



M1 - CSM

Projet Tutoré :
Modélisation de mouvement de foule

Jean POUSSET

24 avril 2025

Table des matières

1	Équation modèle et schéma de résolution	2
1.1	Équation modèle	2
1.2	Schéma de discrétisation du pas de temps : JKO	3
2	Méthode ADMM pour la résolution du problème d'optimisation	5
2.1	Reformulation du problème JKO sous un problème sous contrainte avec Lagrangien	5
2.2	Formulation sous la forme d'un Lagrangien augmenté	7
2.3	Formulation de la méthode ADMM pour notre problème	8
3	Traduction des minimisations des étapes 1 et 2	10
3.1	Étape 1 : un problème de poisson 3D	10
3.2	Étapes 2 : minimisations en tous point $(t, x) \in \Theta$	12
4	Détails d'implémentation	14
4.1	Éléments finis pour l'étape 1	14
4.2	Minimisation en tout points (t, x) pour l'étape 2	16
5	Résultats	17
5.1	Visualisation d'un déplacement de foule simple	17
5.2	Convergence et temps d'exécution	19
6	Bilan et conclusion	20

Introduction

◇ Objectifs du projet tutoré

Ce projet a pour but l'analyse d'articles scientifiques pour ensuite implémenter les méthodes discutées pour un problème de modélisation. J'ai choisi de modéliser le mouvement d'une foule.

◇ Ressource scientifique et choix du modèle

J'ai d'abord exploré des modèles microscopiques qui sont simples à comprendre et à mettre en place. Malheureusement, la complexité de résolution augmente exponentiellement pour ces modèles basés sur les interactions.

Je me suis donc penché sur les modèles macroscopiques. J'ai été limité par le manque d'articles qui soient à la fois en rapport avec les cours de la formation et en même temps détaillent assez les méthodes pour que je puisse les utiliser.

Ce projet est basé sur l'article : *An Augmented Lagrangian Approach to Wasserstein Gradient Flows and Applications* écrit par Jean-David BENAMOU, Guillaume CARLIER et Maxime LABORDE.

Une grande partie de mon travail a été de comprendre et trouver les résultats intermédiaires de cet article. Il y a beaucoup d'explications mathématiques dans ce rapport parce que c'est ce que j'ai produit pour la compréhension de ce rapport. Lorsqu'un élément est donné par l'article je l'indique.

La fin du rapport concerne l'implémentation sur FreeFEM et la visualisation des résultats (graphique et convergence) sur un cas simple.

1 Équation modèle et schéma de résolution

1.1 Équation modèle

◇ Foule \simeq densité sous contrainte

Au niveau macroscopique une foule se modélise parfaitement par une densité $\rho \in \mathcal{P}_2$ l'ensemble des distributions de probabilité de moment 2 fini. Cette densité est naturellement définie pour un temps $t \in [0, T]$ sur un domaine fini de dimension 2 : $\Omega \subset \mathbb{R}^2$.

Pour respecter la réalité physique d'une foule cette densité doit respecter 3 propriétés :

- **Conservation de la masse** : $\int_{\Omega} \rho(t, \cdot) = \int_{\Omega} \rho(0, \cdot), \forall t$
- **Positivité**, il n'y a pas de nombre de personne négative : $\rho(t, x) \geq 0, \forall t, \forall x$
- **Congestion**, il ne peut pas y avoir plus d'un certain nombre de personnes à un endroit : $\rho(t, x) \leq \rho_{\max} := 1, \forall t, \forall x$

◇ EDP du modèle

Cette dernière propriété rajoute une contrainte à notre équation modèle qui rend les méthodes de résolution beaucoup plus complexes que les résolutions des mêmes équations sans cette contrainte.

L'équation proposée dans l'article est de type **advection-diffusion** avec des termes supplémentaires :

$$\partial_t \rho - \Delta \rho = \operatorname{div}(\rho(\nabla V + \nabla p)) \quad (\text{AD})$$

Où :

- $\rho(t, x)$: densité de foule au cours du temps ($\rho \leq \rho_{\max} := 1$)
- $p(t, x)$: pression ($p \geq 0, p(1 - \rho) = 0$)
- $V(x)$: potentiel (positions souhaitées par la foule)

La pression p n'est jamais utilisée/référencée dans l'article ailleurs que pour cette équation. Je ne comprenais pas trop son sens ni comment on devait la traiter. Après avoir lu d'autres articles avec d'autres modélisations j'ai compris que cette pression était introduite dans le but de satisfaire la propriété de congestion.

1.2 Schéma de discrétisation du pas de temps : JKO

◇ Flot de gradient de Wasserstein

L'article énonce le théorème suivant :

L'EDP du modèle (AD) possède une structure de flot de gradient pour l'énergie :

$$\mathcal{E}(\rho) := \int_{\Omega} E(\rho) + \int_{\Omega} \rho V$$

Avec :

$$E(\rho) := \rho \log(\rho) + \chi_{[0,1]}(\rho) = \begin{cases} \rho \log(\rho) & , \rho \in [0, 1] \\ +\infty & , \text{sinon} \end{cases}$$

◇ Schéma de Jordan-Kinderlehrer-Otto (JKO)

Grâce à cette structure de flot de gradient nous pouvons utiliser le schéma de Jordan-Kinderlehrer-Otto pour résoudre notre EDP (AD). Ce schéma consiste à construire, à partir d'une densité initiale ρ^0 , une séquence de densités à temps fixé $\{\rho^k\}_k$ telle que :

$$\rho^{k+1} \in \arg \min_{\rho \in \mathcal{P}_2} \frac{1}{2\tau} W_2^2(\rho, \rho^k) + \mathcal{E}(\rho)$$

L'article ne précise pas vraiment les détails quant au sens de cette minimisation. Je l'interprète comme : « chercher une densité pour le prochain pas de temps : ρ^{k+1} telle que :

- ρ^{k+1} ne soit pas trop différente de ρ^k (**continuité**). C'est contraint grâce à la distance de Wasserstein : $W_2^2(\rho, \rho^k)$ qui mesure la non-similitude des deux densités. Le terme $1/2\tau$ est courant en optimisation pour permettre d'équilibrer la minimisation de deux termes.
- ρ^{k+1} soit plus proche de la position souhaitée par la foule. C'est modélisé par le terme $\int_{\Omega} \rho V$. En effet ce terme est minimal lorsque la foule est concentrée sur les zones où le potentiel V est le plus faible, i.e. où la foule veut aller.
- ρ^{k+1} respecte les contraintes de congestion et de positivité. C'est modélisé par l'indicatrice convexe : $\chi_{[0,1]}(\rho)$ qui élimine toute solution qui ne respecte pas cette contrainte en fixant à $+\infty$ le terme à minimiser dans ce cas là.

Il y a également le terme d'entropie : $\int_{\Omega} \rho \log(\rho)$. Je n'ai pas trop d'idée pour celui-ci à part qu'il pourrait forcer à limiter l'effet de diffusion dans l'étape de JKO. En effet, la diffusion de la densité va augmenter le désordre et donc l'entropie.

2 Méthode ADMM pour la résolution du problème d'optimisation

L'article propose une formulation du problème de minimisation sous la forme d'un lagrangien augmenté afin d'utiliser la méthode AMDM (*Alternating Direction Method of Multipliers*).

2.1 Reformulation du problème JKO sous un problème sous contrainte avec Lagrangien

Cette reformulation est assez longue et se fait en plusieurs étapes. Je liste les 4 éléments les plus importants (à mon sens) qu'ils utilisent pour cette formulation. Dans leurs explications, ils enchaînent les arguments très rapidement sans donner trop d'explications car c'est évident pour un public de chercheurs spécialistes.

Je vais donc indiquer, après chaque argument, les explications supplémentaires que j'ai dû formuler pour comprendre vraiment l'enchaînement de leur raisonnement.

◇ Formule dynamique de Benamou-Brenier pour la distance de Wasserstein

L'article utilise le théorème suivant (dans le cas d'un domaine Ω borné et lisse) :

$$W_2^2(\rho, \rho^k) = \inf_{(\mu, m) \in C} \int_0^1 \int_{\mathbb{R}^d} \frac{\|m(x)\|^2}{\mu(x)} dx dt$$
$$C := \{ (\mu; m) : [0, 1] \times \mathbb{R}^2 \rightarrow (\mathbb{R}_+; \mathbb{R}_+^2) : \begin{aligned} &\partial_t \mu + \operatorname{div}(m) = 0, \\ &\mu|_{t=0} = \rho^k, \\ &\mu|_{t=1} = \rho, \\ &m \cdot \nu = 0 \text{ sur } \partial\Omega, \\ &\mu \geq 0, \\ &(\mu = 0 \Rightarrow m = 0) \} \end{aligned}$$

Cette formulation de la distance de Wasserstein permet de transformer le problème de JKO en un problème convexe variationnel. Pour cela on pose au préalable : $\mu_1 := \mu$ qui remplace la fonction de minimisation ρ . À la convergence, on retrouvera donc $\boxed{\rho^{k+1} = \mu_1}$.

$$\inf_{(\mu, m, \mu_1)} \left\{ \frac{1}{2} \int_0^1 \int_{\mathbb{R}^d} \frac{\|m(x)\|^2}{\mu(x)} dx dt + \tau \mathcal{E}(\mu_1) \right\} \quad (1)$$

Sous les contraintes :

- (i) $\partial_t \mu + \operatorname{div}(m) = 0$
- (ii) $\mu|_{t=0} = \rho^k$
- (iii) $\mu \geq 0$
- (iv) $(\mu = 0 \Rightarrow m = 0)$
- (v) $m \cdot \nu = 0$ sur $\partial\Omega$

◇ **Introduction d'une fonction support d'un convexe pour la prise en compte des contraintes (iii) et (iv)**

Les auteurs introduisent la fonction suivante ($\forall (\mu, m) \in \mathbb{R} \times \mathbb{R}^2$) :

$$\Psi(\mu, m) = \begin{cases} \frac{\|m\|_2^2}{2\mu}, & \mu > 0 \\ 0 & , \text{ si } \mu = 0 \text{ et donc } m = 0 \\ +\infty, & \text{ sinon} \end{cases}$$

qui est la fonction support (i.e. $\Psi(\mu, m) = \sup_{(\alpha, \beta) \in K} \{\alpha\mu + \beta \cdot m\}$) de l'ensemble convexe :

$$K = \left\{ (\alpha, \beta) \in \mathbb{R} \times \mathbb{R}^2 : \alpha + \frac{1}{2} \|\beta\|_2^2 \leq 0 \right\}$$

On voit donc que si l'on remplace dans (1) $\frac{\|m(x)\|_2^2}{\mu(x)}$ par la fonctionnelle $\Psi(\mu, m)$ les conditions (iii) et (iv) sont prises en compte car la nouvelle fonctionnelle prend la valeur $+\infty$ si ces contraintes ne sont pas vérifiées.

◇ **Formulation lagrangienne pour la contrainte (i)**

Les auteurs introduisent la fonction $\phi : [0, 1] \times \mathbb{R}^2 \rightarrow \mathbb{R}$ comme multiplicateur associé à la contrainte (i) : $\partial_t \mu + \operatorname{div}(m) = 0$. Ils donnent ensuite directement une forme de Lagrangien qui prend en compte cette condition sans expliquer ni donner d'indication sur les étapes. J'ai donc essayé de trouver et expliquer ces étapes.

Pour prendre en compte cette condition, on rajoute le terme : $\int_0^1 \int_{\Omega} \phi (\partial_t \mu + \operatorname{div}(m))$ dans l'expression à minimiser. On distribue ensuite ϕ pour avoir 2 terme à intégrer. Si on inverse l'ordre d'intégration par Fubini (fonctions continues sur intervalles bornés) et que l'on intègre par parties en temps on obtient pour le premier terme :

$$\int_0^1 \int_{\Omega} \phi \partial_t \mu = \int_{\Omega} \phi(1, \cdot) \mu(1, \cdot) - \int_{\Omega} \phi(0, \cdot) \mu(0, \cdot) - \int_{\Omega} \int_0^1 \partial_t \phi \mu$$

Avec bien sur notre notation $\mu(1, \cdot) = \mu_1$ et la condition (ii) : $\mu|_{t=0} = \rho^k$. Pour le deuxième on fait l'”intégration par partie” sur le domaine d'espace Ω . Ce dernier est compact à bord lisse, en supposant que ϕ et le champ m sont \mathcal{C}^1 on a la formule de Green en dimension 2 (Théorème de la divergence) :

$$\int_0^1 \int_{\Omega} \phi \operatorname{div}(m) = - \int_0^1 \int_{\Omega} m \operatorname{div}(\phi) + \int_{\partial\Omega} \phi(m \cdot \nu)$$

Avec le terme de bord qui s'annule grâce à la condition (v). Nous avons donc notre problème de minimisation (1) qui peut se formuler sous la forme de la minimisation d'un Lagrangien : $\inf_{(\mu, m, \mu_1)} \sup_{\phi} \tilde{L}(\mu, m, \mu_1; \phi)$ qui s'écrit alors :

$$\inf_{\sigma=(\mu, m, \mu_1)} \left\{ \int_0^1 \int_{\Omega} \Psi(\mu, m) + \tau \mathcal{E}(\mu_1) + \sup_{\phi} \left\{ \int_{\Omega} (\phi(1, \cdot) \mu_1 - \phi(0, \cdot) \rho^k) + \int_0^1 \int_{\Omega} (\partial_t \phi \mu + \operatorname{div}(\phi) m) \right\} \right\} \quad (\text{P})$$

2.2 Formulation sous la forme d'un Lagrangien augmenté

◇ Utilisation du problème dual

À partir de cette forme lagrangienne, les auteurs déduisent une forme duale au problème. Il y a plusieurs étapes et explications avant d'arriver à ce résultat, mais aucune ne m'a demandé un travail de ré-explication. J'ai juste eu besoin des notions sur la dualité Lagrangienne et les transformées de Legendre que j'ai vu dans mes cours d'optimisation précédents.

Je ne vais donc pas re-détailler ces étapes dans ce rapport. Mais simplement présenter le problème dual obtenu. Notons $\sigma := (\mu, m, \mu_1)$ et $q = (a, b, c)$ des fonctions respectivement de même nature que (μ, m, μ_1) . Il faut aussi introduire l'opérateur : $\Lambda : \phi \mapsto (\partial_t \phi, \nabla \phi, -\phi(1, \cdot))$ et la notation \mathcal{E}^* pour la transformée de Legendre de notre énergie \mathcal{E} .

Le problème (P) est dual au problème :

$$\inf_{\phi} F(\phi) + G(\Lambda \phi) \quad (\text{D})$$

Où

$$F(\phi) := \int_{\Omega} \phi(0, \cdot) \rho^k \quad G(q) := \int_0^1 \int_{\Omega} \chi_K(a, b) dx dt + \tau \mathcal{E}^*\left(\frac{c}{\tau}\right)$$

◇ Formulation du Lagrangien augmenté

D'après les auteurs, les conditions d'extrémas primales-duales sont équivalentes à trouver le point selle du Lagrangien suivant :

$$L(\phi, q, \sigma) = F(\phi) + G(q) + \sigma \cdot (\Lambda\phi - q)$$

C'est à dire que (ϕ, σ) respectent les conditions d'optimalité respectivement du dual (D) et du primal (P) si et seulement si $(\phi, q, \sigma) = (\phi, \Lambda\phi, \sigma)$ est un point selle du Lagrangien L. Nous pouvons donc enfin formuler notre lagrangien augmenté pour tout $r > 0$:

$$L_r(\phi, q, \sigma) = F(\phi) + G(q) + \sigma \cdot (\Lambda\phi - q) + \frac{r}{2} \|\Lambda\phi - q\|^2$$

J'ai eu un peu de mal à comprendre le sens des notations de produits scalaire \cdot ou norme au carré $\|\cdot\|^2$ utilisés. Ce sont en fait comme la somme des produits scalaires et normes de fonction (i.e. calcul d'intégrales). Par exemple :

$$\sigma \cdot q = \int_0^1 \int_{\Omega} (\mu(t, x) a(t, x) + m(t, x) \cdot b(t, x)) dx dt + \int_{\Omega} \mu_1(x) c(x) dx$$

$$\|\Lambda\phi - q\|^2 = \int_0^1 \int_{\Omega} ((\partial_t \phi(t, x) - a(t, x))^2 + \|\nabla \phi(t, x) - b(t, x)\|_2^2) dx dt + \int_{\Omega} (\phi(1, x) + c(x))^2 dx$$

2.3 Formulation de la méthode ADMM pour notre problème

◇ Pourquoi cette méthode ?

La forme du Lagrangien peut se décomposer en deux termes qui dépendent tous les deux de σ . Le premier qui est fonction de ϕ : $F(\phi) + \sigma \cdot \Lambda\phi$ et le deuxième fonction de q : $G(q) - \sigma \cdot q$.

Cette décomposition permet d'utiliser la méthode "*Alternating Direction Method of Multipliers*". Comme son nom l'indique, cette méthode consiste à minimiser selon la première variable (ici ϕ), puis minimiser selon la deuxième variable (ici q) puis mettre à jour le multiplicateur de Lagrange (ici σ) et ré-itérer.

J'ai déjà étudié cette méthode dans un cours d'optimisation mais dans le cas de minimisation de variable, jamais pour des fonctions. Dans la pratique, nous utilisons des éléments finis pour approcher ces fonctions et donc le problème devient à dimension finie (de dimension du nombre de noeuds du maillage). Nous avons dans ce cas là des résultats qui garantissent la convergence de la méthode.

◇ Présentation de l'algorithme ADMM pour notre problème

Il faut commencer par choisir q^0 et σ^0 . J'ai choisi en me basant sur des méthodes d'ADMM pour des minimisations de variable de prendre ces fonctions identiquement nulles.

On calcule ensuite la séquence de fonctions (ϕ^n, q^n, σ^n) avec les étapes suivantes :

- **Étape 1 :** Minimisation par rapport à ϕ :

$$\phi^{n+1} = \arg \min_{\phi} \left\{ F(\phi) + \sigma^n \cdot \Lambda \phi + \frac{r}{2} \|\Lambda \phi - q^n\|^2 \right\} \quad (\text{E1})$$

- **Étape 2 :** Minimisation par rapport à q :

$$q^{n+1} = \arg \min_q \left\{ G(q) - \sigma^n \cdot q + \frac{r}{2} \|\Lambda \phi^{n+1} - q\|^2 \right\} \quad (\text{E2})$$

- **Étape 3 :** Mise-à-jour du multiplicateur σ (montée de gradient) :

$$\sigma^{n+1} = \sigma^n + r (\Lambda \phi^{n+1} - q^{n+1}) \quad (\text{E3})$$

◇ Choix du critère d'arrêt

Pour ce qui concerne le critère d'arrêt il n'est pas vraiment indiqué par les auteurs. Le premier qui m'est venu en tête est celui qui indique si on a atteint le point-scelle du Lagrangien, i.e. si $(\phi, q, \sigma) = (\phi, \Lambda \phi, \sigma) \Leftrightarrow q = \Lambda \phi$. J'ai donc pris comme critère d'arrêt :

$$e_P := \|\Lambda \phi^n - q^n\|^2 \leq \varepsilon_P$$

Au cours de mon implémentation j'ai eu beaucoup de problèmes. Je me suis mis donc à douter de ce choix et je me suis replongé dans mes cours sur l'ADMM. Au final il s'avère que c'était un bon critère et qu'il correspond au résidu primal du point de vue de la formulation ADMM pour une minimisation de variable. En effet dans la formulation ADMM pour variable on a la contrainte du Lagrangien : $Ax + Bz - c = 0$. Dans notre cas fonctionnel j'ai identifié par l'opérateur Λ comme l'analogie de A et moins l'identité $-\text{Id}$ l'analogie de B (et $c = 0$).

Cela m'a permis de trouver un autre critère d'arrêt : le résidu dual. Du moins c'est ce que je pensais au début. Pour la formulation ADMM pour des variables on a l'expression du reste dual : $e_D := \|rA^T B(z^{n+1} - z^n)\|^2$. Si on le traduit dans notre formulation fonctionnelle on a :

$$e_D := \|r\Lambda^T(q^n - q^{n+1})\|^2$$

Où j'interprète Λ^T comme l'opérateur qui a $q = (a, b, c)$ va donner la fonction Q telle que $\partial_t Q = a$, $\nabla Q = b$ et $-Q(1, \cdot) = c$. Ce qui revient à résoudre un problème de transport (advection)

avec la donnée au temps final au lieu du temps initial. Je n'étais pas sûr de pouvoir résoudre ce problème et dans tous les cas cela aurait fait beaucoup de calculs pour un simple critère d'arrêt. De plus je n'étais même pas sûr de cette interprétation et je n'ai pas trop trouvé de cours en ligne sur la formulation ADMM pour des problèmes fonctionnels. J'ai donc choisi d'ignorer ce critère d'arrêt.

3 Traduction des minimisations des étapes 1 et 2

3.1 Étape 1 : un problème de poisson 3D

L'article indique simplement que l'étape 1 (E1) revient à un problème de Poisson 3D que je vais indiquer par la suite sans donner aucune explication. Avant ça je vais expliquer d'où cela vient :

◇ Minimisation quadratique → calcul de dérivée

Le problème (E1) est la minimisation par rapport à ϕ d'un terme quadratique par rapport ϕ . Nous allons donc essayer d'annuler la dérivée par rapport à ϕ de ce terme. Pour cela il faut d'abord exprimer cette dérivée "fonctionnelle". J'ai cherché un peu sur internet et dans la plupart des cas elle est exprimée à partir d'une variation, de la même manière que l'on introduit les dérivées partielles. On pose donc $J(\phi) = F(\phi) + \sigma^n \cdot \Lambda\phi + \frac{r}{2}\|\Lambda\phi - q^n\|^2$ et on va développer $J(\phi + \epsilon\psi)$ pour chacun des termes de J .

- 1^{er} terme $F(\phi)$:

$$F(\phi + \epsilon\psi) = F(\phi) + \epsilon \int_{\Omega} \psi(0, \cdot) \rho^k$$

- 2^{ème} terme, produit scalaire $\sigma^n \cdot \Lambda\phi$:

$$\sigma^n \cdot \Lambda(\phi + \epsilon\psi) = \sigma^n \cdot \Lambda\phi + \epsilon (\sigma^n \cdot \Lambda\psi)$$

- 3^{ème} terme, pénalisation quadratique $\frac{r}{2}\|\Lambda\phi - q^n\|^2$:

On utilise ici le fait que la norme à développer est en fait une somme de normes : $\|\Lambda\phi - q^n\|^2 = \|\partial_t\phi - a^n\|_{[0,1] \times \Omega}^2 + \|\nabla\phi - b^n\|_{[0,1] \times \Omega}^2 + \|\phi(1, \cdot) - c^n\|_{\Omega}^2$. Développons alors le premier avec notre variation $\epsilon\psi$:

$$\begin{aligned}
\|\partial_t(\phi + \epsilon\psi) - a^n\|_{[0,1]\times\Omega}^2 &= \|\partial_t\phi - a^n + \epsilon\partial_t\psi\|_{[0,1]\times\Omega}^2 \\
&= \|\partial_t\phi - a^n\|_{[0,1]\times\Omega}^2 + 2\epsilon \int_0^1 \int_{\Omega} (\partial_t\phi - a^n) \partial_t\psi \, dx \, dt + \epsilon^2 \|\partial_t\psi\|_{[0,1]\times\Omega}^2 \\
&= \|\partial_t\phi - a^n\|_{[0,1]\times\Omega}^2 + 2\epsilon \int_0^1 \int_{\Omega} (\partial_t\phi - a^n) \partial_t\psi \, dx \, dt + \mathcal{O}(\epsilon^2)
\end{aligned}$$

Avec cette écriture nous identifions facilement le terme facteur de ϵ et le reste d'ordre 2. On peut reproduire le même raisonnement pour les autres termes qui composent la pénalisation quadratique car ils sont également le carré de la différence entre ϕ ou son gradient et une composante de q^n .

◇ Annulation de dérivée → formulation variationnel

On obtient notre dérivée selon la "direction" ψ qui est la somme des termes facteur de ϵ . Ces termes sont tous des intégrales. On veut avoir cette dérivée nulle pour toutes les "directions" i.e. pour toute fonction ψ . Annuler la dérivée de J par rapport à ϕ revient donc à résoudre $(\forall \psi)$:

$$\begin{aligned}
&\int_0^1 \int_{\Omega} (\mu^n \partial_t\psi + m^n \cdot \nabla\psi + r((\partial_t\phi - a^n)\partial_t\psi + (\nabla\phi - b^n) \cdot \nabla\psi)) \, dx \, dt \\
&+ \int_{\Omega} \left(\psi(0, \cdot) \rho^k + \mu_1^n \psi(1, \cdot) + r\psi(1, \cdot)(\phi(1, \cdot) - c^n) \right) \, dx = 0
\end{aligned}$$

Que l'on peut réécrire :

$$\begin{aligned}
&r \int_0^1 \int_{\Omega} \nabla_{t,x}\phi \cdot \nabla_{t,x}\psi = \int_0^1 \int_{\Omega} (r(a^n, b^n) - (\mu^n, m^n)) \cdot \nabla_{t,x}\psi \\
&- \int_{\Omega} \left(\psi(0, \cdot) \rho^k + \mu_1^n \psi(1, \cdot) + r\psi(1, \cdot)(\phi(1, \cdot) - c^n) \right) \, dx
\end{aligned}$$

Posons $\Theta = [0, 1] \times \Omega$ notre domaine 3D. nous souhaitons utiliser la première identité de Green $(\int_{\Theta} \nabla u \cdot \nabla v = -\int_{\Theta} v \Delta u + \int_{\partial\Theta} v \nabla u \cdot \nu)$ et le théorème de la divergence pour un champ $(\int_{\Theta} \nabla u \cdot f = -\int_{\Theta} u \operatorname{div}(f) + \int_{\partial\Theta} u(f \cdot \nu))$.

On a la frontière de Θ : $\partial\Theta = \partial\Omega \cup \Gamma_0 \cup \Gamma_1$. Où

- $\partial\Omega$: Frontière du domaine d'espace Ω extrudée sur $t \in [0, 1]$.
- $\Gamma_0 := \Theta \cap \{t = 0\}$: Limite "à gauche" pour le temps
- $\Gamma_1 := \Theta \cap \{t = 1\}$: Limite "à droite" pour le temps

Il faut donc préciser le sens du terme à la frontière $\nabla u \cdot \nu$ pour chacune de ces sous frontières. Sur $\partial\Omega$ on a par définition $\nabla u \cdot \nu|_{\partial\Omega} =: \frac{\partial u}{\partial \nu}$. Sur Γ_0 (respectivement Γ_1) on remarque que la normale vaut $(-1, 0, 0)^T$ (resp. $(1, 0, 0)^T$). On a donc $\nabla u \cdot \nu|_{\Gamma_0} =: \partial_t u(0, \cdot)$ et $\nabla u \cdot \nu|_{\Gamma_1} =: \partial_t u(1, \cdot)$

On peut donc réécrire l'équation d'annulation de dérivée :

$$\begin{aligned}
& - \int_{\Theta} r \Delta_{t,x} \phi \, \psi - \int_{\Gamma_0} r \partial_t \phi(0, \cdot) \psi + \int_{\Gamma_1} r \partial_t \phi(1, \cdot) \psi + \int_{\partial\Omega} r \frac{\partial \phi}{\partial \nu} \psi + \int_{\Gamma_1} r \phi(1, 0) \psi \\
& = - \int_{\Theta} \operatorname{div}((a^n, b^n) - (\mu^n, m^n)) \, \psi - \int_{\Gamma_0} (\mu^n(0, \cdot) - r a^n(0, \cdot)) \, \psi \\
& \quad + \int_{\Gamma_1} (\mu^n(1, \cdot) - r a^n(1, \cdot)) \, \psi + \int_{\partial\Omega} (m^n - r b^n) \cdot \nu \, \psi - \int_{\Gamma_0} \rho^k \, \psi + \int_{\Gamma_1} r (\mu_1^n - c^n) \, \psi
\end{aligned}$$

◇ Problème "fort" de la formulation variationnelle

Nous pouvons donc identifier que le problème ci-dessus est une formulation variationnelle ou "faible" d'une EDP sous la fonction test ψ . Je ne suis pas assez calé en analyse variationnelle pour connaître les conditions et le théorème qui permettent de passer de la forme faible à la forme forte. Je suis obligé de supposer que c'est possible.

Pour obtenir l'EDP de la formulation forte on regroupe les termes sous l'intégrale sur Θ . Les termes des intégrales sur $\partial\Omega$, Γ_0 et Γ_1 donnent respectivement les conditions aux limites en espace, en temps à gauche ($t = 0$) et en temps à droite ($t = 1$). Nous obtenons donc le problème de poisson 3D suivant :

ϕ^{n+1} : solution du problème de Poisson **3D** sur $\Theta = [0, 1] \times \Omega$:

$$\boxed{-\Delta_{t,x} \phi = \operatorname{div}_{t,x} \left(\frac{1}{r} (\mu^n, m^n) - (a^n, b^n) \right)}$$

— CL espace : $\frac{\partial \phi}{\partial \nu} = -\frac{1}{r} (m^n - r b^n) \cdot \nu$ (sur $\partial\Omega$)

— CL de temps $t \in \{0, 1\}$:

$$\begin{cases} \partial_t \phi(0, \cdot) = \frac{1}{r} (\rho^k - \mu^n(0, \cdot)) + a^n(0, \cdot) & , \text{ sur } \Gamma_0 \\ \partial_t \phi(1, \cdot) = -\phi(1, \cdot) + \frac{1}{r} (\mu_1^n - \mu^n(1, \cdot)) + (a^n(1, \cdot) - c^n(\cdot)) & , \text{ sur } \Gamma_1 \end{cases}$$

C'est le résultat qui est donné par les auteurs avec un ré-arrangement des égalités et des coefficients r , $1/r$.

3.2 Étapes 2 : minimisations en tous point $(t, x) \in \Theta$

◇ Séparation en deux problèmes indépendants

Le problème de l'étape 2 (E2) est le suivant :

$$q^{n+1} = \arg \min_{q=(a,b,c)} \int_0^1 \int_{\Omega} \chi_K(a,b) dx dt + \tau \mathcal{E}^* \left(\frac{c}{\tau} \right) - \sigma^n \cdot q + \frac{r}{2} \|\Lambda \phi^{n+1} - q\|^2$$

Il faut remarquer ici que si on utilise la décomposition précédente du produit scalaire et de la pénalisation quadratique, on fait apparaître une somme composée de termes qui dépendent (a, b) et d'autres de c . Nous pouvons donc séparer cette étape en un problème de minimisation par rapport à (a, b) et un autre par rapport à c .

◇ Minimisation par rapport à $(a, b) \rightarrow$ projection

Le problème d'optimisation est le suivant :

$$(a^{n+1}, b^{n+1}) = \arg \min_{(a,b)} \int_0^1 \int_{\Omega} \chi_K(a,b) dx dt - (\mu^n, m^n) \cdot (a, b) + \frac{r}{2} \|(\partial_t \phi^{n+1}, \nabla \phi^{n+1}) - (a, b)\|_{[0,1] \times \Omega}^2$$

On peut facilement prendre en compte l'indicatrice convexe $\chi_K(a, b)$ en rajoutant la contrainte $(a, b) \in K$ à notre problème de minimisation. Si l'on développe le carré de la pénalisation quadratique on obtient :

$$\begin{aligned} (a^{n+1}, b^{n+1}) &= \arg \min_{(a,b) \in K} -(\mu^n, m^n) \cdot (a, b) + \frac{r}{2} \|(\partial_t \phi^{n+1}, \nabla \phi^{n+1})\|_{[0,1] \times \Omega}^2 - r(\partial_t \phi^{n+1}, \nabla \phi^{n+1}) \cdot (a, b) + \frac{r}{2} \|(a, b)\|_{[0,1] \times \Omega}^2 \\ &= \arg \min_{(a,b) \in K} \frac{r}{2} \|(\partial_t \phi^{n+1}, \nabla \phi^{n+1})\|_{[0,1] \times \Omega}^2 - ((\mu^n, m^n) + r(\partial_t \phi^{n+1}, \nabla \phi^{n+1})) \cdot (a, b) + \frac{r}{2} \|(a, b)\|_{[0,1] \times \Omega}^2 \end{aligned}$$

Étant donné que nous nous intéressons à l'argument de minimisation nous pouvons diviser tout par r et rajouter le terme $\frac{1}{2} \|(\partial_t \phi^{n+1}, \nabla \phi^{n+1})\|_{[0,1] \times \Omega}^2 + \frac{2}{r} (\partial_t \phi^{n+1}, \nabla \phi^{n+1}) \cdot (\mu^n, m^n)$ sans modifier l'égalité. En regroupant toutes les distributions des carrés des normes on obtient :

$$(a^{n+1}, b^{n+1}) = \arg \min_{(a,b) \in K} \frac{1}{2} \|(\partial_t \phi^{n+1}, \nabla \phi^{n+1}) + \frac{1}{r} (\mu^n, m^n) - (a, b)\|_{[0,1] \times \Omega}^2$$

On a donc notre problème qui se transforme, par définition, en une simple projection sur K . C'est bien le résultat qui était donné directement par les auteurs. Nous détaillerons la projection dans la partie implémentation.

◇ Minimisation pour c

Je me suis rendu compte que j'avais déjà beaucoup écrit. L'explication de cette étape implique d'introduire de nouvelles notations, calculer une transformée (simple) de Legendre et ensuite

reformuler plusieurs fois le problème. Par soucis de temps et d'espace je ne vais pas donc pas détailler cette étape.

4 Détails d'implémentation

4.1 Éléments finis pour l'étape 1

◇ Discrétisation du temps et de l'espace

Nous devons tout d'abord une discrétisation pour notre espace Ω . J'ai choisi de commencer par quelque chose de simple avec un simple carré de 16×16 noeuds. Ensuite pour la discrétisation du temps on construit $\Theta = [0, 1] \times \Omega$ en extrudant notre maillage pour Ω . J'ai pris 8 noeuds pour la discrétisation en temps.

En pratique j'ai directement créé un cube au bonne dimension. Mais il vaut mieux créer Ω et ensuite utiliser la commande FreeFEM buildlayers pour pouvoir extruder n'importe quelle forme de Ω . C'est ce que je voulais faire pour simuler une porte ou un goulot d'étranglement. Malheureusement je n'avais pas trop le temps et la documentation ne donnait pas des exemples très clairs pour la gestion des indices des arrêtes pour les conditions aux bords.

J'ai donc les maillages suivants :

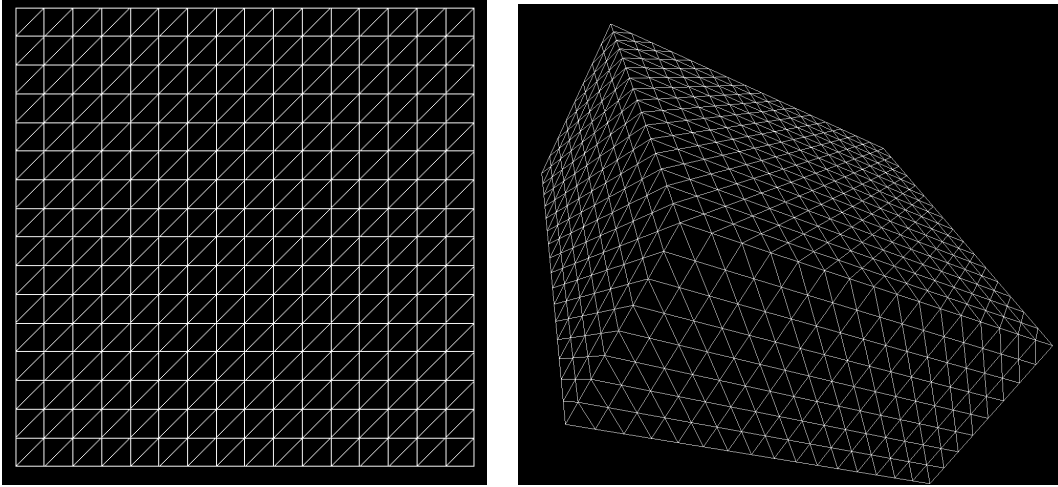


FIGURE 1 – Discrétisation de Ω (droite) \rightarrow extrusion pour Θ (gauche)

◇ Choix de l'interpolation élément finis \mathbb{P}_k

Les auteurs précisent d'utiliser des éléments finis \mathbb{P}_2 pour approcher les ϕ et \mathbb{P}_1 pour les autres fonctions. Je ne comprenais pas vraiment au début et j'ai compris pourquoi en réfléchissant plus sur le sens de l'approximation par éléments finis.

Dans les étapes on utilise les dérivées en temps $\partial_t \phi$ et en espace $\nabla \phi$ de ϕ en les sommant aux autres fonctions.

Dériver une fonction approchée en éléments finis \mathbb{P}_k revient à dériver des polynômes de degrés k . Donc on retrouve une approximation de la dérivée à partir de polynômes de degré $k - 1$. On va donc manquer de précisions si on ne prend le degré au dessus. Et dans notre cas cela revient faire des sommes de fonctions qui sont \mathbb{P}_k avec d'autres qui sont \mathbb{P}_{k-1} .

◇ Dérivées de $\phi \rightarrow$ projection sur \mathbb{P}_1

Les auteurs indiquent qu'il faut comprendre les opérations de dérivée de ϕ comme des projections de \mathbb{P}_2 sur \mathbb{P}_1 . Je comprenais sans comprendre et j'ai utilisé l'opérateur de dérivée basique de FreeFEM pour obtenir mes dérivées $\mathbf{dx}(\)$.

Mais lorsque je faisais cela, je n'avais pas convergence dans mon algorithme. Après avoir passé énormément de temps à revoir tous mes calculs et mes lignes de code je me suis dit que le mot projection avec plus de sens que ce que je pensais.

J'ai donc cherché beaucoup sur internet jusqu'à réussir à tomber sur un code FreeFEM qui traitait des dérivées comme j'en avais besoin.

J'avais juste des lignes de code et j'ai essayé de comprendre ce qu'il fallait faire. Voici ce que j'en ai déduit. Il faut calculer une première formulation variationnelle pour passer de \mathbb{P}_2 à \mathbb{P}_1 . Et une autre que j'appellerai de normalisation de \mathbb{P}_1 .

$$\text{fv}_{21x}(u, v) = \int_{\Theta} \partial_x u v \quad \text{fv}_{11}(u, v) = \int_{\Omega} uv$$

.

Ensuite il faut assembler les matrices d'approximation des ces formulations variationnelles. Pour avoir la projection, il faut donner en entrée de ces formules variationnelles non pas u et v comme on a l'habitude mais les espaces fonctionnels directement. Que l'on note V_{hn} avec n le degré des élément fini. On obtient notre matrice de projection P_{21x} et de normalisation P_{11} de la manière suivante :

$$P_{21x} \simeq \text{fv}_{21x}(V_{h2}, V_{h1}) \quad P_{11} \simeq \text{fv}_{11}(V_{h1}, V_{h1})$$

Pour obtenir notre dérivée par rapport à x de ϕ , il faut appliquer matrices de projection et de normalisation au vecteur $\vec{\phi}$ qui contient les coefficients associés à chaque nœud $= \{\phi_i\}_i$ de

l'approximation de ϕ .

$$(i.e.) \quad \overrightarrow{\partial_x \phi} = P_{11}^{-1} P_{21x} \overrightarrow{\phi}$$

J'ai donc calculé les dérivées de ϕ à chaque itération de cette manière. J'ai du également faire des projections de \mathbb{P}_2 à \mathbb{P}_1 sans dérivées lorsque je devais par exemple additionner $\phi(1, .)$ à d'autres termes \mathbb{P}_1 . Dans ce cas c'est le même principe sauf qu'au lieu d'utiliser fv_{21x} on utilise :

$$fv_{21}(u, v) = \int_{\Theta} u v$$

4.2 Minimisation en tout points (t, x) pour l'étape 2

L'étape 2 consiste initialement à obtenir les fonctions (a, b, c) comme les arguments minimums en tout point (t, x) de fonctionnelles convexes.

◇ Signification de "en tout point (t, x) "

J'ai bloqué longtemps sur cette partie parce que je ne comprenais pas trop comment minimiser en tout point. Je n'avais encore pas trop assimilé la notion d'assemblage que l'on était en train de voir en cours Élément finis. Je n'avais donc pas en tête qu'une fonction s'approchait par des éléments finis tout simplement de la manière suivante :

$$\forall (t, x), \phi(t, x) \simeq \sum_{i=1}^N \phi_i w_i(t, x)$$

Avec N le nombre total de nœuds $n_i = (t_i, x_i)$. En utilisant la propriété qui définit une fonction de base : $w_i(n_j) = \delta_{ij}$ on voit que l'on a en tout nœud d'interpolation : $\phi(n_i) = \phi(t_i, x_i) = \phi_i$.

De cette manière nous pouvons récupérer les valeurs en chaque point d'interpolation pour avoir un problème d'optimisation en chaque nœud. Ainsi nous pouvons définir les coefficients des fonctions $(a_i^{n+1}, b_i^{n+1}, c_i^{n+1})$ et donc directement définir les fonctions $(a^{n+1}, b^{n+1}, c^{n+1})$.

◇ Projection sur K pour obtenir (a^{n+1}, b^{n+1})

On a montré que :

$$(a^{n+1}, b^{n+1}) = \text{proj}_K \left((\partial_t \phi^{n+1}, \nabla \phi^{n+1}) + \frac{1}{r} (\mu^n, m^n) \right)$$

L'opération de projection d'un point sur K nous est tiré par les auteurs d'une autre publication. Je suis allé voir mais c'était très long et trop généraliste pour que j'essaye de le remonter

moi-même. On a :

$$\text{proj}_K(\alpha, \beta) = \begin{cases} (\alpha, \beta), & \text{si : } \alpha + \frac{1}{2}\beta^2 \leq 0 \\ \left(\alpha - \lambda, \frac{\beta}{1-\lambda}\right), & \text{avec } \lambda \text{ racine de } (\star\star) \text{ sinon} \end{cases}$$

$$(\star\star) : (\alpha - \lambda)(1 + \lambda)^2 + \frac{1}{2}\beta^2 = 0$$

Il faut donc calculer la racine positive (il y en a qu'une dans ce cas) d'un polynôme de degré 3. J'ai utilisé une formule de résolution directe pour calculer cette racine. Pour cela il faut exprimer la forme dépressive du polynôme et utiliser la formule de Cardano ou la formulation trigonométrique en fonction de si le discriminant est positif ou non.

Il aurait peut-être été plus rapide (en temps d'exécution) d'utiliser la méthode de Newton mais je n'ai pas eu le temps de comparer.

◇ Minimisation pour obtenir c^{n+1}

Posons $\bar{c}(x) := -\phi^{n+1}(1, x) + \frac{1}{r}\mu_1^n(x)$. On arrive à montrer que la minimisation revient en chaque point (t, x) à prendre

$$c^{n+1}(x) = \begin{cases} \bar{c}(x) - \frac{1}{r}, & \text{si : } \bar{c}(x) \geq \tau(1 + V(x)) + \frac{1}{r} \\ \text{racine de } (\star\star\star), & \text{sinon} \end{cases}$$

$$(\star\star\star) = c(x) + \frac{1}{r} \exp\left(\frac{c(x)}{\tau} - V(x) - 1\right) - \bar{c}(x)$$

Cette fois-ci pour calculer la racine de $(\star\star\star)$ j'ai utilisé la méthode de Newton avec une précision minimale de 10^{-8} et 500 itérations max.

5 Résultats

5.1 Visualisation d'un déplacement de foule simple

◇ Construction du cas de déplacement simple

J'ai construit un exemple simple avec la foule initialement concentrée au centre et un gradient de potentiel qui force à aller vers deux cercles à gauche et à droite. J'ai donc le potentiel suivant :

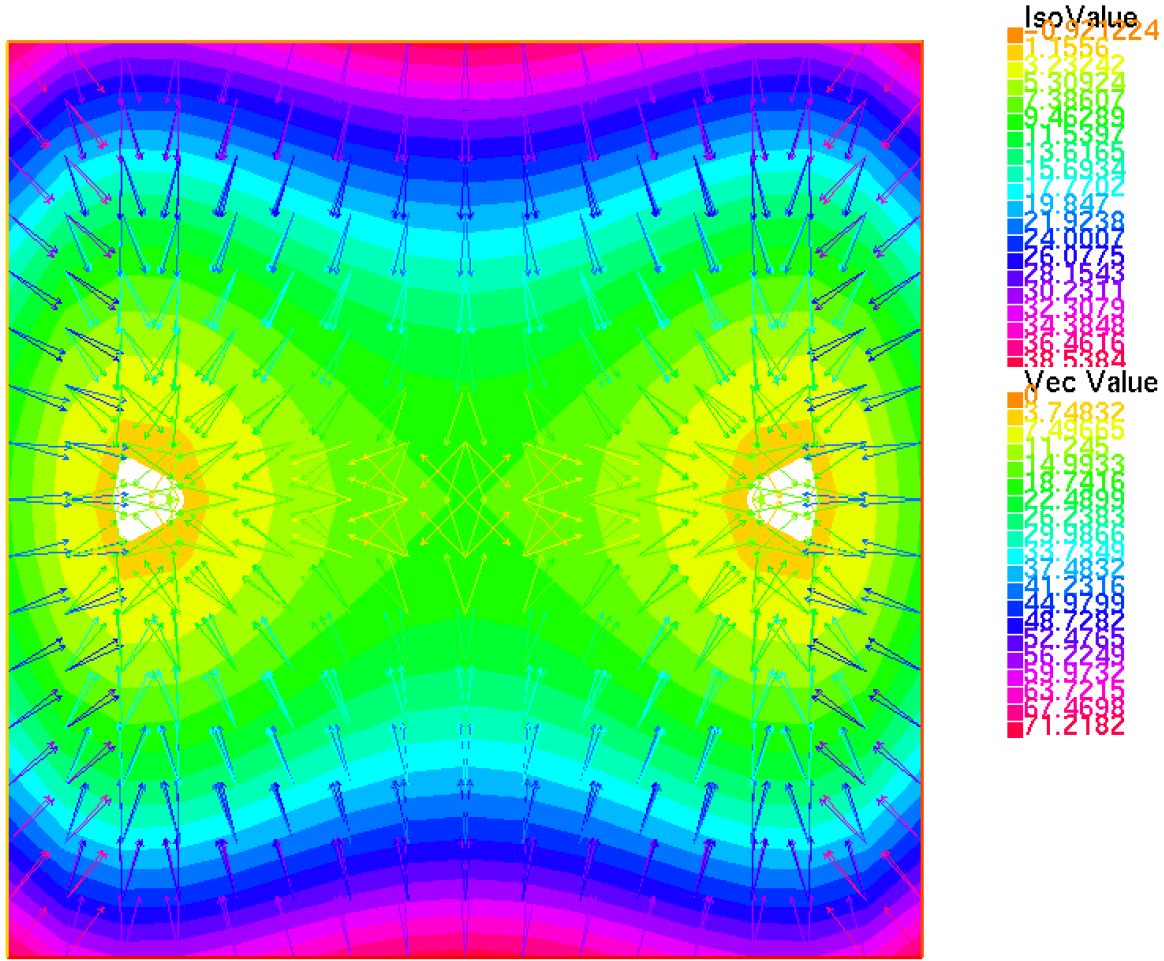


FIGURE 2 – Potentiel V et l'opposé de son gradient : $-\nabla_x V$ sur Ω

◇ Choix du paramètre ρ

Ce paramètre est un peu semblable au pas pour la descente de gradient. Je n'ai pas eu le temps de trop l'étudier. Néanmoins j'ai fait de nombreux tests choisir un ρ ni trop grand ni trop petit afin d'avoir une convergence rapide. J'ai pris $\rho = 0.9$.

◇ Visualisation du déplacement de la foule

En prenant un pas de discrétisation $\tau = 0.01$ on observe un déplacement logique de la foule :

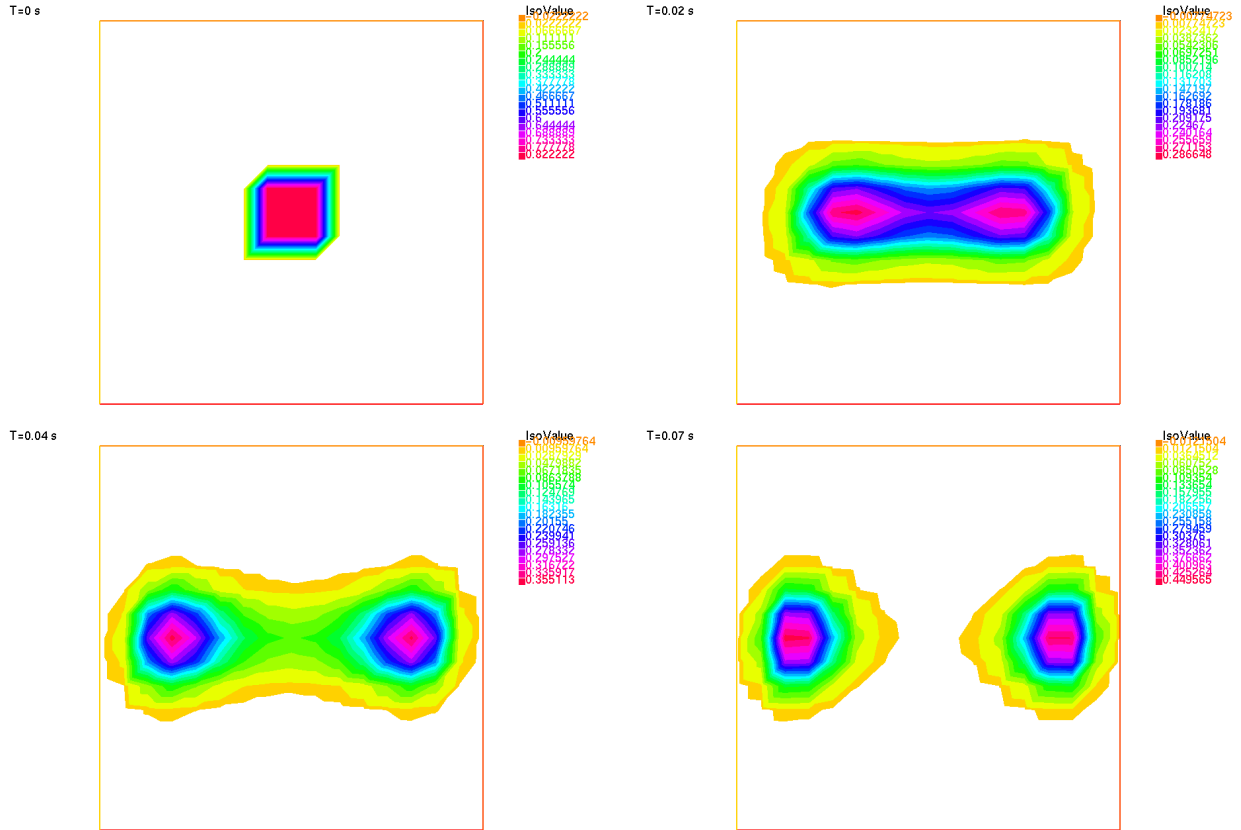


FIGURE 3 – Étapes du mouvement de foule

5.2 Convergence et temps d'exécution

ATTENTION Lorsque l'on exécute le code, il se met en pause automatiquement dès qu'il y a plus de 5 graphes en attente d'être affiché. Donc au bout de quelques itérations il faut aller sur la fenêtre du plot et cliquer sur entrer pour passer au graphe suivant et libérer l'exécution **ATTENTION**.

◇ Convergence de l'algorithme pour JKO

J'ai pu observer la convergence du critère d'arrêt e_p . Mais étant donné que le code n'est pas parallélisé en général, je garde une précision d'arrêt pas trop petite (environ 10^{-3}) pour avoir un temps d'exécution total en dessous des 5 minutes.

J'ai poussé jusqu'à prendre une précision de 10^{-4} ce qui a donné un temps de calcul d'environ

13 minutes. J'avais en moyenne 130 itérations pour une étape de JKO.

Je n'ai malheureusement pas eu le temps de faire un plot pour montrer l'évolution de la convergence du critère d'arrêt. En effet il faut exporter les données sur Matlab ce qui est un peu compliqué à mettre en oeuvre et je n'avais pas trop le temps. Cependant j'affiche le critère de convergence à chaque itération et il semble bien converger de manière plus ou moins logarithmique.

◇ Répartition du temps de calcul

J'ai mesuré les temps de calcul pour l'étape 1 et l'étape 2 afin de connaître ce qui était le plus coûteux dans mes calculs. Sans surprise la résolution du problème de Poisson en 3D est de loin le plus coûteux avec 717s pour 8 étapes de JKO avec précision de 10^{-4} . On a ensuite le reste des opérations avec 19s pour la projection sur K , 18s pour la minimisation de c et quand même 12s pour les opérations de projection sur les dérivées.

6 Bilan et conclusion

Dans ce projet j'ai réussi à faire ce que je voulais, à savoir modéliser le déplacement d'une foule. J'aurais bien aimé prendre un domaine d'espace plus compliqué pour pouvoir modéliser un goulot d'étranglement.

Une grande partie de mon travail a été de comprendre et expliquer les enchaînements d'étapes de l'article. Tous les résultats et explications que j'ai détaillé ici ne sont pas des éléments de l'article mais mon travail personnel. C'est pour cela qu'il m'a semblé important de l'inclure.

L'autre partie a été l'apprentissage de FreeFEM et l'utilisation des éléments finis. Au final j'ai adoré ce projet car il m'a permis de comprendre complètement un article qui me semblait très compliqué, et m'a fait poser beaucoup de questions sur les éléments finis que je maîtrise mieux maintenant.