BACHELOR

Modelling pedestrian kinematics on the basis of the continuity equation

Bovenkerk, Thomas A.

Award date:
2023

Bachelor Final Project

Modelling pedestrian kinematics on the basis of the continuity equation

2021-2022

Bovenkerk, T.A.

Eindhoven University of Technology

Bachelor Final Project

Department of Mathematics and Computer Science

Solving pedestrian kinematics on the basis of the continuity equation

Author:
Thomas Alexander Bovenkerk

Supervisor:
Jan ten Thije Boonkkamp

Eindhoven, January 31, 2023

Abstract

This report will investigate how pedestrians move in crowds. This is important because researching this topic could help prevent perilous situations where people get trampled in a large crowd. This problem will be investigated using computer simulation based on the continuity equation to help answer the question if it is possible reduce or prevent congestions. Using the computer simulation, we found that if there is a way to influence the way pedestrians move and make them walk faster, it is possible to reduce or even prevent congestions. This would be very beneficial to places where large crowds gather in case of an emergency. Even in the case where there is no emergency it would still be beneficial as it would reduce the average density on a path.

Contents

1 Introduction

In crowd dynamics, when crowd density increases, people start walking more slowly and even coming to a complete standstill is not uncommon. When a crowd becomes less dense, people are better able to move around and thus walk faster. Dense crowds cause congestions, which is generally undesirable as this could potentially cause dangerous situations for example when a large crowd has to exit a stadium safely after a football game. The question is if it is possible to reduce or even prevent congestions.

1.1 The Crowdflow project

This project is part of the innovation space challenges. Innovation space is a centre of expertise for challenge-based learning and student entrepreneurship at TU/e. The challenge owner of this project is the TU/e Crowdflow Research Group. Crowdflow is essentially the investigation of crowds by measuring, predicting, and influencing them. In this challenge, along with four other students, crowdflow will be researched. One common problem with crowds is that congestions occur. Congestions usually occur because people start to move slower. The Crowdflow research focused on adding a so-called optic flow element to the crowd dynamics simulation. This optic flow was added in the form of light strips, which turn on and off to create a wave of light. The idea is that this optic flow functions as a form of visual aid, guiding the crowd in the direction of the exit. This occurring wave effect is defined as a nudge. How this was tested can be seen below in Figure 1.

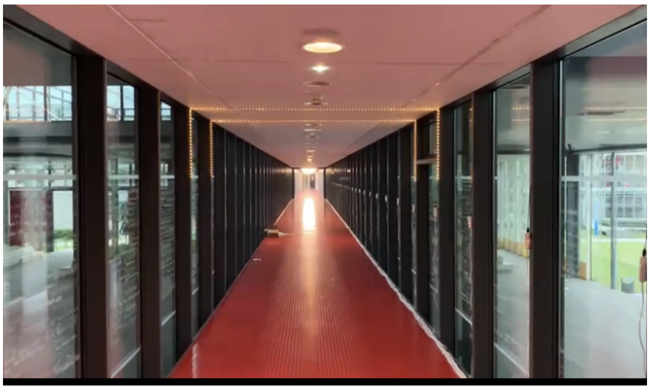


Figure 1: The optic flow

1.2 Problem statement and hypothesis

A widely present problem in the field of crowd dynamics is the congestion that occurs when the crowd density in a walking area increases. Real-world examples include horrible events where many pedestrians were trampled at concerts, sports matches and other large events. Our experiment hypothesises that if an optic flow as described in Section 1.1 is added to a walking path, this will increase the walking speed of people, and subsequently the flow of people going through the path is increased. As a result, this would decrease the number of people on the path, thus reducing the congestion.

1.3 Approach

The goal of this Bachelor Final Project is to analyse how the flow of pedestrians in a 1D and 2D domain changes after adding the nudge. This is done by using a computer model to create a path with pedestrians and let them move along this path. To achieve this goal, the research consists of three steps: the modelling of the crowd flow in a 1D situation Section 3, the modelling of the crowd flow in a 2D situation Section 4 and subsequently the addition of the nudge Section 5. The mathematical modelling and subsequent simulations of the formulated models will be done with MATLAB.

The first step in solving the problem at hand is by describing a pedestrian flow. The one that will be used for this report is the one described by Hughes [2]. In the report, a pedestrian flow of a single type of pedestrian is described. This means that there are only pedestrians and nothing else on the path. This flow depends on three variables. The density of the flow, ρ, is the number of pedestrians per m^2, and the velocity of the flow in the horizontal x-direction moving to the right and left is u and the velocity in the vertical y-direction moving up and down is v.
The equation describing the relationship between these variables is the following:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0. \tag{1.1}$$

Because there are three variables which are all dependent functions of x, y, and t this equation is defined as a partial differential equation and contains partial derivatives. In fluid dynamics, this equation is known as the continuity equation. This equation is often used in fluid mechanics to describe the flow of a fluid. In this project, this equation states that the change of the number of pedestrians in an infinitesimal area is equal to the number of pedestrians that arrive in the region subtracting the ones that leave that area. By solving equation (1.1) it will be possible to gain insight into the way pedestrians move.

1.4 Model

Directly solving partial differential equation (1.1) in this form would be rather challenging. The strategy to solve such an equation will be with the use of numerical analysis. The numerical scheme used for this is the so-called Godunov scheme which will be explained in the next section.

The first step in building a model that analyses the pedestrian flow is creating a 1D simulation in MATLAB. In the simulation, it is assumed that on the path there is only an x-direction and pedestrians move from the left to the right, as such in this situation $v(\rho) = 0$. After this simulation is done it will be extended to a 2D simulation; this simulation will represent a more true-to-life visual representation. It can be verified that the 2D model behaves as expected using the results gathered from the 1D situation. If any problems arise during this stage it is possible to change the model as needed. After this step is completed it is possible to implement the nudge into the simulation and it is then possible to see the results from such a nudge.

2 Constraints and features

When building a model it is important to formulate the assumptions that have been taken into account. These assumptions hold for both the 1D and the 2D models. It is assumed that:

- At the start of a simulation the density of pedestrians along the path is random.
- The maximum number of pedestrians per m^2 is 5. This is called $\rho_{\max}$; at this value pedestrians are no longer able to move
- The top and bottom parts of the path are closed as displayed in Figure 2. Pedestrians cannot exit through them, the only exit is at the end of the path.
- When the path is evacuated pedestrians do not move faster, this can be done by opening the top and bottom edges of the path.
- When the nudge is applied people move 10% faster, this will be applied to the horizontal speed given by $u(\rho)$.
- Pedestrians enter from the left and exit at the end of the path, other edges are closed.
- Pedestrians do not go for the shortest route. They are more inclined to move to a lower-density area.
- The speed in the x-direction and the speed in the y-direction of the pedestrians do not have the same value. The primary direction of movement is in the x-direction.
- The maximum walking speed of pedestrians in the x-direction is 1.25 m s^{-1}.
- The maximum walking speed of pedestrians in the y-direction is 0.4 m s^{-1}.
- In the 1D model the path has a length of $L_x = 100$m in the x-direction and a length of $L_y = 30$m in the y-direction.
- In the 2D model The path has a length of $L_x = 30$m in the x-direction and a length of $L_y = 30$m in the y-direction.
- Pedestrians do not move slower or faster when close to the edge of the path, contrary to how most fluid dynamic models behave.
- Pedestrians are compressible, as in this model they are treated as a fluid. Otherwise, the density would be constant everywhere. This would not provide interesting results. The speed of the pedestrians changes each based on the density.
- A path is considered congested if there is a density of more than 2.75 pedestrians per m^2 for at least 5 m.

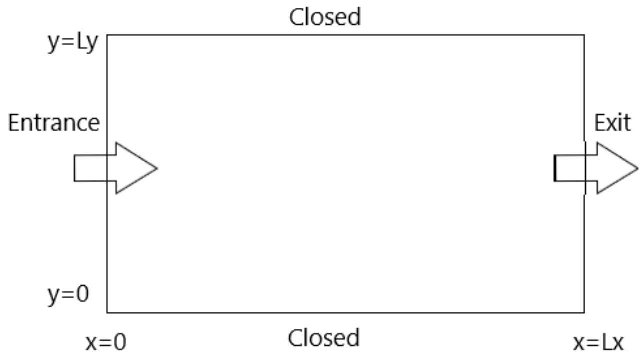


Figure 2: Rectangular path

2.1 Simplification

The model is a simplification of a real-world situation. Predicting how pedestrians walk is very difficult. However, with the assumptions stated above a working model has been created. In a more realistic model, pedestrians would have a direction they need to go. In the current model, they move more as a fluid would, this is fine for large crowds. In real life, pedestrians can see from a distance where a path is crowded and choose to avoid those areas before they get there. This is not the case in our model. In real life, pedestrians also have varying speeds, as not everyone walks at the same speed, this is also something that is not used in the model.

3 1D model

The continuity equation is a first-order partial differential equation. This equation describes the change of density in an infinitesimal area. In this section, this equation will be solved for a 1D situation. This is done by modelling a path of pedestrians.

3.1 Variables

Before it is explained how the model works, the variables of the model will be explained first. The first variable is Lx, which is the length of the path. In the path, there is a certain density ρ at every point, i.e., the number of pedestrians per meter. The variable t is the time at. Finally, pedestrians have a certain speed dependent on the density. The speed function is defined as a function of the density, namely

$$u(\rho) = u_{\text{max}}\Big(1 - \frac{\rho}{\rho_{\text{max}}}\Big). \tag{3.1}$$

Using this speed function we see that

$$u(0) = u_{\text{max}}, \ u(\rho_{\text{max}}) = 0.$$

This means that when $\rho = \rho_{\text{max}}$ pedestrians are standing still. Conversely, when $\rho = 0$ pedestrians reach their maximum speed.

3.2 Numerical scheme

Because solving first-order partial differential equations is usually very complicated, the strategy of finding a numerical approximation to the solution is used. This will make it easier to solve the equation and create a simulation in MATLAB. In the previous section equation (1.1) was introduced. In this section, the 1D case will be analysed. The equation at hand is a first-order partial differential equation with a boundary condition at the beginning of the path. For t a discretization of time is used, and for x a discretization of the domain is used. To put this plainly, this means that t will be divided N_t parts of length Δt. The domain of x will be divided in N_x parts of length Δx, Δx can be rewritten to $\Delta x = \frac{L_x}{N_x}$. Now, the equations will be considered in the points $t_n = n \cdot \Delta t$ and $x_j = j \cdot \Delta x$ for $n = 0, \ldots, N_t$ and $j = 0, \ldots, N_x$, respectively. For convenience, the following notation is introduced:

$$\rho(t_n, x_j) \approx \rho_j^n.$$

The superscript refers to the time level, whilst the subscript refers to the position in space. It is important to remember that ρ_j^n is a numerical approximation of $\rho(t_n, x_j)$. Each cell of x has a length of Δx. Each of these cells has its own density ρ, and thus also its velocity. This is illustrated in Figure 3 below.

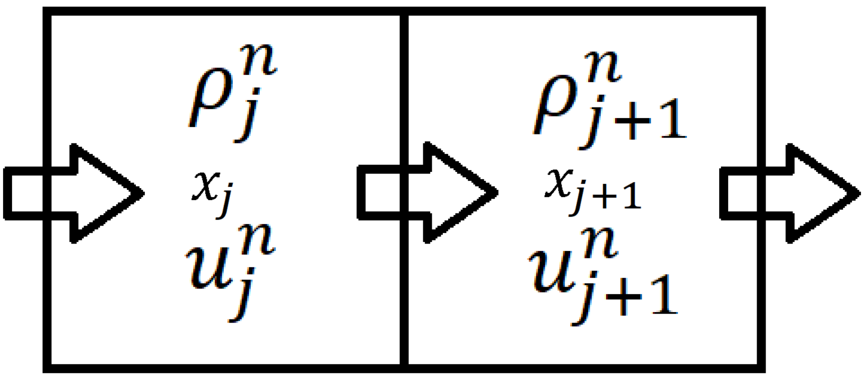


Figure 3: Two adjacent cells

It was mentioned in Section 1 that in the 1D situation the speed in the y-direction is zero, as such $v = 0$. Applying this to (1.1) gives us the equation for the pedestrian flow for a 1D situation. The equation then reads:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) = 0. \tag{3.2}$$

This can be written as

$$\rho_t + (\rho u)_x = 0. \tag{3.3}$$

In computational fluid dynamics this equation, and also (1.1) are also known as one of the Euler equations. These are equations in fluid dynamics named after Leonard Euler [3]. They describe how the pressure, velocity, and density of a moving fluid are related to each other. They can be seen as a more simplified version of the Navier-Stokes equations, these are a set partial differential equations which describe the motion of viscous fluid. The Euler ignore viscosity making them more simplified. Therefore the Euler equations are only an approximation of a fluid problem. The equations (1.1) and (3.2) are formulated in divergence form. These equations can be solved using the Riemann solvers, named after Bernhard Riemann. These are initial value problems which consist of an equation in divergence form and initial piecewise constant data of the domain. Riemann problems are often used to solve complex nonlinear equations, the Euler equations are a good example of this.

This system has one boundary condition, which is that at the entrance of the path there will be an inflow of pedestrians, this inflow is given. At the exit of the path, there will be an outflow of pedestrians meaning they leave the system, this outflow follows from the solution and may not be prescribed. Furthermore, when looking at the flow of pedestrians, the initial distribution of pedestrians needs to be known otherwise no comment can be made about the propagation of the flow, as such:

$$\begin{aligned} \text{Boundary condition: } & \rho(t,0) = \rho_0(t) \text{ with } t \geq 0, \\ \text{Initial condition: } & \rho(0,x) = \rho^0(x) \text{ with } 0 \leq L_x. \end{aligned}$$

It is now time to discuss the numerical scheme that will be used during this report. To solve a Riemann problem, a Riemann solver is often used. In this case, the way it will be solved is by using the so-called Godunov scheme, named after Sergei Godunov. This is a conservative numerical scheme. This means that this scheme yields a solution that will obey the physical conservation laws. This scheme approximates the Riemann problems at each cell boundary between two cells.

The Godunov scheme is first-order accurate for both space and time. First, q is defined by:

$$\rho = q = 1 = \frac{\rho}{\rho_{\max}}. \tag{3.4}$$

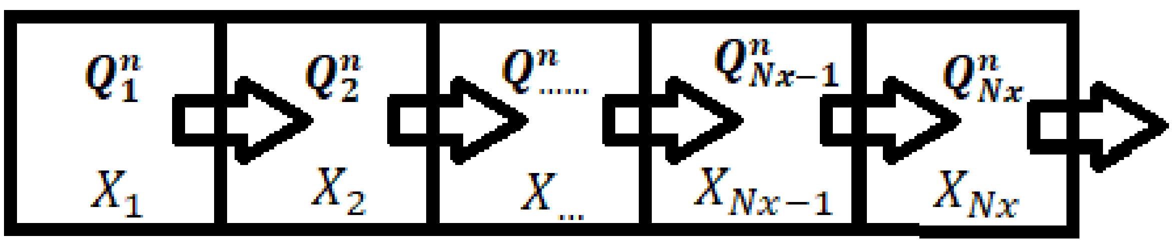


Figure 4: The complete path

In Figure 4 the flow along the path is described. At the last cell N_x, pedestrians exit the path and at the first cell pedestrians enter the path. The direction of movement is indicated by an arrow.

From this it is clear that at $q = 0$, the density ρ reaches its maximum value $\rho_{\max}$, and for $q = 1$, the density $\rho = 0$ Using (3.4), the speed function can also be defined as a function of q and it reads:

$$u = u_{\max} q. \tag{3.5}$$

This q then satisfies a conservation law, and when applied to (3.3) the resulting equation is:

$$q_t + f(q)_x = 0. \tag{3.6}$$

The flux is defined as the number of pedestrians that pass through a certain point per unit of time, and per unit area and it is defined as:

$$f(q) = u_{\max} q(q - 1). \tag{3.7}$$

By introducing q the flux function (3.7) which is a convex equation as $f''(q) > 0$. The conservation law (3.6) can be discretized to the following form by looking at the integral form of this equation over the control volume $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, which gives us:

$$\int_{t_n}^{t_{n+1}} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \Big(q_t + f(q)_x\Big) \mathrm{d}x \mathrm{d}t = 0. \tag{3.8}$$

This equation is evaluated to the following form according to [5]:

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \Big(q(t_{n+1}, x) - q(t_n, x)\Big) \mathrm{d}x + \int_{t_n}^{t_{n+1}} \Big(f(q(t, x_{j+\frac{1}{2}})) - f(q(t, x_{j-\frac{1}{2}}))\Big) \mathrm{d}t = 0. \tag{3.9}$$

These integrals will be evaluated using the finite volume method and the time-averaged flux function, respectively. The finite volume method will approximate the value of q_j^n by taking the average value of j at interval t and is given by:

$$q_j^n = \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} q(t_n, x) \mathrm{d}x. \tag{3.10}$$

Applying (3.10) to the left integral of (3.9) leads to:

$$\Delta x \Big(q_j^{n+1} - q_j^n\Big) = \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \Big(q(t_{n+1}, x) - q(t_n, x)\Big) \mathrm{d}x. \tag{3.11}$$

In the Godunov scheme the time-averaged flux function is:

$$F_{j+\frac{1}{2}}^n = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} f(q(t, x_{j+\frac{1}{2}})) \mathrm{d}t. \tag{3.12}$$

In this report (3.12) will be referred to as the exact numerical flux. By applying (3.12) to the right integral of (3.9) we obtain:

$$\Delta t \Big(F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n\Big) = \int_{t_n}^{t_{n+1}} \Big(f(q(t, x_{j+\frac{1}{2}})) - f(q(t, x_{j-\frac{1}{2}}))\Big) \mathrm{d}t. \tag{3.13}$$

Combining equations (3.9), (3.11), and (3.13) gives:

$$\Delta t(F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n) + \Delta x(q_j^{n+1} - q_j^n) = 0.$$

Diving both parts by Δt and Δx leads to

$$\frac{q_j^{n+1} - q_j^n}{\Delta t} + \frac{F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n}{\Delta x} = 0, \tag{3.14}$$

which can be rewritten to:

$$q_j^{n+1} = q_j^n - \frac{\Delta t}{\Delta x}\Big(F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n\Big). \tag{3.15}$$

In the Godunov scheme, $F_{j+\frac{1}{2}}^n$ and $F_{j-\frac{1}{2}}^n$ are defined as the numerical flux. To find the values of $F_{j+\frac{1}{2}}^n$ and $F_{j-\frac{1}{2}}^n$ the numerical Godunov flux function F is introduced.

$$F_{j+\frac{1}{2}}^n = F(q_j^n, q_{j+1}^n). \tag{3.16}$$

Figure 5 below illustrates what the situation at the cell boundary looks like.

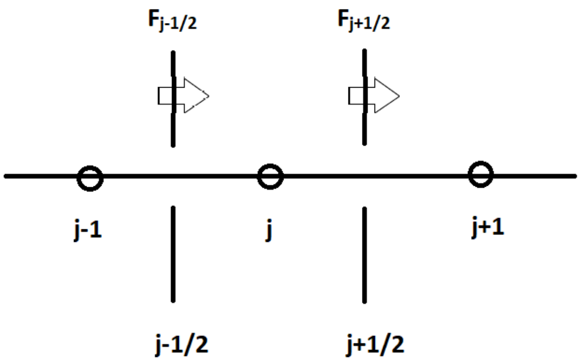


Figure 5: Numerical flux

The choice of whether to use q_j^n, q_{j+1}^n for the numerical flux function depends on several factors. During a simulation, it is possible to experience a shock wave or a rarefaction wave. A good example of when a rarefaction and a shock wave occur is the following. When a traffic light turns red the first vehicle stops moving and the vehicles behind that first vehicle have a higher speed and when they brake they are more clustered together. Contrary when the light turns green, the first vehicle starts to move and the vehicles behind are moving at a slower speed, increasing the distance between the vehicles. An example of this can be seen in Figure 6 below.

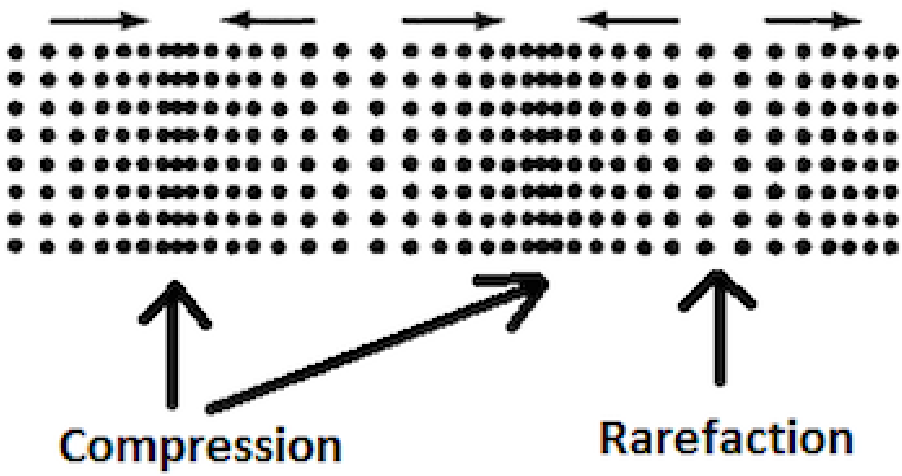


Figure 6: Shock wave and rarefaction wave [1]

The compression areas represent a shock wave.

To determine whether a shock wave or a rarefaction wave occurs one needs to consider the derivative of the flux (3.7), which is given by:

$$a(q) = f'(q) = u_{\max}(2q - 1). \tag{3.17}$$

For simplicity, the notation $F(q_l, q_r)$ is introduced with $q_l = q_j^n$, and $q_r = q_{j+1}^n$. If there occurs a situation where $a(q_l) \geq a(q_r)$ then a shock wave arises.

$$F(q_l, q_r) = \begin{cases} f(q_l) & \text{if } s \geq 0, \\ f(q_r) & \text{if } s < 0. \end{cases} \tag{3.18}$$

The property that q is convex is paramount for this since $a(q_l) > a(q_r) \Leftrightarrow q_l > q_r$ The equation to calculate the speeds of the shock wave is given by:

$$s = \frac{f(q_r) - f(q_l)}{q_r - q_l} = u_{\max}(q_r + q_l - 1). \tag{3.19}$$

On the contrary, when $a(q_l) < a(q_r) \Leftrightarrow q_l > q_r$, a rarefaction wave occurs. The property that q is convex allows for the use of the following results.

$$F(q_l, q_r) = \begin{cases} f(q_l) & \text{if } a(q_l) \geq 0, \\ f(w(0)) & \text{if } a(q_l) < 0 < a(q_r), \\ f(q_r) & \text{if } a(q_r) < 0. \end{cases} \tag{3.20}$$

For the second condition $a(w(\eta)) = \eta$ holds. To find this value it is required to calculate the value of $w(0)$.

$$a(w(\eta)) = \eta.$$

Substituting this into Equation (3.17) leads to:

$$u_{\max}\Big(2w(\eta) - 1\Big) = \eta.$$

Divide both sides by $u_{\max}$ and isolating $w(\eta)$.

$$w(\eta) = \frac{1}{2}\Big(1 + \frac{\eta}{u_{\max}}\Big).$$

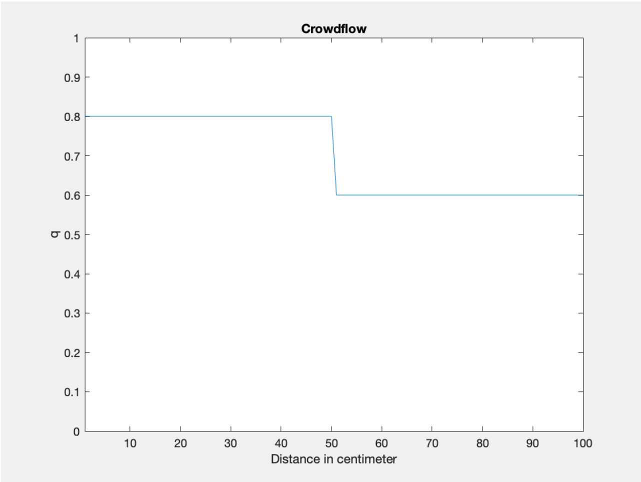
Subsituting η=0 gives.

$$w(0) = \frac{1}{2}.$$

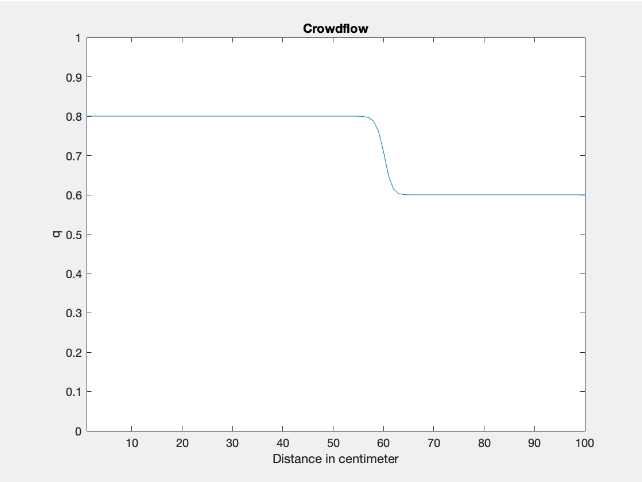
Insert this back into Equation 3.7 gives the result of:

$$f\Big(\frac{1}{2}\Big) = -0.3125.$$

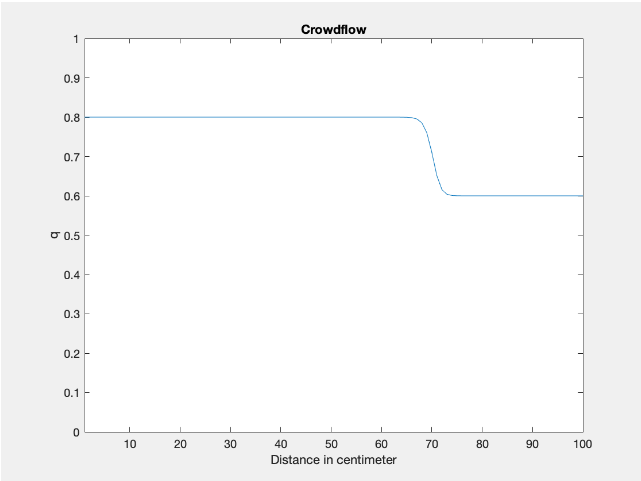
Using the results above it is now possible to solve a Riemann Problem. In both situations, there is a piecewise constant initial data that has a single discontinuity. This is in both cases at $x = 0.5$ meter. In the example $u_{\max} = 1$.



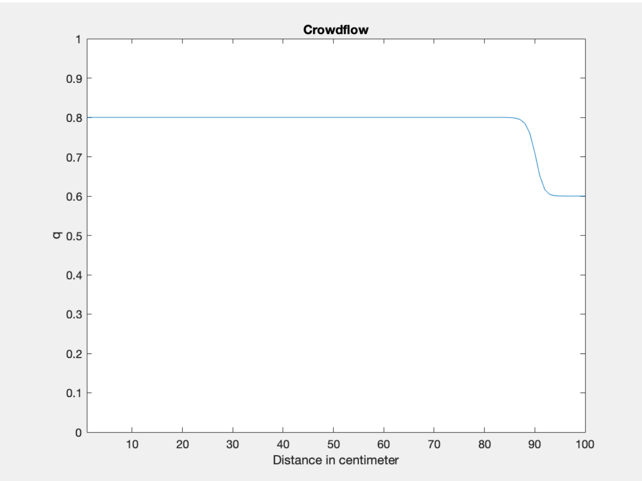
(a) t=0



(b) t=0.25



(c) t=0.5

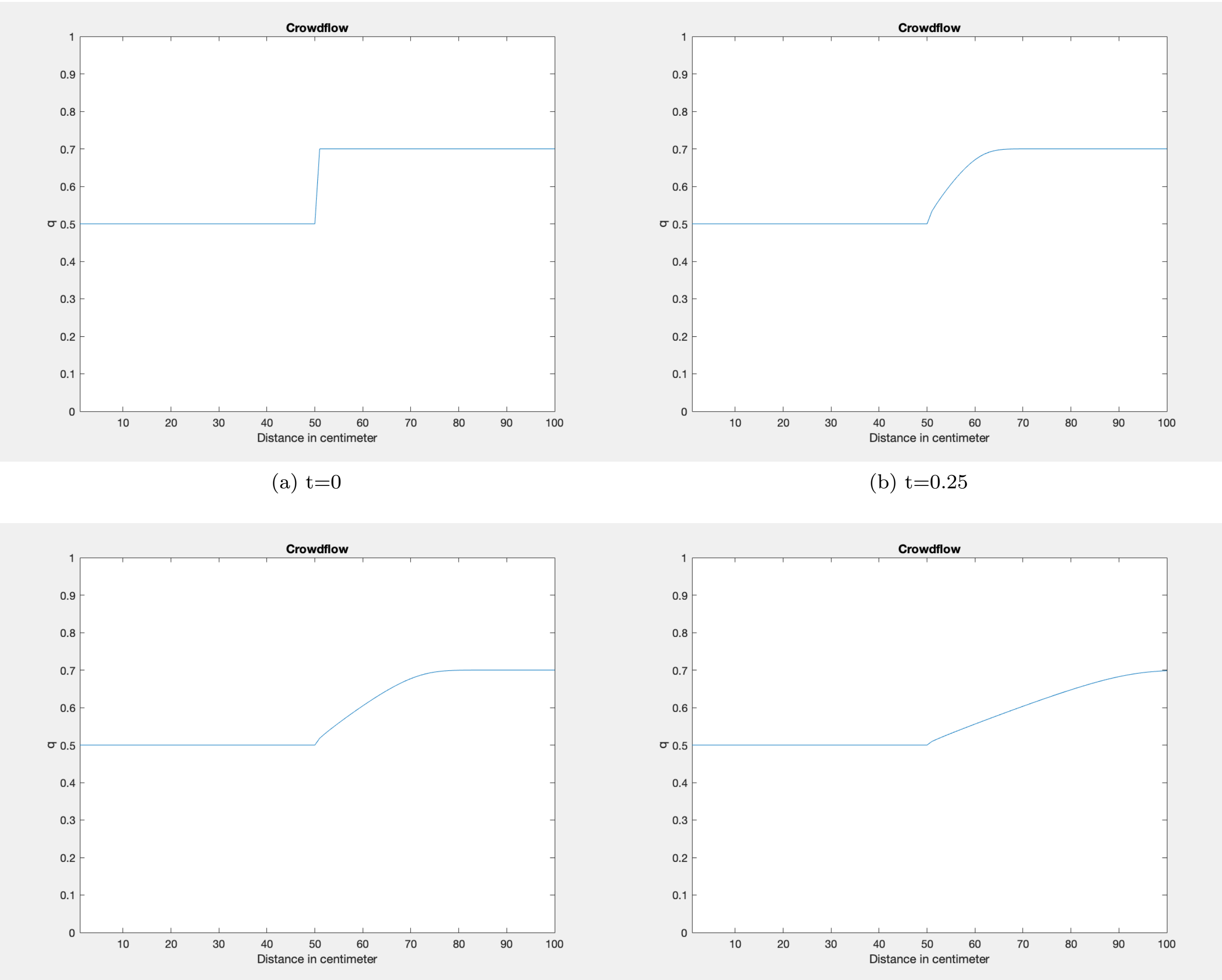


(d) t=1

Figure 7: Shockwave

In Figure 7 $q_l = 0.8$ and $q_r = 0.6$ so $q_l > q_r$ and a shock wave occurs. The shock speed can be calculated using (3.18) and results in a shock speed of $q_l + q_r - 1 = 0.4$. This means that after 1 second the discontinuity should move from $x = 0.5$ to $x = 0.9$. It can be seen from Figure 7d that this is indeed the case.

In Figure 8 $q_l = 0.5$ and $q_r = 0.7$ so $q_l < q_r$ and a rarefaction wave occurs.



(a) t=0 (b) t=0.25

(c) t=0.5 (d) t=1

Figure 8: Rarefaction wave

It is also important to realise that the choice of Δx and Δt are not random. They depend on each other, and the choice of these values needs to meet the requirements of the numerical scheme. This is given by:

$$\frac{\Delta t}{\Delta x}|f'(q)| < 1. \tag{3.21}$$

$f'(q)$ is given by:

$$f'(q) = u_{\max}(2q - 1). \tag{3.22}$$

Plugging this into (3.21) gives.

$$\frac{\Delta t}{\Delta x}|u_{\max}(2q - 1)| < 1.$$

The maximum value of $u_{\max}|2q - u_{\max}|$ occurs when either $q = 0$ or $q = 1$. By taking $q = 1$ this results in:

$$u_{\max}\left|\frac{\Delta t}{\Delta x}\right| < 1.$$

3.3 Numerical results

Now that the numerical scheme has been explained it can be used with the program MATLAB. MATLAB will simulate the pedestrian flow at each time step. With the simulation, it is possible to compute the density at each point. At the start of a simulation, an $n \times j$ random matrix will be initiated, this matrix represents the starting densities, and will store the densities of the pedestrian flow at each point, and at each time level. In this matrix, n represents the time steps, and j represents the steps in space. In each column of the matrix, the value q at each point x_j will be stored. It will then be translated back to the density ρ. The first column depicts the inflow of pedestrians into the path. Each row of the matrix represents the pedestrian flow at each time level. With this, the 'Crowdflow' can be simulated and analyzed using the code found in A.1.

3.4 Results

Validation of the model is done by running the code for different situations and checking for potential errors. Verification is an important step when building a model. However, it is not easy to compare this model to a real-life path. Instead, the first results in this section will verify that the model behaves as expected and that there are no irregularities.

Five distinct situations will be simulated, these simulations will start at $t = 0$. The parameters of each simulation are different. In all cases $\Delta t = 0.75$ and $\Delta x = 1$.

- In the first simulation shown in Figure 9 there will be no inflow of pedestrians meaning $\rho(t, 0) = 0$ and the existing pedestrians will be placed randomly on the path. The expected result is that after several time steps the path will be empty.
- In the second simulation shown in Figure 10 an empty path with constant inflow will be tested. In this case, the inflow is set to $\rho = 2.5$. The expectation is that the path will fill with pedestrians but that the density will not become higher than the value in the inflow.
- In the third simulation shown in Figure 11 it is the first time that there is congestion. In the beginning, the density was randomly distributed from $\rho = 4.5$ to $\rho = 5$. The inflow is set to $\rho = 0.5$ so the number of pedestrians coming into the path is very low. The expected result is that the path empties.
- In the fourth simulation shown in Figure 12 the initial situation is similar to that of the third simulation the only difference is that the speed of pedestrians in this simulation has been increased by 10% to see if it is possible to reduce the time the path is congested. This will increase $u_{\max}$ to 1.375 in the simulation. The expectation is that the congestion will dissolve faster than in the previous simulation.
- The fifth and final simulation is shown in Figure 13. The inflow has been changed from a constant flow to an inflow that is dependent on the time.

3.4.1 No inflow scenario

When the simulation shown in Figure 9 begins, peaks of high density can be seen as is expected with a random distribution. These high peaks quickly dissipate. However, as can be seen in Figure 9b at $t = 16.5$ these peaks cause certain areas of the path to have a higher density. In Figure 9c the effect of no inflow of people is visible, with no inflow, the path starts to empty, and the small areas of higher density do not cause a congestion as the density is not high enough. The peaks of higher densities also start to decrease when comparing the peaks of Figure 9d to Figure 9b. Finally at $t = 168.75$ the path is nearly empty as shown in Figure 9f. In this simulation because $\rho(t, 0) = 0$ results in $q_l > q_r$ as $q(t, 0) = 1$ at the beginning of the path. Because of this, a shock wave arises at the beginning of the path. This effect continues for the entire path as it quickly empties.

The speed of the shock wave in Figure 9b can be calculated as $q_l = 1$ and $q_r \approx 0.4$ so $s \approx 0.5\text{m}^2\ \text{s}^{-1}$. As a result, it is expected that the shock wave will have caused the pedestrians to move 14.5 meters going from Figure 9b to Figure 9c. Using the shock speed from Figure 9b it is expected that the path empties in 200 seconds. This is not the case. However, this can be attributed to the fact the density is not equal to 0.4 all the time as it can be seen that in Figure 9d that $q_r \approx 0.45$ which results in a shock speed of $s \approx 0.5625\text{m}^2\ \text{s}^{-1}$, which means that path would empty in approximately 172 seconds. This is very close to the actual time it takes for the path to empty.

3.4.2 Constant inflow scenario

The results of the second simulation can be seen in Figure 10. It can be seen in Figure 10b that at $t = 15$ already 20% of the path has started to fill up with pedestrians. In Figure 10c at $t = 52.5$, this has increased to 80%. This increases to 100% at $t = 75$ as can be seen in Figure 10d. After this, the density along the path slowly increases as can be seen in Figures 10e, and 10f. However, the density in a cell will never exceed the value of the inflow. Eventually, the path reaches a balanced state where the densities no longer change.

Contrary to the case of Simulation 1, since $q(t, 0) = 0.5$ a situation arises where $q_l < q_r$ as such in the simulation the effect of a rarefaction wave can be seen. This effect continues for the entirety of the path.

3.4.3 Low inflow scenario

In the third simulation shown in Figure 11, the effect of a high density can be seen. In the initial distribution, the density is very high as can be seen in Figure 11a. What happens first from the initial state is that values of the density are spread a little more evenly over the next 22.5 seconds. However, as can be seen in Figure 11b there is a difference in the densities as some areas have a higher density and some areas have a lower density. At $t = 102.75$, in Figure 11c it can be seen that at the end of the path the density is lower than at the beginning; this could be because the beginning of the path slowly starts to empty. In the situation, $q_l > q_r$ at the beginning of the path creates a shock wave just like in the first simulation. In the middle of the path the reverse happens and $q_l < q_r$ which creates a rarefaction wave. At $t = 264$ most of the path balances out and at the beginning of the path the densities now rapidly start to decrease as can be seen in Figure 11d. This effect of the density rapidly decreasing continues in Figure 11e. Finally at $t = 465$ in Figure 11f almost all of the high density has disappeared. It is noticeable that even with a very low inflow it takes a long time for the path to empty.

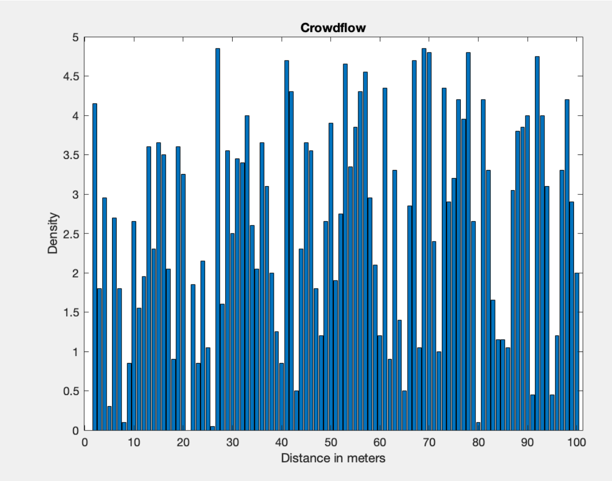
The shock speed of the simulation can also be calculated using $q_l = 0.9$ and $q_r = 0.3$ this results in a shock speed of $s = 0.2\text{m}^2\ \text{s}^{-1}$, with this shock speed the shock wave will have moved the pedestrians 93 meters at $t = 465$. This seems to fall in line with the result of the simulation.

3.4.4 Increased speed scenario

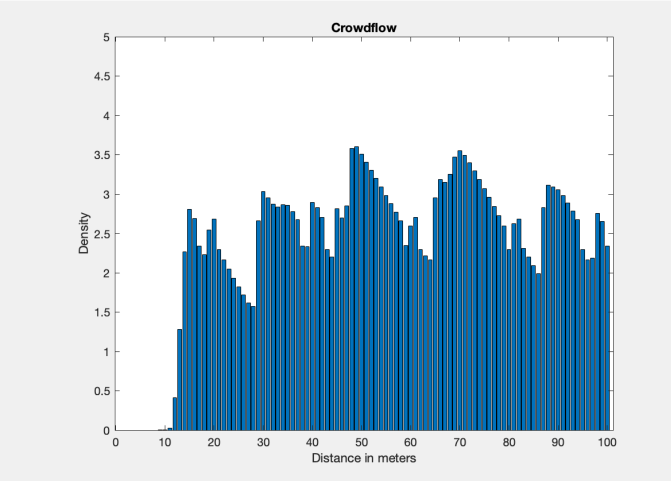
In the situation described by Figure 12 it can be seen that a congestion can take a long time to resolve. The inflow at the entrance of the path plays a role in this. Another factor is the speed at which pedestrians walk. By increasing the speed of pedestrians they exit the path faster and move out of congested areas faster. However, this also means that if pedestrians move faster they will go through the path faster. As a result, if there already is a congested area in the path, then they could create a congestion of greater length. In this case, moving faster has an adverse effect on the flow of the path. It can be seen in Figure 12b that pedestrians leave the path faster than in Figure 11b. This is also the case in the next Figure 12c where the density is lower than in Figure 11c. In Figure 12d the average density of the path is lower than in Figure 11d. However, the density at the end of the path with increased speed is the same as the density at the end of the path without the increased speed. Very noticeable from this result is that an increase in speed of 10% leads to the fact that, in Figure 12f with the increased speed at $t = 414.75$ the congestion is dissolved as opposed to the situation in Figure 11f where the congestion dissolves at $t = 465$, in this case, the congestion dissolves 11% faster.

3.4.5 Periodic inflow scenario

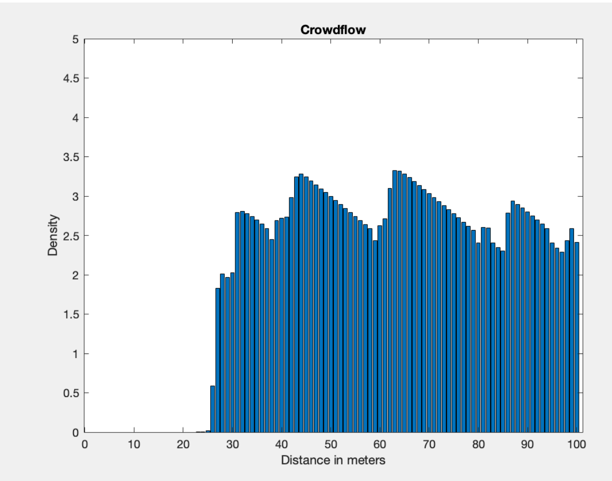
In the situation of Figure 13, a periodic inflow is added. At the start of the simulation in Figure 13b, at $t = 27$, this is not that noticeable. However, at $t = 102$ in Figure 13c it can be seen that at the beginning of the path the density is increasing, this is because the inflow is almost at $q = 0.5$, just like in the second simulation this creates a rarefaction wave because again $q_l < q_r$. In the next figure, Figure 13d at $t = 110$ the inflow is now lower again and which creates a situation where $q_l > q_r$ as such a shock wave arises. This effect where there will first be a shock wave for a while and then a rarefaction wave will keep on happening. In Figure 13e it can be seen that these shock waves and rarefaction waves in combination with the periodic inflow create a wave effect. At $t = 300$ in Figure 13f, this wave effect spans the entirety of the path. Because of the periodic inflow, there are periods when barely any pedestrians enter the path, as such the density along the path is fairly low. Limiting the inflow could also be a good way to prevent congestions.



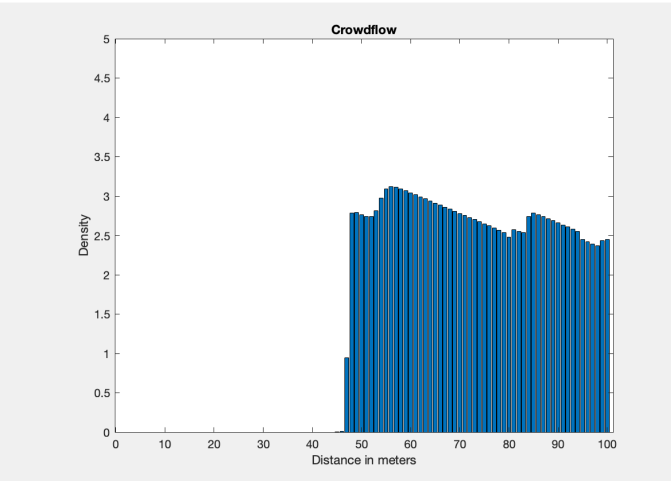
(a) t=0



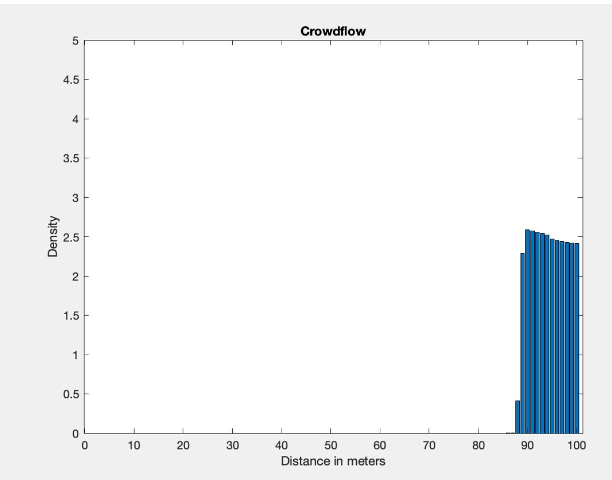
(b) t=16.5



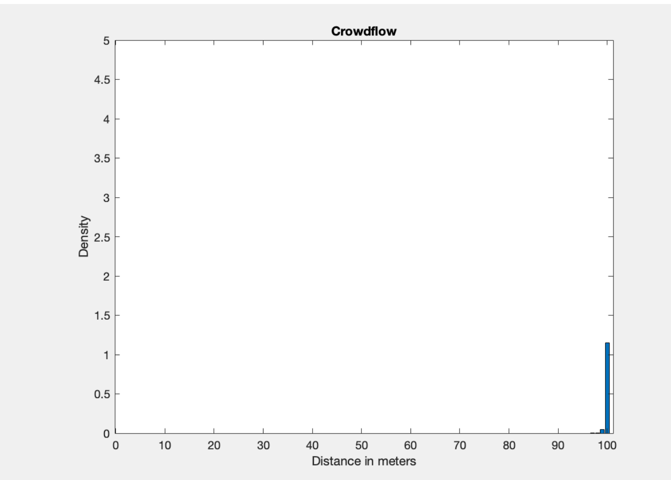
(c) t=37.5



(d) t=75

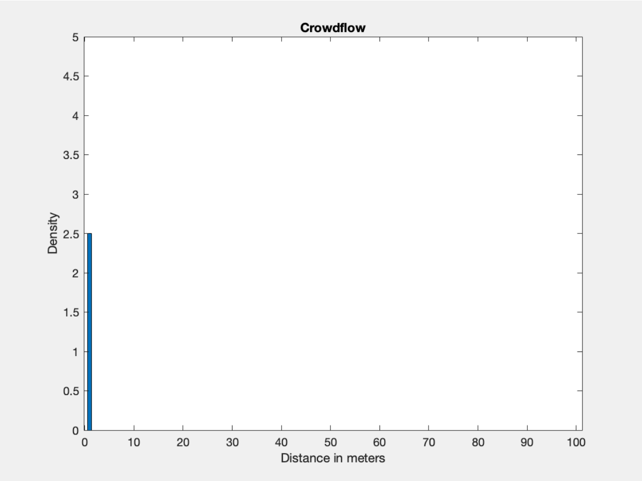


(e) t=150

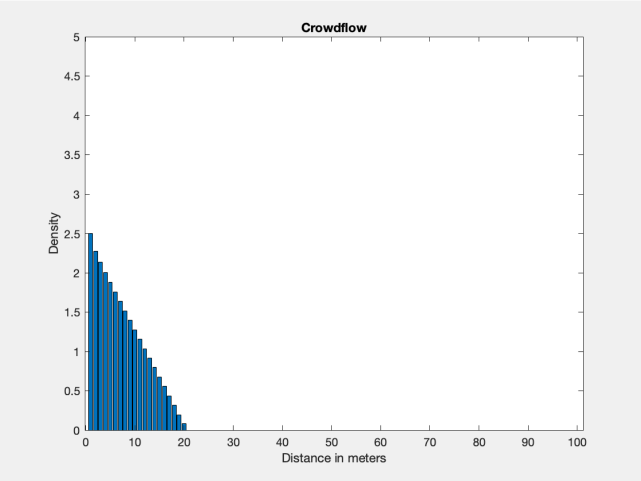


(f) t=168.75

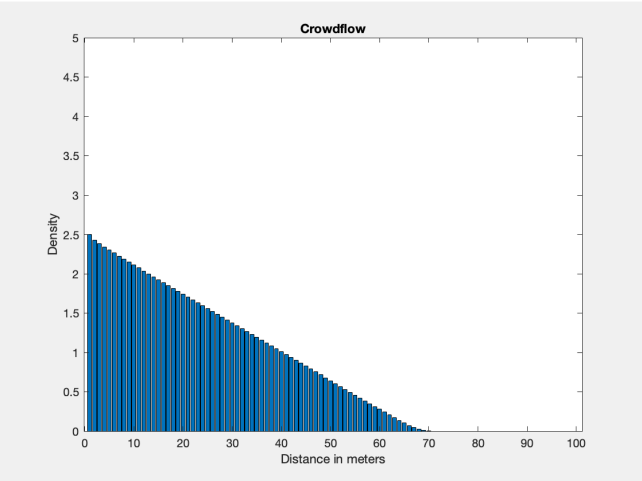
Figure 9: No inflow of people



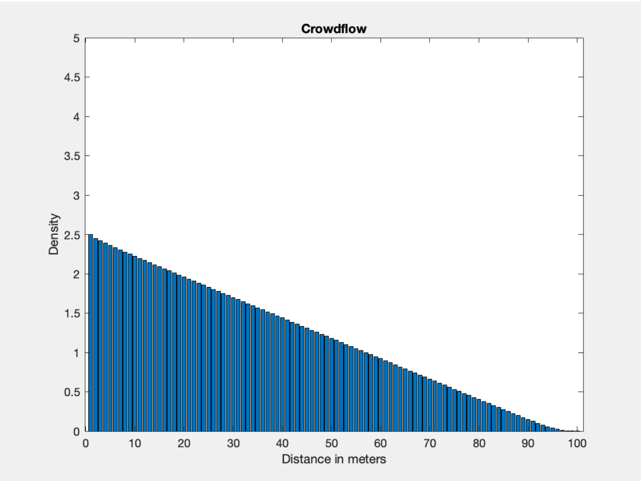
(a) t=0



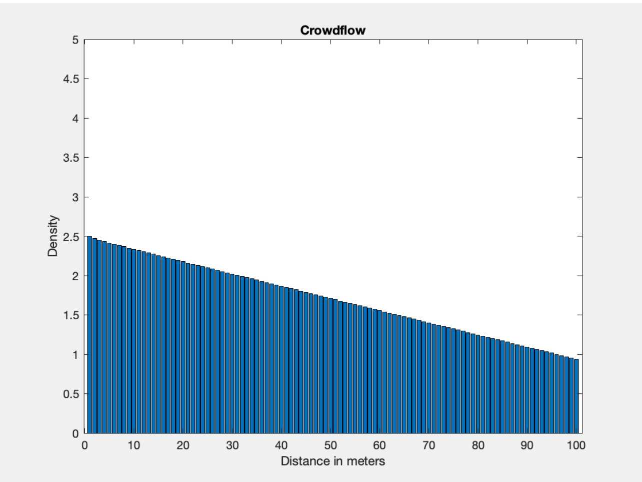
(b) t=15



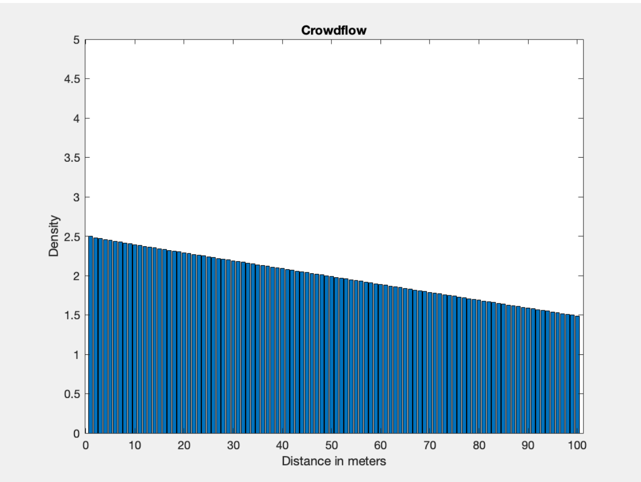
(c) t=52.5



(d) t=75

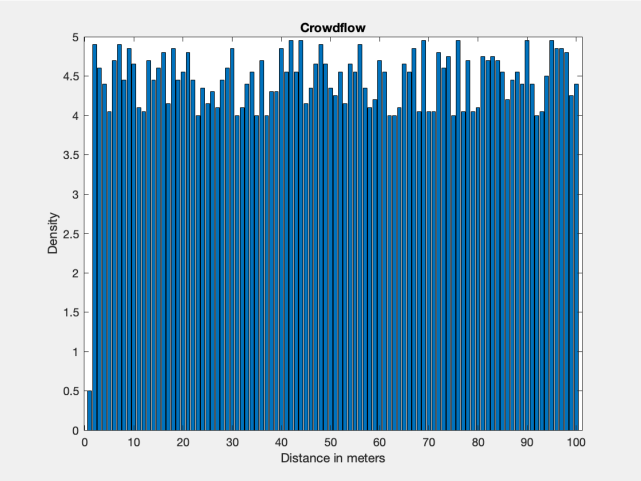


(e) t=127.5

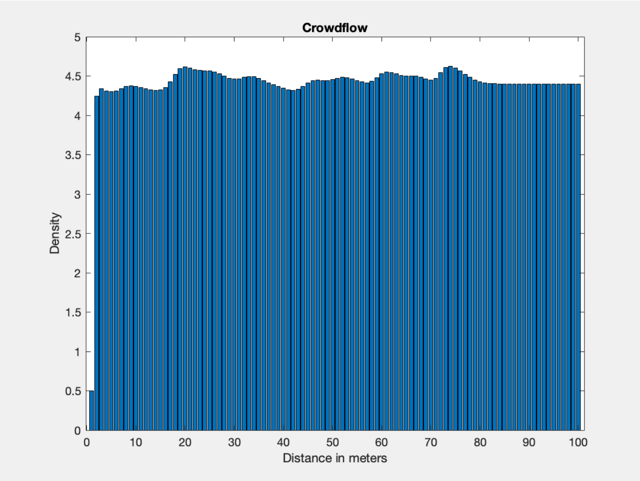


(f) t=198

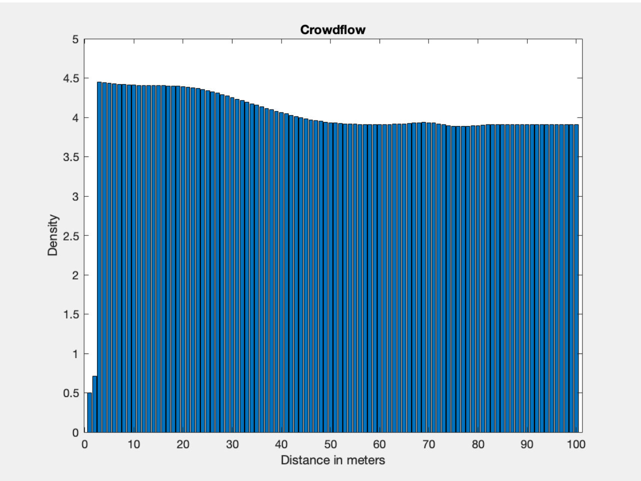
Figure 10: Initially empty path



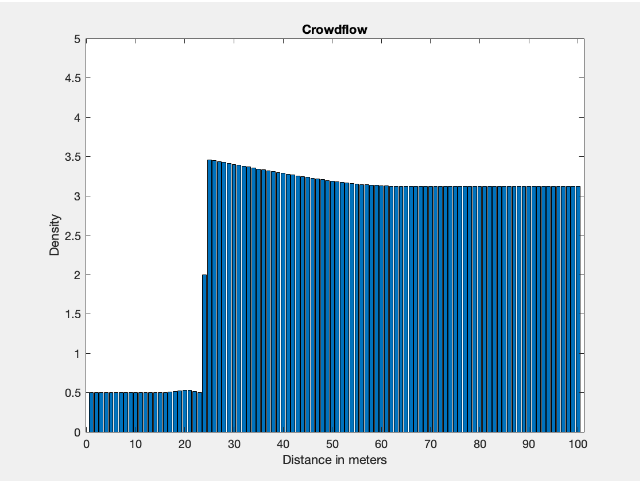
(a) t=0



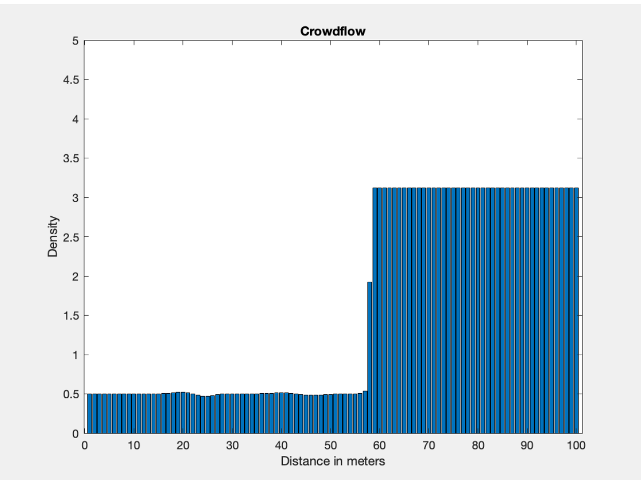
(b) t=22.5



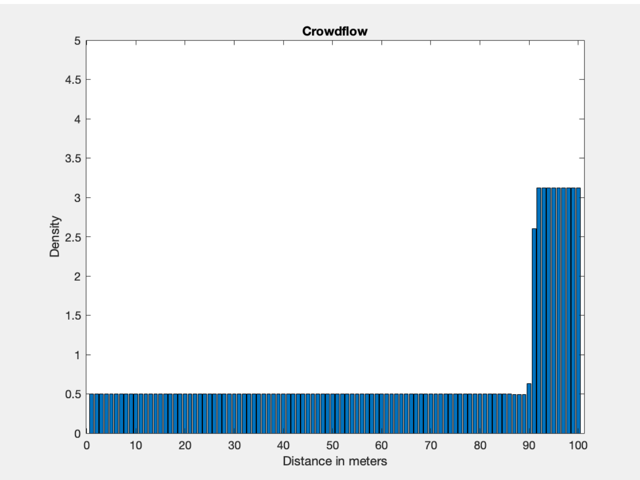
(c) t=102.75



(d) t=264

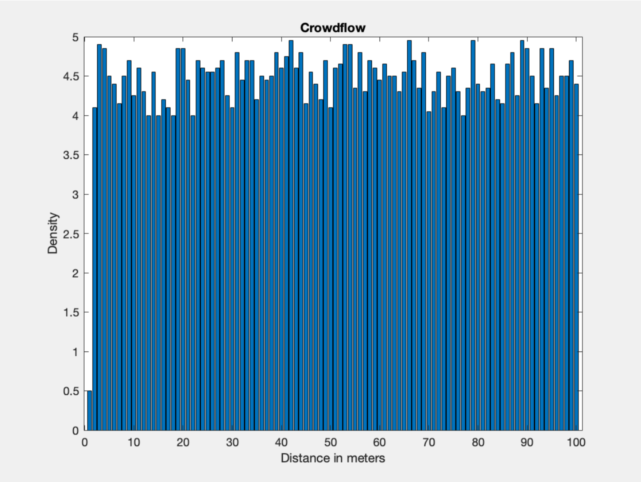


(e) t=369

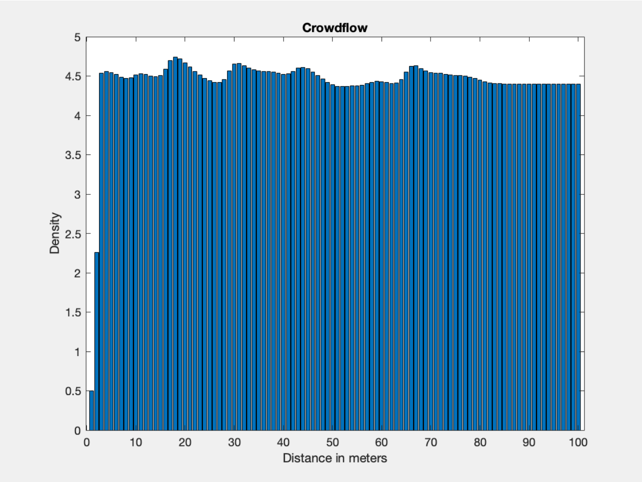


(f) t=465

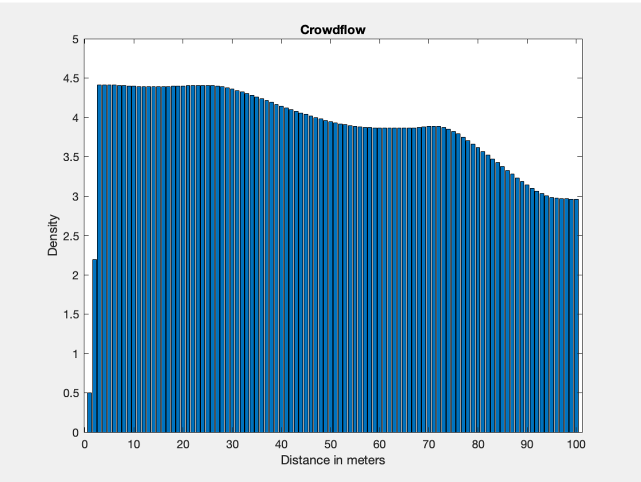
Figure 11: Congestion with low inflow



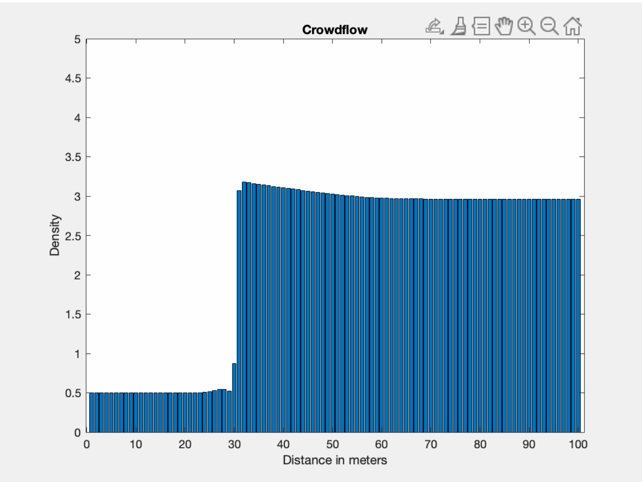
(a) t=0



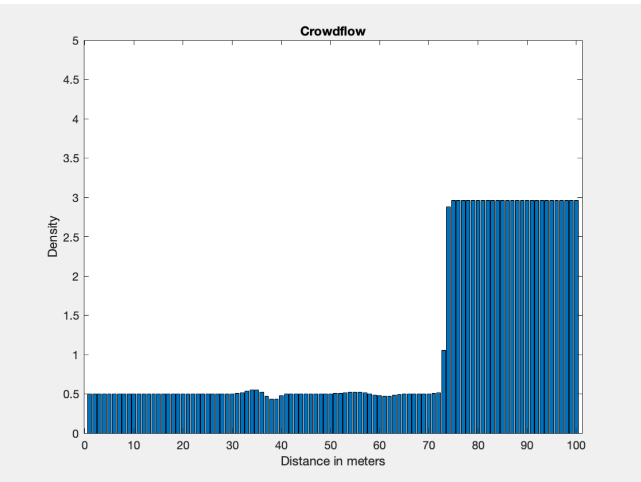
(b) t=22.5



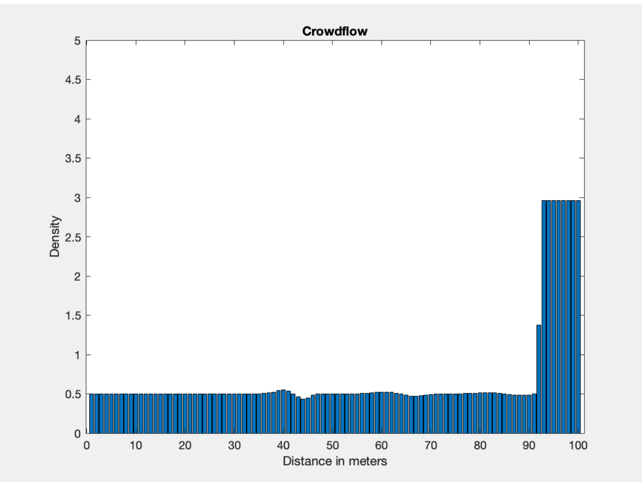
(c) t=102.75



(d) t=264

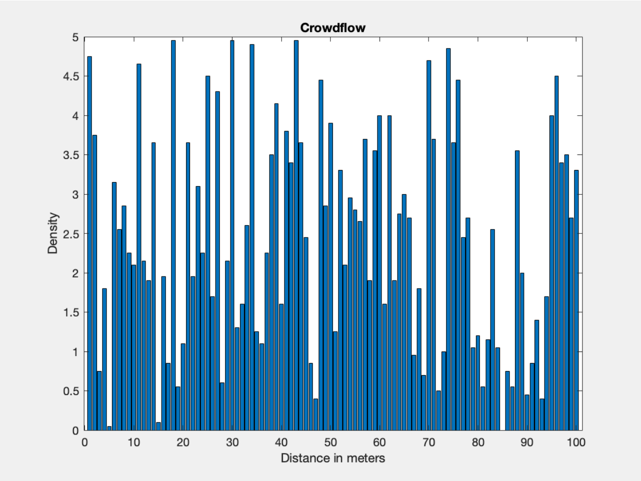


(e) t=369

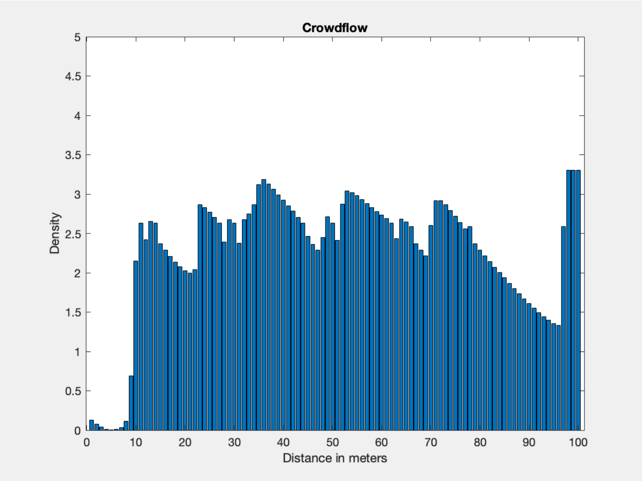


(f) t=414.75

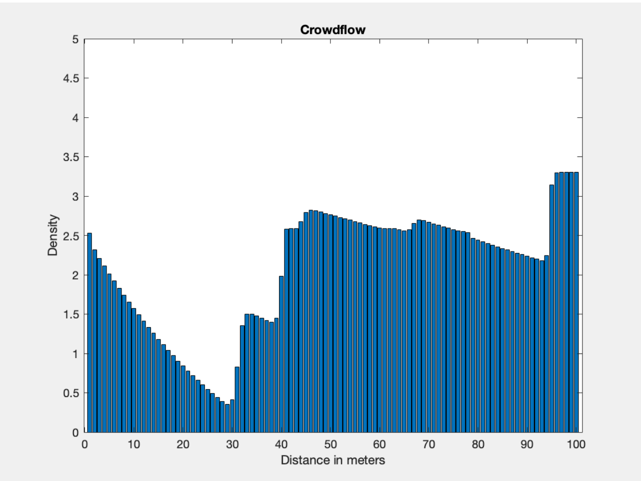
Figure 12: Congestion with increased speed



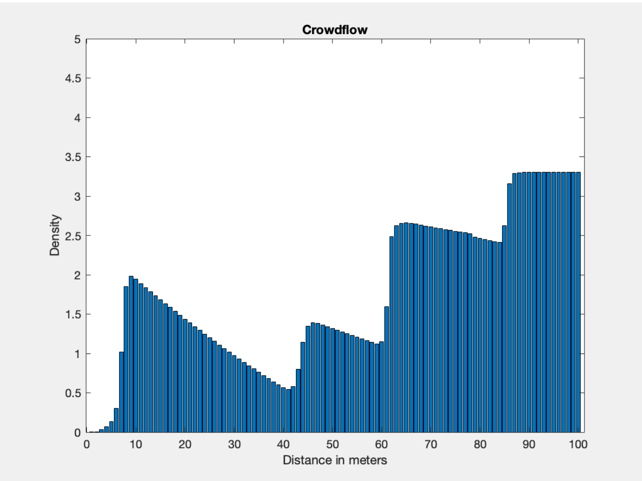
(a) t=0



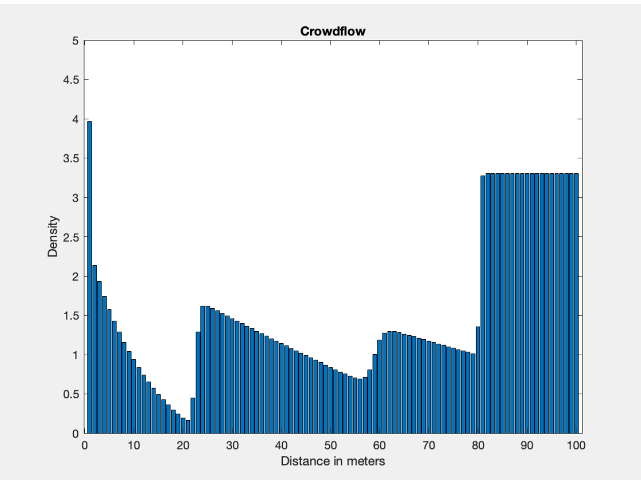
(b) t=27



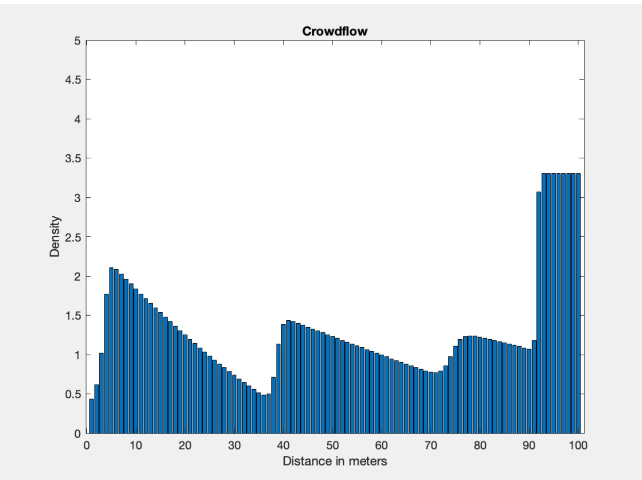
(c) t=102



(d) t=110



(e) t=232.5



(f) t=300

Figure 13: Periodic inflow

4 2D model

In the 2D situation, pedestrians can move in both vertical and horizontal directions. By extending the 1D model to a situation where pedestrians have both a vertical, and horizontal speed, a 2D model can be created. This model also uses the Godunov scheme applied to (1.1).

4.1 Variables

First, the variables of the 2D model will be explained, they are similar to the variables of the 1D model except for some extensions. The first two variables are L_x, and L_y, this is the length of the path in the x-direction and the y-direction, respectively. As with the 1D model, each point in the path has its density ρ the number of pedestrians per m^2. The variable t remains unchanged. Lastly, pedestrians have a certain speed dependent on the density. However, in the 2D situation, there are two speed functions which both are defined as functions of the density in their cell. The speed in the x-direction is the same as in the 1D situation but the speed in the y-direction is different. These speed functions are given by:

$$u(\rho) = u_{\max}\Big(1 - \frac{\rho}{\rho_{\max}}\Big), \tag{4.1}$$

$$v(\rho) = \begin{cases} v_{\max}(1 - \frac{\rho}{\rho_{\max}}) & \text{Downward movement,} \\ -v_{\max}(1 - \frac{\rho}{\rho_{\max}}) & \text{Upward movement.} \end{cases} \tag{4.2}$$

Whether the speed in the vertical y-direction is positive or negative is determined randomly.

The same properties hold as for the 1D situation:

$$u(0) = u_{\max},\ u(\rho_{\max}) = 0.$$
$$v(0) = \pm v_{\max},\ v(\rho_{\max}) = 0.$$

4.2 Numerical scheme

The path in this 2D model will be numerically approximated using the Godunov scheme. Since an extra dimension is present in these equations, a discretization of the t, $\underline{\mathrm{x}}$ and $\underline{\mathrm{y}}$ domains is required for the approximation. The domains of t, x and y will be divided in N_t, N_x and N_y parts of lengths Δt, Δx and Δy, respectively. The equations will again be considered in the points $t_n = n \cdot \Delta t$, $x_j = j \cdot \Delta x$ and $y_k = k \cdot \Delta y$ for $n = 1, \ldots, N_t$, $j = 1, \ldots, N_x$ and $k = 1, \ldots, N_y$, respectively. Again using the notation introduced in a previous section.

$$q(t_n, x_j, y_k) \approx q^n_{k,j}$$

It is again important to realise that $q^n_{k,j}$ is a numerical approximation of the exact solution.

The boundary conditions for the 2D situation are an extension of the boundary conditions in the 1D situation resulting in:

$$\text{Boundary condition: } \rho(t, 0, y_k) = \rho_{0,y_k}(t) \text{ with } t \geq 0,$$
$$\text{Initial condition: } \rho(0, x, y) = \rho^0(x, y).$$

The inflow of pedestrians works the same way in the 2D situation as in the 1D situation. Furthermore, there exist a few extra boundary conditions in the 2D situation. Pedestrians cannot exit the path at the edges. When at the top edge of the path the speed v upwards is 0 and conversely when at the bottom edge of the path the speed v downwards is 0, respectively. This will make it so that pedestrians do not exit out of the path at the top and the bottom.

Using again the same q introduced in (3.4) which satisfies the conservation law. This is then applied to (1.1) and the resulting equation reads:

$$q_t + f(q)_x + g(q)_y = 0. \tag{4.3}$$

The definition of the flux remains unchanged and is defined as:

$$\frac{\#\text{pedestrians}}{m^2 \cdot s},$$

Which is the number of pedestrians that enter or exit a certain area per second. The equation for this is given by:

$$f(q) = u_{\max} q(q-1), g(q) = \pm v_{\max} q(q-1). \tag{4.4}$$

This is similar to the situation in the 1D case. However when $v(\rho) = -v_{\max}$ the speed function

$$v(\rho) = -v_{\max}(1 - \frac{\rho}{\rho_{max}}) \leq 0, v_{\max}.$$

this would then result in.

$$f(q) = -v_{\max} q(q-1).$$

Using this equation it is possible to calculate the derivative, which is given by:

$$a(q) = v_{\max}(2q-1), |a(q)| \leq v_{\max}.$$

However, this means that the second derivative $f''(q) = -2v_{\max} < 0$ as a result the flux function is no longer convex but concave. The solution of the Riemann Problem and the expression of the Godunov flux are the same except that

$$a(q_l) > a(q_r) \Leftrightarrow q_l < q_r,$$
$$a(q_l) < a(q_r) \Leftrightarrow q_l > q_r.$$

Equation (4.3) can be discretized by solving the integral form of this equation which reads:

$$\int_{y_{k-\frac{1}{2}}}^{y_{k+\frac{1}{2}}} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \int_{t_n}^{t_{n+1}} \Big(q_t + f(q)_x + g(q)_y \Big) \mathrm{d}t\mathrm{d}x\mathrm{d}y = 0. \tag{4.5}$$

This equation is again evaluated using the results given by Teyssier in [5] over the control volumes $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}] \times [y_{k-\frac{1}{2}}, y_{k+\frac{1}{2}}]$ and results in:

$$\begin{aligned} &\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \int_{y_{k-\frac{1}{2}}}^{y_{k+\frac{1}{2}}} \Big(q(t_{n+1}, x, y) - q(t_n, x, y) \Big) \mathrm{d}y\mathrm{d}x \\ &+ \int_{t_n}^{t_{n+1}} \int_{y_{k-\frac{1}{2}}}^{y_{k+\frac{1}{2}}} f(q(t, x_{j+\frac{1}{2}}, y)) - f(q(t, x_{j-\frac{1}{2}}, y)) \mathrm{d}y\mathrm{d}t \\ &+ \int_{t_n}^{t_{n+1}} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} g(q(t, x, y_{k+\frac{1}{2}})) - f(g(t, x, y_{k-\frac{1}{2}})) \mathrm{d}x\mathrm{d}t = 0. \end{aligned} \tag{4.6}$$

These integrals will be solved using the finite volume approximation and the time-averaged flux function respectively. The finite volume method says:

$$q^n_{k,j} = \frac{1}{\Delta y} \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \int_{y_{k-\frac{1}{2}}}^{y_{k+\frac{1}{2}}} q(t_n, x, y) \mathrm{d}y\mathrm{d}x. \tag{4.7}$$

Applying (4.7) to the first integral of (4.6) leads to:

$$\Delta y \Delta x (q^{n+1}_{k,j} - q^n_{k,j}) = \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \int_{y_{k-\frac{1}{2}}}^{y_{k+\frac{1}{2}}} q(t_{n+1}, x, y) - q(t_n, x, y) \mathrm{d}y\mathrm{d}x. \tag{4.8}$$

The flux functions are now time and space averaged and give:

$$F^n_{k,j+\frac{1}{2}} = \frac{1}{\Delta t} \frac{1}{\Delta y} \int_{t_n}^{t_{n+1}} \int_{y_{k-\frac{1}{2}}}^{y_{k+\frac{1}{2}}} f(q(t, x_{j+\frac{1}{2}}, y)) \mathrm{d}t\mathrm{d}y. \tag{4.9}$$

$$G^n_{k+\frac{1}{2},j} = \frac{1}{\Delta t} \frac{1}{\Delta x} \int_{t_n}^{t_{n+1}} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} g(q(t, x, y_{k+\frac{1}{2}})) \mathrm{d}t\mathrm{d}x. \tag{4.10}$$

Applying (4.9) to the second integral of (4.6)

$$F^n_{k,j+\frac{1}{2}} - F^n_{k,j-\frac{1}{2}} = \frac{1}{\Delta t} \frac{1}{\Delta y} \int_{t_n}^{t_{n+1}} \int_{y_{k-\frac{1}{2}}}^{y_{k+\frac{1}{2}}} f(q(t, x_{j+\frac{1}{2}}, y)) - f(q(t, x_{j-\frac{1}{2}}, y))) \mathrm{d}t\mathrm{d}y. \tag{4.11}$$

Applying (4.10) to the integral of (4.6) leads to:

$$G^n_{k+\frac{1}{2},j} - G^n_{k+\frac{1}{2},j} = \frac{1}{\Delta t}\frac{1}{\Delta x}\int_{t_n}^{t_{n+1}}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} g(q(t,x,y_{k+\frac{1}{2}})) - g(q(t,x,y_{k-\frac{1}{2}}))\mathrm{d}t\mathrm{d}x. \tag{4.12}$$

Combining equations (4.8) and (4.11), and (4.12) gives:

$$\Delta y\Delta x(q^{n+1}_{k,j} - q^n_{k,j}) + \Delta y\Delta t(F^n_{k,j+\frac{1}{2}} - F^n_{k,j-\frac{1}{2}}) + \Delta x\Delta t(G^n_{k+\frac{1}{2},j} - G^n_{k-\frac{1}{2},j}) = 0 \tag{4.13}$$

$$= q^{n+1}_{k,j} - q^n_{k,j} + \frac{\Delta t}{\Delta x}\Big(F^n_{k,j+\frac{1}{2}} - F^n_{k,j-\frac{1}{2}}\Big) + \frac{\Delta t}{\Delta y}\Big(G_{k+\frac{1}{2},j} - G_{k-\frac{1}{2},j}\Big) = 0. \tag{4.14}$$

From this it can be seen that the 2D Godunov equations are:

$$q^{n+1}_{k,j} = q^n_{k,j} - \frac{\Delta t}{\Delta x}\Big(F^n_{k,j+\frac{1}{2}} - F^n_{k,j-\frac{1}{2}}\Big) - \frac{\Delta t}{\Delta y}\Big(G^n_{k+\frac{1}{2},j} - G^n_{k-\frac{1}{2},j}\Big). \tag{4.15}$$

In this situation in the Godunov scheme $F^n_{k,j+\frac{1}{2}}$, $F^n_{k,j-\frac{1}{2}}$, $G^n_{k+\frac{1}{2},j}$, and $G^n_{k-\frac{1}{2},j}$ are defined as the numerical flux.

The way to solve equation (4.15) is to use the 1D Godunov flux for both F and G. There exist two paths, one where pedestrians move in a direction to the right, and one where they move in an upwards and downwards direction. In other words, the 1D Godunov scheme is applied to every direction of the path.

The choice of whether to use $q^n_{k,j}, q^n_{k,j+1}$ for the numerical flux will be the same as for the 1D situation as explained in 3.18 and 3.20.

For simplicity the notation $F(q_l, q_r)$ is used again. Using the conditions that have been explained in the 1D model, it is now possible to solve equation (4.3).

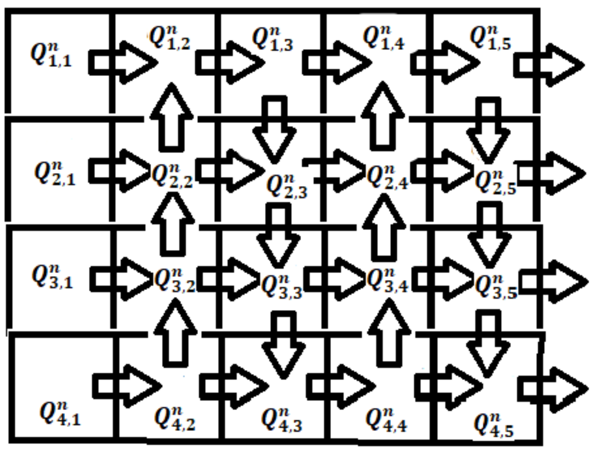


Figure 14: Complete path 2D

In Figure 14 an image of what a simulation at a certain time-step could look like, in this case, x has 5 grid points and y has 4 grid points. The pedestrians enter the path from the left at $x = 1$ and they leave at the last grid point $x = 5$

4.3 Numerical results

The program MATLAB will again be used to simulate the flow of pedestrians. The difference from the 1D model where each time-step all the points in the x-direction were calculated is that in the 2D model each time-step calculates all the points in the x-direction and the y-direction. In the beginning, a $k \times j \times n$ multi-dimensional matrix will be initiated, and each time-step will create a new $k \times j$ matrix. Each row represents the points in the x-direction of the path. While at the same time each column represents the points in the y-direction of the path. The results of this matrix represent the numerical solutions to equation (4.3). The codes can be found in A.2.

4.4 Results 2D

The 2D model will also be validated and verified by looking at the results of the 2D simulation, and seeing if they are correct. The first results specifically will be used to validate the model. Four distinct situations will be simulated, these simulations will start at $t = 0$. The parameters of each simulation are different. In all cases $\Delta t = 0.2$, $\Delta x = 1$, and $\Delta y = 1$

- The first simulation is similar to the one shown in Figure 9. In Figure 15 there will be no inflow of pedestrians meaning $\rho(t, 0, y) = 0$ and the existing pedestrians will be placed randomly on the path. The expected result is that after several time steps the path will be empty.
- In the second simulation shown in Figure 16 two empty paths with constant inflows will be tested and compared. In this case, the inflow of the path on the left is set to $\rho = 2.5$ and the inflow of the path on the right is set to $\rho = 0.75$.
- In the third simulation shown in Figure 17 a small portion of the path has a high density. The rest of the path is distributed randomly with the inflow being $\rho(t, 0, y) = 2.5$. In this scenario, the effect of a congestion can be seen. The congestion could either spread to the whole path or dissolve.
- In the fourth simulation shown in Figure 18 the entire path has a high density and the inflow has been set to $\rho = 2.5$.

4.4.1 No inflow scenario 2D

In Figure 15 the first simulation is the same situation as in the 1D model. The densities of the path are randomly distributed and there is currently no inflow of pedestrians. In this situation, there are again areas where the density is higher as can be seen in Figure 15b. Similarly to the 1D case, a shock wave arises. However due to the higher density at the beginning of the path the shock speed at the beginning is different as $q_L = 1$ and $q_r \approx 0.35$ this results in a shock speed of $s \approx 0.525\text{m}^2\ \text{s}^{-1}$. In Figures 15b and 15c the pedestrians have moved around 8 meters, if they moved with the shock speed given above they should have moved 7.665 meters in that time frame. After this the density at the beginning of the path reduces to around $\rho = 2.5$ in Figure 15d, this results in $q_r \approx 0.4$ and as a result, the shock speed is $s \approx 0.5\text{m}^2\ \text{s}^{-1}$. With this shock speed, the pedestrians should have moved 12 meters going from $t = 36$ to $t = 60$. This also seems to be the case. However, it is not easy to calculate when the path will be empty since when the density lowers, the shock speed also decreases.

4.4.2 Constant inflow scenario 2D

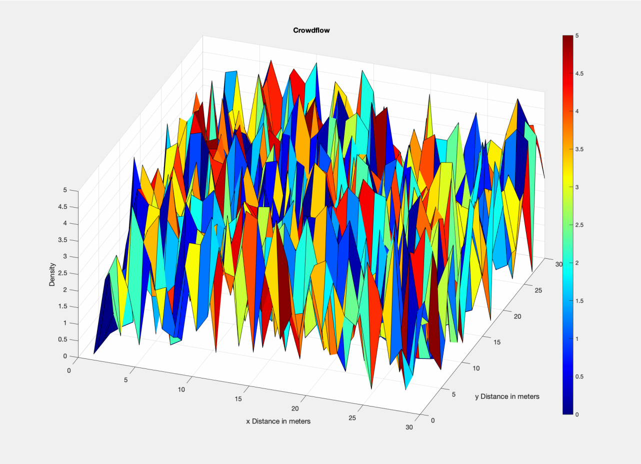
In Figure 16 there is an empty path. However, there are two distinct situations. Both paths have a constant inflow but the path on the left has an inflow of $\rho = 2.5$ and the path on the right has an inflow of $\rho = 0.75$. Comparing Figures 17c and 17d, it is clear how much the inflow impacts the rest of the path in the 2D situation. It is important to realise that if $\rho = 2.5$ then $q = 0.5$. 0.5 is also the value for which $a(q)$ reaches its maximum value and thus this is the value for which the flux is highest. This is why the path in Figure 17e fills up faster.

4.4.3 Congestion scenario

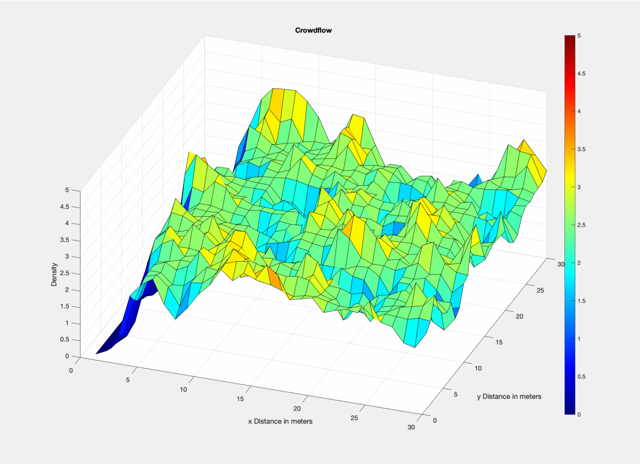
In Figure 17, it is explored how congestions occur. If the path is empty at the start a congestion is not likely to occur. One way to start a congestion is to let the situation at $t = 0$ be congested. In Figure 17b it can be seen that the density immediately starts to reduce. However, because of the high density, pedestrians are not able through the path quickly. This results in a congestion occurring, as can be seen in Figure 17c where the congestion has spread to a longer area of the path. This is also the case for Figure 17d. However, the density has now started to decrease and after a while, the congestion dissolves. The effect of the congestion can still be seen in the path as the density is still more than in the scenario in Figure 17e. In the situation, a shock wave arises, however in this case the shock speed is negative.

4.4.4 True congestion scenario

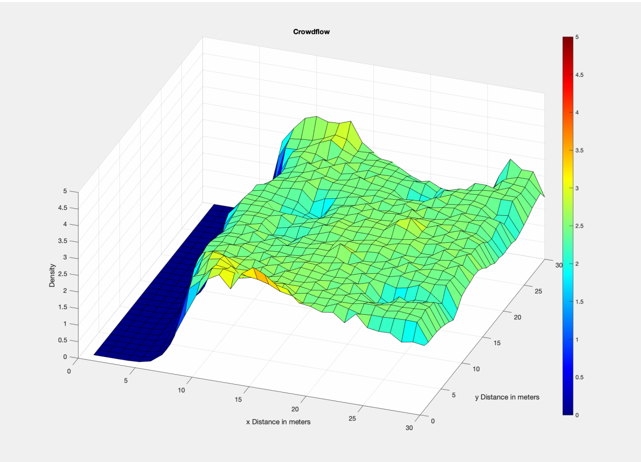
In Figure 18, a situation is shown where the entire path is congested. It can be seen that in this case, it is even possible for a congestion to never dissolve. If the inflow is too high. Even after 400 seconds, the path is still congested. This is because the inflow is too high for the congestion to dissolve.



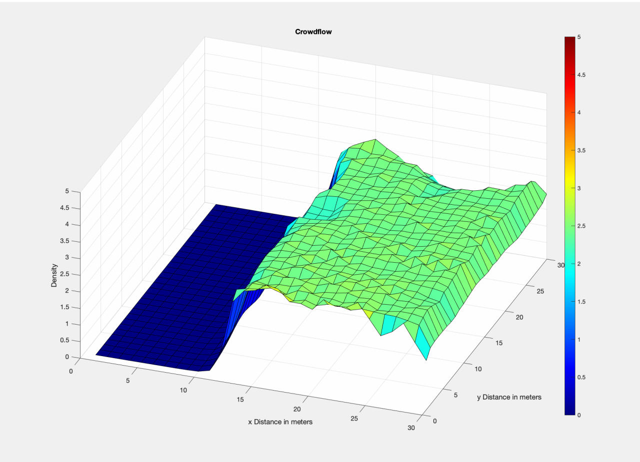
(a) t=0



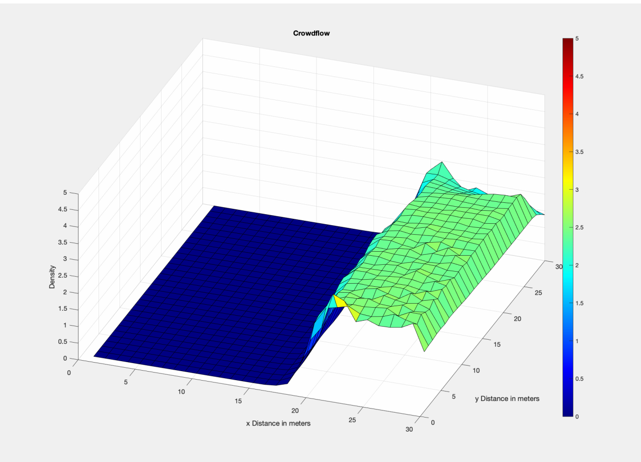
(b) t=5.4



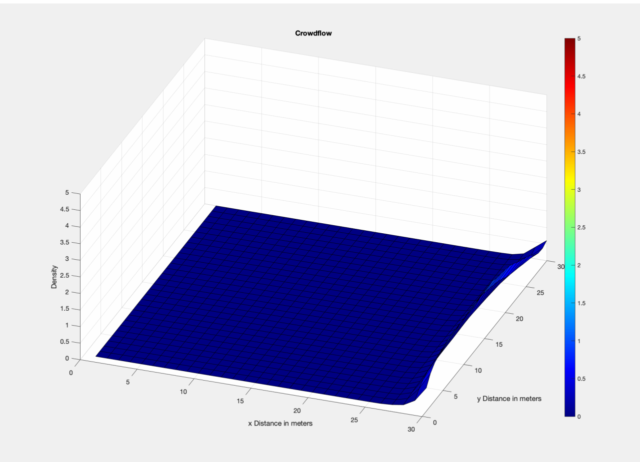
(c) t=20



(d) t=36

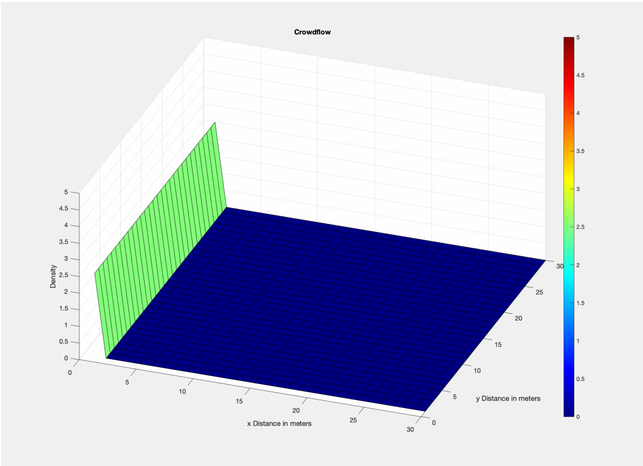


(e) t=60

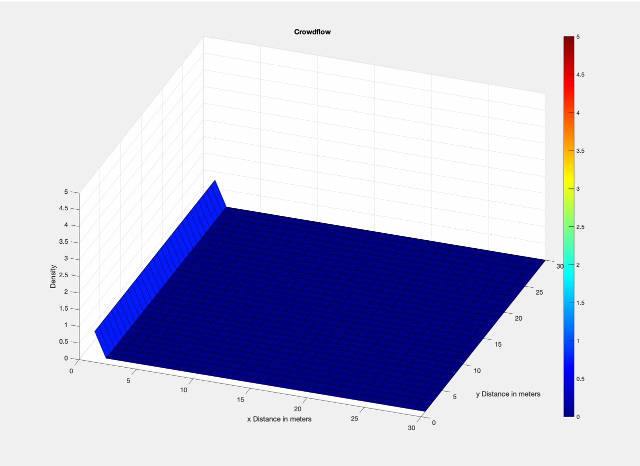


(f) t=93

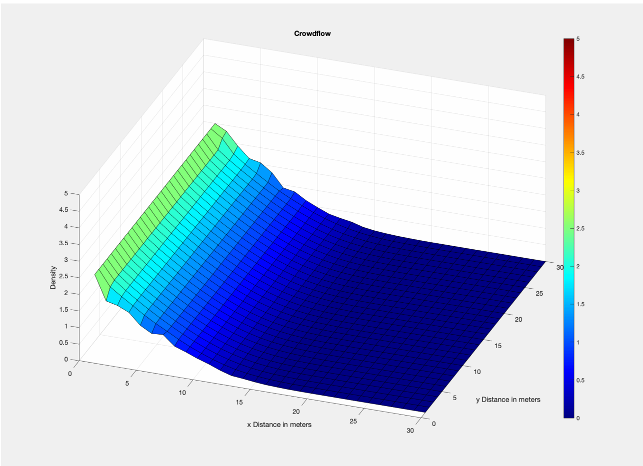
Figure 15: Empty path 2D



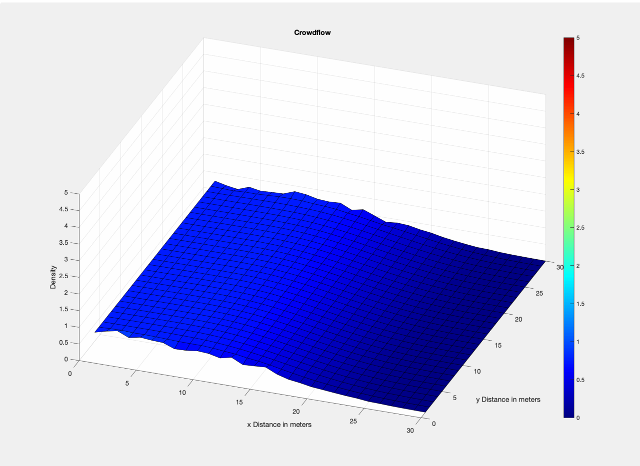
(a) t=0, inflow $\rho = 2.5$



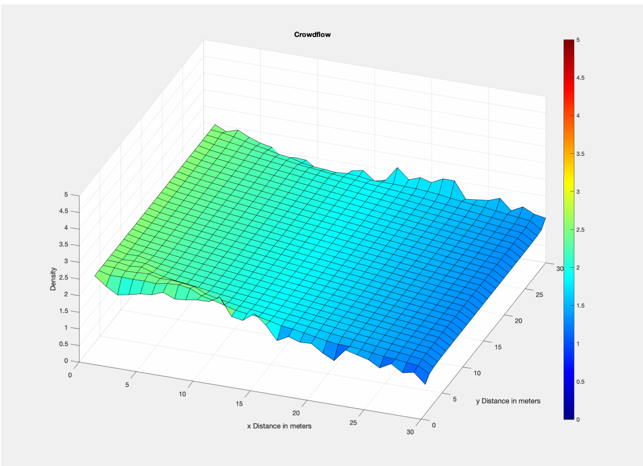
(b) t=0, inflow $\rho = 0.5$



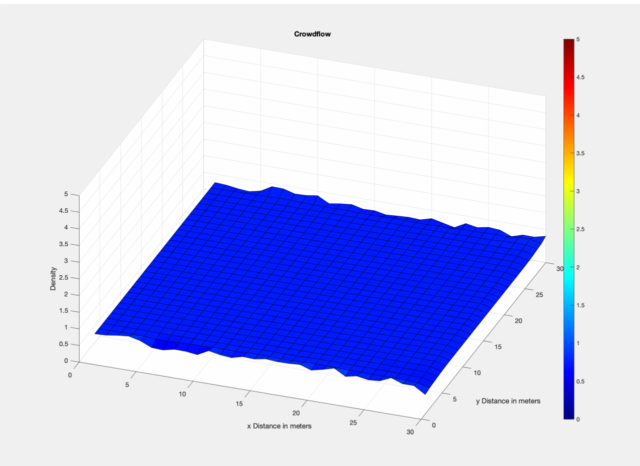
(c) t=16, inflow $\rho = 2.5$



(d) t=16, inflow $\rho = 0.5$

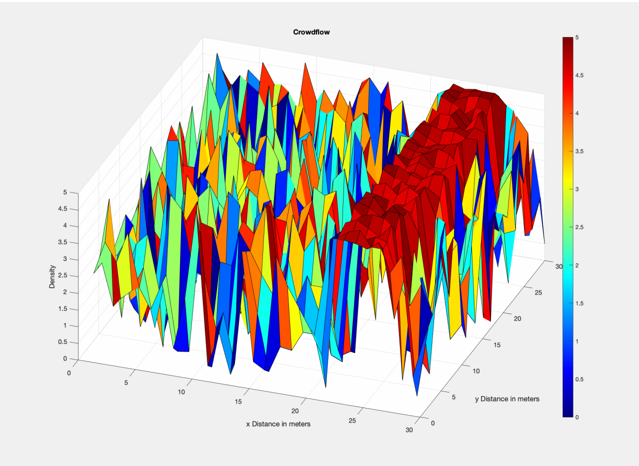


(e) t=100, inflow $\rho = 2.5$

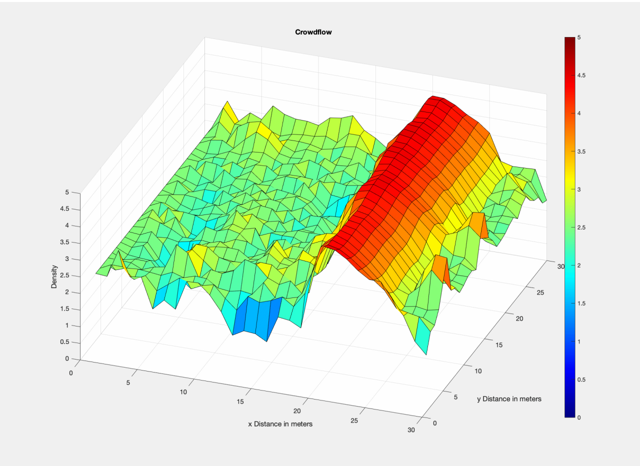


(f) t=100, inflow $\rho = 0.5$

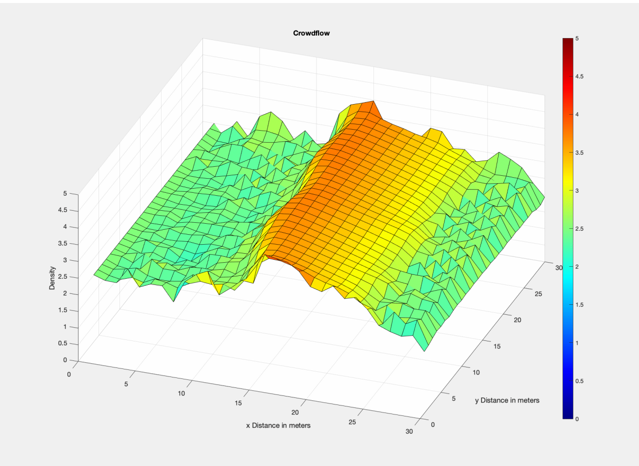
Figure 16: Inflow comparison



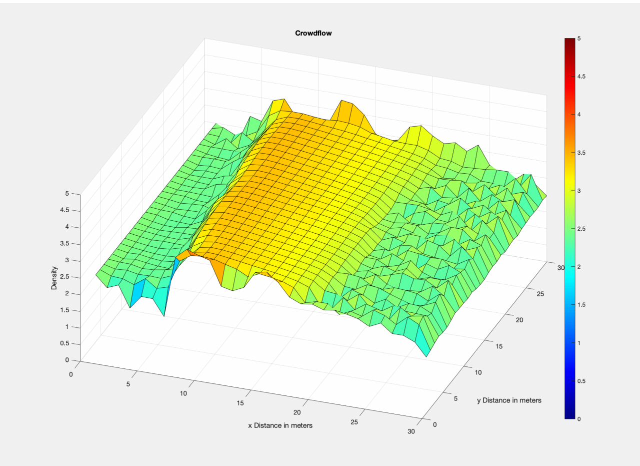
(a) t=0



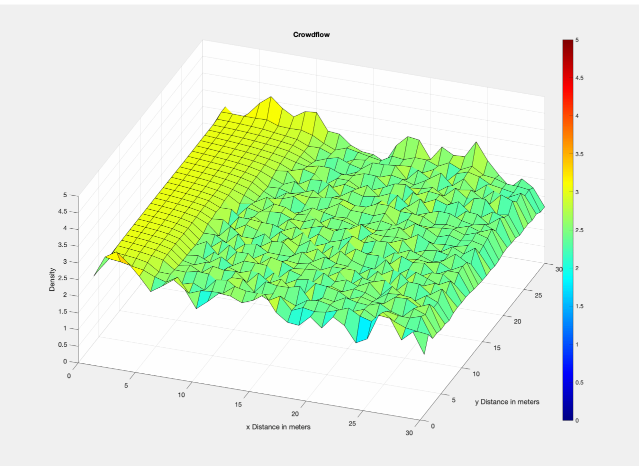
(b) t=5



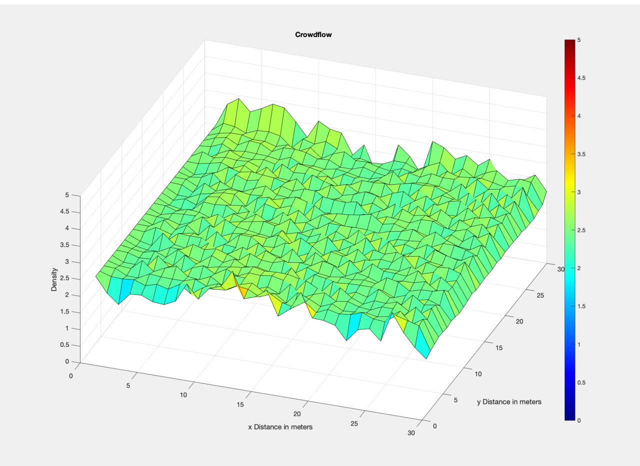
(c) t=20



(d) t=41

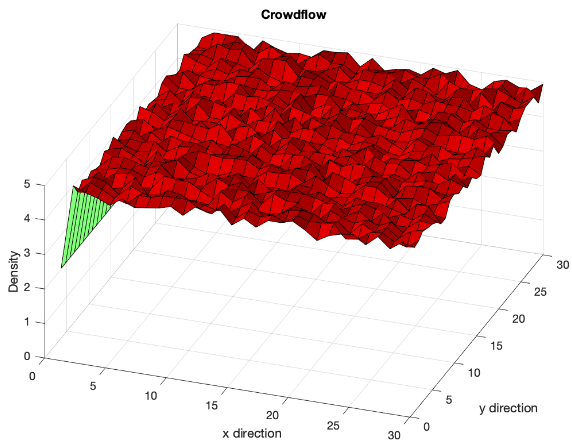


(e) t=100

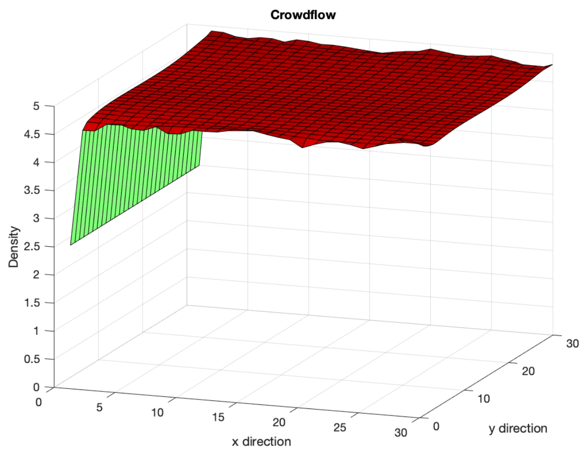


(f) t=140

Figure 17: Effect of a congestion



(a) t=0



(b) t=400

Figure 18: Congestion forever

5 Nudge

During the ISBEP the main problem statement was centred around reducing congestion and thinking of ways to reduce congestion. If it is possible to reduce the density of the path then there would occur fewer congestions. The research was done to look for ways to influence the way pedestrians walked. This nudge as it is called will serve that purpose.

The nudge that has been investigated is a nudge that uses optic flow in the form of a light effect that creates a wave of light moving over the head of pedestrians. This works opposite to the road markings in the centre of the road that is meant to make people aware of their current speed as described in [6]. These lines make people slow down if the lines pass them by too quickly.

The effect of the nudge was tested in a corridor at the TU/e. It was done by tracking people when the nudge was active and inactive and comparing the results. From the research, it seemed that pedestrians that walked there when the nudge was active did not speed up as described by J. Pullen in [4]. However, it is still interesting to see whether or not a higher speed leads to a decrease in density. To see how an increased speed affects the path it will be assumed that the nudge increases pedestrian speed by 10%.

5.1 Numerical scheme

The numerical scheme is unchanged compared to the scheme for the 2D model. The only thing that has been added is the variable Lightf. This variable determines how much faster pedestrians move when they are influenced by the nudge. In this report lightf has been set to 1.1, meaning that pedestrians move 10% faster when influenced by the nudge. The nudge only increases the speed of pedestrians in the x-direction. The way this works is that u_{max} is multiplied by the variable lichtf.

5.2 Results

The results are gathered in the same way as in the previous section. The differences are that in this case for the grid points between $10 < y < 25$ and $x > 1$ the nudge is in effect and that the pedestrians in that part of the path are moving faster than the rest. The nudge will start when $x > 1$. This is done so that the number of people entering the path will be the same as in the situation where there is no nudge active. This will make it easier to compare different situations.

To investigate the effect of the nudge two distinct situations will be compared. The starting density will stay the same for each of these simulations and is randomly distributed. The starting inflow is also the same for each of these situations namely $\rho = 0.5$. Furthermore, $\Delta x = 1$, $\Delta y = 1$, and $\Delta t = 0.1$

- In the he first situation see Figures 19 and 20 the situation without and with the nudge will be compared. Two normal paths with random densities will compare the effect of the nudge.
- The second simulation shown in Figure 21 will test if the nudge can solve congestions faster. It will now be applied to a situation where there is already a congestion similar to the situation which occurred in

Figure (17) from the previous section and compare the densities.

5.2.1 Nudge scenario

In Figure 20 the nudge is active as can be seen. The middle part is where the nudge is active it can be seen that the density is lower than in the rest of the path.

The density of pedestrians at the top and the bottom of the path is now also slightly less crowded than it was before. To answer the question if the nudge is effective, the average densities of the two paths will be compared at different time points.

Table 1: Nudge comparison normal path

Time in seconds	30	120	200
Nudge	2.4409	2.2387	2.0914
No nudge	2.5135	2.5097	2.5177

From Table 1 it is possible to compare the paths at different time points. It is not hard to see that the average density of the path without the nudge is higher compared to the density of the path with the nudge, even at the beginning of the simulation after only 30 seconds the effect can be seen as the average density is 3% lower. This increases to 11% after 120 seconds. After 200 seconds this even increases further to 17%.

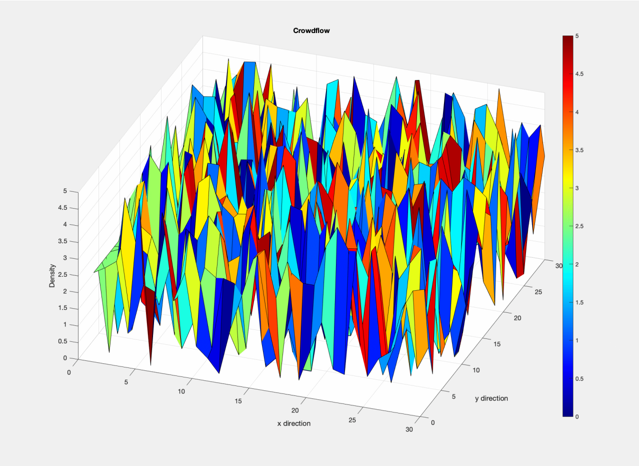
5.2.2 Nudge applied to congestion scenario

The results for the nudge in such a situation are still positive as can be seen in Table 2, even though there are some areas where the density is higher than without the nudge, on average there is an improvement to the density. The average densities are compared in the table below.

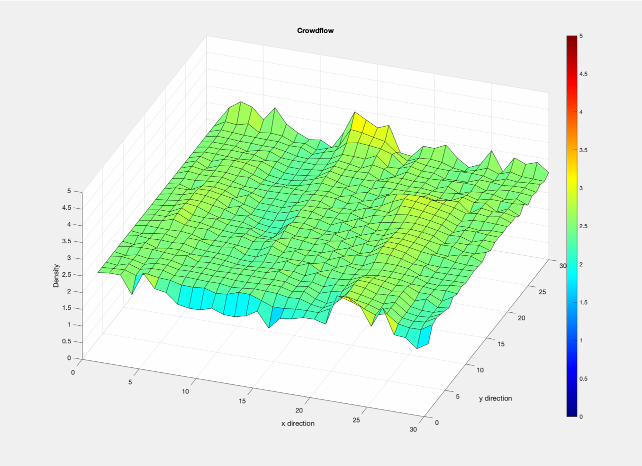
Table 2: Nudge comparison congestion

Time in seconds	7	20	30	40	55
Nudge	2.9620	2.9312	2.9068	2.8836	2.8488
No nudge	2.9877	3.0083	3.0307	3.0572	3.0710

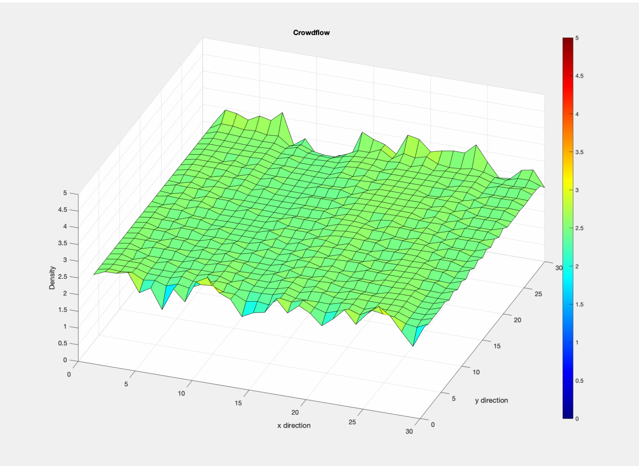
From the table, it can be seen that if a nudge speeds up people by 10% this will decrease the average density by 4.1% after 30 seconds. This increases to 7.1% after 55 seconds. More importantly, it can be seen that without the nudge the average density increases because there is a congestion in the path. In the situation with the nudge, however, the average density decreases.



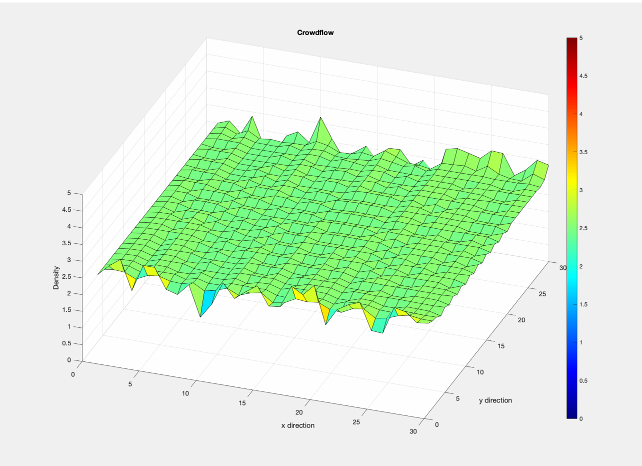
(a) t=0.05



(b) t=30

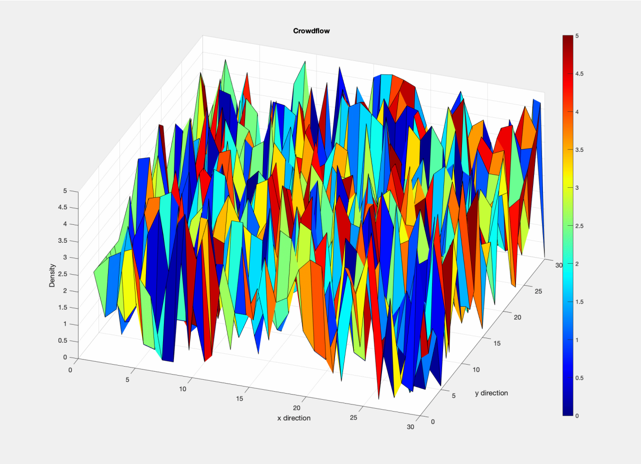


(c) t=120

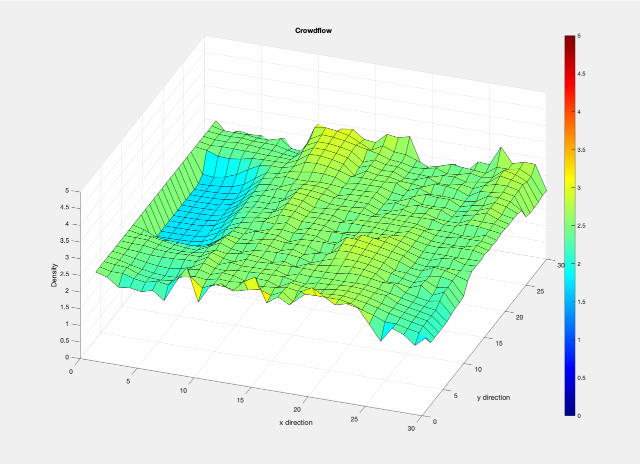


(d) t=200

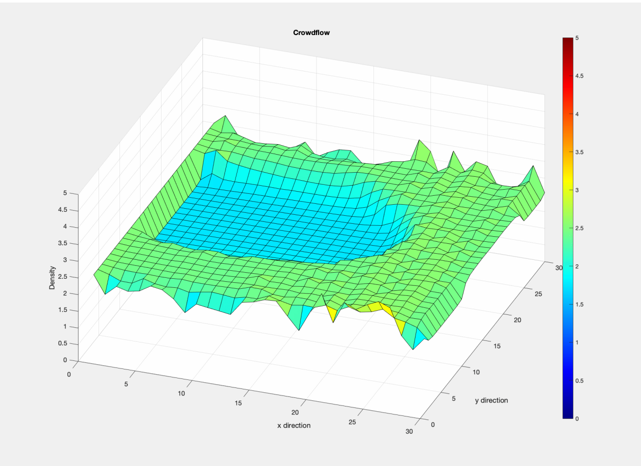
Figure 19: Path without nudge



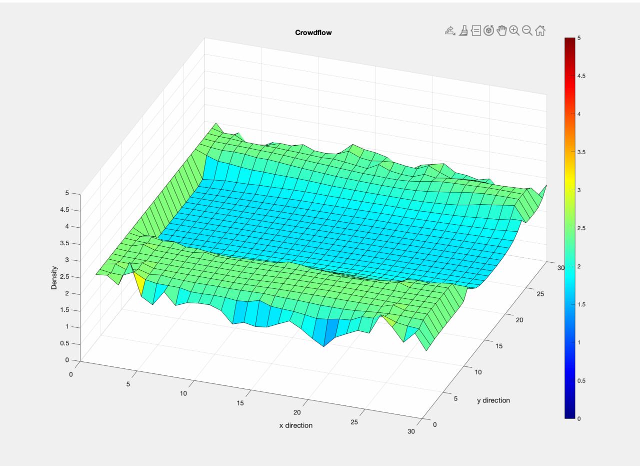
(a) t=0



(b) t=30

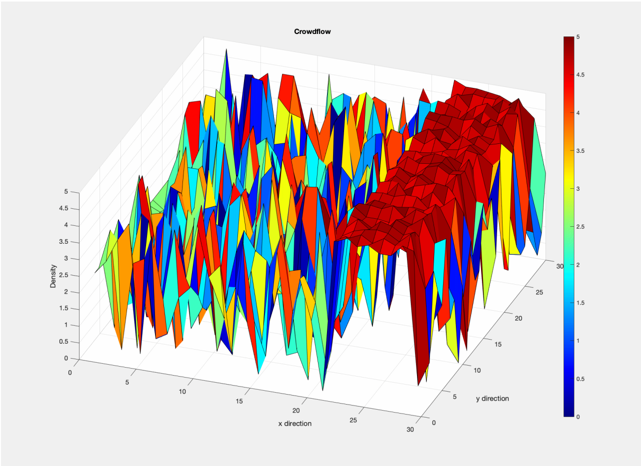


(c) t=120

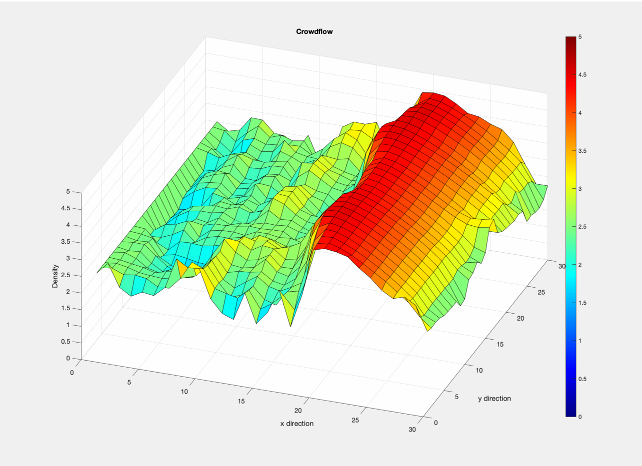


(d) t=200

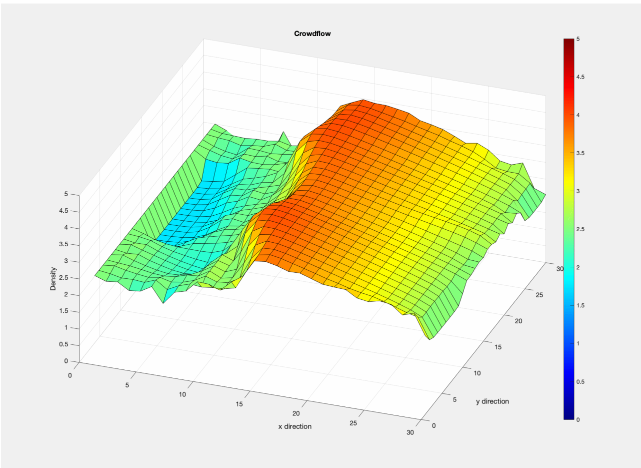
Figure 20: Path with nudge



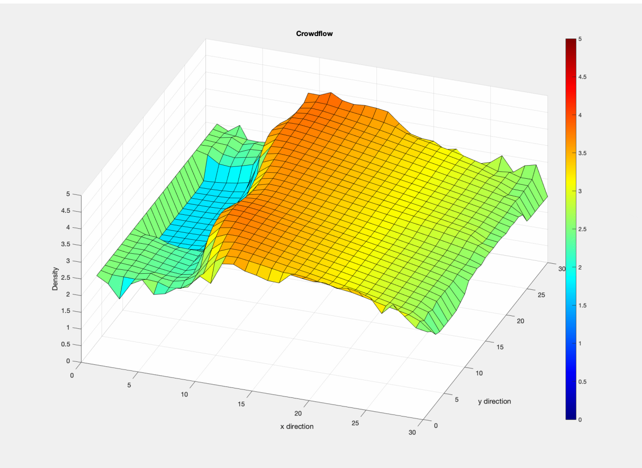
(a) t=0



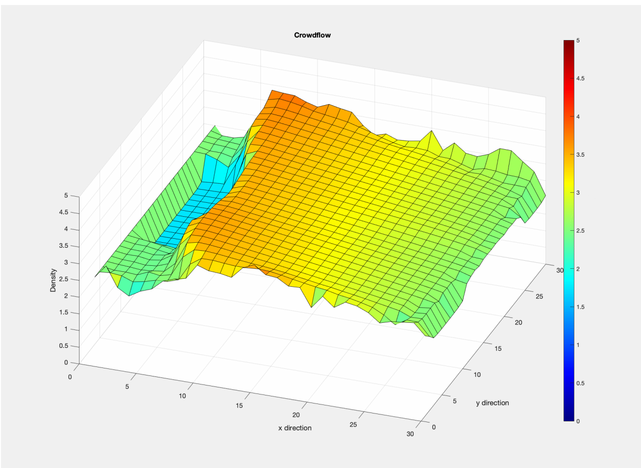
(b) t=7



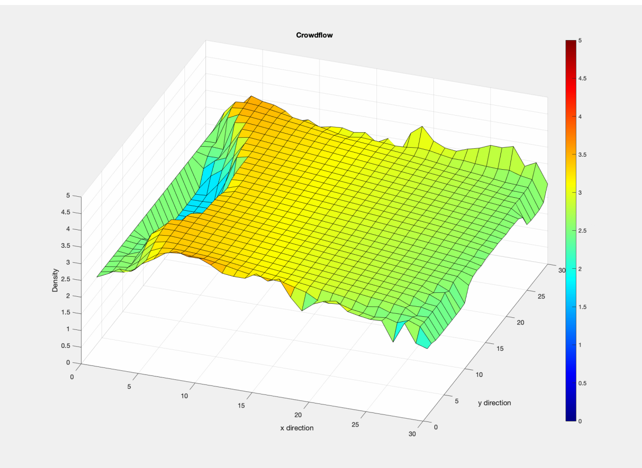
(c) t=20



(d) t=30



(e) t=40



(f) t=55

Figure 21: Congestion with nudge

6 Conclusion

6.1 Main results

The 1D model gives an insight into how pedestrians move in a crowd. It is a relatively simple model that helps verify the model. In the 1D simulation the effect of the shock wave is easy to observe as was seen in Figure 10. In Figure 11 it was noticeable that when a congestion occurs it can take a long time before it is dissolved. The time that it takes for a congestion to dissolve can be reduced by increasing the speed of pedestrians as was seen in Figure 12. Another way to reduce congestion is to implement a limit on how many people can enter a path as is seen in Figure 13.

The 2D model gives an insight into a more realistic model how pedestrians walk. By adding the different directions that pedestrians are able to walk in, it is easy to see what the effect of a congestion is on an entire path. For simple simulations it is still possible to see the effect of a shock wave as is seen in Figure 15. Figure 16 displays the effect of the inflow of a path. In Figure 17 the lingering effect of congestion was seen. Finally in Figure 18 it can be seen that some congestions cannot be solved if the inflow is too high. In the 2D model it can clearly be seen that dissolving a congestion as quickly as possible is the most beneficial for the flow of the path as the effect of a congestion can hinder the flow of the path for a long time.

By extending the 2D model to a model where the nudge can be applied gives interesting results. Even though there are some points in the path where the density increases, on average the density decreases. The nudge not only helps reduce density but it also helps to solve congestions faster. Similarly to the 1D path where a congestion was resolved quicker when the pedestrians move faster. Though in the 1D situation an increase in speed of 10% was a lot more substantial than in the 2D situation.

6.1.1 Answer to the problem statement

From the theoretical results gathered from the simulations, if a nudge exists that would make people move 10% faster it would definitely be worth to implement. However, since there are some areas where the density increases it would be best if the nudge would be applied to the entire path and not just a small portion of the path.

There are some problems for such a nudge. It is that it is very difficult to test if it works. The negative effects of the nudge have not been tested, it could be possible that the pedestrians walking in the opposite direction of the nudge are slowed down by the nudge. The last problem with the nudge is that most pedestrians are not consciously walking, some are on their phone or are simply not paying attention. All of these factors could decrease the effectiveness of the nudge. However, implementing a nudge based on optical flow is relatively cheap and is potentially very beneficial if one is able to find one the works.

7 Discussion

When I started this project it seemed like a difficult task. This was because solving the continuity equation which is a partial differential equation is not something that is often done during the Applied Mathematics bachelor. The first problem that arose when the continuity equation was being solved was that at first to solve it a standard upwind scheme was being used. For the 1D situation this resulted in:

$$\rho_j^{n+1} = \rho_n^j - \frac{\Delta t}{\Delta x}\Big(\rho_j^n u_j^n - \rho_{j-1}^n u_{j-1}^n\Big) = 0.$$

By solving this equation using MATLAB it is again possible to simulate pedestrians along a path similarly to what has been done in the report. However, there is one problem with this equation. When it gets very crowded in point j the flux of pedestrians going out of that point decreases. As ρ_{max} is approached the outgoing flux goes to zero. However, the incoming flux does not decrease and a situation is possible where $\rho_j > \rho_{max}$ this is a contraction to what has been assumed. This is when the Godunov scheme was introduced as this has been used before for the modelling of pedestrians.

The 2D Godunov scheme was not so easy to implement as it seemed in the beginning. In the slides from ([5]) such an equation is solved by first doing a step in the x-direction and then in the y-direction. In the next time step first compute a y-step and then an x-step. In this report I have drawn inspiration from that method but instead of doing it at a fixed rate the y-step direction is random.

Since the model is a simplification there were some limitations such as pedestrians only walking one way in the x-direction. For future research it would be interesting to see if the model that was built could be extended. This could be done for example by adding different types of pedestrians such as old or young pedestrians, these pedestrians would then have different maximum speeds. Another option would be to change the shape of the path or add areas to the path where pedestrians cannot go. A disadvantage of the nudge is that it will not work if people are not paying attention to the nudge by being on their phones, it might even be the case that people get used to a certain nudge if it is used often. The theoretical results were promising. However, further research is recommended to draw a definitive conclusion.

8 References

[1] K. Alsinan. Shock wave and rarefaction wave.

[2] Roger L. Hughes. A continuum theory for the flow of pedestrians. Vol. 1306, 2001.

[3] NASA. Euler equations. `https://www.grc.nasa.gov/www/k-12/airplane/eulereqs.html`, accessed June 5, 2022.

[4] Jeroen Pullen. Determining the influence of light-based accelerating optic flow manipulations on pedestrians over time.

[5] O. Agertz R. Teyssier. The Godunov method, 2009. Lecture slides 4 Computational Astrophysics.

[6] N. Malhotra S. G. Charlton, N. J. Starkey. Using road markings as a continuous cue for speed choice. *Accident Analysis Prevention*, 117:288–297, 8 2018.

A MATLAB Code

A.1 1D simulation

```
clear
clc
umax = 1.25;
pmax = 5;
L=100;
Nx=100;
dx = L/(Nx-1);
Nt=3-00;
tmax = 180;
dt = tmax/Nt;

rho=zeros(Nt,Nx);
q=zeros(Nt,Nx);
F=zeros(Nt,Nx);
a=zeros(Nt,Nx);
s=zeros(Nt,Nx);

q(1,:)=1/100*randi([1,100],1,(Nx));

% for n=1:Nt
%      q(n,1)=abs(sin(n/100));
% end

q(:,1)=0.5;


for n=1:(Nt)
    for j=(1:Nx)
        a(n,:)=2.5*q(n,:)-umax;
        f(n,j)=(umax*q(n,j)*(q(n,j)-1));
        if j<Nx && q(n,j)==q(n,j+1)
            F(n,j)=umax*(q(n,j)*(q(n,j)-1));
        end
        if j<Nx && a(n,j)>a(n,j+1)
            s(n,j)=(umax*q(n,j+1)*(q(n,j+1)-1)-
            (umax*q(n,j)*(q(n,j)-1)))/(q(n,j+1)-q(n,j));
            if s(n,j)>0
                F(n,j)=umax*q(n,j)*(q(n,j)-1);
            elseif s(n,j)<0
                F(n,j)=umax*q(n,j+1)*(q(n,j+1)-1);
             elseif s(n,j)==0
                 F(n,j)=umax*q(n,j)*(q(n,j)-1);
            end
        end

        if j<Nx && a(n,j)<a(n,j+1)
            if a(n,j)>=0
                F(n,j)=umax*q(n,j)*(q(n,j)-1);
            elseif a(n,j+1)<0
                F(n,j)=umax*q(n,j+1)*(q(n,j+1)-1);
            elseif a(n,j)<0 && 0<a(n,j+1)
                F(n,j)=-0.3125;
            end
        end

        if j > 1 && j<Nx
            q(n+1,j)=q(n,j)-(dt/dx)*(F(n,j)-F(n,j-1));
        elseif j == Nx
            q(n+1,j)=q(n,j)-(dt/dx)*(umax*(q(n,j)*(q(n,j)-1))-F(n,j-1));
        end
    rho(n,j)=(1-q(n,j))*pmax;
```

```
    end
end



h = figure;
axis tight manual
filename = '1dgodunov.gif';
for n = 1:Nt
    bar(rho(n,:))
    ylim([0 5])
    title('Crowdflow')
    xlabel('Distance')
    ylabel('Density')
    drawnow
    frame = getframe(h);
    im = frame2im(frame);
    [imind,cm] = rgb2ind(im,256);
    if n == 1
        imwrite(imind,cm,filename,'gif', 'Loopcount',inf);
    else
        imwrite(imind,cm,filename,'gif','WriteMode','append');
    end
end
```

A.2 2D simulation

```

clear
clc
umax = 1.25;
vmax = 0.4;
pmax = 5;
Lx=30;
Ly=30;
Nx=30;
Ny=30;
dx = Lx/(Nx-1);
dy = Ly/(Ny-1);
Nt=5000;
tmax = 90;
dt = tmax/Nt;



rho=zeros(Ny,Nx,Nt);
q=zeros(Ny,Nx,Nt);
F=zeros(Ny,Nx,Nt);
G=zeros(Ny,Nx,Nt);
G=zeros(Ny,Nx,Nt);
a=zeros(Ny,Nx,Nt);
s=zeros(Ny,Nx,Nt);
V=randi([1,2],1,Nx,Nt);






q(:,:,1)=1/100*randi([1,100],Ny,(Nx),1);

q(:,1,:)=0.5;



```

```
for n=1:(Nt)
    for j=(1:Nx)
        for k=(1:Ny)
            a(:,:,n)=2.5*q(:,:,n)-umax;


            if j<Nx && q(k,j,n)==q(k,j+1,n)
                F(k,j,n)=umax*(q(k,j,n)*(q(k,j,n)-1));

            end
            if j<Nx && a(k,j,n)>a(k,j+1,n)
                s(k,j,n)=(umax*q(k,j+1,n)*(q(k,j+1,n)-1)
                -(umax*q(k,j,n)*(q(k,j,n)-1)))/(q(k,j+1,n)-q(k,j,n));
                if s(k,j,n)>0
                    F(k,j,n)=umax*q(k,j,n)*(q(k,j,n)-1);
                elseif s(k,j,n)<0
                    F(k,j,n)=umax*q(k,j+1,n)*(q(k,j+1,n)-1);
                elseif s(k,j,n)==0
                    F(k,j,n)=umax*q(k,j,n)*(q(k,j,n)-1);
                end
            end
            if j<Nx && a(k,j,n)<a(k,j+1,n)
                if a(k,j,n)>=0
                    F(k,j,n)=umax*q(k,j,n)*(q(k,j,n)-1);
                elseif a(k,j+1,n)<0
                    F(k,j,n)=umax*q(k,j+1,n)*(q(k,j+1,n)-1);
                elseif a(k,j,n)<0 && 0<a(k,j+1,n)
                    F(k,j,n)=-0.3125*1.1;
                end
            end


            if V(1,j,n)==1
                if k<Ny && q(k,j,n)==q(k+1,j,n)
                    G(k,j,n)=vmax*(q(k,j,n)*(q(k,j,n)-1));

                end
                if k<Ny && a(k,j,n)>a(k+1,j,n)
                    s(k,j,n)=(vmax*q(k+1,j,n)*(q(k+1,j,n)-1)
                    -(vmax*q(k,j,n)*(q(k,j,n)-1)))/(q(k+1,j,n)-q(k,j,n));
                    if s(k,j,n)>0
                        G(k,j,n)=vmax*q(k,j,n)*(q(k,j,n)-1);
                    elseif s(k,j,n)<0
                        G(k,j,n)=vmax*q(k+1,j,n)*(q(k+1,j,n)-1);
                    elseif s(k,j,n)==0
                        G(k,j,n)=vmax*q(k,j,n)*(q(k,j,n)-1);
                    end
                end
                if k<Ny && a(k,j,n)<a(k+1,j,n)
                    if a(k,j,n)>=0
                        G(k,j,n)=vmax*q(k,j,n)*(q(k,j,n)-1);
                    elseif a(k+1,j,n)<0
                        G(k,j,n)=vmax*q(k+1,j,n)*(q(k+1,j,n)-1);
                    elseif a(k,j,n)<0 && 0<a(k+1,j,n)
                        G(k,j,n)=-0.3125;
                    end
                end

            elseif V(1,j,n)==2
                if k>1 && q(k,j,n)==q(k-1,j,n)
                    G(k,j,n)=vmax*(q(k,j,n)*(q(k,j,n)-1));

                end
                if k>1 && a(k,j,n)>a(k-1,j,n)
```

```
                s(k,j,n)=(vmax*q(k-1,j,n)*(q(k-1,j,n)-1)
                -(vmax*q(k,j,n)*(q(k,j,n)-1)))/(q(k-1,j,n)-q(k,j,n));
                if s(k,j,n)>0
                    G(k,j,n)=vmax*q(k,j,n)*(q(k,j,n)-1);

                elseif s(k,j,n)<0
                    G(k,j,n)=vmax*q(k-1,j,n)*(q(k-1,j,n)-1);

                elseif s(k,j,n)==0
                    G(k,j,n)=vmax*q(k,j,n)*(q(k,j,n)-1);
                end
            end
            if k>1 && a(k,j,n)<a(k-1,j,n)
                if a(k,j,n)>=0
                    G(k,j,n)=vmax*q(k,j,n)*(q(k,j,n)-1);
                elseif a(k-1,j,n)<0
                    G(k,j,n)=vmax*q(k-1,j,n)*(q(k-1,j,n)-1);
                elseif a(k,j,n)<0 && 0<a(k-1,j,n)
                    G(k,j,n)=-0.3125;
                end
            end
        end
    end
end



umaxr=1.25*1.1;
umaxb=0;
umaxo=0;
if vmax==0
    G(:,:,:)=0;
end
for j=(1:Nx)
    for k=(1:Ny)
        if V(1,j,n)==1
            if j > 1 && k>1 && k<Ny && j<Nx
                q(k,j,n+1)=q(k,j,n)-(dt/dx)*(F(k,j,n)-F(k,j-1,n))
                -(dt/dy)*(G(k,j,n)-G(k-1,j,n));
            elseif k==1 && j==Nx
                q(k,j,n+1)=q(k,j,n)-(dt/dx)*(umaxr*(q(k,j,n)*(q(k,j,n)-1))-F(k,j-1,n))
                -(dt/dy)*(umaxb*(q(k,j,n)*(q(k,j,n)-1)))-(dt/dy)*(G(k,j,n));
            elseif k==Ny && j==Nx
                q(k,j,n+1)=q(k,j,n)-(dt/dx)*(umaxr*(q(k,j,n)*(q(k,j,n)-1))-F(k,j-1,n))
                -(dt/dy)*(umaxo*(q(k,j,n)*(q(k,j,n)-1))-G(k-1,j,n));
            elseif j > 1 && k==1
                q(k,j,n+1)=q(k,j,n)-(dt/dx)*(F(k,j,n)-F(k,j-1,n))
                -(dt/dy)*(umaxb*(q(k,j,n)*(q(k,j,n)-1)))-(dt/dy)*(G(k,j,n));
            elseif j > 1 && k==Ny
                q(k,j,n+1)=q(k,j,n)-(dt/dx)*(F(k,j,n)-F(k,j-1,n))
                -(dt/dy)*(umaxo*(q(k,j,n)*(q(k,j,n)-1))-G(k-1,j,n));
            elseif j==Nx
                q(k,j,n+1)=q(k,j,n)-(dt/dx)*(umaxr*(q(k,j,n)*(q(k,j,n)-1))-F(k,j-1,n))
                -(dt/dy)*(G(k,j,n)-G(k-1,j,n));
            end
        elseif V(1,j,n)==2
            if j > 1 && k>1 && k<Ny && j<Nx
                q(k,j,n+1)=q(k,j,n)-(dt/dx)*(F(k,j,n)-F(k,j-1,n))
                -(dt/dy)*(G(k,j,n)-G(k+1,j,n));
            elseif k==1 && j==Nx
                q(k,j,n+1)=q(k,j,n)-(dt/dx)*(umaxr*(q(k,j,n)*(q(k,j,n)-1))-F(k,j-1,n))
                -(dt/dy)*(umaxb*(q(k,j,n)*(q(k,j,n)-1))-G(k+1,j,n));
            elseif k==Ny && j==Nx
                q(k,j,n+1)=q(k,j,n)-(dt/dx)*(umaxr*(q(k,j,n)*(q(k,j,n)-1))-F(k,j-1,n))
                -(dt/dy)*(G(k,j,n))-(dt/dy)*(umaxo*(q(k,j,n)*(q(k,j,n)-1)));
            elseif j > 1 && k==1
```

```
                    q(k,j,n+1)=q(k,j,n)-(dt/dx)*(F(k,j,n)-F(k,j-1,n))
                    -(dt/dy)*(umaxb*(q(k,j,n)*(q(k,j,n)-1))-G(k+1,j,n));
                elseif j > 1 && k==Ny
                    q(k,j,n+1)=q(k,j,n)-(dt/dx)*(F(k,j,n)-F(k,j-1,n))
                    -(dt/dy)*(G(k,j,n))-(dt/dy)*(umaxo*(q(k,j,n)*(q(k,j,n)-1)));
                elseif j==Nx
                    q(k,j,n+1)=q(k,j,n)-(dt/dx)*(umaxr*(q(k,j,n)*(q(k,j,n)-1))
                    -F(k,j-1,n))-(dt/dy)*(G(k,j,n)-G(k+1,j,n));

                end
            end
            rho(k,j,n)=(1-q(k,j,n))*pmax;

        end
    end
end



fileN = 'emptypathrho075.gif';
H=figure('Position', [ 500 500 1000 1000]);
for index=1:Nt
    surf(1:Nx,1:Ny,rho(:,:,index));
    title('Crowdflow')
    xlabel('x direction')
    ylabel('y direction')
    zlabel('Density')
    view(20,45)
    zlim([0,5])
    colormap jet;
    caxis([0,5]);
    hold off;
    colorbar;
    pause(0.1);

    Frame = getframe(H);
    im = frame2im(Frame);
    [imind, CM] = rgb2ind(im,256);

    if index == 1
        imwrite(imind, CM,fileN,'gif', 'Loopcount',inf);
    else
        imwrite(imind, CM,fileN,'gif','WriteMode','append');
    end
end
```

A.3 Nudge simulation

```

clear
clc
lichtf=1.1;
umax = 1.25;
vmax = 0.4;
pmax = 5;
Lx=30;
Ly=30;
Nx=30;
Ny=30;
dx = Lx/(Nx-1);
dy = Ly/(Ny-1);
Nt=5000;
tmax = 90;
dt = tmax/Nt;
```

```
rho=zeros(Ny,Nx,Nt);
q=zeros(Ny,Nx,Nt);
F=zeros(Ny,Nx,Nt);
G=zeros(Ny,Nx,Nt);
G=zeros(Ny,Nx,Nt);
a=zeros(Ny,Nx,Nt);
s=zeros(Ny,Nx,Nt);
V=randi([1,2],1,Nx,Nt);

q(:,:,1)=1/100*randi([1,100],Ny,(Nx),1);
q(:,1,:)=0.5;



for n=1:(Nt)
    for j=(1:Nx)
        for k=(1:Ny)
            a(:,:,n)=2.5*q(:,:,n)-umax;
            if j>1 && k>3 && k<9 && j<Nx && q(k,j,n)==q(k,j+1,n)
                F(k,j,n)=lichtf*umax*(q(k,j,n)*(q(k,j,n)-1));
            elseif j<Nx && q(k,j,n)==q(k,j+1,n)
                F(k,j,n)=umax*(q(k,j,n)*(q(k,j,n)-1));;
            end
            if j>1 && k>3 && k<9 && j<Nx && a(k,j,n)>a(k,j+1,n)
                s(k,j,n)=(umax*q(k,j+1,n)*(q(k,j+1,n)-1)
                -(umax*q(k,j,n)*(q(k,j,n)-1)))/(q(k,j+1,n)-q(k,j,n));
                if s(k,j,n)>0
                    F(k,j,n)=lichtf*umax*q(k,j,n)*(q(k,j,n)-1);
                elseif s(k,j,n)<0
                    F(k,j,n)=lichtf*umax*q(k,j+1,n)*(q(k,j+1,n)-1);
                elseif s(k,j,n)==0
                    F(k,j,n)=lichtf*umax*q(k,j,n)*(q(k,j,n)-1);
                end
            elseif j<Nx && a(k,j,n)>a(k,j+1,n)
                s(k,j,n)=(umax*q(k,j+1,n)*(q(k,j+1,n)-1)
                -(umax*q(k,j,n)*(q(k,j,n)-1)))/(q(k,j+1,n)-q(k,j,n));
                if s(k,j,n)>0
                    F(k,j,n)=umax*q(k,j,n)*(q(k,j,n)-1);
                elseif s(k,j,n)<0
                    F(k,j,n)=umax*q(k,j+1,n)*(q(k,j+1,n)-1);
                elseif s(k,j,n)==0
                    F(k,j,n)=umax*q(k,j,n)*(q(k,j,n)-1);
                end
            end
            if j>1 && k>3 && k<9 && j<Nx && a(k,j,n)<a(k,j+1,n)
                if a(k,j,n)>=0
                    F(k,j,n)=lichtf*umax*q(k,j,n)*(q(k,j,n)-1);
                elseif a(k,j+1,n)<0
                    F(k,j,n)=lichtf*umax*q(k,j+1,n)*(q(k,j+1,n)-1);
                elseif a(k,j,n)<0 && 0<a(k,j+1,n)
                    F(k,j,n)=-0.3125*lichtf;
                end
            elseif j<Nx && a(k,j,n)<a(k,j+1,n)
                if a(k,j,n)>=0
                    F(k,j,n)=umax*q(k,j,n)*(q(k,j,n)-1);
                elseif a(k,j+1,n)<0
                    F(k,j,n)=umax*q(k,j+1,n)*(q(k,j+1,n)-1);
                elseif a(k,j,n)<0 && 0<a(k,j+1,n)
                    F(k,j,n)=-0.3125;
                end
            end
```

```
            if V(1,j,n)==1
                if k<Ny && q(k,j,n)==q(k+1,j,n)
                    G(k,j,n)=vmax*(q(k,j,n)*(q(k,j,n)-1));
                end
                if k<Ny && a(k,j,n)>a(k+1,j,n)
                    s(k,j,n)=(vmax*q(k+1,j,n)*(q(k+1,j,n)-1)
                    -(vmax*q(k,j,n)*(q(k,j,n)-1)))/(q(k+1,j,n)-q(k,j,n));
                    if s(k,j,n)>0
                        G(k,j,n)=vmax*q(k,j,n)*(q(k,j,n)-1);
                    elseif s(k,j,n)<0
                        G(k,j,n)=vmax*q(k+1,j,n)*(q(k+1,j,n)-1);
                    elseif s(k,j,n)==0
                        G(k,j,n)=vmax*q(k,j,n)*(q(k,j,n)-1);
                    end
                end
                if k<Ny && a(k,j,n)<a(k+1,j,n)
                    if a(k,j,n)>=0
                        G(k,j,n)=vmax*q(k,j,n)*(q(k,j,n)-1);
                    elseif a(k+1,j,n)<0
                        G(k,j,n)=vmax*q(k+1,j,n)*(q(k+1,j,n)-1);
                    elseif a(k,j,n)<0 && 0<a(k+1,j,n)
                        G(k,j,n)=-0.3125;
                    end
                end

            elseif V(1,j,n)==2
                if k>1 && q(k,j,n)==q(k-1,j,n)
                    G(k,j,n)=vmax*(q(k,j,n)*(q(k,j,n)-1));
                end
                if k>1 && a(k,j,n)>a(k-1,j,n)
                    s(k,j,n)=(vmax*q(k-1,j,n)*(q(k-1,j,n)-1)
                    -(vmax*q(k,j,n)*(q(k,j,n)-1)))/(q(k-1,j,n)-q(k,j,n));
                    if s(k,j,n)>0
                        G(k,j,n)=vmax*q(k,j,n)*(q(k,j,n)-1);
                    elseif s(k,j,n)<0
                        G(k,j,n)=vmax*q(k-1,j,n)*(q(k-1,j,n)-1);
                    elseif s(k,j,n)==0
                        G(k,j,n)=vmax*q(k,j,n)*(q(k,j,n)-1);
                    end
                end
                if k>1 && a(k,j,n)<a(k-1,j,n)
                    if a(k,j,n)>=0
                        G(k,j,n)=vmax*q(k,j,n)*(q(k,j,n)-1);
                    elseif a(k-1,j,n)<0
                        G(k,j,n)=vmax*q(k-1,j,n)*(q(k-1,j,n)-1);
                    elseif a(k,j,n)<0 && 0<a(k-1,j,n)
                        G(k,j,n)=-0.3125;
                    end
                end
            end
        end
    end

    umaxr=1.25;
    umaxb=0;
    umaxo=0;
    for j=(1:Nx)
        for k=(1:Ny)
            if V(1,j,n)==1
                if j > 1 && k>1 && k<Ny && j<Nx
                    q(k,j,n+1)=q(k,j,n)-(dt/dx)*(F(k,j,n)-F(k,j-1,n))
                    -(dt/dy)*(G(k,j,n)-G(k-1,j,n));
                elseif k==1 && j==Nx
                    q(k,j,n+1)=q(k,j,n)-(dt/dx)*(umaxr*(q(k,j,n)*(q(k,j,n)-1))-F(k,j-1,n))
                    -(dt/dy)*(umaxb*(q(k,j,n)*(q(k,j,n)-1)))-(dt/dy)*(G(k,j,n));
                elseif k==Ny && j==Nx
```

```
                    q(k,j,n+1)=q(k,j,n)
                    -(dt/dx)*(umaxr*(q(k,j,n)*(q(k,j,n)-1))-F(k,j-1,n))
                    -(dt/dy)*(umaxo*(q(k,j,n)*(q(k,j,n)-1))-G(k-1,j,n));
                elseif j > 1 && k==1
                    q(k,j,n+1)=q(k,j,n)-(dt/dx)*(F(k,j,n)-F(k,j-1,n))
                    -(dt/dy)*(umaxb*(q(k,j,n)*(q(k,j,n)-1)))
                    -(dt/dy)*(G(k,j,n));
                elseif j > 1 && k==Ny
                    q(k,j,n+1)=q(k,j,n)-(dt/dx)*(F(k,j,n)-F(k,j-1,n))
                    -(dt/dy)*(umaxo*(q(k,j,n)*(q(k,j,n)-1))-G(k-1,j,n));
                elseif k>3 && k<9 && j==Nx
                    q(k,j,n+1)=q(k,j,n)-(dt/dx)*(lichtf*umaxr*(q(k,j,n)*(q(k,j,n)-1))
                    -F(k,j-1,n))-(dt/dy)*(G(k,j,n)-G(k-1,j,n));
                elseif j==Nx
                    q(k,j,n+1)=q(k,j,n)-(dt/dx)*(lichtf*umaxr*(q(k,j,n)*(q(k,j,n)-1))
                    -F(k,j-1,n))-(dt/dy)*(G(k,j,n)-G(k-1,j,n));
                end
            elseif V(1,j,n)==2
                if j > 1 && k>1 && k<Ny && j<Nx
                    q(k,j,n+1)=q(k,j,n)-(dt/dx)*(F(k,j,n)-F(k,j-1,n))
                    -(dt/dy)*(G(k,j,n)-G(k+1,j,n));
                elseif k==1 && j==Nx
                    q(k,j,n+1)=q(k,j,n)-(dt/dx)*(umaxr*(q(k,j,n)*(q(k,j,n)-1))-F(k,j-1,n))
                    -(dt/dy)*(umaxb*(q(k,j,n)*(q(k,j,n)-1))-G(k+1,j,n));
                elseif k==Ny && j==Nx
                    q(k,j,n+1)=q(k,j,n)-(dt/dx)*(umaxr*(q(k,j,n)*(q(k,j,n)-1))-F(k,j-1,n))
                    -(dt/dy)*(G(k,j,n))-(dt/dy)*(umaxo*(q(k,j,n)*(q(k,j,n)-1)));
                elseif j > 1 && k==1
                    q(k,j,n+1)=q(k,j,n)-(dt/dx)*(F(k,j,n)-F(k,j-1,n))
                    -(dt/dy)*(umaxb*(q(k,j,n)*(q(k,j,n)-1))-G(k+1,j,n));
                elseif j > 1 && k==Ny
                    q(k,j,n+1)=q(k,j,n)-(dt/dx)*(F(k,j,n)-F(k,j-1,n))-(dt/dy)*(G(k,j,n))
                    -(dt/dy)*(umaxo*(q(k,j,n)*(q(k,j,n)-1)));
                elseif k>3 && k<9 && j==Nx
                    q(k,j,n+1)=q(k,j,n)-(dt/dx)*(lichtf*umaxr*(q(k,j,n)*(q(k,j,n)-1))
                    -F(k,j-1,n))-(dt/dy)*(G(k,j,n)-G(k+1,j,n));
                elseif j==Nx
                    q(k,j,n+1)=q(k,j,n)-(dt/dx)*(umaxr*(q(k,j,n)*(q(k,j,n)-1))-F(k,j-1,n))
                    -(dt/dy)*(G(k,j,n)-G(k+1,j,n));

                end
            end
            rho(k,j,n)=(1-q(k,j,n))*pmax;

        end
    end
end

fileN = 'emptypathrho075.gif';
H=figure('Position', [ 500 500 1000 1000]);
for index=1:Nt
    surf(1:Nx,1:Ny,rho(:,:,index));
    title('Crowdflow')
    xlabel('x direction')
    ylabel('y direction')
    zlabel('Density')
    view(20,45)
    zlim([0,5])
    colormap jet;
    caxis([0,5]);
    hold off;
    colorbar;
    pause(0.1);
    Frame = getframe(H);
    im = frame2im(Frame);
    [imind, CM] = rgb2ind(im,256);
```

```
    if index == 1
        imwrite(imind, CM,fileN,'gif', 'Loopcount',inf);
    else
        imwrite(imind, CM,fileN,'gif','WriteMode','append');
    end
end
```