Università degli Studi di Ferrara
Corso di Laurea in Fisica

Statistical learning and simulating the paths of walking pedestrians

Relatori:
Prof. Alessandro Drago
Prof. Federico Toschi

Correlatore:
Dr. Alessandro Corbetta

Laureando:
Dario Chinelli

24 marzo 2022

Anno Accademico 2020 – 2021

Abstract

[ENG] The dynamics of pedestrian changes considerably depending on the surrounding space, not just for the intrinsic chaotic movements that people does walking but also due to the reciprocal collisions and environment condition. We have considered some scenarios to implement models and a tools that can gives us simulations of the movements of a single pedestrian. In order to properly simulate a pedestrians' dynamic, is to have information about the probability to change direction after every step, in every positions of the trajectory. This approach is linked to the path integral in the way that: given a trajectory, it's possible to say with certain probability where the next step is. This mathematical approach is computationally expensive, even more with the big amount of data we are using. So we started implementing a discrete system and a easy model and than we moved to more complex model. In total we get four types of models: two time dependent and tow independent. From now on we'll call those: D2Q9 and D2Q9Q9 the firsts two; TD2Q9 and TD2Q9Q9 the others two.

[ITA] L'obiettivo scientifico è stato creare un metodo, ispirato al Lattice-Boltzman, con cui apprendere, a partire da dati reali, la dinamica pedonale e quantificarla in termini di matrici di transizione su reticolo. L'obiettivo fondamentale è riuscire a quantificare il campo di probabilità, trovato utilizzando diversi modelli. Questo campo ci permette di studiare la dinamica e creare simulazioni di pedoni e traiettorie le cui statistiche sono indistinguibili per costruzione dalle statistiche delle traiettorie reali

Contents

Chapter 1

Introduction

The aim of this work is to clarify the possibility to analyze real life datasets and simulate the pedestrian crowd starting from the Lattice model.

1.1 Assimilating pedestrian dynamics

1.2 Challenges

Starting from real data how can we define a good model to simulate a pedestrian in the crowd flow? In this type of system there is a multitude of *forces* that determinate the path of a single pedestrian. So let's take into account a single pedestrian P that walks in a certain space. The first type of interaction is the structure where P can or cannot walk thought, that is defined as the whole domain Ω. The second interaction is between P and the other pedestrians. Every pedestrian needs a personal space all around, that is variable due the circumstance and it is not easy to be analytically determinate. A third type of interaction are random events along the P's path, real world events. ...

How to visualise the pedestrian's path using a multi-dimensional histogram? It is possible to plot every single trajectory, but this lead to a chaotic data representation and not so functional nor readable. It is also easily possible to plot the *heatmap* of a dataset to analyse the most "walked" areas. Even if this second plot choice can takes into account more trajectories than the first and still be readable, it has a problem. This second lead to a representation where the time dependency is completely lost. ...

1.3 Relevance

Chapter 2

Pedestrian dynamics measurements

2.1 Idea of the recording technique

The recording techniques employed are two: Kinect based technology and ProRail security cameras. In both cases the field of view is covered using multiple cameras working together. Starting from the row images, each object is tracked down along its entire path. This is possible using imaging recognition software. The software give as output a data-frame with coordinates and time for each pedestrian.

2.1.1 Kinect: depth map based pedestrian tracking

2.1.2 ProRail cameras

2.2 Presenting datasets

2.2.1 Glow experiment

The experiment was realized during the Glow Expo in 2017. The idea is that people that enter from a single entrance have to chose between two possible exits, because of the obstacle in the middle. In the (Figure 2.1) is represented a digital reconstruction of the domain.

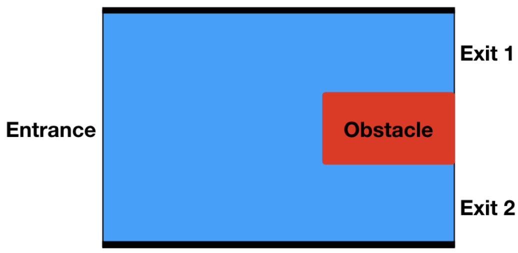


Figure 2.1: Reconstruction of the *glow* experiment, point of view from above.

2.2.2 Utrecht Centraal (Floorefield 10)

In collaboration with ProRail - the company responsible of the train's stations in Netherlands - we had the possibility to use data from the Utrecht's train station. The (Figure 2.2) represents the camera's point of view of the analyzed domain in the station. This is an interesting spot given that this square has three free sides where people could walk through. It's also a huge corridor and an highly crossed spot, that increase the statistic. With this domain we could study if the simulations we're doing are correct also in cases where people with different directions cross the same coordinates in the map. In fact - as described in the following paragraphs - the representation with the 3-dimensional histogram shows us a sort of *cross* X.

Figure 2.2: Utrecht Centraal, cameras point of view (Floorfield 10)

2.3 Technique of measure

Technique of measure documentation intro here

2.3.1 Kinect sensor

Technique of measure with Kinect sensor documentation intro here

2.3.2 ProRail data

Technique of measure with ProRail data documentation intro here

Chapter 3

Background

3.1 Markov Chain

A *Markov Chain* is a stochastic model. It describes the future outcome state based on the present state. In other words, the present state determinate the probabilities for every possible future outcome. The model's representation is a *stochastic matrix*, from now on called P matrix. The matrix's entries P_{ij} has as row-index i the starting state and as column-index j the ending state of the system.

(a) The diagram of a two-state Markov Chain

(b) The transition matrix, also named the Markov matrix

Figure 3.1: From the Markov diagram to the Markov matrix of a tow-state system

A two-state Markov chain is the most basic model which can be used for the illustration of the Markov process. The diagram in (Figure 3.1a) represents the possibility that the system has to change from both states. For instance, from the state W the system can move to the B state with the big black arrow or can remain in the W state with the small white arrow. The entries in the Markov Matrix in (Figure 3.1b) are positives numbers that represent the probability of changing state.

3.2 Cellular-Automata

Cellular-Automata documentation here

3.3 Lattice-Boltzman

Lattice-Boltzman documentation here

Chapter 4

Propose data assimilation technique

4.1 Approximation of path integrals: 3D histogram

Approximation of path integrals: 3D histogram documentation here

4.2 Learning transition matrices from data

In this study there is a total of four models: *D2Q9*, *D2Q9Q9*, *TD2Q9*, *TD2Q9Q9*. The firsts two are only dependent by the position in space, also called *time-independents*. The others two are dependent by the position and time, also called *time-dependents*. Whereas there is also a distinction between the D2Q9s and the D2Q9Q9s. For the D2Q9s what it's doing is considering the velocity from a cell to another, so just the change in position. For the D2Q9Q9s it's also considering the acceleration, so the change in velocity. The starting point of each one is the dataset, collected form a real life situation. Since each of them are entirely based on real world pedestrian's path in a crowd, those models simulates an *effective potential* (EP). This potential considers the imposed limit due to the presence of others pedestrians, such as pedestrians tend to not collide each others. It also considers the boundary condition given by the structural environment. The strong point of this EP is that is generated by the real world observation and not built by hand. With the aim of reproducing realistic pedestrians movements, synthetic paths are created from the models. Every model generate one trajectory that simulate just one pedestrian in a statistical crowd. When simulating more paths it consider pedestrian that walks alone in the crowd. This model doesn't consider the interaction made by the others simulated pedestrians.

Notation Lets assume $\gamma = \gamma(\vec{x}_c)$ a pedestrian's path, where $\vec{x}_c = (x_c, y_c)$ has a bi-dimensional spacial dependancy. Given a field Ω_c, the continuous space where pedestrians are tracked, the path γ in that space has a start position A and an end position B. The field Ω is than divided into *rectangular* cells, dividing the real space along x, whit maximum extension indicated as L_x, in a certain number of cells D_x; as well for the y-direction, with obvious notation: L_y and D_y. After this discretization is obtained a *grid space* Ω_g. Where every path γ is converted from continuous $\gamma = \gamma(x_c, y_c)$ to discrete coordinate $\gamma = \gamma(x_g, y_g)$, referred to the *grid*. To lighten up the notation when speaking of *grid space* it is simply used (x, y) in reference to the discrete grid position.

The standard D2Q9 configuration In reference to the (Figure 4.1). This *map* is set for each position (x_0, y_0) in the grid space and it represents the eight neighbors and the central position where a pedestrian could go. Each direction will be associated to a certain transition probability.

When a trajectory change position, in the grid space (Figure 4.2), from $P_0 = (x_0, y_0)$ to $P_1 = (x_1, y_1)$ is associated a transition. The transition is identified by a number $k = 0, 1, ..., 8$ such that is unique. It is derived from the series of coordinates for each trajectory and each step in time. When the calculation is made for each step, for every position in time is also associated a transition number, that represents where is going to go in the next step.

If this transition is associated to the change in position it identify a certain velocity, as vector, with a certain direction. Iterating this procedure to the entire pedestrian's trajectory it is possible to get something like what's illustrated in the (Figure 4.3). In that figure it is possible to distinguish

6	2	5
3	0	1
7	4	8

Figure 4.1: Transitions associated to possible movements from the center cell. The figure identify the k nine values for each cell.

$(-1,+1)$	$(+0,+1)$	$(+1,+1)$
$(+1,+0)$	$(+0,+0)$	$(+1,+0)$
$(-1,-1)$	$(+0,-1)$	$(+1,-1)$

Figure 4.2: Given the initial position at the center square, this is a representation of the change in coordinates to the next cell.

the path in the continuous space and the discrete path in the grid space. It also shows the direction of the next movement for each position with arrows that are consistent with the velocity arrows in each position. The numbers are the value of the k-index in each position, it is solid with the maps above. This lead the discussion directly to the first model $D2Q9$ in the next paragraph.

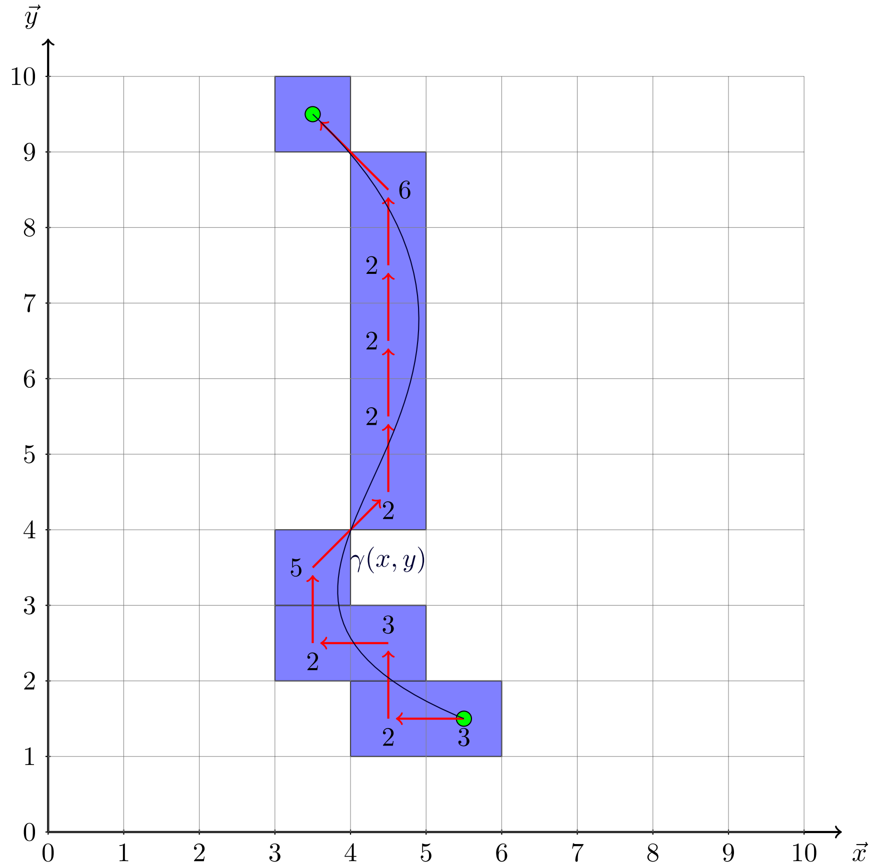


Figure 4.3: This illustration represents a trajectory in the *continuous space* as the blue line γ. That path γ is discretized in the *grid space*, represented by the blue cells. The red arrows represents the change from a cell to the next. The numbers are the associated to the D2Q9 indexes to those moves, also called *k-directions*

4.2.1 Model D2Q9

The simplest model considered here is called $D2Q9-model$. This model is a time-independent and it consider the velocity of the pedestrian. Given a starting position (x_0, y_0) in the field Ω. It uses the nine closest possible positions where a pedestrian could go from that point. With the $D2Q9-model$ is than possible to know, for each position (x_0, y_0), the probability to go up, down, left, right or a combination of those movements.

Transitions from the initial position $P_0 = (x_0, y_0)$ to the next closest cell in the grid P_k are defined by the index k. So that the index k gives the direction of the transition. To explicit all the transitions from P_0 to P_k, those transformation are defined in (Equation 4.1) and represented as diagram in (Figure 4.4):

$$\begin{aligned}
P_0 \to P_0 : &\quad (x_0, y_0) \to (x_0, y_0) \\
P_0 \to P_1 : &\quad (x_0, y_0) \to (x_0 + 1, y_0) \\
P_0 \to P_2 : &\quad (x_0, y_0) \to (x_0, y_0 + 1) \\
P_0 \to P_3 : &\quad (x_0, y_0) \to (x_0 - 1, y_0) \\
P_0 \to P_4 : &\quad (x_0, y_0) \to (x_0, y_0 - 1) \\
P_0 \to P_5 : &\quad (x_0, y_0) \to (x_0 + 1, y_0 + 1) \\
P_0 \to P_6 : &\quad (x_0, y_0) \to (x_0 - 1, y_0 + 1) \\
P_0 \to P_7 : &\quad (x_0, y_0) \to (x_0 - 1, y_0 - 1) \\
P_0 \to P_8 : &\quad (x_0, y_0) \to (x_0 + 1, y_0 - 1)
\end{aligned} \tag{4.1}$$

Considering the (Figure 4.4) all the transitions are associated to a specific k. This is a particular Markov Chain (Ref.Chap. 3.1) where there are a total of *nine* states. Between these states the transitions always and only start from the P_0 state to go to the others P_k states or itself. The same

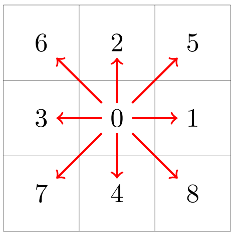


Figure 4.4: The possible transitions, represented as vectors. For the D2Q9 model those vector are also representing the velocity vectors.

concept is graphically represented with the diagram in (Figure 4.5). To every transition is associated a certain probability to happen. Formally this probability is given by the initial and the final states: p_{if}. Since there always is the same starting state, it is possible to omit it. So that the probability of the transition from P_0 to P_k is expressed by p_k, where the index k points to the ending state. It means that for each position in Ω it's possible to say how likely is to "step forward" or "turn right" and so on. Then, once in the new position, it's again possible to say the most probable direction that the pedestrian will choose. The same prediction is applicable to the whole space, mapped by the real datas.

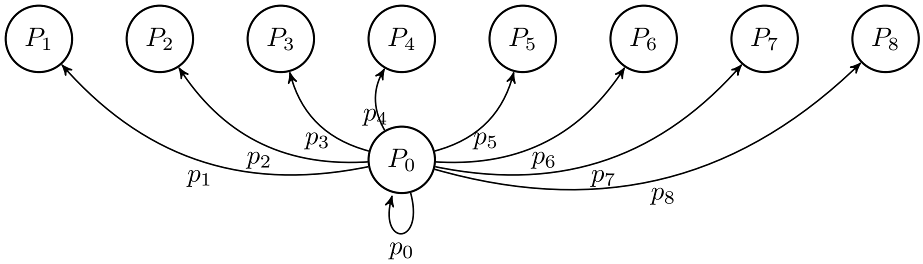


Figure 4.5: The Markov Chain diagram of the system. The states are indicated with circles and labeled with P_k. The transitions are indicated with arrows and labeled with theirs probability p_k

With this structure it is then possible to create a tensor A with three indices. Taking into account the simplest model, as above, the relative *tensor* is A_{xyk}. Where every entries is the probability p to move along the k direction from the location (x, y). Since the aim of every models is to simulate a pedestrian in the crowd, this tensor is the key to get to the result. In general it's not easy to represent the tensor A in all following models, but it's possible for this first one as plotted in (Figure 4.6). It shows a 3×3 matrix of figures. Each one is referred to a certain value of the k-index. Each figure's position is in reference to the usual $D2Q9$ map, similarly as in (Figure 4.1).

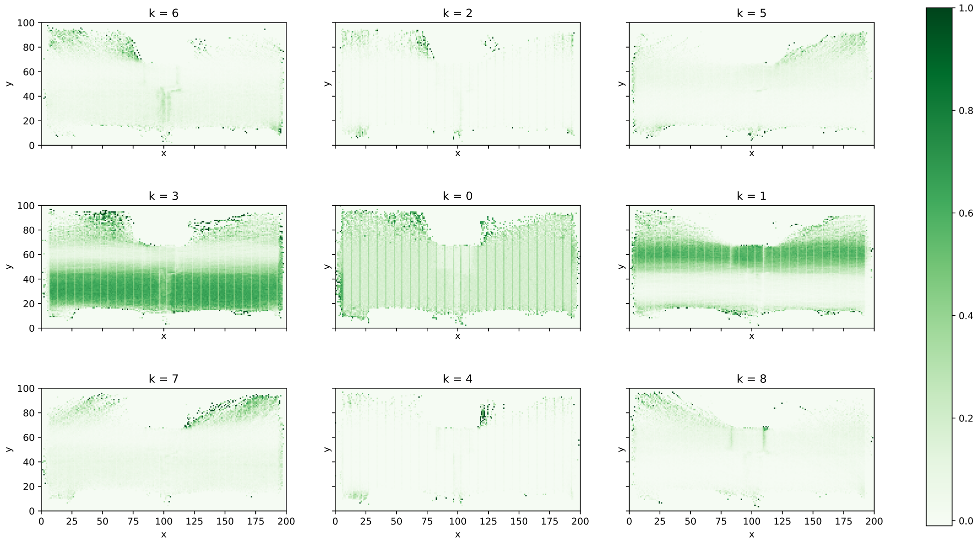


Figure 4.6: Each figure represents the probability in every position to move along a certain direction, defined by the k-index.

4.2.2 Model D2Q9Q9

The conceptual step forward of the study is to consider the next position, but also the previous one. Similarly to the previous model, this is a *time-independent* model. Hence given a trajectory γ in the grid space of a pedestrian that make a transition for each time step. For each point P_0 of γ it is possible to determinate where it was before at P_{-1} and where is going to be after at P_{+1}. The index that represents the *next* position is k, meanwhile the index that represents the *previous* position is h. For instance it is given the table of the coordinates and the two indexes related to the (Figure 4.5) in the (Table 4.1). The *tensor* associated to this model is characterized by a total of

Time step	x_g	y_g	k-index	h-index
1	5	1	3	0
2	4	1	2	1
3	4	2	3	4
4	3	2	2	1
5	3	3	5	4
6	4	4	2	7
7	4	5	2	4
8	4	6	2	4
9	4	7	2	4
10	4	8	6	4
11	3	9	0	8

Table 4.1: This is tabulated the trajectory of the same illustrative pedestrian as above in (Figure 4.5). Here is expressed the position time to time, the index of the following move and the index of the previous move.

four indexes as A_{xykh}. Every element of this tensor is now representing a certain probability to move away from a state to another. Taking into account the example in the (Table 4.1), and considering this the only one possible trajectory. It is easy to see that the probability at $A_{xykh} = A_{3,3,5,4} = 1$ is maximum in the position $(3, 3)$. The probability for every other k, h in the same position is zero, $A_{3,3,k,h} = 0 \quad for\, k \neq 5,\, h \neq 4$. Instead, if two trajectories pass by the same position in the grid but with different directions, this probability is distributed along two directions. This is the scenario represented in (Figure 4.8), where there are two trajectories. Those pass by the same cell at different times, but leave in the model *D2Q9Q9* a strong influence. In this case $A_{2,2,5,7} = 1$ and $A_{2,2,6,8} = 1$. For the previous model *D2Q9* in the same position it would be, with A_{xyk}, $A_{2,2,5} = 0.5$ and $A_{2,2,6} = 0.5$.

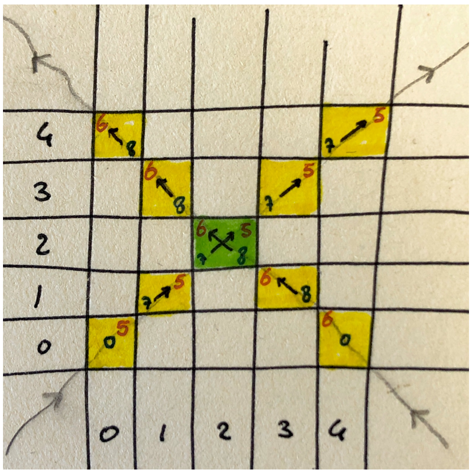


Figure 4.7: (TO DO WITH TIKZ) Two paths passing by the same cell, in green, with different directions. The numbers in red are defining the values of the k-index for every step. The numbers in blue are the values of the h-index for every step.

Pedestrian	Time step	x	y	k	h
Ped 1	1	0	0	5	0
Ped 1	2	1	1	5	7
Ped 1	3	2	2	5	7
Ped 1	4	3	3	5	7
Ped 1	5	4	4	5	7

Pedestrian	Time step	x	y	k	h
Ped 2	1	0	4	6	0
Ped 2	2	1	3	6	8
Ped 2	3	2	2	6	8
Ped 2	4	3	1	6	8
Ped 2	5	4	0	6	8

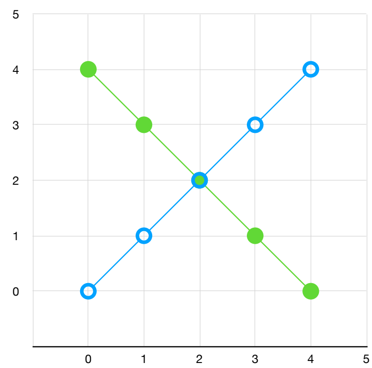


Figure 4.8: (TO DO WITH TABLES) Two paths passing by the same cell, with different directions.

4.2.3 Model TD2Q9

This module takes into account all the tools offered by the *D2Q9 model*. But time is now relevant and so this is a *time-dependent* model.

Time It is important to describe properly what is *time* in this study. Lets start from what is not: time is not the universal time, like UTC. Time here is discrete and it's defined also as *time step*. It is divided in seconds, using the *unix time* or *UNIX Epoch time*. Every step in time define a new state along the time axes, it is possible to imagine it as a new dimension. For each pedestrian path, time start at the entrance in the field and ends at the exit of it. So time is relative to each trajectories and not global.

The definition of a *state* is not just by the position in space but is given by x, y, t. In this model, pedestrians moves along three axes: two dimensions in space and one in time. The *tensor* representing the probability to move is defined by A_{txyk}. With this structure it is possible to associate the velocity to the time step. This gives the possibility to differentiate when a trajectory is going to exit or is just entered, when giving the probability to move. Lets make an example and consider a position close to the map border $P_b = (x_b, y_b)$, something like in (Figure 4.9). If it's not known the time of this position P_b the probabilities to go to the center of the map or out of it are non zero. So it's not possible, given P_b, to really distinguish if the pedestrian is going out or not. But if the time is taken into account it's necessary to distinguish if the pedestrian is at the beginning of it's path or at the end. Lets start again from the position P_b. If it's at the beginning in time steps, the more probable move will be to the center. If some time is passed inside the map, it will have higher probability to go out from the map.

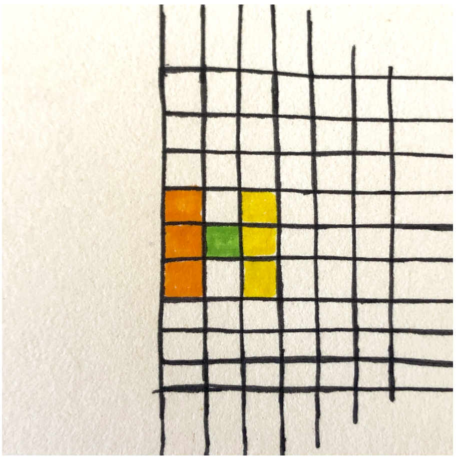


Figure 4.9: The boundary position P_b considered is the green cell. When time is *small*, the trajectory is at the beginning, yellow positions are more likely than orange positions. When time is *big*, the trajectory is at the ending, yellow positions are less likely than orange positions.

4.2.4 Model TD2Q9Q9

This model is the extension of the previous *TD2Q9* to the *D2Q9Q9* method, where acceleration is taken into account. With this method it is associated a tensor A_{txykh}, with obvious notation in reference to the previous paragraphs.

4.3 Simulations

Simulations documentation here

4.3.1 Trajectories simulation

4.3.2 Distribution simulation

Chapter 5

Results

5.1 Comparison of assimilation methods

5.1.1 Real data - D2Q9

5.1.2 Real data - D2Q9Q9

5.1.3 Real data - TD2Q9

5.1.4 Real data - TD2Q9Q9

5.2 Simulation with generative models

The aim of this paragraph is the comparison between the real dataset utilized to produce the models and some simulations. The following quantities are taken into account to compare the results. Those are plotted as histograms, to empathize the statistical approach.

(i) The magnitude of the velocity vector along the $\vec{x}$ axes, plotted as 1-dimensional histogram;

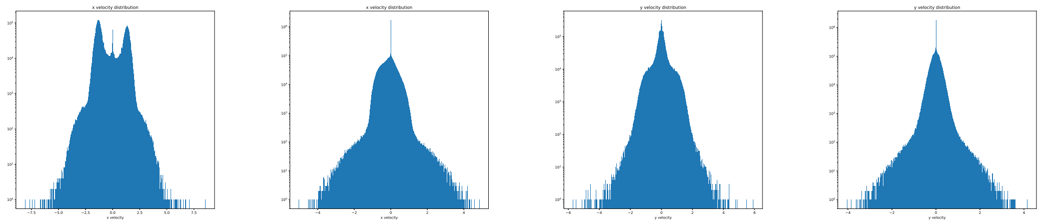
(ii) the magnitude of the velocity vector along the $\vec{y}$ axes, plotted as 1-dimensional histogram;

(iii) the correlation between the position along the $\vec{x}$ axes and the magnitude of the velocity vector along the same axes, plotted as heat-map or 2-dimensional histogram;

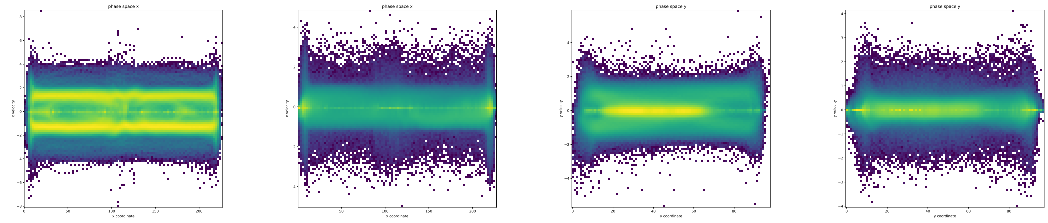
(iv) the correlation between the position along the $\vec{y}$ axes and the magnitude of the velocity vector along the same axes, plotted as heat-map or 2-dimensional histogram.

(v) the heat-map of the positions along $\vec{x}$ and $\vec{y}$ axis of all paths that have passed though, plotted as 2-dimensional histogram;

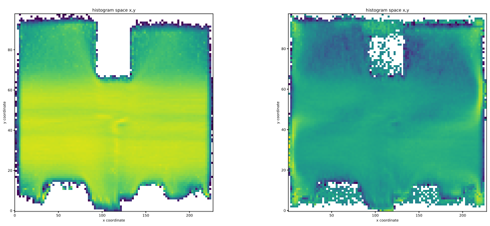
5.2.1



(a) Real data hist vx (b) Simulation data hist vx (c) Real data hist vy (d) Simulation data hist vy



(e) Real data hist2d xVx (f) Simulation data hist2d xVx (g) Real data hist2d yVy (h) Simulation data hist2d yVy

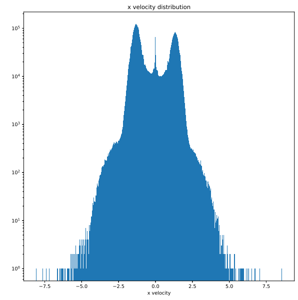


(i) Real data hist2d Pxy (j) Simulation data hist2d Pxy

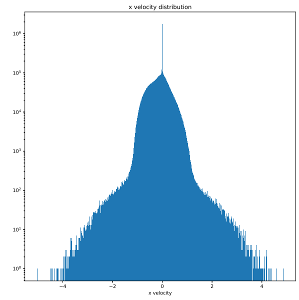
Figure 5.1: TEST – (i) & (ii) & (iii) & (iv) & (v) - simD2Q9 - All plots in one page.

5.2.2 Real data - D2Q9

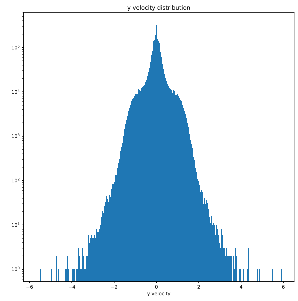
This paragraph's reference are the following: (Figure 5.2), (Figure 5.3), (Figure 5.4)



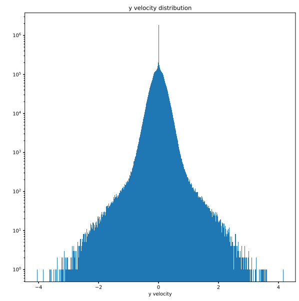
(a) Real data hist vx



(b) Simulation data hist vx

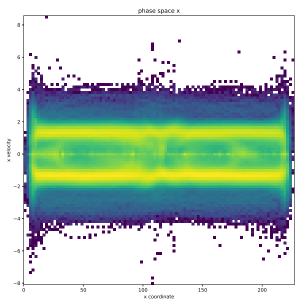


(c) Real data hist vy

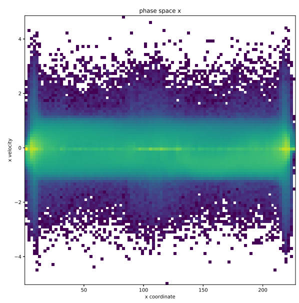


(d) Simulation data hist vy

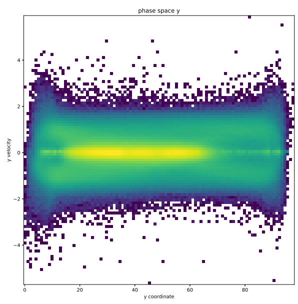
Figure 5.2: (i) & (ii) - simD2Q9 - The magnitude of the velocity vector along the $\vec{x}$ and $\vec{y}$ axis, plotted as 1-dimensional histograms.



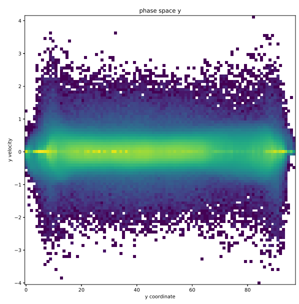
(a) Real data hist2d xVx



(b) Simulation data hist2d xVx

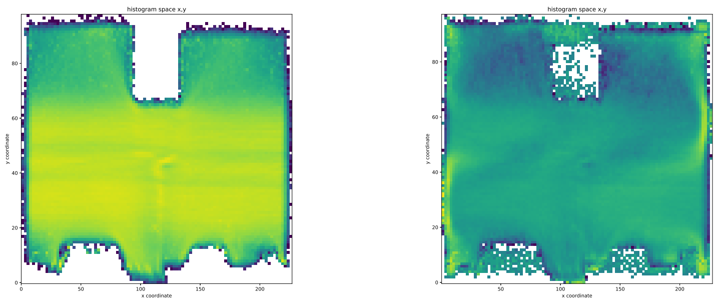


(c) Real data hist2d yVy



(d) Simulation data hist2d yVy

Figure 5.3: (iii) & (iv) - simD2Q9 - The correlation between the position along the $\vec{x}$ and $\vec{y}$ axis and the magnitude of the velocity vector along the same correspondent axes, plotted as heat-map or 2-dimensional histograms.



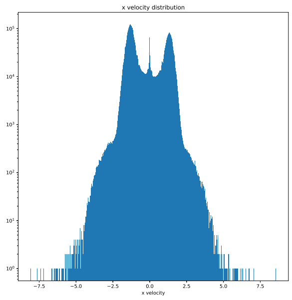
(a) Real data hist2d Pxy

(b) Simulation data hist2d Pxy

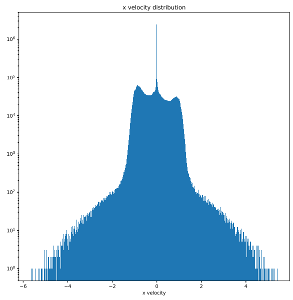
Figure 5.4: (v) - simD2Q9 - The heat-map of the positions along $\vec{x}$ and $\vec{y}$ axis of all paths that have passed though, plotted as 2-dimensional histogram.

5.2.3 Real data - D2Q9Q9

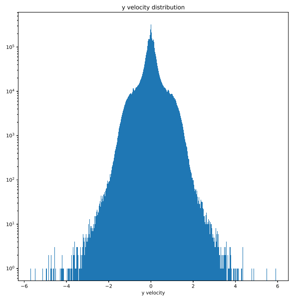
This paragraph's reference are the following: (Figure 5.5), (Figure 5.6), (Figure 5.7)



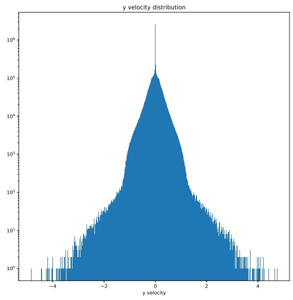
(a) Real data hist vx



(b) Simulation data hist vx

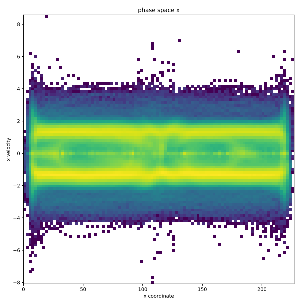


(c) Real data hist vy

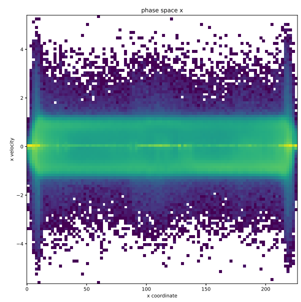


(d) Simulation data hist vy

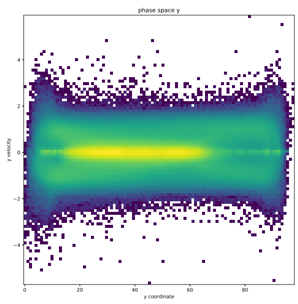
Figure 5.5: (i) & (ii) - simD2Q9Q9 - The magnitude of the velocity vector along the $\vec{x}$ and $\vec{y}$ axis, plotted as 1-dimensional histograms.



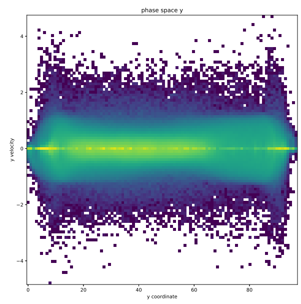
(a) Real data hist2d xVx



(b) Simulation data hist2d xVx

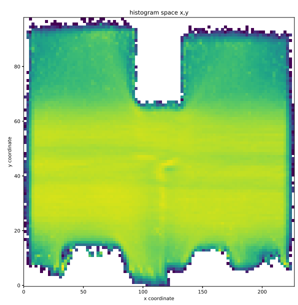


(c) Real data hist2d yVy

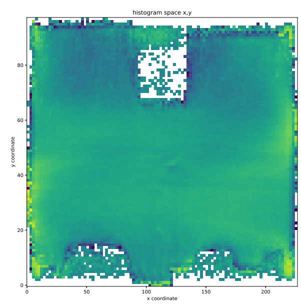


(d) Simulation data hist2d yVy

Figure 5.6: (iii) & (iv) - simD2Q9Q9 - The correlation between the position along the $\vec{x}$ and $\vec{y}$ axis and the magnitude of the velocity vector along the same correspondent axes, plotted as heat-map or 2-dimensional histograms.



(a) Real data hist2d Pxy

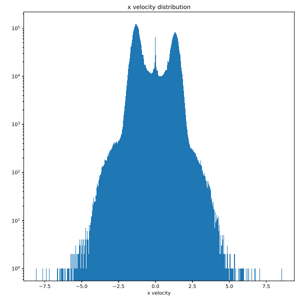


(b) Simulation data hist2d Pxy

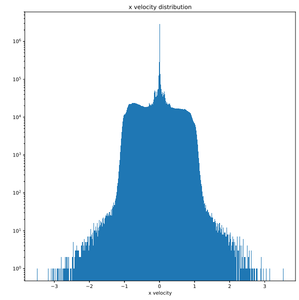
Figure 5.7: (v) - simD2Q9Q9 - The heat-map of the positions along $\vec{x}$ and $\vec{y}$ axis of all paths that have passed though, plotted as 2-dimensional histogram.

5.2.4 Real data - TD2Q9

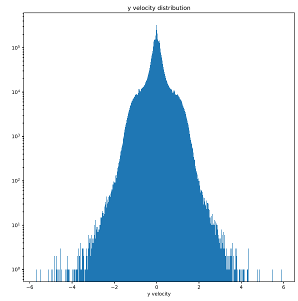
This paragraph's reference are the following: (Figure 5.8), (Figure 5.9), (Figure 5.10)



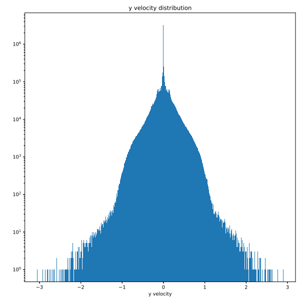
(a) Real data hist vx



(b) Simulation data hist vx

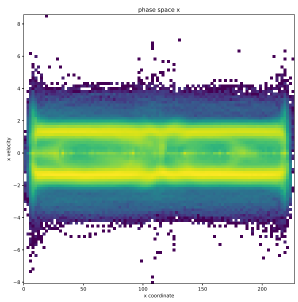


(c) Real data hist vy

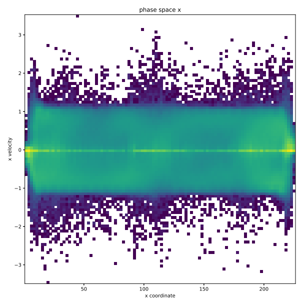


(d) Simulation data hist vy

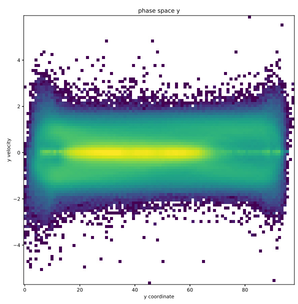
Figure 5.8: (i) & (ii) - simTD2Q9 - The magnitude of the velocity vector along the $\vec{x}$ and $\vec{y}$ axis, plotted as 1-dimensional histograms.



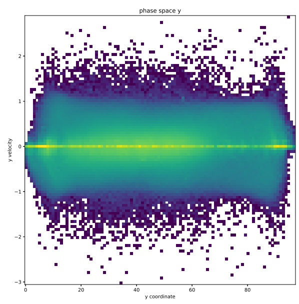
(a) Real data hist2d xVx



(b) Simulation data hist2d xVx

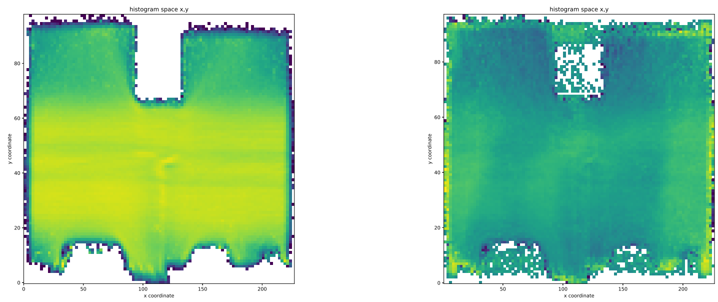


(c) Real data hist2d yVy



(d) Simulation data hist2d yVy

Figure 5.9: (iii) & (iv) - simTD2Q9 - The correlation between the position along the $\vec{x}$ and $\vec{y}$ axis and the magnitude of the velocity vector along the same correspondent axes, plotted as heat-map or 2-dimensional histograms.



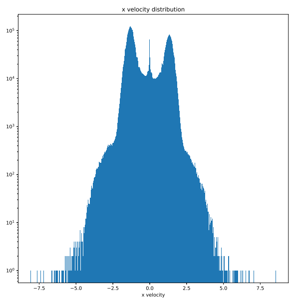
(a) Real data hist2d Pxy

(b) Simulation data hist2d Pxy

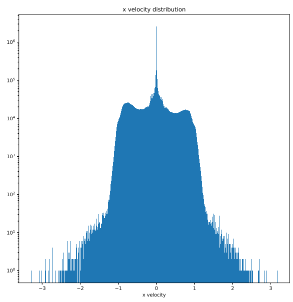
Figure 5.10: (v) - simTD2Q9 - The heat-map of the positions along $\vec{x}$ and $\vec{y}$ axis of all paths that have passed though, plotted as 2-dimensional histogram.

5.2.5 Real data - TD2Q9Q9

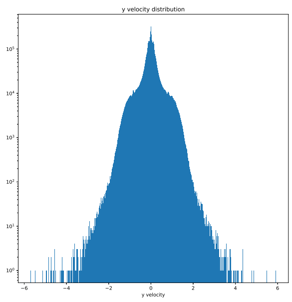
This paragraph's reference are the following: (Figure 5.11), (Figure 5.12), (Figure 5.13)



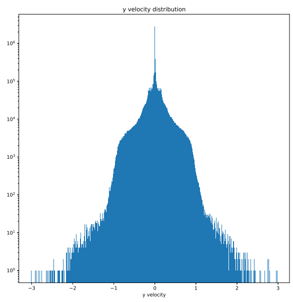
(a) Real data hist vx



(b) Simulation data hist vx

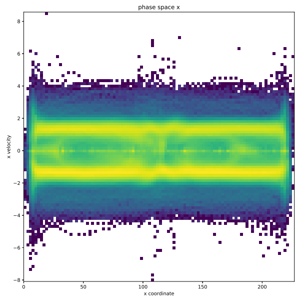


(c) Real data hist vy

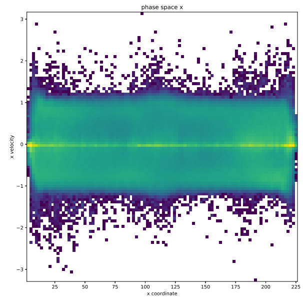


(d) Simulation data hist vy

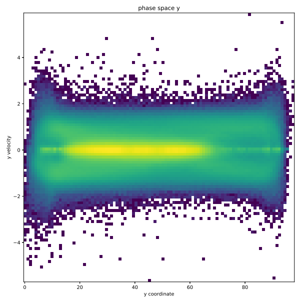
Figure 5.11: (i) & (ii) - simTD2Q9Q9 - The magnitude of the velocity vector along the $\vec{x}$ and $\vec{y}$ axis, plotted as 1-dimensional histograms.



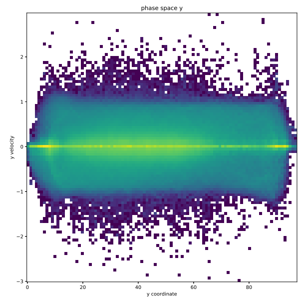
(a) Real data hist2d xVx



(b) Simulation data hist2d xVx

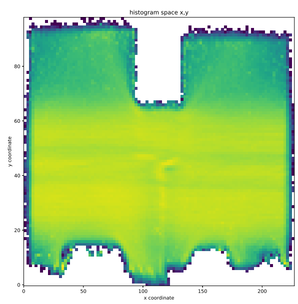


(c) Real data hist2d yVy

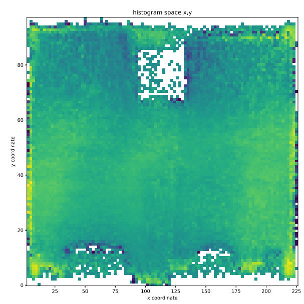


(d) Simulation data hist2d yVy

Figure 5.12: (iii) & (iv) - simTD2Q9Q9 - The correlation between the position along the $\vec{x}$ and $\vec{y}$ axis and the magnitude of the velocity vector along the same correspondent axes, plotted as heat-map or 2-dimensional histograms.



(a) Real data hist2d Pxy



(b) Simulation data hist2d Pxy

Figure 5.13: (v) - simTD2Q9Q9 - The heat-map of the positions along $\vec{x}$ and $\vec{y}$ axis of all paths that have passed though, plotted as 2-dimensional histogram.

Chapter 6

Discussion

Discussion documentation here

Chapter 7

Appendix

7.1 Package pathintegralanalytics - code description

Package pathIntegralAnalytics documentation here

7.1.1 Explanation of the library

Explanation of the library documentation here

7.1.2 UML diagram

UML pathIntegralAnalytics documentation code here