# The Lagrangian coordinate system and what it means for two-dimensional crowd flow models

Femke van Wageningen-Kessels, Ludovic Leclercq, Winnie Daamen, Serge P. Hoogendoorn

▶ **To cite this version:**

Femke van Wageningen-Kessels, Ludovic Leclercq, Winnie Daamen, Serge P. Hoogendoorn. The Lagrangian coordinate system and what it means for two-dimensional crowd flow models. Physica A: Statistical Mechanics and its Applications, 2015, 19 p. 10.1016/j.physa.2015.09.048 . hal-01215762v1

**HAL Id: hal-01215762**

**https://hal.science/hal-01215762v1**

Submitted on 14 Oct 2015 (v1), last revised 11 May 2017 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# The Lagrangian coordinate system and what it means for two-dimensional crowd flow models

Femke van Wageningen-Kessels, Ludovic Leclercq,
Winnie Daamen and Serge P. Hoogendoorn

April 21, 2015

**Abstract**

A continuum crowd flow model is solved using the Lagrangian coordinate system. The system has proven to give computational advantages over the traditional Eulerian coordinate system for (one-dimensional) road traffic flow. Our extension of the model and simulation method to (two-dimensional) crowd flow paves the way to explore the advantages for crowd flow simulation. This paper provides a first exploration and shows that a model and simulation method for two-dimensional crowd flow can be developed using Lagrangian numerical techniques and that it leads to accurate simulation results.

**keywords** Crowd flow, Model, Simulation, Lagrangian coordinates

## 1 Introduction

Crowd flow models are used to describe, understand and predict collective behaviour of crowds. Roughly, three types of models exist [Bellomo et al., 2012, Duives et al., 2013]: microscopic models in which the movements of individual pedestrians are described and traced, mesoscopic models in which position and velocity of individual pedestrians is described using probability distributions and macroscopic (or continuum) models in which variables are aggregated and which are the focus of this study.

Continuum models describe the dynamics of crowds as a continuum flow, in terms of average speed, velocity and density [Hughes, 2002]. Recent research shows that continuum models can reproduce self-organisation and certain important dynamic phenomena such as lane formation and diagonal striping [Jiang et al., 2012, Hoogendoorn et al., 2014, 2015], which is in contrast to previous claims. Furthermore, simulations based on continuum models have the potential to significantly reduce computation time whilst keeping high accuracy. This makes them useful for a larger range of applications, including real time state estimation and prediction for crowd management and optimisation of control strategies, under normal operating conditions and for evacuations. For this, the solutions to the model equations need to be calculated using efficient (i.e. both fast and accurate) computational methods.

The Lagrangian coordinate system has been applied to traffic flow models [Aw et al., 2002, Leclercq et al., 2007]. Van Wageningen-Kessels et al. [2010], Yuan et al.

[2012] show that the system has many advantages over the traditional Eulerian coordinate system, including more accurate reproduction of shock waves and more efficient state estimation based on trajectory data. It is expected that the advantages also hold in the case of two-dimensional crowd flow models resulting in more efficient simulation results than with methods based on the Eulerian coordinate system such as proposed by Dogbé [2008], Huang et al. [2009], Jiang et al. [2012], Hänseler et al. [2014]. In this contribution we focus on showing how crowd simulations can be done using the Lagrangian coordinate system. We also give some insight into the advantages of Lagrangian crowd simulations. Future studies will have to provide a definitive answer to the question under which circumstances the Lagrangian coordinate system is superior to the Eulerian coordinate system for crowd simulation.

Our main contribution is the proposal of an extension of Lagrangian simulation methods previously applied for one-dimensional traffic flow models to two-dimensional crowd flow models. We introduce the reformulation of the two-dimensional continuum crowd flow model into Lagrangian coordinates (Section 2), develop a numerical simulation method (Section 3) and show that it leads to meaningful simulation results (Section 4). The contribution is summarised and open questions and future research directions are discussed in Section 5.

# 2 Crowd flow model and its Lagrangian formulation

In a traditional, Eulerian (fixed) coordinate system the density at location $\vec{x}$ and time $t$ is calculated:

$$\rho = \rho(\vec{x}, t) \tag{1}$$

In contrast, the main idea of the Lagrangian (or moving) coordinate system is that the location $\vec{x}$ of the $n$-th particle (i.e. a vehicle or a pedestrian) is calculated at time $t$:

$$\vec{x} = \vec{x}(n, t) \tag{2}$$

In vehicular traffic the $n$-th vehicle usually follows the $(n-1)$-th vehicle. Such a simple ordering of pedestrians is usually not possible and limits the model formulation. We show how the Lagrangian model can be formulated nevertheless, but without a closed-form conservation equation.

## 2.1 Conservation model

Our approach includes a physical explanation of the conservation of particles. In Eulerian coordinates, the one-dimensional conservation equation is well-known:

$$\frac{\partial \rho}{\partial t} + \frac{\partial q}{\partial x} = 0 \tag{3}$$

with $\rho$ the density (vehicles per meter), and $q$ the flow (vehicles per second unit). The Lagrangian equivalent of the one-dimensional conservation equation (3) is as follows:

$$\frac{\partial(1/\rho)}{\partial t} + \frac{\partial v}{\partial n} = 0 \tag{4}$$

with $v = q/\rho$ the vehicle speed (meter per second). To derive the Lagrangian equivalent (4) of the one-dimensional conservation (3) equation in Eulerian coordinates, a Moskowitz function can be defined [Newell, 1993]. Therefore, a numbering of the vehicles $n = 1, \ldots, N$ is used and their position as a function of vehicle number $x(n)$ is defined [Aw et al., 2002, Leclercq et al., 2007, van Wageningen-Kessels et al., 2010]. However, for crowd flow, such a numbering of the pedestrians is not possible and pedestrians can not be numbered consistently. This prevents a continuous mapping of the pedestrians with respect to space and time. Therefore, an alternative approach is proposed.

### 2.1.1 Two-dimensional conservation model

The two-dimensional extension of the Eulerian conservation equation (3) is well-known:

$$\frac{\partial \rho}{\partial t} + \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} = 0 \tag{5}$$

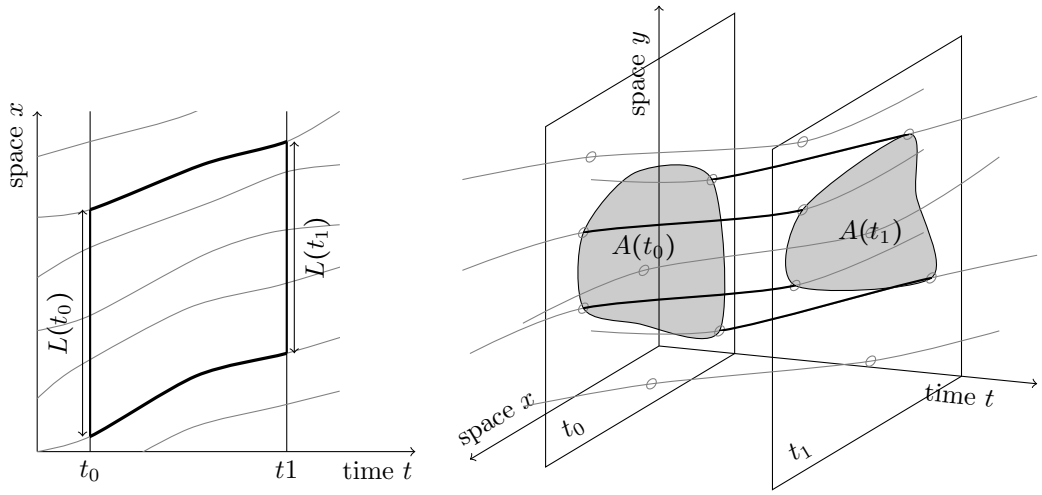
with $\rho$ the density (pedestrians per square meter), and $q_x$ and $q_y$ the flow in $x$- and $y$-direction, respectively (pedestrians times meter per second).

### 2.1.2 Two-dimensional Lagrangian conservation model

We use the Lagrangian approach to solve the conservation equation in two dimensions (5). Therefore, fixed portions of particles (i.e. groups of vehicles or pedestrians) are defined and their position and speed is traced over time, see Figure 1. Note that the portion does not necessarily include an integer number of particles but instead the number of vehicles or pedestrians can take any positive value. In one-dimensional road traffic flow, the portion of vehicles occupies a certain length of road $L(t)$ at time $t$. In two-dimensional crowd flow, the portion of pedestrians occupies a certain surface $A(t)$ at time $t$. The evolution of the size of such portions (i.e. the evolution of $L(t)$ or $A(t)$ over time) and their location only depends on the movement of the particles located at the boundary. Therefore, we have to assume that the boundary particles remain at the boundary and that there is no switching of positions, i.e. vehicles or pedestrians do not overtake each other. This assumption is true if the speed profile is smooth enough and the evolution is followed over a small time period. The inflow into the portion and the outflow out of the portion are by definition zero. Therefore, we can apply the conservation principle and we find for one-dimensional flow:

$$L(t_1) = L(t_0) + \int_{t_0}^{t_1} \left( v_{\text{first}}(t) - v_{\text{last}}(t) \right) \mathrm{d}t \tag{6}$$

with $v_{\text{first}}$ and $v_{\text{last}}$ the speed of the first and the last vehicle in the portion, respectively.




(a) One-dimensional road traffic flow.

(b) Two-dimensional crowd flow.

**Figure 1:** Following of the evolution of a fixed portion of particles (vehicles or pedestrians) over time. Gray lines are trajectories.

The same principles can be applied to find the area occupied by a portion of the pedestrian in two-dimensional crowd flows. Therefore, we first consider the movement of one pedestrian $j$, which is at location $\vec{x}_j(t_0)$ at time $t = t_0$ and moves with velocity $\vec{v}_j(\vec{x}, t)$. At time $t = t_1 > t_0$ its location will be:

$$\vec{x}_j(t_1) = \vec{x}_j(t_0) + \int_{\tau=t_0}^{t_1} \vec{v}_j(\tau)\, \mathrm{d}\tau \tag{7}$$

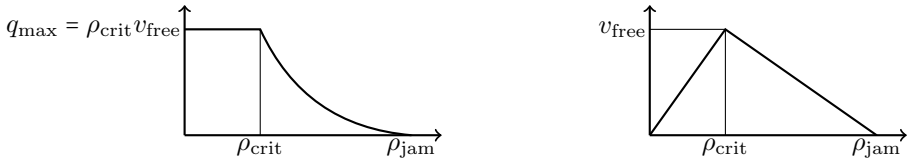
Secondly, we consider a region $\Omega_i$ with $N_i(t) = \int_{\vec{x}\in\Omega_i(t)} \rho(\vec{x}, t)\, \mathrm{d}\vec{x}$ pedestrians in it. The area of this region equals:

$$A_i(t) = \int_{\vec{x}\in\Omega_i(t)} 1\, \mathrm{d}\vec{x} = \oint_{\vec{x}\in\Gamma_i(t)} \vec{x}\, \mathrm{d}\vec{x} \tag{8}$$

with $\Gamma$ the boundary of the region. In the Lagrangian coordinate system, the region moves with the pedestrians, the pedestrians on the boundary, are assumed to stay on the boundary, the flux over the boundary is zero, and thus the number of pedestrians does not change ($N(t_1) = N(t_0)$). However, the region and its boundary move if the speed of the pedestrians on the boundary is nonzero ($\Gamma_i(t_1) \neq \Gamma_i(t_0)$). Using both (7) and (8), we can compute the area $A(t_1)$ at time $t = t_1 > t_0$, see also Figure 1(b):

$$\begin{aligned} A_i(t_1) &= \underbrace{\oint_{\vec{x}\in\Gamma_i(t_1)} \vec{x}\, \mathrm{d}\vec{x}}_{\text{new area}} = \underbrace{\oint_{\vec{x}\in\Gamma_i(t_0)} \vec{x}}_{\text{old area}} + \underbrace{\int_{\tau=t}^{t_1} v(\vec{x}, \tau)\, \mathrm{d}\tau\, \mathrm{d}\vec{x}}_{\text{change in area}} \\ &= A_i(t_0) + \oint_{\vec{x}\in\Gamma_i(t_0)} \int_{\tau=t_0}^{t_1} v(\vec{x}, \tau)\, \mathrm{d}\tau\, \mathrm{d}\vec{x} \end{aligned} \tag{9}$$

The above concept of change in area occupied by a group of pedestrians according to the velocity of the pedestrians at the boundary, forms the basis of the numerical computation method, as outlined in Section 3.




(a) Density-speed. (b) Density-flow.

**Figure 2:** Two representations of the bi-linear fundamental diagram.

## 2.2 Velocity: speed and route choice

To complete the model, we also need to describe the velocity $\vec{v}(\vec{x},t)$. We propose a simple velocity model, which may be applied either in Lagrangian or in Eulerian coordinates. The velocity is calculated using a simplified version of the approach proposed by Hoogendoorn et al. [2014, 2015]:

$$\vec{v}(\vec{x},t) = V(\rho(\vec{x},t))\,\vec{e}(\vec{x},t) = V(\rho(\vec{x},t))\,\frac{\vec{e}_{stat}(\vec{x},t) + \vec{e}_{dyn}(\rho(\vec{x},t))}{\|\vec{e}_{stat}(\vec{x},t) + \vec{e}_{dyn}(\rho(\vec{x},t))\|} \tag{10}$$

with $V(\rho)$ the fundamental relation and $\vec{e}$ the route choice (i.e. walking direction). The route choice $\vec{e}$ is composed of the static route choice component $\vec{e}_{stat}$ and the dynamic (local, state dependent) route choice component $\vec{e}_{dyn}$.

We are well aware that other, possibly more realistic, speed and route choice models exist. However, all models include key variables such as speed, density and density gradient, just like our simple model. This allows us to develop a numerical method, which could also be applied to more refined models. We emphasise that the simple version of the velocity model will suffice to illustrate the numerical method, which is the focus of our current study.

### 2.2.1 Speed

In our simplified approach, the speed only depends on density, following a bi-linear fundamental relation, see also Figure 2:

$$V(\rho) = \begin{cases} v_{free} & \text{if } \rho \in [0, \rho_{crit}] \\ \frac{\rho_{crit} v_{free}}{\rho_{jam} - \rho_{crit}} \left( \frac{\rho_{jam}}{\rho} - 1 \right) & \text{if } \rho \in (\rho_{crit}, \rho_{jam}] \end{cases} \tag{11}$$

with $v_{free}$ free flow speed, $\rho_{crit}$ critical density and $\rho_{jam}$ jam density.

### 2.2.2 Route choice: static and dynamic

The static route choice component $\vec{e}_{stat}$ depends on the layout of the facility and possibly other factors. It can, for example reflect the shortest route to the destination. In the simulations (Section 4) we apply three different static route choices, which are described in more detail there.

For the dynamic route choice we apply a simple model reflecting that pedestrians avoid high density regions:

$$\vec{e}_{dyn} = -\beta_{dyn} \nabla \rho \tag{12}$$

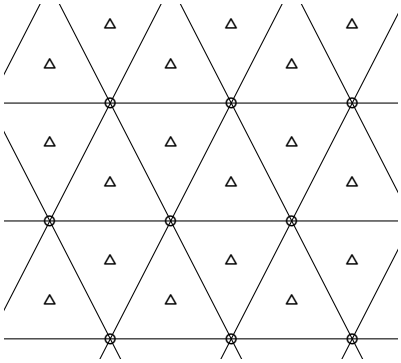


**Figure 3:** A staggered triangular grid. Density $\rho_{\vartriangle}$ is calculated at the cell centres, velocity $\vec{v}_{\circ}$ is calculated at the vertices.

with $\beta_{\text{dyn}} \geq 0$ the weight of the dynamic route choice. The dynamic route choice component steers pedestrians in the direction with the steepest negative density gradient., i.e. away from other pedestrians.

# 3 Simulation approach

The main idea of the simulation approach using the Lagrangian coordinate system is that a moving grid is applied. In particular, Lagrangian cells correspond to a fixed portion of the crowd and the grid moves with the pedestrians in that portion. In the following we discuss how the grid is constructed, how the velocity of the pedestrians and the vertices of the grid is computed and how the grid is moved each time step $\Delta t$. Furthermore, we discuss when and how the grid is reconstructed (remeshing).

## 3.1 The grid

We assume triangular grid cells for simplicity, but the methods can be extended to other polygon-shaped grid cells. The grid is staggered, meaning that some variables are calculated at the cell centres, others at the vertices, see Figure 3. We do not consider any variables calculated at the edges. The cell centres are defined as the centroid of the cells. Variables calculated at the cell centres are indicated with a subscripted triangle ($\vartriangle$). Variables calculated at the vertex are indicated with a subscripted circle ($\circ$). The main variable to be calculated at the cells centres is density ($\rho_{\vartriangle}$). The main variable to be calculated at the vertices is velocity ($\vec{v}_{\circ}$). This is consistent with the cell definition: calculating the velocity at the vertices allows the cells to be moved, calculating the density at the interior is natural as each cell contains a fixed number of pedestrians.

## 3.2 Velocity of grid cell vertices

At the start of each time step, the location of the grid cells and their vertices and the number of pedestrians within each cell is known from the previous time step. Recall that the number of pedestrians in the portion can take any positive value and is not integer. From the location of the vertices and the number of pedestrians in a cell, we

approximate the velocity of the grid vertices, as outlined below. We first discuss the method for an inner vertex, later the adaptations for a boundary vertex are discussed.

### 3.2.1 Density in grid cells

The average density in the grid cell is calculated:

$$\rho_{\triangle} = \frac{N_{\triangle}}{A_{\triangle}} \tag{13}$$

and serves as an approximation of the density at the grid cell centre. To simplify notation, we leave out the time index in this section wherever possible. $N_{\triangle}$ is the number of pedestrians in the grid cell (remains constant over time). $A_{\triangle}$ is the area of the grid cell (may change each time step). In general, the area is computed as in (8), but because the grid cell is triangular we can use:

$$A_{\triangle} = \tfrac{1}{2} \left| (x_{j_1} - x_{j_3})(y_{j_2} - y_{j_1}) - (x_{j_1} - x_{j_2})(y_{j_3} - y_{j_1}) \right| \tag{14}$$

with the locations of the vertices $\{j_1, j_2, j_3\} \in J$ of the triangle (see also Figure 4(a)):

$$\vec{x}_{\circ,j} = \begin{pmatrix} x_j \\ y_j \end{pmatrix} \tag{15}$$

### 3.2.2 Speed at vertices

To obtain the speeds from the fundamental relation, the densities at the vertices are needed. The density at the vertex is an interpolation of the densities at the centroids of the surrounding cells, see Figure 4. The centroids (cell centres) $\vec{x}_{\triangle,i}$ are calculated as the centre of mass of the cells:

$$\vec{x}_{\triangle,i} = \tfrac{1}{3} \sum_{j \in J_i} \vec{x}_{\circ,j} \tag{16}$$

with $j \in J_i$ the indices of the vertices of cell $i$. The density at a vertex $\rho_{\circ}(x_{\circ}, y_{\circ})$ is calculated as a linear interpolation of the densities in the surrounding cells $\rho_{\triangle}(x_{\triangle}, y_{\triangle})$. This results in:

$$\rho_{\circ}(x_{\circ}, y_{\circ}) = c_1 x_{\circ} + c_2 y_{\circ} + c_3. \tag{17}$$

with parameters $c_1$, $c_2$ and $c_3$ determined by the linear interpolation function, which uses a truncated Taylor series [D'Errico, 2012]. Subsequently, the speed of the vertices are approximated using the fundamental relation $v_{\circ} = V(\rho_{\circ})$, see equation (11). In this simple algorithm we do not take into account anisotropy (the fact that pedestrians react differently to pedestrians in front than to those to the side and to the back). The method could possibly be improved by including anisotropy using weights for each of the surrounding cells.

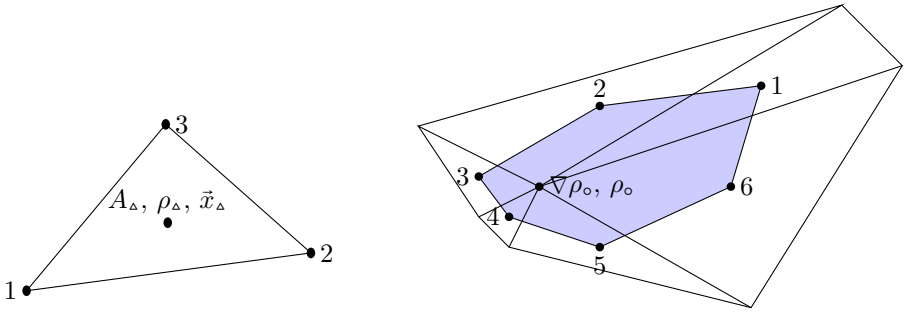
### 3.2.3 Route choice at vertices

For the dynamic route choice component we need the density gradient at the vertices. These can be obtained directly from the parameters of the linear interpolation (17), see also Figure 4:

$$\nabla \rho_\circ = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \tag{18}$$

We use $\vec{e}_{\mathrm{dyn},\circ} = -\beta_{\mathrm{dyn}} \nabla \rho_\circ$ (see equation (12)) to calculate the dynamic route choice component.

The static route choice component $\vec{e}_{\mathrm{stat}}$ is supposed to be given at any location and time.




(a) Vertex numbering to compute area, density and centre of a cell.

(b) Cell numbering to compute density and density gradient at vertex.

**Figure 4:** Examples of vertex and cell numbering. This is used to compute the area, density and centre of a cell (a) and to compute the density and density gradient at a vertex (b).

### 3.2.4 Velocity of vertices

We can now approximate the velocity of the vertex (see equation (10)):

$$\vec{v}_\circ = v_\circ \frac{\vec{e}_{\mathrm{stat},\circ} - \beta_{\mathrm{dyn}} \nabla \rho_\circ}{\|\vec{e}_{\mathrm{stat},\circ} - \beta_{\mathrm{dyn}} \nabla \rho_\circ\|} \tag{19}$$

### 3.2.5 Boundary conditions

At the boundaries, vertices are only partly surrounded by cells and density and density gradient can not be computed straightforwardly as in (17) and (18). A simple approach to deal with boundaries is introduced. Alternative, more accurate numerical boundary conditions may be developed in future research. In the current approach, ghost cells are introduced, see Figure 5. They have their centroids opposite of the centroids of the surrounding cells of the boundary vertex. Furthermore, since we are only dealing with free boundaries in the test cases in the next section, the ghost cells have zero densities. To calculate the density gradient, the ghost cells are included in the linear regression model as in (17) and the density gradient is as in (18). However, when calculating density, ghost cells are only included in the regression model (17) if they are located in the walking direction according to the static route choice.

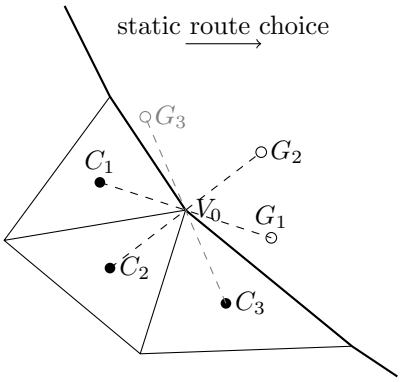



**Figure 5:** Example of boundary with ghost cells. $C_1$, $C_2$ and $C_3$ are the centroids of the cells neighbouring boundary vertex $V_0$. The centroids of their respective ghost cells are located at $G_1$, $G_2$ and $G_3$. Only ghost centroids $G_2$ and $G_3$ are in the static route choice walking direction and they are taken into account when computing the density at boundary vertex $V_0$. All ghost cells are taken into account when calculating the density gradient.

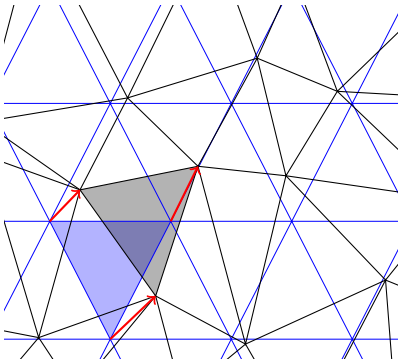


**Figure 6:** Example of movement of the grid during one time step. The old grid is shown in thin blue lines. The new grid in thick black lines. One grid cell is highlighted and the figure shows how it moves and deforms, following the velocities of its corners (red arrows).

## 3.3 Moving of grid cells

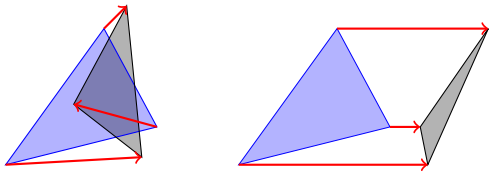
Once the velocity of the vertices is known, they can be moved. Therefore, we assume that the velocity remains as computed in (19) during the time step. The vertices are moved as follows:

$$\vec{x}_\circ^{\,\text{new}} = \vec{x}_\circ^{\,\text{old}} + \Delta t \vec{v}_\circ \tag{20}$$

This is illustrated in Figure 6. We assume that the velocity of the edges is a linear interpolation of the velocities of the neighbouring vertices. Therefore, edges remain straight and the cell remains triangular.

## 3.4 Remeshing the grid

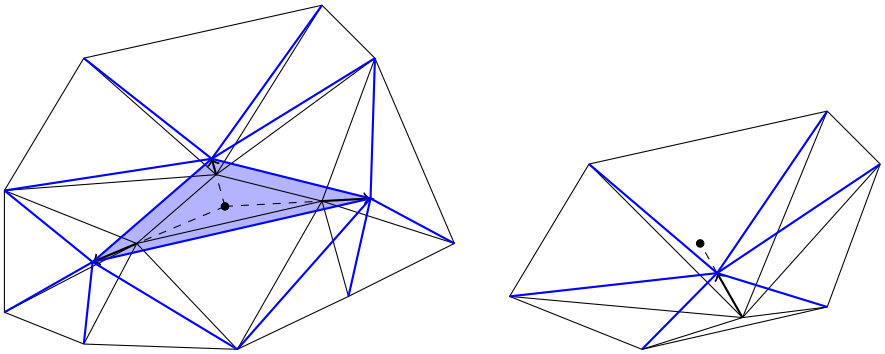
The vertices of an individual grid cell, will usually have different velocities. Therefore, the grid cell will change its shape. If this deforms the grid cells 'too much' and the grid becomes too irregular, it is be necessary to remesh. Furthermore, densities may become too large (i.e. above jam density) and remeshing becomes necessary for this reason. Finally, if vertices move in opposite directions or if they have large speed



(a) Opposite directions.

(b) Different speeds.

**Figure 7:** Cells 'flip' if the velocities of their vertices (red arrows) are very different. Blue cell with thin lines: old, gray cell with thick lines: new.



(a) Enlarging grid cell.

(b) Moving vertex in inner domain.

**Figure 8:** Illustrations of the remeshing approaches. Thin black line indicates the old mesh, thick blue ones the new mesh, arrows indicate the movement of the vertex or vertices. In Figure 8(a), the dot indicates the centroid of the cell that is to be enlarged, in Figure 8(b), the dot is the centroid of the polygon consisting of all triangles shown in the figure.

differences, the cell may 'flip' (see Figure 7).

In the remeshing process a new mesh is formed and the pedestrians in the old grid cells, are assigned to new grid cells in such a way that the density at each location $\vec{x}$ remains (approximately) the same and the total number of pedestrians in the computational domain does not change (conservation of pedestrians). Too achieve a computationally efficient remeshing process, only vertices and cells that satisfy certain remeshing criteria are remeshed, while others are left unchanged.

The objective of the remeshing is twofold: the density in each cell is to be kept below (or at) jam density and the grid is to be kept regular with the shape of the cells 'close enough' to an equilateral triangle. To achieve the first objective, grid cells that have densities close to jam density are enlarged, see Figure 8(a). To achieve the second objective vertices are moved if the are 'too close' to other vertices, see Figure 8(b). The combination of both the cell enlargement and the moving of vertices, prevents the cells from flipping. The remeshing procedures are detailed below.

### 3.4.1 Enlarging grid cells

A grid cell is enlarged if its density becomes too large:

$$\rho > \alpha_\rho \rho_{\text{jam}} \tag{21}$$

with $\alpha_\rho \in \left(\frac{\rho_{\text{crit}}}{\rho_{\text{jam}}}, 1\right]$ a parameter. Too enlarge the grid cells, all vertices of the cell are moved outward. They follow the line from the centroid ($\vec{x}_c$) of the grid cell through the original position of the vertex ($\vec{x}_i$), see Figure 8(a). The new position is:

$$\vec{x}_i^* = (1 + \beta_\rho)\vec{x}_i - \beta_\rho \vec{x}_c \tag{22}$$

with $\beta_\rho > 0$ a parameter. Currently, an appropriate value for $\beta_\rho$ is determined by trial and error for the simulations. A more rigorous approach, possibly taking into account the actual traffic state can be developed in the future.

### 3.4.2 Moving vertices to keep grid regular

Furthermore, a vertex $\vec{x}_i$ is moved if its position is far away from the centroid of the surrounding polygon $\vec{x}_c$, compared to its distance to any of its neighbouring vertices $\vec{x}_j$:

$$\|\vec{x}_i - \vec{x}_c\| > \alpha_c \max_j \|\vec{x}_i - \vec{x}_j\| \tag{23}$$

with $\alpha_c > 0$ a parameter. The vertex is then moved towards the centroid of its surrounding polygon, see Figure 8(b). Its new position is:

$$\vec{x}_i^* = (1 - \beta_c)\vec{x}_i + \beta_c \vec{x}_c \tag{24}$$

with $\beta_c \in (0, 1)$ a parameter. Just like with $\beta_\rho$, the value of $\beta_c$ is currently determined by trial and error for the simulations, leaving room for more efficient approaches.

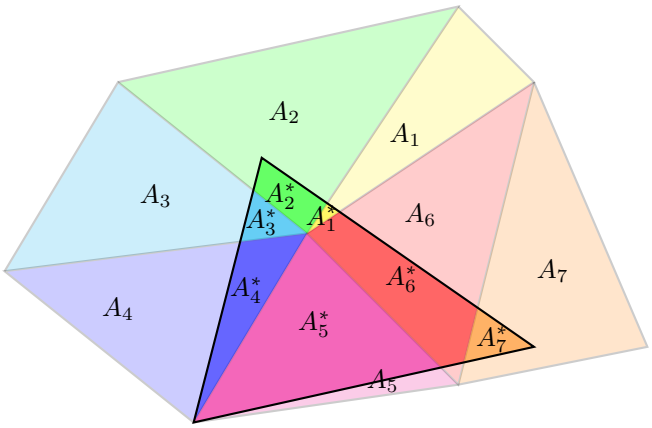
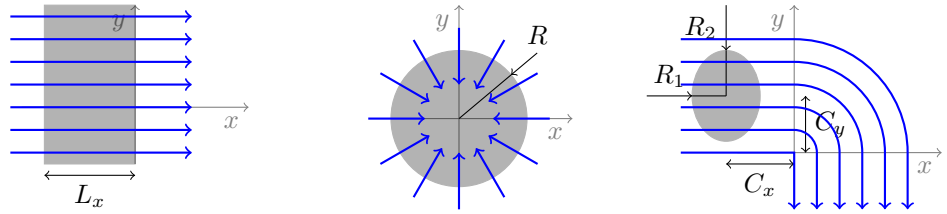



**Figure 9:** Example of recalculating densities after remeshing. The old mesh consists of (lightly coloured) triangles $A_i$. One triangle of the new mesh is drawn with thick black lines. It overlaps with 7 triangles of the old mesh. The intersections of the old and new triangles are brightly coloured and are indicated with $A_i^*$.

### 3.4.3 Recalculation the densities

After the new grid has been determined, the density in each of the reshaped grid cells is computed. Figure 9 shows an example. For the remeshing, density in the $j$-th




(a) Case 1: initial jam will dissolve as pedestrians walk to the right.

(b) Case 2: pedestrians are initially positioned in circle, static route choice attracts pedestrians to its centre.

(c) Case 3: pedestrian are initially positioned in ellipse, will walk to right and then turn right around a corner.

**Figure 10:** Initial state and static route choice. Gray area indicates original position of pedestrians. Thick blue arrows indicate static route choice walking direction.

triangle of the new mesh $\rho_j^{\text{remeshed}}$ is calculated as follows:

$$\rho_j^{\text{remeshed}} = \frac{N_j^{\text{remeshed}}}{A_j^{\text{remeshed}}} = \frac{\sum_i \frac{A_i^*}{A_i} N_i}{A_j^{\text{remeshed}}} \tag{25}$$

with $A_i$ the area of the $i$-th triangle of the old mesh, $N_i$ the number of pedestrians in that triangle and $A_i^*$ the area of the intersection between the $i$-th triangle of the old mesh and the $j$-th triangle in the new mesh.

# 4 Simulation and results

We perform simulations with the newly developed method to show that crowds can indeed be simulated using Lagrangian techniques and that this approach gives good solutions. We consider three test cases. In the first and second case we show that the numerical solution using the newly developed Lagrangian simulation method approximates the analytical solution well. Furthermore, the second test case shows the influence of the weight of the dynamic route choice component. The third test case shows a more realistic setting where a group of pedestrians rounds a corner.

## 4.1 Test set up

In all three test cases we use remeshing settings and a bi-linear fundamental diagram with parameters as in Table 1 and initial state and static route choices as in Figure 10. The other settings are different for each of the cases, as Table 1 shows, and they are described in more detail below.

### 4.1.1 Case 1 ('straight')

Test case 1 is used to show the numerical accuracy, that is: the ability of the Lagrangian simulation method to approximate the exact solution of the model equations. In this simple test case, initially, there is a high density area, which could resemble a jam in a wide corridor. At the front of the jam, pedestrians start walking and the jam dissolves. We ignore the influence of the dynamic route choice (i.e. we set its weight to zero: $\beta_{\text{dyn}} = 0$), which enables us to calculate the analytical solution using

**Table 1:** Parameters of test cases

**Fundamental diagram parameters**

| | | | |
|---|---|---|---|
| free flow speed | $v_{\text{free}}$ | 1.3 | m/s |
| critical density | $\rho_{\text{crit}}$ | 1.35 | peds/m$^2$ |
| jam density | $\rho_{\text{jam}}$ | 5.4 | peds/m$^2$ |

**Remeshing settings**

| | | |
|---|---|---|
| threshold enlarge cell | $\alpha_\rho$ | 1.0 |
| factor enlarge cell | $\beta_\rho$ | 0.1 |
| threshold move vertex towards centroid | $\alpha_c$ | 0.33 |
| factor move vertex towards centroid | $\beta_c$ | 1.0 |

**Case dependent settings: route choice**

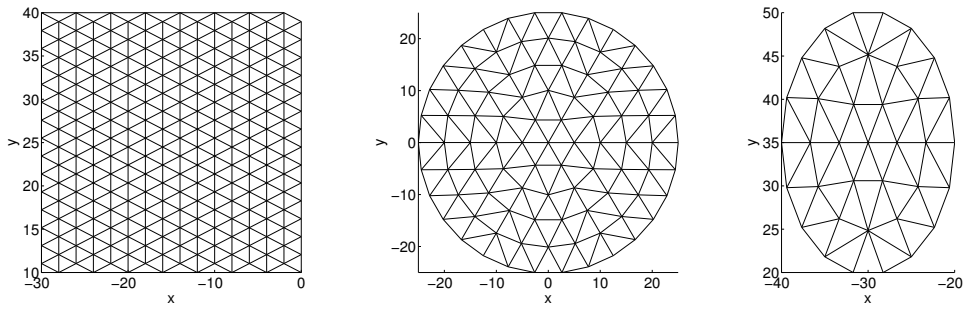
| | | Case 1 | Case 2 | Case 3 | |
|---|---|---|---|---|---|
| static | $\vec{e}_{\text{stat}}$ | $\begin{pmatrix}1\\0\end{pmatrix}$ | $\frac{-1}{\sqrt{x^2+y^2}}\begin{pmatrix}x\\y\end{pmatrix}$ | see Figure 10(c) | |
| dynamic | $\beta_{\text{dyn}}$ | 0 | $[5, 10, 20, 40, 80]$ | 20 | m$^2$/ped |

**Case dependent settings: initial condition**

| | | Case 1 | Case 2 | Case 3 | |
|---|---|---|---|---|---|
| density | $\rho(x, 0)$ | $0.75\rho_{\text{jam}} = 4.05$ | $0.4\rho_{\text{crit}} = 0.54$ | $0.6\rho_{\text{crit}} = 0.81$ | peds/m$^2$ |
| position | | $L_x = 80$ | $R = 25$ | $R_x$=−30 $R_y$=35 $C_x$=10 $C_y$=15 | m |
| nr of peds | $N_{\text{tot}}$ | 3637 | 1058 | 380 | |

**Case dependent settings: mesh and numerical settings**

| | | Case 1 | Case 2 | Case 3 | |
|---|---|---|---|---|---|
| initial cell area | $A$ | 2.1 | $[10.9, 14.3]$ | $[7.3, 9.6]$ | m$^2$ |
| peds per cell | $N$ | 8.4 | $[5.9, 7.7]$ | $[5.9, 7.8]$ | |
| nr of cells | $I$ | 434 | 150 | 54 | |
| nr of vertices | $P$ | 240 | 91 | 37 | |
| time step size | $\Delta t$ | $[0.2, 0.5, 1, 2]$ | 0.1 | 0.5 | |




(a) Case 1 (b) Case 2 (c) Case 3

**Figure 11:** Initial meshes.

kinematic wave theory and compare it with the simulation result. Furthermore, we compare the results for different time step sizes to get more insight into the accuracy.

#### 4.1.2 Case 2 ('circular')

Test case 2 is also used to show the numerical accuracy, now with taking the dynamic route choice term into account. It also shows the influence of the weight of the dynamic route choice on the outcome of the model. In this test case, initially, there is a circular high density area. The static route choice represents an attractor at the centre of this area: that is where the pedestrians would want to go. The static route choice component is normalised such that its length always equals one: $\|\vec{e}_{\text{stat}}\| = 1$. However, the pedestrians will also walk away from the centre because it is too crowded, which is represented by the dynamic route choice term. For different weights of the dynamic route choice term, the stationary state in which the crowd does not move anymore is computed analytically and compared to the simulation results.

#### 4.1.3 Case 3 ('corner')

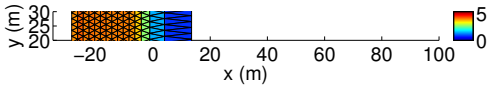
Test case 3 is used to show that the numerical method also works for (slightly) more realistic cases in which a corner is involved. The static route choice is chosen according to equidistant trajectories around the corner. And again, the static route choice component is normalised such that its length always equals one: $\|\vec{e}_{\text{stat}}\| = 1$. The analytical solution is not computed, but we will discuss the plausibility of the results.

### 4.2 Initialization

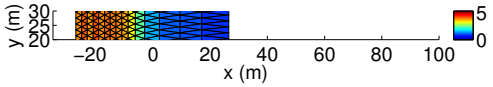
For the initialisation of the test cases, a mesh is created manually, depending on the initial positioning of the pedestrian. The initial meshes are shown in Figure 11. A certain number of pedestrians is assigned to each cell according to the initial density. The values are shown in Table 1.

### 4.3 Test results

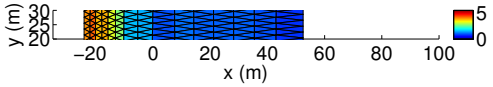
Test results are shown in Figures 12–15. The results for Case 1 (Figures 12 and 13) show that the numerical method is able to reproduce how the jam dissolves and



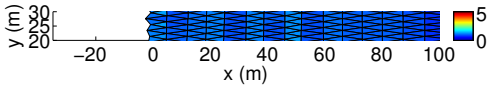

(a) Time $t$ = 10 s




(b) Time $t$ = 20 s

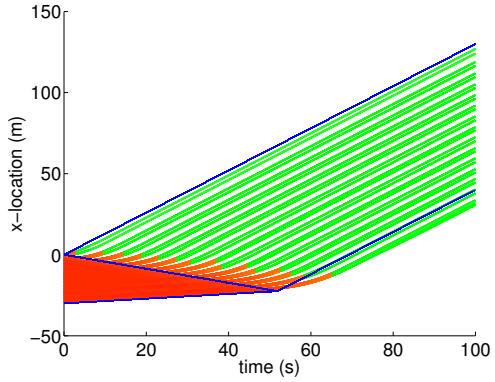



(c) Time $t$ = 40 s

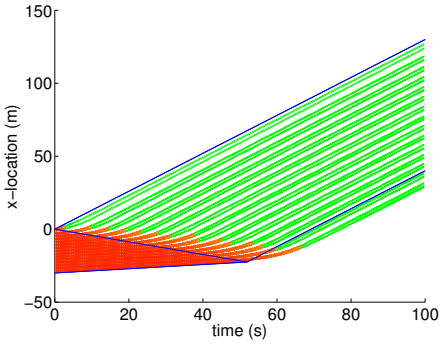



(d) Time $t$ = 80 s

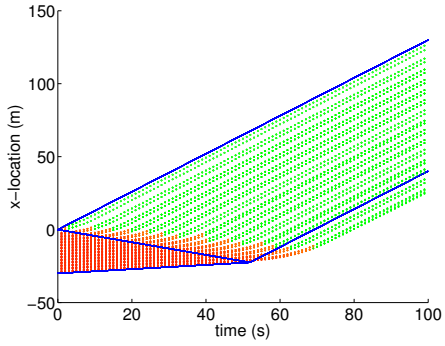
**Figure 12:** Case 1, densities at different times, with time step size $\Delta t$ = 1 s.



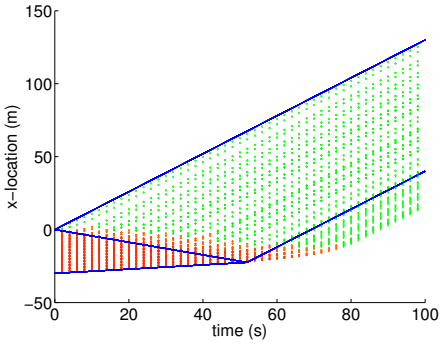

(a) $\Delta t$ = 0.2 s




(b) $\Delta t$ = 0.5 s

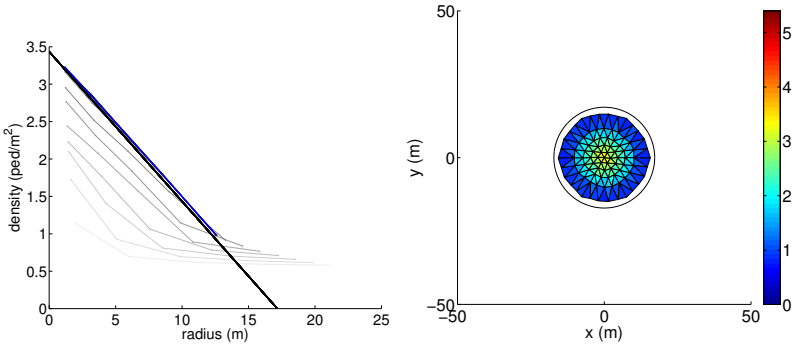



(c) $\Delta t$ = 1 s

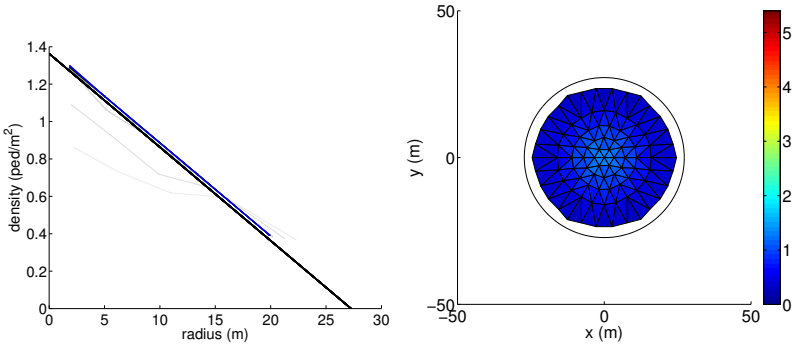



(d) $\Delta t$ = 2 s

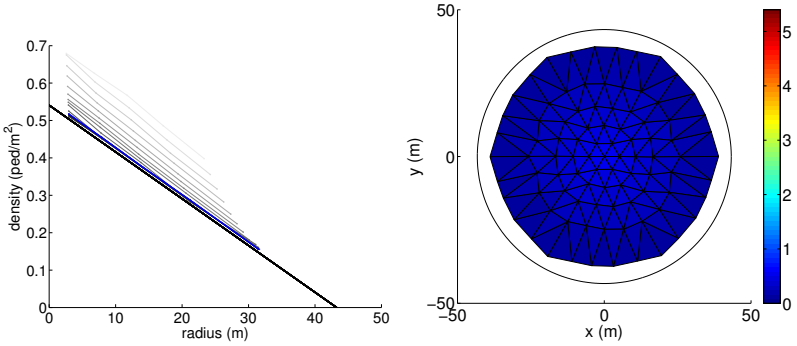
**Figure 13:** Case 1, development of densities at cell centres over time, with different time step sizes $\Delta t$. Colored dots indicate locations of cell centres each time step. Red indicates high density, gradually changing into orange for lower densities, until critical density. Green indicates density below critical. Blue lines indicate analytical solution.




(a) Weight dynamic route choice $\beta_{\text{dyn}} = 5$.

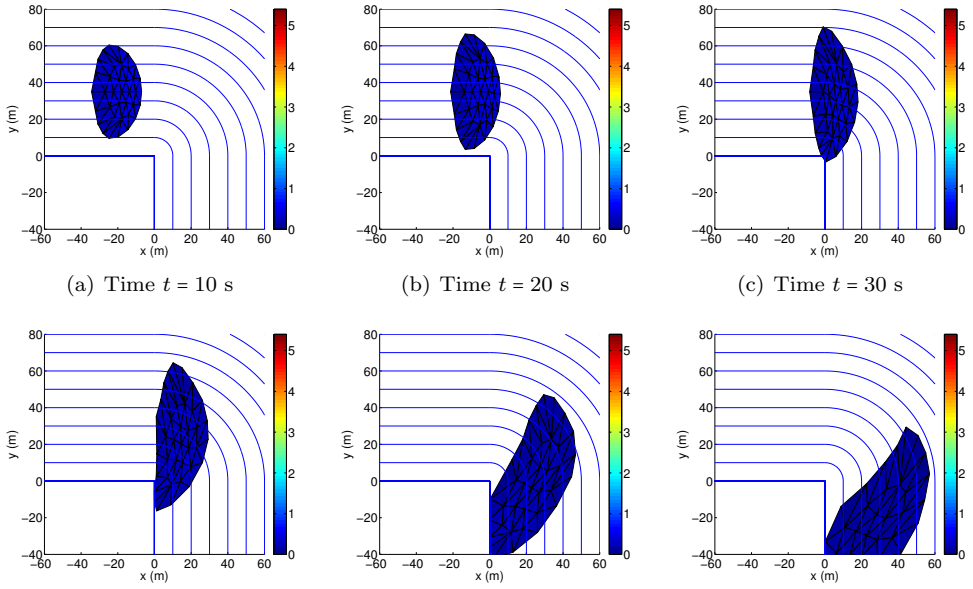



(b) Weight dynamic route choice $\beta_{\text{dyn}} = 20$.




(c) Weight dynamic route choice $\beta_{\text{dyn}} = 80$.

**Figure 14:** Densities for Case 2, for various weights of the dynamic route choice. Left: average density at certain radii. Shown every second: light colors for first time steps, gradually getting darker. Black thick line: density in steady state according to analytical solution. Blue: density after 15 seconds. The plot shows that the density barely changes anymore and this is considered the numerically computed steady state. Right: space-density plots. Only density after 15 seconds (numerical steady state) is shown. Black circle indicates radius of group in steady state according to analytical solution.




(a) Time $t$ = 10 s (b) Time $t$ = 20 s (c) Time $t$ = 30 s

(d) Time $t$ = 40 s (e) Time $t$ = 60 s (f) Time $t$ = 80 s

**Figure 15:** Case 3: Densities at different times.

pedestrians start walking. Figure 13 also includes a comparison with the analytical solution. This comparison shows that there is some delay in when the pedestrians start to walk, resulting in the front of the congestion wave travelling backward slower than expected. Comparison over different time step sizes, shows that this delay decreases for small time step sizes and that more accurate results can be obtained using smaller time steps, which indicates that the method is consistent.

The results for Case 2 (Figure 14) show that the numerical method is well able to approximate the stationary solution. After 15 seconds the solution barely changes anymore and this is considered to be the numerical steady state. The results on the left show that the numerical steady state (blue lines) and analytical steady state (black lines) are almost identical. Furthermore, the test results show that the solution depends heavily on the weight of the dynamic route choice. This can be seen best on the right hand side, where the radius of the circle depends on the weight. If it has a low value ($\beta_{\text{dyn}}$ = 5, Figure 14(a)), the pedestrians stay closely together (in this example in an area with radius of about 17 meter), while a high value ($\beta_{\text{dyn}}$ = 80, Figure 14(c)) makes the pedestrians spread out over a much larger area (in this example with radius of about 43 meter). Therefore, in most applications, it will be important to calibrate this parameter well.

The results for Case 3 (Figure 15) show that the method is also capable of simulations with slightly more complex layout. The pedestrians first walk to the right, while spreading out a bit more than in the initial solution. They round the corner in a way similar to observations with low densities. The results show that the numerical method allows for changes of the shape of the pedestrian group. Furthermore, the numerical scheme reproduces what we expect from the behavioural hypothesis in the model.

# 5 Summary and future work

A numerical simulation method for crowd flow based on the Lagrangian coordinate system is introduced. The method uses a triangular mesh that moves according to rules dictated by the traffic flow model. In order to prevent too high densities and too much deformation of the mesh, a remeshing procedure is introduced. Test results show that this leads to plausible simulation results and that high accuracy can be obtained.

The Lagrangian simulation method can be applied to solve continuum crowd flow models. It can be applied to a simple model as presented here, but also to more advanced models. Further research is necessary to quantify the efficiency in terms of accuracy and computational speed, and to compare the efficiency with that of alternative methods such as those based on Eulerian grids fixed in space[Hänseler et al., 2014]. Certain aspects can be studied further, such as the approach to free boundaries and the remeshing procedure. This can include a sensitivity analysis of the remeshing parameters. Furthermore, future research will focus on alternative settings of the numerical method, including meshes consisting of generic polygons and alternative remeshing procedures. Future extensions of the method include the introduction of closed boundary conditions (including walls and obstacles), inflow and outflow boundary conditions and multiple groups that have different walking or route choice characteristics or destinations.

# Acknowledgements

This research is performed as part of the The Netherlands Organisation for Scientific Research (NWO) Aspasia grant of Daamen (Daamen, Van Wageningen-Kessels, grant number 016.125.440) and as part of the NWO-VICI project 'Travel Behavior And Traffic Operations In Case Of Exceptional Events' (Hoogendoorn, grant number 453-08-006).

# References

A. Aw, A. Klar, M. Rascle, and T. Materne. Derivation of continuum traffic flow models from microscopic follow-the-leader models. *SIAM Journal on Applied Mathematics*, 63(1):259–278, 2002.

N. Bellomo, B. Piccoli, and A. Tosin. Modeling crowd dynamics from a complex system viewpoint. *Mathematical Models and Methods in Applied Sciences*, 22(2), 2012.

J. D'Errico. Polyfitn matlab function, 2012. URL `http://www.mathworks.com/matlabcentral/fileexchange/34765-polyfitn`. Updated in April 2014.

C. Dogbé. On the numerical solutions of second order macroscopic models of pedestrian flows. *Computers & Mathematics with Applications*, 56(7):1884–1898, 2008.

D. C. Duives, W. Daamen, and S. P. Hoogendoorn. State-of-the-art crowd motion simulation models. *Transportation Research Part C: Emerging Technologies*, 37: 193–209, 2013.

F. S. Hänseler, M. Bierlaire, B. Farooq, and T. Mühlematter. A macroscopic loading model for time-varying pedestrian flows in public walking areas. *Transportation Research Part B: Methodological*, 69:60–80, 2014.

S. P. Hoogendoorn, F. L. M. van Wageningen-Kessels, W. Daamen, and D. C. Duives. Continuum modelling of pedestrian flows: From microscopic principles to self-organised macroscopic phenomena. *Physica A: Statistical Mechanics and its Applications*, 416:684–694, 2014.

S. P. Hoogendoorn, F. L. M. van Wageningen-Kessels, W. Daamen, and M. Sarvi. Continuum theory for pedestrian traffic flow: Local route choice modelling and its applications. 2015. Accepted for ISTTT21 and subsequent publication in Transportation Research Series.

L. Huang, S.C. Wong, M. Zhang, C.-W. Shu, and W. H. K. Lam. Revisiting Hughes' dynamic continuum model for pedestrian flow and the development of an efficient solution algorithm. *Transportation Research Part B: Methodological*, 43(1):127–141, 2009.

R. L. Hughes. A continuum theory for the flow of pedestrians. *Transportation Research Part B: Methodological*, 36(6):507–535, 2002.

Y. Jiang, S.C. Wong, P. Zhang, R. Liu, Y. Duan, and K. Choi. Numerical simulation of a continuum model for bi-directional pedestrian flow. *Applied Mathematics and Computation*, 218(10):6135–6143, 2012.

L. Leclercq, J.A. Laval, and E. Chevallier. The Lagrangian coordinates and what it means for first order traffic flow models. In R. E. Allsop, M. G. H. Bell, and B. G. Heydecker, editors, *Transportation and Traffic Theory 2007*, pages 735–753, Oxford, 2007. Elsevier.

G. F. Newell. A simplified theory of kinematic waves in highway traffic (part I-III). *Transportation Research Part B: Methodological*, 27(4):281–313, 1993.

F. L. M. van Wageningen-Kessels, J. W. C. van Lint, S. P. Hoogendoorn, and C. Vuik. Lagrangian formulation of a multi-class kinematic wave model. *Transportation Research Record: Journal of the Transportation Research Board*, 2188:29–36, 2010.

Y. Yuan, J. W. C. van Lint, R. E. Wilson, F. L. M. van Wageningen-Kessels, and S. P. Hoogendoorn. Real-time Lagrangian traffic state estimator for freeways. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):59–70, 2012.