Pedestrian Circulation

Introduction

It's important to know how pedestrians move, or "circulate", the built environment, perhaps inside complex buildings such as hospitals or schools, or outside in large shopping malls. Even in confined spaces such as planes and submarines. An understanding of pedestrian circulation is important for many reasons from the evacuation of buildings in an emergency, to designing buildings with the optimal use of floor space to let the building work. The model we shall use is Helbing's "Social Force Model" where pedestrians change their direction and speed of travel according to how close they are to other pedestrians and also obstacles. Figures 1 and 2 show a couple of scenarios.

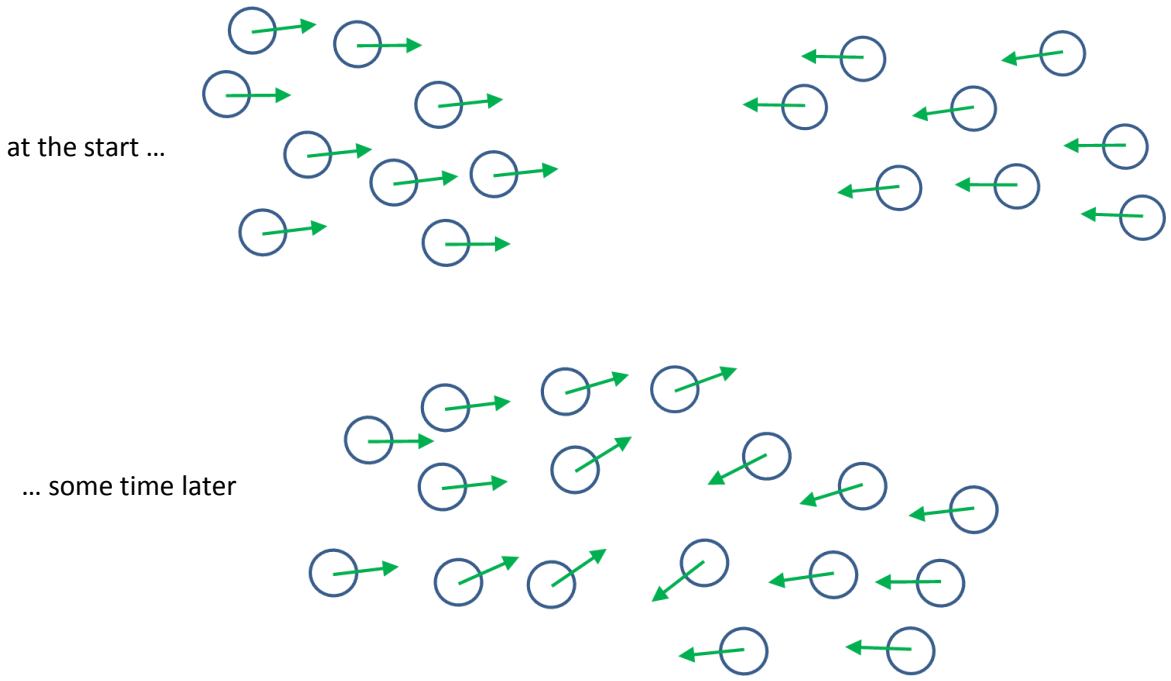


Figure 1. Top shows two crowds of pedestrians approaching each other, the green arrows show their individual velocities. Bottom shows a possible arrangement when they collide, the left pedestrians device to move to their left, and the right group also turn to their left.

Conventions used in this document

1. Vector arrows. All velocity vectors are shown in green, all force vectors are shown in red, and all position arrows are shown in **black**

2. Vectors and scalars. All vectors are shown in bold face, like this velocity vector $\boldsymbol{v}$. All scalars (numbers) are not bold, like this distance d.

3. Subscripts and superscripts. In a multi-agent system we need to label which agent we are talking about. This agent will be labelled i, e.g., the velocity of this agent is written $\boldsymbol{v}_i$. We also need to talk about which entity is *acting on* our principal agent, e.g., when a wall exerts a force on an agent then we write this as $\boldsymbol{f}_{iW}$.

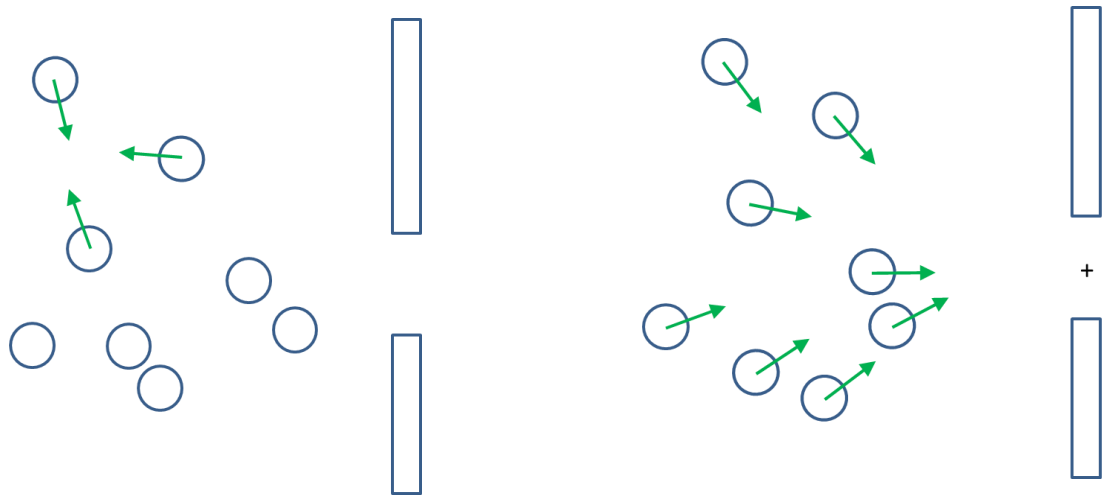
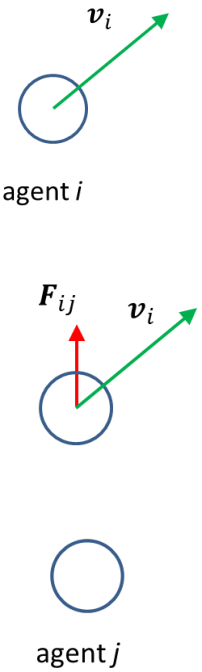


Figure 2. Evacuation. Left shows people at a meeting in a room, three are coming together; there are two stationary pairs and a singleton. When the evacuation alarm sounds, they all move towards the centre of the opening in the wall (small cross). Note how their velocity vectors point towards this cross.

Overview of the Dynamics

To determine how the green velocity vectors change as pedestrians interact, we use the concept of *force* which ties up with our approach to M&S seen so far. First the concepts, without the maths. Figure 3 (top) shows agent *i* moving with a certain velocity (speed with direction). Then agent *j* appears (middle) and applies a repulsive force to agent *i*. This is the vertical red arrow. The effect of this force is to add some extra velocity onto agent *i*'s velocity *in the same direction as the force.* This is shown in the last diagram (bottom) where the new velocity of agent *i* is shown.



To calculate the dynamics of the agents, we first get the acceleration from the force

$$a_x = F_x/m$$

then we use this to update the velocity

$$v_x = v_x + a_x \Delta t$$

and finally we use the velocity to update the position of the agent

$$x = x + v_x \Delta t$$

The code to do this is shown in the box below. A similar set of expressions and code statements must be done for the *y* direction

```
aX = fX/mass;
vX += aX*dT;
x += vX*dT;
```

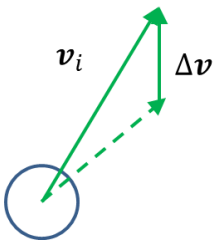


Figure 3. shows how the interaction force changes the agent's velocity

Three Types of Force

Pedestrians (agents) experience three basic types of force: (i) a force moving them to their desired location (attractive), (ii) repulsive forces on them from other pedestrians, (iii) repulsive forces on them from obstacles such as walls.

Force to a Desired Location

Let's start by looking at how the force works, the "effect" of the force. This is shown in Figure 4 which shows the behaviour of a single agent with time. Agent *i* is moving with a certain velocity (green arrow) then decides to move towards the target (yellow blob). So the target exerts a force (red arrow). The effect of the force is to rotate the agent's velocity vector, and also to change its length; the agent speeds up, for a while (details later). Note also the direction of the force; it does not point to the target! It points a little to one side. This is because it has to remove the upward motion of the agent. This will become clearer in a while.

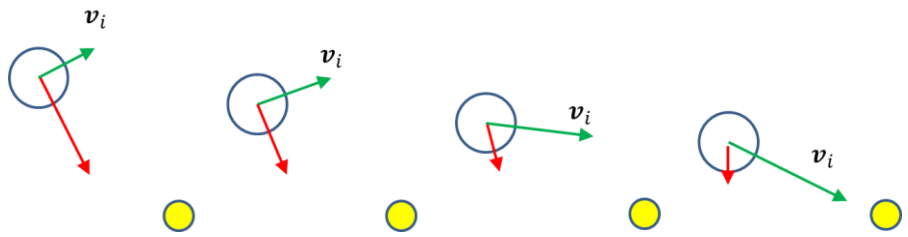


Figure 4. Agent approaching a target. The target exerts a force rotating the agent's velocity vector.

So we have to change the direction and length of the velocity vector $\boldsymbol{v}_i$. First the direction. Here we work with the vector of unit length (=1) which points to the target, in the desired direction. This is $\boldsymbol{e}_i^{des}$. Then we have the desired speed of movement, typically around 0.6 m/s (Helbing, 2000). Let's call this v_i^{des}. Note that this is a *scalar*. To work out an expression for the force, we start with this

$$v_i^{des}\boldsymbol{e}_i^{des} - \boldsymbol{v}_i$$

On the left is the desired velocity and on the right is the actual velocity. This is positive when the left term is larger than the right term, in other words the agent is moving too slow. This causes the agent to speed up, so its velocity $\boldsymbol{v}_i$ increases. This increase continues until the right term matches the left, so the force is zero, and the agent is moving towards the target with the desired speed. A snapshot of this is shown in Figure 5.

Figure 5. Pedestrian attracted to target. Left shows the pedestrian's actual velocity, it's moving up and to the right, and also its desired velocity pointing to the target. Right shows the difference between desired and actual which generates the force vector shown in red.

We have to finish the expression for the force to target. The Let's look at this, then see what it means.

$$\boldsymbol{f}_{target,i} = m_i \frac{v_i^{des}\boldsymbol{e}_i^{des} - \boldsymbol{v}_i}{\tau_i}$$

So we have divide the velocity difference $v_i^{des}\boldsymbol{e}_i^{des} - \boldsymbol{v}_i$ by a time constant τ_i (pronounced "tauw") which determines how quickly the agent's velocity reacts, this is 0.5 sec (Helbing, 2000). Dividing velocity by time gives us an acceleration. Then we multiply by the agent's mass which converts acceleration into force by virtue of Newton's Second Law. All done! The code for this force is shown in the box below.

```
// Get the vector from this agent to the target using components
nX = target.x - x;
nY = target.y - y;
// Get the length of this vector
dist = (float)Math.sqrt(nX*nX + nY*nY);
// Divide the vector by its length to get a unit vector
nX = nX/dist;
nY = nY/dist;
// Compute the force vector, vDes is the scalar desired velocity
fX += (vDes*nX - vX)*mass/tau;
fY += (vDes*nY - vY)*mass/tau;
```

Force from Obstacles – Walls.

This is a repulsive force which tends to keep pedestrians clear of walls. The situation is shown in Figure 6 which shows a pedestrian approaching a wall obliquely, then how its velocity vector will change. The force on the pedestrian will always be in the same direction, perpendicular to the wall, but the magnitude (size) to the force will change.

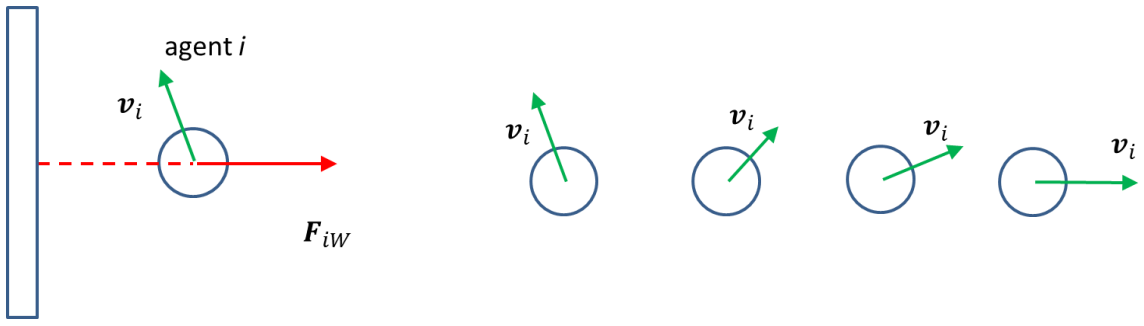


Figure 6. Wall avoidance. On the left an agent approaches a wall and experiences a force perpendicular to the wall, in this case to the right. On the right a series of snapshots showing the agent's velocity changing, becoming aligned perpendicular to the wall.

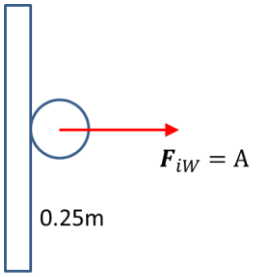
Let's now have a look at how the force on the pedestrian from the wall is computed. The expression for the force has two terms on the right, the magnitude of the force F_{mag} and the direction of the force $\boldsymbol{n}_{iW}$ which is a vector of length 1 perpendicular to the wall; this is in the direction of the red arrow in Figure 6. So we have

$$\boldsymbol{f}_{iW} = F_{mag}\boldsymbol{n}_{iW}$$

All we need to do now is to understand how the magnitude of the force depends on how close the pedestrian is to the wall; the closer to the wall, then the larger the force. If the radius of the pedestrian is r_i and the distance of this pedestrian to the wall is d_{iW} then the magnitude of the force is given by the expression

$$F_{mag} = A_i e^{[(r_i - d_{iW})/B_i]}$$

With values of A = 2000 N and B = 0.08m then plotting F_{mag} as a function of d_{iW} produces the graph shown in Figure 7. When the pedestrian is touching the wall, d_{iW} =0.25m, the pedestrian radius, then the force is greatest at the value of parameter A. As the pedestrian moves further away from the wall, the force drops sharply, so this is a *short-range* force. It is very small indeed at a distance of 0.5m, and at 1m away from the wall, there is practically no force at all. This makes sense; imagine yourself walking down a corridor in Charles Hastings, how close do you get to the wall before you feel uncomfortable? The code for the force calculation is shown in the box below for the case of a vertical wall, where the force is in the x-direction.



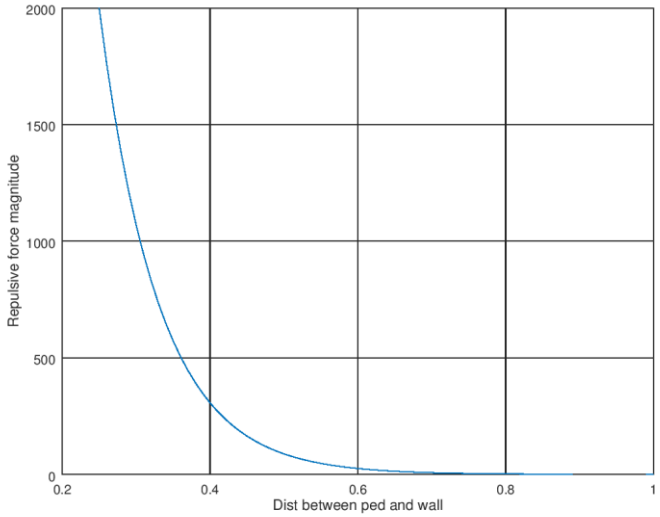


Figure 7. Force on Pedestrian from a Wall. Right shows the variation of force with distance between the centre of the pedestrian and the wall surface. Left shows max force when pedestrian is touching the wall

```
// compute the vector from the wall to the pedestrian
nX = x - wall.xCentre;
// compute the length of this vector
dist = (float)Math.sqrt(nX*nX);
// Normalize vector by dividing by its length
nX = nX/dist;
// Now compute the force magnitude
fMag = A*Math.exp((radius - dist)/B);
```

Forces between Pedestrians

When you are walking around, you tend not to get too close to other people, and certainly not collide with them, both would be impolite. Adjusting your velocity to stay at a reasonable distance from other people is modelled with a pedestrian-pedestrian repulsive force. How this force works is very culture-dependent; in my experience Russian pedestrians get much closer than English pedestrians, I wonder what happens in China or Japan or the US? The force is also dependent on the situation, compare moving along a high street with moving through an underground datarion.

The inter-pedestrian force is very similar to the pedestrian-wall force calculation. The situation is shown in Figure 8 where we are looking at the force **on** agent *i* **from** agent *j.* The direction of the force is on a vector connecting the agents, shown as the red dashed-line.

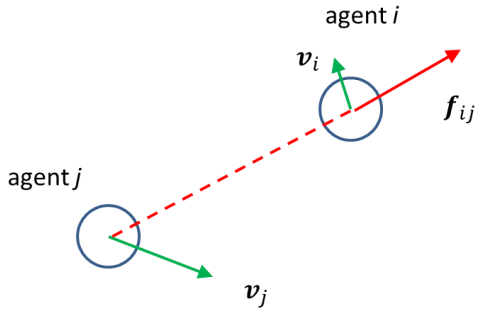


Figure 8. We focus on agent *i* and look at the repulsive force $\boldsymbol{f}_{ij}$ exerted on agent *i* by a second agent *j*.

The expression for the inter-agent force is just like the agent-wall force, but instead of the wall, we have the second agent *j* who is applying the force on us, agent *i.* So it looks like this

$$\boldsymbol{f}_{ij} = F_{mag}\boldsymbol{n}_{ij}$$

The vector $\boldsymbol{n}_{ij}$is the unit vector pointing from agent *j* to agent *i* andF_{mag} is the magnitude (size) of the force. How do we calculate all of this? Well, again we follow Helbing (2000). First we have to calculate $\boldsymbol{n}_{ij}$. Look at Figure 9; the locations of the two agents are shown relative to the space origin (0,0). We find the vector difference between the agents $\boldsymbol{r}_i - \boldsymbol{r}_j$and then divide this by the size of this difference d_{ij} which gives us a unit vector (length 1.0) in the correct direction pointing from agent-*j* to agent-*i*. This vector is

$$\boldsymbol{n}_{ij} = (\boldsymbol{r}_i - \boldsymbol{r}_j)/d_{ij}$$

The expression for the force magnitude is similar to the wall force

$$F_{mag} = A_i e^{[(r_{ij}-d_{ij})/B_i]}$$

where r_{ij} is the sum of the radii of the two agents and d_{ij} is the distance between them. This is shown in Figure 8. The code to generate this force is shown in the box below.

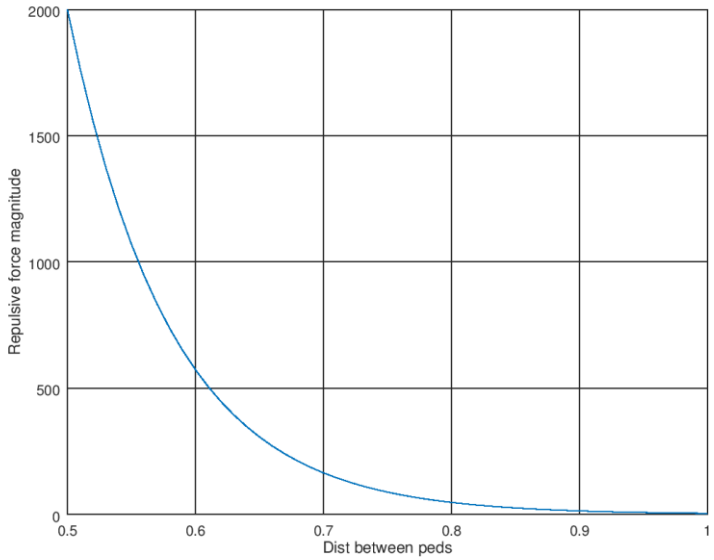


Figure 9. Force between two pedestrian which is maximum when they are in contact.

```
// Get vector between agents
nX = this.x - other.x;
nY = this.y - other.y;
// Compute the length of this vector
dist = (float)Math.sqrt(nX*nX + nY*nY);
// Normalize vector
nX = nX/dist;
nY = nY/dist;
//Sum the agents radii
float sumR = this.radius + other.radius;
// Calc force components
fMag = A*Math.exp((sumR - dist)/B);
fX += (float)fMag*nX;
fY += (float)fMag*nY;
```