;R_4(j,i)*(UR-U_4(j,i))/tow;0];
        Q_new_04=q_new_04+D_t*f_s;

        R_04(j,i)=Q_new_04(1);
        U_04(j,i)=Q_new_04(2)*(b-R_04(j,i)^(g+1))/...
            (b*R_04(j,i)+0.000001);
        V_04(j,i)=Q_new_04(3)*(b-R_04(j,i)^(g+1))/...
            (b*R_04(j,i)+0.000001);
    end
end
i=2:N+1;
j=2:N+1;
R_04(1,i)=R_04(2,i);   R_04(N+2,i)=R_04(N+1,i);
R_04(j,1)=R_04(j,2);   R_04(j,N+2)=R_04(j,N+1);
```

```
    U_04(1,i)=U_04(2,i);   U_04(N+2,i)=U_04(N+1,i);
    U_04(j,1)=-U_04(j,2);   U_04(j,N+2)=-U_04(j,N+1);
    V_04(1,i)=-V_04(2,i);   V_04(N+2,i)=-V_04(N+1,i);
    V_04(j,1)=V_04(j,2);   V_04(j,N+2)=V_04(j,N+1);

    for j=2:N+1
        for i=2:N+1
            [GL]=FORCE(R_04(j,i),R_04(j-1,i),U_04(j,i),U_04(j-1,i),...
                V_04(j,i),V_04(j-1,i),D_t,D_x,g,b,1);
            [GR]=FORCE(R_04(j+1,i),R_04(j,i),U_04(j+1,i),U_04(j,i),...
                V_04(j+1,i),V_04(j,i),D_t,D_x,g,b,1);

            pp1=U_04(j,i)*R_04(j,i)^(g+1)/(b-R_04(j,i)^(g+1));
            pp2=V_04(j,i)*R_04(j,i)^(g+1)/(b-R_04(j,i)^(g+1));
            Qold_34=[R_04(j,i);R_04(j,i)*(U_04(j,i)+pp1);R_04(j,i)...
                *(V_04(j,i)+pp2)];
            q_new_34=Qold_34-D_t/(D_x)*[GR-GL];

            R_n(j,i)=q_new_34(1);
            V_n(j,i)=q_new_34(3)*(b-q_new_34(1)^(g+1))/...
                (b*q_new_34(1)+0.000001);
            VR=vff*(1-R_n(j,i)/1);%*sign(V_n(j,i));
            g_s=[0;0;R_n(j,i)*(VR-V_n(j,i))/tow];
            Q_new_34=q_new_34+D_t*g_s;

            R_new(j-1,i-1)=Q_new_34(1);
            U_new(j-1,i-1)=Q_new_34(2)*(b-Q_new_34(1)^(g+1))/...
                (b*Q_new_34(1)+0.000001);
            V_new(j-1,i-1)=Q_new_34(3)*(b-Q_new_34(1)^(g+1))/...
                (b*Q_new_34(1)+0.000001);
        end
    end
end function [F]=FORCE(Rr,Rl,Ur,Ul,Vr,Vl,Dt,Dx,g,b,M);
Pl_1=Rl^(g+1)*Ul/(b-Rl^(g+1)); Pr_1=Rr^(g+1)*Ur/(b-Rr^(g+1));
Pl_2=Rl^(g+1)*Vl/(b-Rl^(g+1)); Pr_2=Rr^(g+1)*Vr/(b-Rr^(g+1)); if
M==0
    F_R=[Rr*Ur; Rr*Ur*(Ur+Pr_1); Rr*Ur*(Vr+Pr_2)];
    F_L=[Rl*Ul; Rl*Ul*(Ul+Pl_1); Rl*Ul*(Vl+Pl_2)];
    q_r=[Rr; Rr*(Ur+Pr_1); Rr*(Vr+Pr_2)];
    q_l=[Rl; Rl*(Ul+Pl_1); Rl*(Vl+Pl_2)];

    F_LF=0.5*(F_L+F_R)+0.5*Dx/Dt*(q_l-q_r);
```

```
    Q_RI=0.5*(q_l+q_r)+0.5*Dt/Dx*(F_L-F_R);
    r=Q_RI(1);
    u=Q_RI(2)*(b-r^(g+1))/(b*r+0.000001);
    v=Q_RI(3)*(b-r^(g+1))/(b*r+0.000001);
    p1=r^(g+1)*u/(b-r^(g+1));
    p2=r^(g+1)*v/(b-r^(g+1));
    F_RI=[r*u; r*u*(u+p1); r*u*(v+p2)];
    F=0.5*(F_LF+F_RI);
else
    G_R=[Rr*Vr; Rr*Vr*(Ur+Pr_1); Rr*Vr*(Vr+Pr_2)];
    G_L=[Rl*Vl; Rl*Vl*(Ul+Pl_1); Rl*Vl*(Vl+Pl_2)];
    q_r=[Rr; Rr*(Ur+Pr_1); Rr*(Vr+Pr_2)];
    q_l=[Rl; Rl*(Ul+Pl_1); Rl*(Vl+Pl_2)];

    G_LF=0.5*(G_L+G_R)+0.5*Dx/Dt*(q_l-q_r);   %Lax
    Q_RI=0.5*(q_l+q_r)+0.5*Dt/Dx*(G_L-G_R);
    rr=Q_RI(1);
    uu=Q_RI(2)*(b-rr^(g+1))/(b*rr+0.000001);
    vv=Q_RI(3)*(b-rr^(g+1))/(b*rr+0.000001);
    pp1=rr^(g+1)*uu/(b-rr^(g+1));
    pp2=rr^(g+1)*vv/(b-rr^(g+1));
    G_RI=[rr*vv; rr*vv*(uu+pp1); rr*vv*(vv+pp2)];   % Richm...
    F=0.5*(G_LF+G_RI);                    % Force
end
```

### 9.1.4 Third System Model

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Third System Model by Lax scheme    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ForcedZhang
clear all
%%%%%%%%%%%%%%%%%%%%%%%%%
%   Initial condition  %
%%%%%%%%%%%%%%%%%%%%%%%%%
 D_t=0.04;D_x=0.2; D_y=0.2; max=20;time=150; row_m=1;Vm=1.36;Vmm=-1.36;
 x=[0:D_x:max];y=[0:D_y:max];
 N=length(x);
y1=0; for yy=0:D_y:max
    x1=0;y1=y1+1;
    for xx=0:D_x:max
```

```
        x1=x1+1;
        R_new(y1,x1)=1*exp(-1/2*((xx-10)^2+(yy-10)^2));
        V_new(y1,x1)=Vm*(1-R_new(y1,x1)/row_m);
        U_new(y1,x1)=Vmm*(1-R_new(y1,x1)/row_m);
    end
end Ri=R_new;Ui=U_new;Vi=V_new;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for t=1:time
    R(2:N+1,2:N+1)=R_new(1:N,1:N);
    V(2:N+1,2:N+1)=V_new(1:N,1:N);
    U(2:N+1,2:N+1)=U_new(1:N,1:N);
    for i=2:N+1
        for j=2:N+1
%    Setting B.C, ghost cells and speed=0 at the boundary  & Calculating
%    the fluxes
            if j==2;   %left B.C
                R(i,j-1)=R(i,j);
                V(i,j-1)=-V(i,j);
                U(i,j-1)=U(i,j);
            end
            if j==N+1;  % Right B.C
                R(i,j+1)=R(i,j);
                V(i,j+1)=-V(i,j);
                U(i,j+1)=U(i,j);
            end
            if i==2;  %bottom
                R(i-1,j)=R(i,j);
                V(i-1,j)=V(i,j);
                U(i-1,j)=-U(i,j);
            end
            if i==N+1;  % Top
                R(i+1,j)=R(i,j);
                V(i+1,j)=V(i,j);
                U(i+1,j)=-U(i,j);
            end

            [Q_new]=FORCE(R(i,j-1),R(i,j+1),R(i-1,j),R(i+1,j),...
                V(i,j-1),V(i,j+1),V(i-1,j),V(i+1,j),U(i,j-1),U(i,j+1),...
                U(i-1,j),U(i+1,j),D_t,D_y,D_x,Vm,Vmm,row_m);

            R_new(i-1,j-1)=Q_new(1);
            V_r_n=Vm*(1-Q_new(1)/row_m);
```

```
            U_r_n=Vmm*(1-Q_new(1)/row_m);
            V_new(i-1,j-1)=Q_new(2)/Q_new(1)+V_r_n;
            U_new(i-1,j-1)=Q_new(3)/Q_new(1)+U_r_n;
        end
    end

end

function [q_new]=FORCE(RL,RR,RB,RU,VL,VR,VB,VU,UL,UR,UB,UU,Dt,...
    Dy,Dx,vf,vff,Rm);
            Vl=vf*(1-RL/Rm);
            Ul=vff*(1-RL/Rm);
            Vr=vf*(1-RR/Rm);
            Ur=vff*(1-RR/Rm);
            Vb=vf*(1-RB/Rm);
            Ub=vff*(1-RB/Rm);
            Vu=vf*(1-RU/Rm);
            Uu=vff*(1-RU/Rm);

            ql=[RL;RL*(VL-Vl);RL*(UL-Ul)];
            qr=[RR;RR*(VR-Vr);RR*(UR-Ur)];
            qb=[RB;RB*(VB-Vb);RB*(UB-Ub)];
            qu=[RU;RU*(VU-Vu);RU*(UU-Uu)];

            fl=[RL*VL;RL*VL*(VL-Vl);RL*VL*(UL-Ul)];
            fr=[RR*VR;RR*VR*(VR-Vr);RR*VR*(UR-Ur)];

            gb=[RB*UB;RB*UB*(VB-Vb);RB*UB*(UB-Ub)];
            gu=[RU*UU;RU*UU*(VU-Vu);RU*UU*(UU-Uu)];
 %   Updatting the density "R"
            q_new=(ql+qr+qb+qu)./4-(Dt/(2*Dx)*(fr-fl)+Dt/(2*Dy)*(gu-gb));
```

## 9.2 Chapter 2 Matlab Program Code

### 9.2.1 One-patch control

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Control: One patch matlab Code  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=100;L=1; d_x=L/n;d_t=0.002; K=1.0; rm=1; h=d_x;
```

```
% you can change d_t, gain K, and jam density rm.
x=[0:d_x:1]; N=length(x);  % Corridor length
x1=0;                      % Start of the initial condition
for xx=0:d_x:1 x1=x1+1; Ro(x1)=0.6*exp(-10*(xx-0.6)^2);
end                        % End of the initial condition

r(1,1:n+1)=Ro(1:n+1);

for t=1:3500;      % Total time = t* d_t
% We first find the integral using trapezoid rule every time step
    sum=0;
    for c=1:n
        sum=sum+(r(t,c)+r(t,c+1));
    end
% Next we compute the PDE
    for j=2:n+1
         if j==2;r(t,j-1)=0*r(t,j);end
         if j==n+1 ; r(t,j+1)=1*r(t,j);end
         q(t,j+1)=r(t,j+1)*(1-r(t,j+1)./rm);
         q(t,j-1)=r(t,j-1)*(1-r(t,j-1)./rm);
         r(t+1,j)=0.5*(r(t,j+1)+r(t,j-1))-(d_t/d_x)/2*(K*h/2*sum)...
             *(q(t,j+1)-q(t,j-1))/((r(t,n+1).*(1-r(t,n+1)./rm))...
             -(r(t,2).*(1-r(t,2)./rm)));
    end
end
```

### 9.2.2 Five-patch control

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Control: 5 patches matlab Code  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=100;L=1; d_x=L/n;d_t=0.000005; h=d_x; K=1; rm=1;
% you can change d_t, gain K, and jam density rm.
x=[0:d_x:1]; N=length(x);  % Corridor length
x1=0;                      % Start of the initial condition
for xx=0:d_x:1
x1=x1+1;
Ro(x1)=0.6*exp(-10*(xx-0.6)^2);
end                        % End of the initial condition

r(1,1:n+1)=Ro(1:n+1);
```

```
for t=1:20000;            % Total time = t* d_t
% We first find the integral using trapezoid rule every time step
% for the five patches.
    sum_1=0;sum_2=0;sum_3=0;sum_4=0;sum_5=0;
    for c=1:n/5
        sum_1=sum_1+h/2*(r(t,c)+r(t,c+1))*1/0.2;
    end
    for c=n/5+1:n*2/5
        sum_2=sum_2+h/2*(r(t,c)+r(t,c+1))*1/0.2;
    end
    for c=n*2/5+1:n*3/5
        sum_3=sum_3+h/2*(r(t,c)+r(t,c+1))*1/0.2;
    end
    for c=n*3/5+1:n*4/5
        sum_4=sum_4+h/2*(r(t,c)+r(t,c+1))*1/0.2;
    end
    for c=n*4/5+1:n
        sum_5=sum_5+h/2*(r(t,c)+r(t,c+1))*1/0.2;
    end
% Next we compute the PDE
    for j=2:n+1
         if j==2;r(t,j-1)=0*r(t,j);end  % left B.C. is no inflow
         if j==n+1 ; r(t,j+1)=1*r(t,j);end  % right B.C is outflow
         q(t,j+1)=r(t,j+1)*(1-r(t,j+1)./rm); % calculate the fluxes
         q(t,j-1)=r(t,j-1)*(1-r(t,j-1)./rm);
% Next we update each patch
         if j<=n/5; sum=sum_1;
             q_R=r(t,n/5).*(1-r(t,n/5)./rm);
             q_L=r(t,2).*(1-r(t,2)./rm);

             r(t+1,j)=0.5*(r(t,j+1)+r(t,j-1))-(d_t/d_x)/2*(K*sum)...
                 *(q(t,j+1)-q(t,j-1))/(q_R-q_L);
         end

         if j>n/5 & j<=2*n/5; sum=sum_2;
             q_L=r(t,n/5+1).*(1-r(t,n/5+1)./rm);
             q_R=r(t,2*n/5).*(1-r(t,2*n/5)./rm);
            r(t+1,j)=0.5*(r(t,j+1)+r(t,j-1))-(d_t/d_x)/2*(K*sum)...
                *(q(t,j+1)-q(t,j-1))/(q_R-q_L);
         end
```

```
        if j>2*n/5 & j<=3*n/5; sum=sum_3;
            q_L=r(t,2*n/5+1).*(1-r(t,2*n/5+1)./rm);
            q_R=r(t,3*n/5).*(1-r(t,3*n/5)./rm);
            r(t+1,j)=0.5*(r(t,j+1)+r(t,j-1))-(d_t/d_x)/2*(K*sum)...
               *(q(t,j+1)-q(t,j-1))/(q_R-q_L);
        end

        if j>3*n/5 & j<=4*n/5; sum=sum_4;
            q_L=r(t,3*n/5+1).*(1-r(t,3*n/5+1)./rm);
            q_R=r(t,4*n/5).*(1-r(t,4*n/5)./rm);
            r(t+1,j)=0.5*(r(t,j+1)+r(t,j-1))-(d_t/d_x)/2*(K*sum)...
                *(q(t,j+1)-q(t,j-1))/(q_R-q_L);
        end

        if j>4*n/5; sum=sum_5;
            q_L=r(t,4*n/5+1).*(1-r(t,4*n/5+1)./rm);
            q_R=r(t,n+1).*(1-r(t,n+1)./rm);
            r(t+1,j)=0.5*(r(t,j+1)+r(t,j-1))-(d_t/d_x)/2*(K*sum)...
                *(q(t,j+1)-q(t,j-1))/(q_R-q_L);
        end
    end
end
```

# Vita

Sadeq J. Al-nasur received his B.S. degree in electrical engineering from Kuwait University, Kuwait, the M.S. degree in electrical engineering from Pennsylvania State University, University Park, in 1998 and 2002, respectively. In September 2003, he started pursuing the Ph.D. degree in electrical engineering at the Virginia Polytechnic Institute and State University, Blacksburg under the supervision of Dr. Pushkin Kachroo.