Rapport Physique Moderne

CAZENAVE Dorian,GENDE-PAMBOU Kevin, HILL Zak, SERRANO Lucas

Mai 2025

Sommaire

1 Introduction

Lors d'une évacuation, les personnes se trouvant dans le bâtiment doivent sortir vers l'extérieur. La vitesse de sortie est primordiale pour permettre au plus grand nombre de survivre. Une étude [1] montre que s'il y a un obstacle devant la sortie, l'évacuation se fait plus rapidement que s'il n'y avait pas d'obstacles. Ce résultat est assez surprenant, car on pourrait penser le contraire. Notre étude se concentrera sur l'observation de ce phénomène.

2 Objectif

Notre objectif est d'étudier une foule en mouvement selon différents cas en modélisant ceux-ci. Nous allons d'abord modéliser le cas où on fait évacuer N-personnes par un seul accès de sortie en s'apppuyant sur le modèle de Helbing pour pouvoir appliquer les forces sociales puis on effectuera d'autres modélisations en rajoutant un ou plusieurs obstacles devant la sortie. On tracera les graphes représentant la vitesse de trois personne (la personne bleu, rose et vert).

3 Méthodes

Pour réaliser ce projet, nous utiliserons uniquement le langage Python pour modéliser l'expérience et pour obtenir les résultats de l'étude. Nous utiliserons les bibliothèques Matplotlib [3] (pour tracer des graphiques de courbes) et Pygame [2] (pour réaliser des modélisations dynamiques). Pour réaliser cette modélisation, on va établir l'équation différentielle de la vitesse d'une personne suivant le modèle d'Helbing:

$$\begin{cases} \vec{F_1} = m_i \frac{v_i^0(t)\vec{e_i^0}(t) - \vec{v_i}(t)}{t_i} \text{ avec } \vec{e_i^0}(t) \text{ correspondant à la direction désirée du pion et } \vec{v_i}(t) \\ \text{le vecteur de la vitesse réelle du pion} \\ \vec{F_2} = e^{\frac{-d_{ij}}{B_i}} \vec{n_{ij}} \text{ avec } d_{ij} \text{la distance entre le piéton i et j et } \vec{n_{ij}} \text{ le vecteur unitaire allant de j vers i.} \\ \vec{F_3} = e^{\frac{-d_0^i}{B_i}} \vec{n_0^i} \end{cases}$$

En appliquant le PFD, on obtient alors

$$m_i \frac{d\vec{v_i}(t)}{dt} = \vec{F_1} + \sum_{j \neq i} \vec{F_2} + \sum_{\Omega} \vec{F_3} \text{ avec } \Omega \text{ l'ensemble des obstacles}$$

$$\frac{d\vec{v_i}(t)}{dt} = \frac{1}{m_i} . (\vec{F_1} + \sum_{j \neq i} \vec{F_2} + \sum_{\Omega} \vec{F_3}) \quad (1)$$

Ainsi, on arrive à une équation différentielle d'ordre 1 donc on peut résoudre cette équation différentielle. Nous allons utiliser la méthode de résolution de

Runge-Kutta d'ordre 4 pour tracer les différents graphes de la vitesse des piétons car on a testé les différentes méthodes de résolution (Euler, RK2 et RK4) et on a remarqué la méthode RK4 était bien plus précis que les autres (voir le graphique tracé du programme eq-diff.py).

Pour choisir la position des piétons dans la pièce, nous avons pensé naïvement les disposer de manière aléatoire. Or, cette première approche peut provoquer des chevauchements de piétons à l'instant t=0, ce qui rendrait le modèle incohérent. Pour résoudre ce problème, nous avons décidé de les faire apparaître à une distance minimale (de chaque point).

4 Fonctionnalité de la simulation

Nom fichier	**Fonctionnalité**
fonction.py	Contient les fonctions essentielles à la simulation
simu.py	Le fichier pour lancer les différentes simulations
solve.py	Contient les méthodes Euler, Runge-Kutta 2 et 4 de résolution d'équations différentielles
variable.py	Contient toutes les variables de la simulation
eqdiff.py	Contient la résolution d'une équation différentielle

Table 1: Description des fichiers de la simulation.

La simulation possède des actions permettant d'interagir avec elle en utilisant le clavier (voir tableau ci-dessous)

Touche du clavier	**Action**
espace	Met en pause ou redémarre la simulation
r	Relance la simulation
g	Affiche un graphique de vitesses des points caractéristiques

Table 2: Les différentes touches d'intéraction de la simulation

4.1 Simulation 1 - Cas sans obstacle devant la sortie

La simulation 1 se compose d'une pièce rectangulaire sans obstacles (hormis les autres personnes), avec une seule porte de sortie. Ici nous l'avons rempli de 100 personnes.

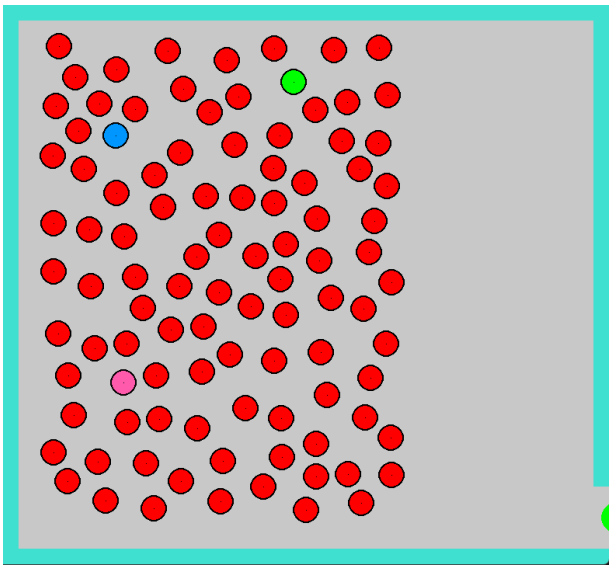


Figure 1: Simulation 1

4.2 Simulation 2 - Cas avec 1 obstacle devant la sortie

La simulation 2 se compose d'une pièce rectangulaire comportant un obstacle, une table devant l'unique porte de sortie.

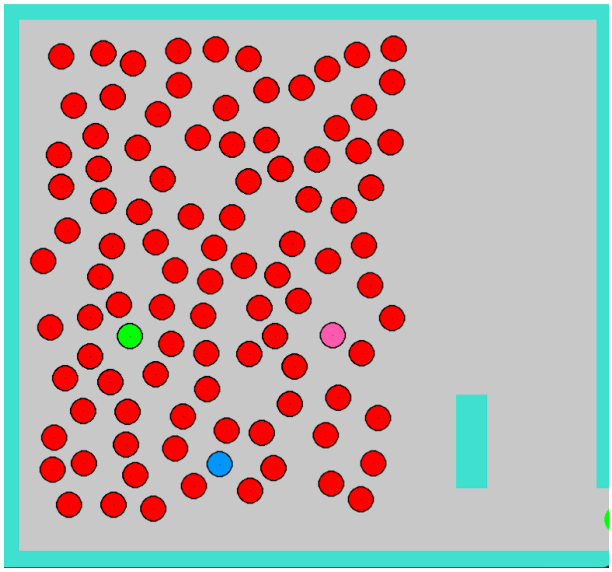


Figure 2: Simulation 2

4.3 Simulation 3 - Evacuation d'une salle de classe

La simulation 3 est une salle de classe de type collège ou lycée. Elle est de forme rectangulaire et comporte 3 rangées de tables (avec le bureau du professeur devant). Ici, les obstacles étant nombreux, la vitesse moyenne des personnes devrait logiquement diminuée par rapport à la simulation 1 ou 2.

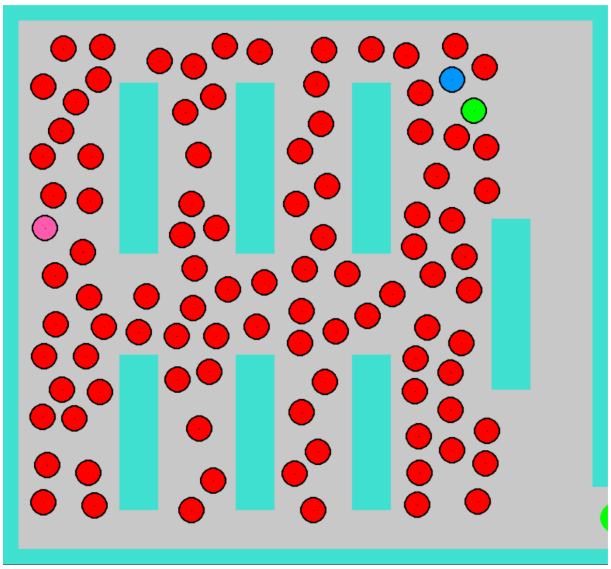


Figure 3: Simulation 3 - Salle de classe

4.4 Protocole de résultats

Nous lancerons chaque simulation 10 fois pour lisser les données. Nous noterons à chaque fois le temps pour que les 100 individus sortent de la pièce. Nous observerons également le graphique de vitesse de 3 personnes caractéristiques (de couleur verte, bleu et rose).

5 Résultats

Voici les résultats obtenus lors de l'execution de la simulation:

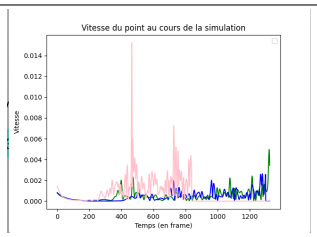
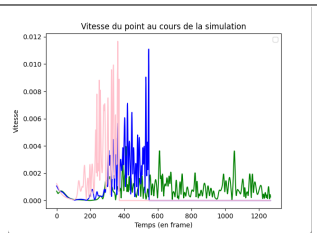
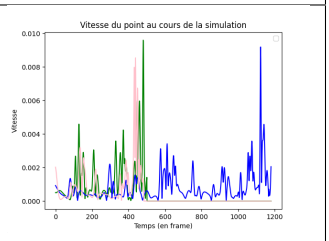
	Simulation 1	Simulation 2	Simulation 3
Temps pour sortir (en seconde)	52.5	58.5	123
Graphique de vitesse			

Table 3: Résultats des simulations 1, 2 et 3

6 Analyse

On remarque que la simulation la plus rapide est la 1, ne comportant aucun obstacle, ce qui semble logique. Ensuite, la simulation 2, qui comporte 1 seul obstacle. Enfin, la simulation 3, comportant beaucoup plus d'obstacles que les deux autres.

On observe alors que plus il y a d'obstacle, plus la vitesse de sortie est lente. Cependant, d'après l'étude [1], l'obstacle aurait dû accélerer la sortie des piétons

Plusieurs hypothèses peuvent être avancées pour expliquer cette contradiction comme son positionnement ou sa taille trop imposante.

On remarque également que les dernières personnes à évacuer peuvent être lentes, souvent bloquées derrière un obstacle ou un mur. Ce problème est inhérent à notre modélisation des forces sociales. En effet, dans la réalité, un humain utiliserait des notions basiques de path-finding [4] (voir illustration ci-dessous) pour se repérer. Dit autrement, il contournerait les murs pour aller directement à la sortie.

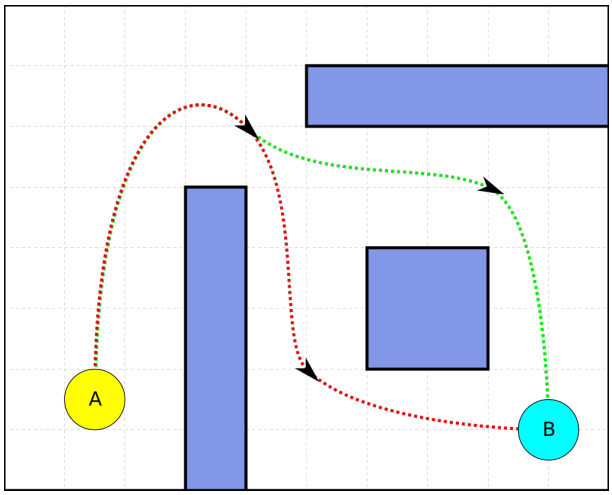


Figure 4: Ilustration du path-finding

7 Conclusion

Ainsi, on observe que les obstacles ralentissent l'évacuation.
On peut alors conjecturer que plus une pièce possède une densité d'obstacle élèvée, plus la vitesse d'évacuation sera lente. On peut également noter que la non-implémentation du path-finding dans le projet simplifie trop la réalitée, ce qui provoque des temps d'évacuations plus lents.

8 Bibliographie

References

[1] Examining effect of architectural adjustment on pedestrian crowd flow at bottleneck *https://arxiv.org/pdf/1808.07439*

[2] Documentation Pygame https://www.pygame.org/docs/

[3] Documentation Matplotlib.pyplot `https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html`

[4] Le concept de path-finding (ou recherche de chemin en français) `https://fr.wikipedia.org/wiki/Recherche_de_chemin`