ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Complex Social Systems:
Modeling Agents, Learning and Games

Project Report

Modeling pedestrian flow in a crowded building

Battista P. [1], Biella A. [2], Ferradini C. A. [3], Fioroni L. [4], Guillemet V. [5]

Abstract

Collective phenomena in pedestrian crowds can be modelled by means of molecular-dynamic-like simulations based on a generalised force model of interacting agents. These forces are not directly exerted by the environment, but they result from the internal motivations of each agent to perform certain actions. The model we consider includes a term that describes the acceleration towards the desired velocity vector of motion, then we include a term that accounts for the distance that each agent keeps from other agents and the borders. Finally, the randomness of human behaviour is included as a stochastic term. The resulting equations of motion are non linearly coupled Langevin equations that can be solved using computer simulations.

Zürich
6^{th} December 2021

[1]pbattista@student.ethz.ch
[2]abiella@student.ethz.ch
[3]cferradini@student.ethz.ch
[4]lfioroni@student.ethz.ch
[5]vguillemet@student.ethz.ch

Agreement for free-download

We hereby agree to make our source code for this project freely available for download from the web pages of the SOMS chair. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

Battista P.

Biella A.

Ferradini C. A.

Fioroni L.

Guillemet V.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

MODELING PEDESTRIAN FLOW IN A CROWDED BUILDING

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	First name(s):
BATTISTA	PIERA
BIELLA	ALESSANDRO
FERRADINI	CARLA ANAIS
FIORONI	LORENZO
GUILLEMET	VINCENT

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date

ZÜRICH, 06/12/2021

Signature(s)

Piera ~~Bat~~

Carla A. Ferradini

Lorenzo Fioroni

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Contents

1 Introduction and Motivations

Being able to predict the behaviour of pedestrian crowds brings together many different disciplines such as social sciences, physics and computer sciences. The importance of the study of this field is evident from the many applications it has, especially in building safety and risk prevention. Successfully predicting the movement and the actions of people when in danger can help prevent accidents and minimise the fatalities due to crowd panic. Additionally, the simulations of crowd movements are often used when constructing structures with the purpose of accommodating high flows of pedestrians, such as crossroads, stations or stadiums.

These phenomena have been profusely studied by researchers with the use of many different models, due to the fact that approximations need to be made and different approaches can be taken. These models are often grouped in three categories based on their granularity, i.e. how much importance is given to the single pedestrian. The three categories are macroscopic, mesoscopic and microscopic models [7]. The macroscopic model focuses on the overall "flow" of pedestrians without paying much attention to their single interactions. This type of model is computationally light and fast, since there is no need to represent and loop over every agent. Some examples can be found in models such as hydrodynamics models which compare the pedestrian flow to liquid flows [3, 6]. Microscopic models, on the other hand, have more success in describing the complex interactions of each agent in the simulation. A large number of degrees of freedom concerning the interaction of each agent with others or the environment and even single characteristics of different groups of agents can be implemented to make the model as realistic as possible. Examples of microscopic simulations can be found in the cellular automata model [9, 2] or the social force model [5]. These models describe in a more accurate way the reality of crowd motion. An intermediate approach between macroscopic and microscopic, i.e. mesoscopic models, such as gaseous models, can be found as in [8].

In this project, the microscopic model of the social force model is investigated as in [5] and implemented. The choice was made as to be able to crate a simulation of pedestrian flow that could replicate the "real-life" interactions of single agents. In particular, we investigated the evacuation of a high school classroom with different configurations of school desks.

In Section 2 the social force model is outlined, while in Section 3 the code used to implement it and solve the resulting equations is described. The results of the study are shown in Section 4 and in Section 5 further applications of this method are discussed.

2 Description of the Model

The social force model aims to describe stochastic behavioural models in terms of Langevin equations, hence it can be applied to the description of pedestrian behaviour. A pedestrian is always confronted with the same situations while walking in a building or outside, therefore his/her actions are often automatic and predictable. Then it is straightforward to assume that the rules of pedestrian behaviour can be described in terms of equations of motion.

Given a set of agents, labelled with a number α, the quantity $\vec{F}_\alpha(t)$, which describes the temporal changes of the preferred velocity of agent α, $\vec{w}_\alpha(t)$, can be interpreted as a *social force*. This quantity describes the motivation to act of the pedestrian, since a variation of $\vec{F}_\alpha(t)$ can be seen as a reaction to the information that the agent obtains from the environment.

Assuming that the agents move in a limited region of space Σ, we can introduce the main effects that determine the motion of a pedestrian α:

- **(Goal)** Agent α wants to reach a destination $\vec{r}_\alpha^0 \in \Sigma$ in the easiest possible way, which most of the time coincides with the shortest one. This optimal path can then be approximated as a polygon with edges $\{\vec{r}_\alpha^i\}_{i=1,\dots,n} \subset \Sigma$, such that $\vec{r}_\alpha^n = \vec{r}_\alpha^0$.
  Then the agent's motion in empty space at time t would be:
  $$\vec{e}_\alpha(t) := \frac{\vec{r}_\alpha^k - \vec{r}_\alpha(t)}{||\vec{r}_\alpha^k - \vec{r}_\alpha(t)||} \tag{1}$$
  where $\vec{r}_\alpha(t)$ is the position of the agent at time t.
  If the agent is unperturbed, he/she will walk into $\vec{e}_\alpha(t)$ with a *desired speed* v_α^0. Otherwise, he/she will experience a deviation of the actual velocity $\vec{v}_\alpha(t)$ from the desired one, $\vec{v}_\alpha^0(t) = v_\alpha^0 \vec{e}_\alpha(t)$, which will then lead to a tendency to reach $\vec{v}_\alpha^0(t)$ again with a *relaxation time* τ_α.
  This process gives rise to an acceleration term:
  $$\vec{F}_\alpha^0(\vec{v}_\alpha^0(t), \vec{v}_\alpha(t)) := \frac{\vec{v}_\alpha^0(t) - \vec{v}_\alpha(t)}{\tau_\alpha} \tag{2}$$
- **(Private sphere)** The motion of a pedestrian α is influenced by the presence of other agents. In particular, people usually feel uncomfortable when they get too close to a stranger and therefore they would try to keep a certain distance from other agents while walking. In the social force model, this results in a repulsive force from every other agent β:
  $$\vec{f}_{\alpha\beta}(\vec{r}_{\alpha\beta}) := -\nabla_{\vec{r}_{\alpha\beta}} V_{\alpha\beta}(b(\vec{r}_{\alpha\beta})) \tag{3}$$

where $\vec{r}_{\alpha\beta}$ is the distance from agent α to agent β and V is a potential. The function b is given by:

$$2b := \sqrt{(||\vec{r}_{\alpha\beta}|| + ||\vec{r}_{\alpha\beta} - v_\beta \Delta t \vec{e}_\beta||)^2 - (v_\beta \Delta t)^2} \tag{4}$$

i.e. the semi-minor axis of the ellipse delimiting the private sphere of the moving agent.

- **(Border effects)** The pedestrian keeps a certain distance from borders of buildings, since when it walks close to them it would have to pay attention to the environment not to get hurt.

  Therefore, every border B generates a repulsive effect:

  $$\vec{F}_{\alpha B}(\vec{r}_{\alpha B}) := -\nabla_{\vec{r}_{\alpha B}} U_{\alpha B}(||\vec{r}_{\alpha B}||) \tag{5}$$

  where $\vec{r}_{\alpha B}$ is the distance from the agent to the border and U is a potential.

All these effects are independent and influence agents' decisions at the same time, therefore their total effect is given by the sum of all of them:

$$\vec{F}_\alpha(t) := \vec{F}^0_\alpha(\vec{v}_\alpha, v^0_\alpha \vec{e}_\alpha) + \sum_\beta \vec{F}_{\alpha\beta}(\vec{e}_\alpha, \vec{r}_\alpha - \vec{r}_\beta) + \sum_B \vec{F}_{\alpha B}(\vec{e}_\alpha, \vec{r}_\alpha - \vec{r}^\alpha_B) \tag{6}$$

The social force model is then defined by:

$$\frac{\mathrm{d}\vec{w}_\alpha}{\mathrm{d}t} := \vec{F}_\alpha(t) + fluctuations \tag{7}$$

where the fluctuations take into account possible random variations of the agent's behaviour.

3 Implementation

The social force model was implemented in Python. The simulation includes three main modules acting together: the environment where the agents move, the agents themselves and their movement inside the environment. Each of these features is coded independently in different files and used together in `main.py`.

In the following the main functions of each feature's file are described.

3.1 Environment

The file `Environment.py` builds the settings where the agents move. A `Wall` is an environmental item defined as a segment in the plane, it takes as arguments two 2-dimensional numpy arrays containing the coordinates of the endpoints of the segment, the decay length of the force field due to a specific instance of the wall and a scale factor for the force intensity. An `Obstacle` is an environmental object defined as a point in the plane, it takes as arguments a 2-dimensional numpy array containing the coordinates of the obstacle, the decay length and a scale factor for the force intensity due to a specific instance of the obstacle.

Then the class `Environment` defines the environment where the agents move. In order to populate the environment with walls and obstacles, the functions `add_wall`, `add_obstacle`, `add_wall_from_polygonal` and `add_wall_from_curve` are provided. The function `compile` computes the force field due to the environmental items that were previously added. In particular, it generates a grid with a discretization length, previously given, and computes the force in every point of the grid. To avoid integration, which is computationally expensive, an approximation has been used. Each segment of the walls has been split in small sub-segments of fixed length, d, then the total force is given by the sum of each sub-segment's contribution. The force at position $\vec{r}$ due to each of them is of following form:

$$\vec{f}_{env}(\vec{r}) = Id\frac{\vec{r}-\vec{r}_{\text{proj}}}{||\vec{r}-\vec{r}_{\text{proj}}||^2}e^{-||\vec{r}-\vec{r}_{\text{proj}}||/R} \tag{8}$$

where I is the given scale factor for the intensity and R is the decay length of the item's force field and $\vec{r}_{\text{proj}}$ is the projection of $\vec{r}$ onto the sub-segment.

Then two quintic interpolation models retrieve the value of the field in an arbitrary point of the plane.

3.2 Agent

The file `Agent.py` contains the class `Agent` that defines the behaviour of each pedestrian during the simulation. The function `compute_force` computes the sum of the forces at the agent's position as in Eq. (6) using the previously computed force field of the environment. Where the repulsive force between two agents is given by the potential:

$$V_{\alpha\beta}(b) = V^0_{\alpha\beta}e^{-b/\sigma} \tag{9}$$

where $V^0_{\alpha\beta}$ is the scale factor and σ is the decay length of the potential. The resulting force is then rotated by an random angle θ distributed as Gaussian with zero as mean value and standard deviation of $\frac{\pi}{6}$.

Then, the `move` function updates speed and position of the agent according to the final force computed. If the new speed exceeds a fixed value v_{max}, then it takes this value instead of the new one.

Finally, the function `plot_trajectory` plots the path of the agent from the beginning of the simulation to the end.

3.3 Simulation

The file `Simulation.py` contains the `Simulation` class which accounts for the whole simulation. It takes as arguments the environment in which the agents are moving and the time step of the simulation, then it provides a function to add agents to the simulation and it automatically compiles the environment.

The function `move` implements a step of the simulation from time t to $t + \Delta t$, calling the function `agent.move` for each active agent.

The function `run` implements the whole simulation from the initial time till every agent reaches its goal or till a maximum time t_{max}. At every time step, it first checks if any new agent has to join the simulation at t and eventually adds it/them, then it calls the function `move`.

The evolution of the system can be visualised with `plot` and `save_frames`.

The function `mean_TimeToGoal` computes the average over all the agents of the time needed by each of them to reach its goal.

3.4 Choice of hyper-parameters

The agent-based model numerically implemented as explained above depends on a quantity of hyper-parameters. These are to be fine-tuned in order for the simulation to be a good representation of the reality. Doing such a work is far from trivial and requires empirical data or divine inspiration. As a consequence we chose to take nearly all these values as the same from [5]. More specifically the values that we used can be found in the following table.

Name	v^0_α	τ_α	σ	$V^0_{\alpha\beta}$	R
Value	$\mathcal{N}(1.34,\, 0.26)$	0.5	0.3	2.1	1

Table 1: Table showing the hyper-parameters used in the simulations.

4 Simulation Results and Discussion

To test the implemented model, different scenarios where compared, leading to the final classroom simulation. The common surroundings to all the tests is a square room delimited by walls of 10 m of length. A variable number of exits (1, 2 or 4) is used to study the outcome of different numbers of crossing points. A total of 50 agents, inserted uniformly at random in the first 10 seconds of simulation, move in every environment that we analysed. After the tests, a rectangular classroom is simulated with different table configurations to investigate which one is quicker to evacuate and hence safer in case of emergency.
The performance of each configuration is quantified by the *mean time to goal* $\langle t \rangle$,

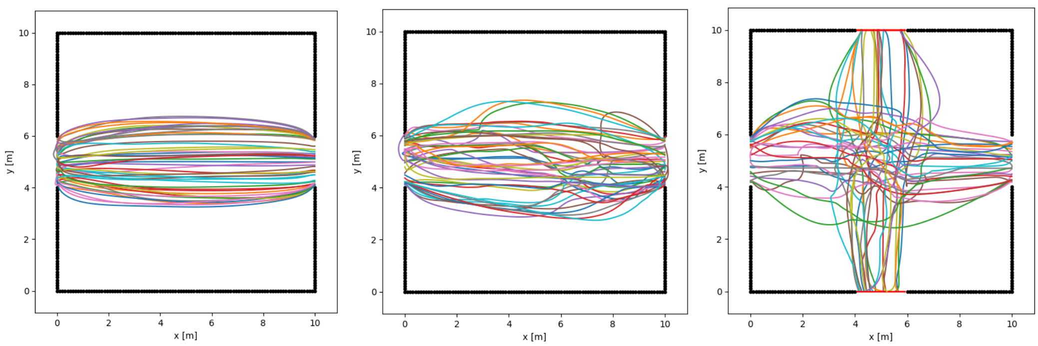
Environment	$\langle t_1 \rangle$ [s]	$\langle t_2 \rangle$ [s]	$\langle t_4 \rangle$ [s]
Empty room	7.27	8.12	10.90
Wedge	7.01	12.09	10.67
Pillars	8.85	9.07	10.18
Random objects	16.41	12.77	21.33
Grid	7.11	6.61	16.70
Shifted grid	7.01	9.77	13.97

Table 2: Table showing the mean time that an agent takes to travel to its destination exit rounded to two decimal places. $\langle t_i \rangle$ measures the time of the configuration with i exits.

shown in Table 2. This represents the average time over all the agents needed to leave the room. The complexity of the environment is progressively increased. The graphs shown in this section illustrate uniquely the final full trajectories of the agents. that are already enough to observe general trends. In addition, animations of the various environments tested can be found on `github` [1].

4.1 Empty room with varying number of exits

The first and simplest test of the model is done by studying an empty room with 1, 2 or 4 exits. The exits are placed one in front of the other to maximise the number of interactions that the agents will have. In Figure 1a, when only one exit is present, the agents directly move from the entrance to the exit along the fastest route. In addition, as expected, as soon as more exits are added and hence the number of interactions between the agents rises, as in Figures 1b and 1c, the mean time of travel increases as shown in Table 2. Figure 1 over all shows how the flow of agents through four



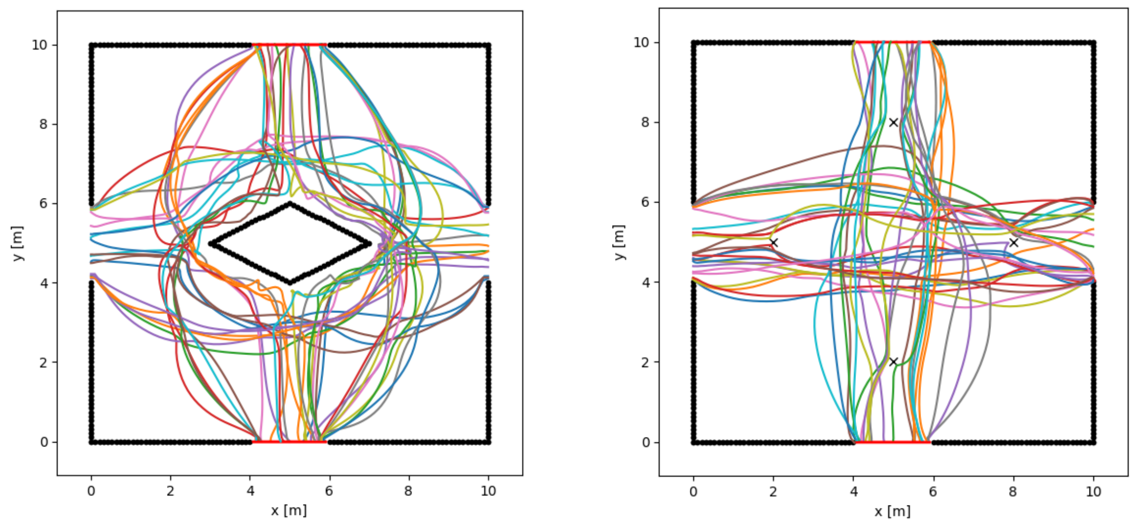
(a) One exit (b) Two exits (c) Four exits

Figure 1: Simulation of an empty room with 1, 2, and four exits. The coloured lines represent the trajectories of the 50 agents simulated. In addition, the force field of the environment is shown.

opposite doors is not efficient in an empty room. The following sections will hence explore how the situation changes with the addition of different obstacles in the same room. From now on, only simulations with four exits will be investigated in order to have the highest possible number of encounters between agents.

4.2 Wedge and obstacles in front of exits

Since the point of interaction of the agents is the centre of the square room, the next environment simulated has a wedge there as shown in Figure 2a. The resulting trajectory lines though do not appear smooth as along the sides of the wedge agents still get stuck between the wall and other incoming agents. The mean time of travel increases compared to the empty room environment as shown in Table 2.

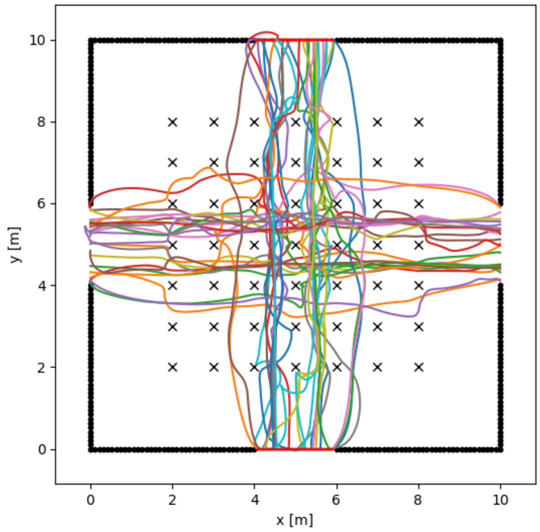


(a) Four exits with one central wedge

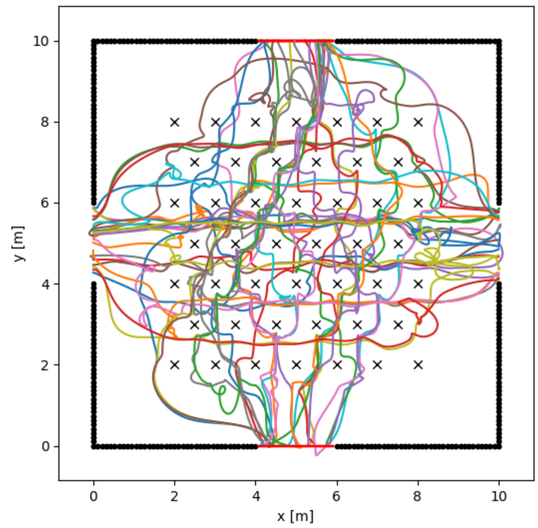
(b) Four exits with an obstacle in front of each exit

Figure 2: Static trajectories of 50 agents in a room with four entries and exits. The "flows" in Figure b with pillars in front of the destinations are much more stable than those in Figure a with a wedge wall.

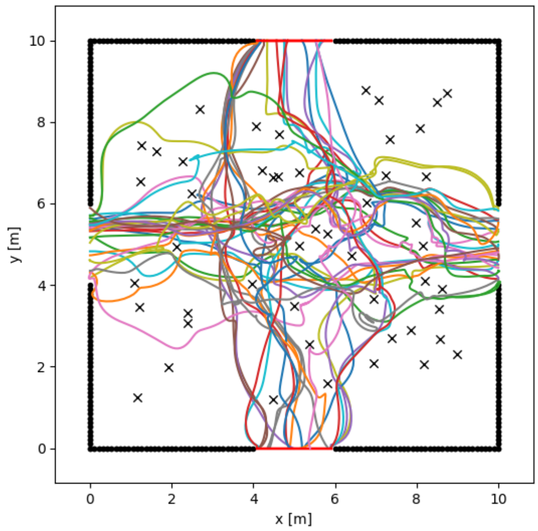
Since the addition of an object in the middle of the room did not improve the flow of people, what is then implemented is to add an obstacle, which could represent a pillar, near the entrances/exits. This allows two "streams" of agents per entrance to arise (since some will go on the left and some on the right of the obstacle). Breaking the path of agents into smaller ones results in organised trajectories with very little deviations from the fastest travel path. The efficiency of this environment is proved by its low mean time of travel as shown in Table 2 confirming the findings in Ref. [4].



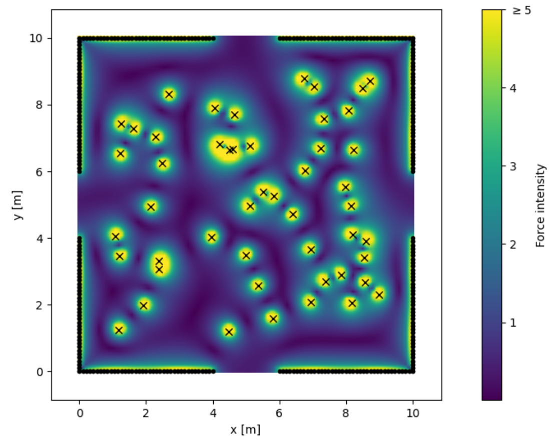
(a) Ordered grid of obstacles



(b) Shifted grid of obstacles



(c) 49 random obstacles



(d) Environment force field for the random obstacles

Figure 3: Simulation of an empty room with 1, 2, and four exits. The coloured lines represent the trajectories of the 50 agents simulated. In addition, the force field of the environment is shown.

All of the previous simulations represent the movement of agents in a empty room (if not for a wedge or four pillars) but this is almost never the case in real life examples. For instance, in an office room, there will be desks on which people work and other objects such as plants or furniture that are in the way of the agents' movements. To represent and investigate this scenario in the next sections a higher number of obstacles are added to the environment.

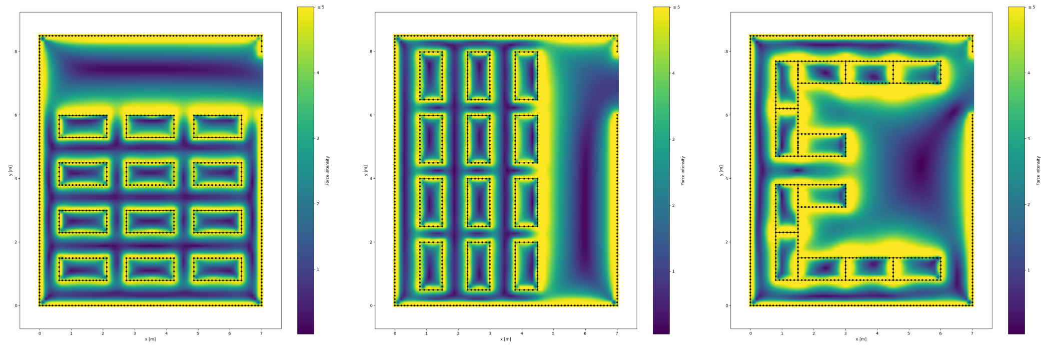
4.3 Random obstacles and grids of obstacles

To replicate the presence of multiple obstacles in a room, 49 objects were used. Three different configurations are simulated to understand how the change of environment affects the agents' behaviour. Firstly, a set of 49 random objects was generated as shown in Figure 3c. This particular configuration appears to be the worst so far with an average time of 21.33 seconds in an environment with four entries and exits. This is visible as well from the force field graph in Figure 3d, which shows a big number of inhomogeneities that result in the slowing down of agents. It is then important to arrange obstacles in a room in an organised manner to maximise the performance of the chosen environment. In Figure 3a is shown a simulation with the use of an ordered grid, which appears to have a very good efficiency with time 16.70 seconds. Comparing this arrangement to the one in Figure 3b there is not much difference to be seen. Shifting the rows by half a distance between columns, resulting in a "slalom-like" path, doesn't change the mean time of the agents all that much. But compared to the randomly placed obstacles, these results show the key role that the correct placement of the same number of obstacles plays in the flow of people through a room. If there is a need for a specific room to be evacuated quickly it is better for it to be arranged in such a way that all the possible obstacles appear in a neat and organised manner.

4.4 Optimal desk configuration in a classroom

In this section, the optimal configuration of 12 desks in a classroom is investigated. For the same room of 7 by 8.5 meters, three different configurations of desks are tested. In the first one, the students are facing the short side of the room, in the second one they are facing the long side and in the last one the tables are arranged in a U shape. For each one of the configurations, 24 agents have to evacuate the room starting in front of their desks. The mean time $\langle t \rangle$ is still used as a quality measure and it is shown in Table 3.

In figure Fig. 4 it is shown the force field computed in the three cases. The simulations have been run until all the agents reached the exit. Figure 5 illustrates the trajectories resulting from one of the simulations. Whereas the values for the *mean time to goal*, obtained averaging over 9 independent simulations, are given in Table 3.

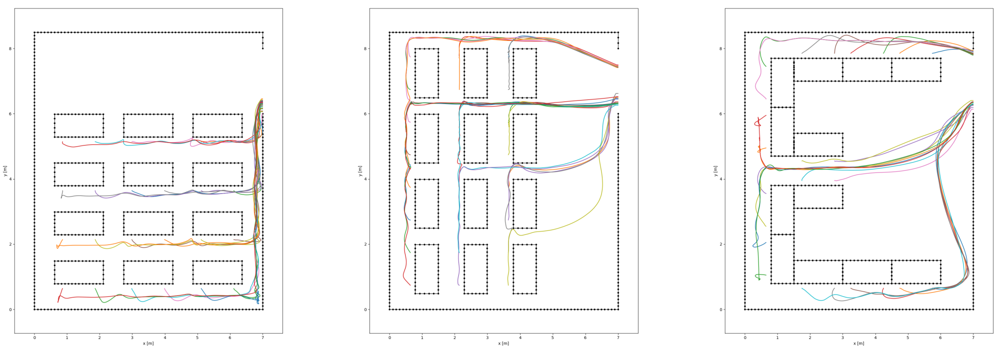


(a) Configuration with students facing the short side of the room

(b) Configuration with students facing the long side of the room

(c) U-shaped configuration

Figure 4: Force field for each of the three considered configurations. The classroom is 7 by 8.5 meters and each desk is 1.5 by 0.7 meters.



(a) Configuration with students facing the short side of the room

(b) Configuration with students facing the long side of the room

(c) U-shaped configuration

Figure 5: Agents' trajectories resulting from a simulation for each of the three considered configurations.

Configuration	$\langle t \rangle$ [s]	$\sigma_{\langle t \rangle}$ [s]
Facing short side	10.58	0.04
Facing short side	12.4	0.2
U-shaped	8.5	0.1

Table 3: Table showing the mean time needed to evacuate the classroom. The average is computed over a set of 9 independent runs.

Note that the fluctuations in Tab. 3 are relatively small compared to the absolute values, meaning that the model is robust with respect to the fluctuations in (7).

Above all the three configuration tested, the one resulting in the lowest mean time to goal is with the desks are arranged in a U-shaped fashion. It is probably the case because of the presence of the central lane, which provides the agents with a direct way out of the room.

In addition to the trajectories in Fig. 5, in the `GitHub` repository are saved the animations of the agents' simulated motion. It is worth mentioning that some of the agents got stuck in positions for which the implemented model doesn't produce meaningful results. As example can be found in the animation relative to the U-shaped configuration. The last agent reaching the exit walks indeed too slowly to regard his motion as a realistic one in an emergency situation. This behaviour is probably due to the model, which aims to move the agent towards the exit in a straight line, without taking into account his surroundings.

5 Summary and Outlook

In conclusion, the social force model for individual behaviour is a valid tool to describe pedestrian motion in terms of Langevin equations. Given the equations that regulate the motion of each agent, computer simulations can be exploited to solve them and analyse specific situation of everyday life. The results obtained might give interesting insights on optimal configurations in evacuation scenarios as well as methods to enhance the flow of pedestrians in crowded buildings.

Further measurements of the variables of pedestrian motion should be performed in order to improve the social force model and better describe human behaviour. Moreover, the repulsive forces from other agents and the borders should be modified and motivated in terms of human perceptions and emotions; for example, the potential from a wall in a museum should look like the potential of the hydrogen atom, with a divergence at short distances and an equilibrium point, where the agent can appreciate the opera. On the other hand, in a very crowded party room the repulsion from the borders might depend to each agent's personality: it should drop fast to zero for shy people who don not want to be in the middle of the party while it could be the opposite for others. This model does not account for attractive forces between agents (friends), but it could be easily implemented by adding an attractive term in(6).

Furthermore, the study of pedestrian behaviour is not the only social system that can be described with a social force model. Opinion formation, traffic dynamics and the spread of an epidemic can be modelled by introducing other forces/parameters or abstract behavioural spaces.

6 References

[1] Battista ; Biella ; Ferradini ; Fioroni ; Guillement: *ETH GESS: Modeling pedestrian flow in a crowded building.* – URL `https://github.com/LorenzoFioroni/ETH-GESS`

[2] Feng, Shumin ; Ding, Ning ; Chen, Tao ; Zhang, Hui: Simulation of pedestrian flow based on cellular automata: A case of pedestrian crossing street at section in China. In: *Physica A: Statistical Mechanics and its Applications* 392 (2013), 07, S. 2847–2859

[3] Helbing, Dirk: A Fluid Dynamic Model for the Movement of Pedestrians. In: *Complex Systems* 6 (1998), 06

[4] Helbing, Dirk ; Buzna, Lubos ; Johansson, Anders ; Werner, Torsten: Self-Organized Pedestrian Crowd Dynamics: Experiments, Simulations, and Design Solutions. In: *Transportation science* 39 (2005), Nr. 1, S. 1–24. – ISSN 0041-1655

[5] Helbing, Dirk ; Molnár, Péter: Social force model for pedestrian dynamics. In: *Physical Review E* 51 (1995), May, Nr. 5, S. 4282–4286. – ISSN 1095-3787

[6] Henderson, L.F.: On the fluid mechanics of human crowd motion. In: *Transportation Research* 8 (1974), Nr. 6, S. 509–515. – ISSN 0041-1647

[7] Kormanová, Anna: A Review on Macroscopic Pedestrian Flow Modelling. In: *Acta Informatica Pragensia* 2 (2013), 12, S. 39–50

[8] Shi, Meng ; Ming Lee, Eric W. ; Ma, Yi: A Newly developed Mesoscopic Model on Simulating Pedestrian Flow. In: *Procedia Engineering* 211 (2018), S. 614–620. – 2017 8th International Conference on Fire Science and Fire Protection Engineering (ICFSFPE 2017). – ISSN 1877-7058

[9] Wolfram, Stephen: Cellular automata as models of complexity. In: *Nature (London)* 311 (1984), Nr. 5985, S. 419–424. – ISSN 0028-0836