

Documentation Technique : Simulateur de Mécanique des Fluides (LBM)

Créé par : Hamza Boudhair

20 novembre 2025

Table des matières

1	Introduction	2
2	Fondements Théoriques (Les Équations)	2
2.1	Modèle D2Q9	2
2.2	Équation de Lattice Boltzmann (LBE)	2
2.3	Fonction d'Équilibre	2
2.4	Variables Macroscopiques	2
2.5	Géométrie : Transformation de Joukowski	3
3	Guide de l'Interface Utilisateur	4
3.1	Contrôles de Simulation	4
3.2	Paramètres d'Écoulement (Flow Parameters)	4
3.3	Rendu et Contrôles (Rendering & Controls)	4
4	Détails de l'Implémentation (Code)	5
4.1	Structures de Données Optimisées	5
4.2	La Boucle de Simulation (Fonction <code>step</code>)	5
4.2.1	1. Collision et Calcul Macroscopique	5
4.2.2	2. Streaming (Propagation)	5
4.2.3	3. Gestion des Obstacles (Bounce-Back)	5
4.3	Visualisation (Vorticité)	6
5	Conclusion	6

1 Introduction

Ce document détaille le fonctionnement mathématique et informatique du simulateur de fluides interactif basé sur la méthode de Boltzmann sur réseau (Lattice Boltzmann Method - LBM). Cette application web permet de visualiser en temps réel les écoulements laminaires et turbulents autour d'un profil d'aile généré par transformation de Joukowski.

2 Fondements Théoriques (Les Équations)

Le moteur physique ne résout pas directement les équations de Navier-Stokes macroscopiques, mais utilise une approche mésoscopique cinétique.

2.1 Modèle D2Q9

La simulation utilise une grille discrète en 2 dimensions avec 9 vitesses discrètes (D2Q9). Pour chaque nœud de la grille, la fonction de distribution $f_k(\vec{x}, t)$ représente la probabilité de trouver une particule se déplaçant dans la direction k .

Les 9 vecteurs vitesse \vec{c}_k sont :

- $k = 0$: $(0, 0)$ (Repos)
- $k = 1..4$: $(\pm 1, 0), (0, \pm 1)$ (Voisins directs)
- $k = 5..8$: $(\pm 1, \pm 1)$ (Diagonales)

2.2 Équation de Lattice Boltzmann (LBE)

L'évolution du fluide est régie par l'équation discrète avec l'opérateur de collision BGK (Bhatnagar-Gross-Krook) :

$$f_k(\vec{x} + \vec{c}_k \Delta t, t + \Delta t) = f_k(\vec{x}, t) - \frac{1}{\tau} [f_k(\vec{x}, t) - f_k^{eq}(\vec{x}, t)] \quad (1)$$

Où :

- τ est le temps de relaxation, lié à la viscosité cinématique ν par : $\nu = (\tau - 0.5)c_s^2 \Delta t$.
- f_k^{eq} est la fonction de distribution à l'équilibre.

2.3 Fonction d'Équilibre

L'équilibre thermodynamique est défini par une expansion de Maxwell-Boltzmann :

$$f_k^{eq} = w_k \rho \left[1 + \frac{3(\vec{c}_k \cdot \vec{u})}{c^2} + \frac{9(\vec{c}_k \cdot \vec{u})^2}{2c^4} - \frac{3\vec{u}^2}{2c^2} \right] \quad (2)$$

Où w_k sont les poids associés au modèle D2Q9 (4/9 pour le centre, 1/9 pour les axes, 1/36 pour les diagonales).

2.4 Variables Macroscopiques

Les grandeurs physiques observables sont recalculées à chaque pas de temps à partir des moments de f_k :

$$\text{Densité : } \rho = \sum_{k=0}^8 f_k \quad (3)$$

$$\text{Vitesse : } \vec{u} = \frac{1}{\rho} \sum_{k=0}^8 f_k \vec{c}_k \quad (4)$$

2.5 Géométrie : Transformation de Joukowski

Le profil d'aile est généré en transformant un cercle dans le plan complexe z vers le plan physique ζ :

$$\zeta = z + \frac{1}{z} \quad (5)$$

En décentrant le cercle initial, on contrôle l'épaisseur et la cambrure de l'aile.

3 Guide de l'Interface Utilisateur

Cette section explique la fonction de chaque contrôle disponible dans la barre latérale (Sidebar).

3.1 Contrôles de Simulation

Simulation Type (Menu déroulant)

Permet de changer le mode de visualisation :

- **Vorticity** : Affiche le rotationnel (tourbillons). Utile pour voir la turbulence.
- **Velocity** : Affiche la magnitude de la vitesse.
- **Pressure** : Affiche le champ de pression (basé sur la densité ρ).

Simulation Type (Menu déroulant)

Choix de la palette de couleurs (Cyan, Plasma, Jet, etc.) pour adapter la visualisation aux besoins (artistique ou scientifique).

3.2 Paramètres d'Écoulement (Flow Parameters)

Flow Speed (Mach) - Slider

Contrôle la vitesse d'entrée du fluide (u_0).

- Valeur basse : Écoulement lent et stable.
- Valeur haute : Écoulement rapide, mais risque d'instabilité numérique si > 0.4 .

Viscosity - Slider

Contrôle la viscosité cinématique du fluide (paramètre ν).

- **Haute viscosité (Miel)** : Écoulement laminaire, très stable, pas de tourbillons.
- **Basse viscosité (Air/Eau)** : Écoulement turbulent, nombre de Reynolds élevé, apparition d'allées de tourbillons (Vortex shedding).

Angle of Attack (AoA) - Slider

Change l'angle d'incidence de l'aile par rapport au flux. Un angle élevé provoque un décrochage (séparation de la couche limite).

3.3 Rendu et Contrôles (Rendering & Controls)

Contrast - Slider

Amplifie les couleurs de la visualisation. Utile pour voir les micro-structures turbulentes.

Boutons Presets (Cruise, Stall, Hover)

Applique instantanément une configuration prédéfinie (ex : "Stall" met un angle d'attaque élevé et une basse viscosité pour forcer le décrochage).

Reset

Réinitialise tout le champ de fluide à l'état initial (écoulement uniforme).

Pause/Resume

Figé ou relance la boucle de calcul.

4 Détails de l'Implémentation (Code)

4.1 Structures de Données Optimisées

Pour garantir la performance (60 FPS) dans le navigateur, nous utilisons des tableaux typés unidimensionnels (Float32Array) plutôt que des tableaux d'objets classiques.

```

1 let Nx = 400, Ny = 200; // Taille de la grille
2 const Q = 9; // 9 vitesses par cellule
3
4 // f stocke les 9 populations pour chaque cellule
5 let f = new Float32Array(Nx * Ny * Q);
6 // Tableaux macroscopiques
7 let rho = new Float32Array(Nx * Ny);
8 let ux = new Float32Array(Nx * Ny);
9 let uy = new Float32Array(Nx * Ny);

```

Listing 1 – Allocation Mémoire

4.2 La Boucle de Simulation (Fonction step)

Cette fonction est appelée plusieurs fois par image. Elle réalise les étapes clés de la LBM.

4.2.1 1. Collision et Calcul Macroscopique

On calcule ρ et \vec{u} , puis on applique la relaxation vers l'équilibre.

```

1 // Exemple simplifi de la collision
2 for (let i = 0; i < Nx * Ny; i++) {
3   // Calcul densité r (rho)
4   let r = f[idx] + f[idx+1] + ... + f[idx+8];
5
6   // Relaxation BGK
7   // f[k] += omega * (f_eq - f[k])
8 }

```

4.2.2 2. Streaming (Propagation)

Les particules se déplacent vers les cellules voisines. Nous gérons ici les conditions aux limites périodiques ou murs.

4.2.3 3. Gestion des Obstacles (Bounce-Back)

Si une cellule est marquée comme solide dans le tableau `mask`, les particules entrantes sont renvoyées dans la direction opposée, simulant une condition d'adhérence (No-slip condition).

```

1 if (mask[next_index]) {
2   // Rebond : la particule repart d'o elle vient
3   fNew[current_index + opposite_direction] = f[current_index +
4   direction];
5 }

```

4.3 Visualisation (Vorticit  )

Le rendu est effectu   pixel par pixel en calculant le *Curl* de la vitesse.

$$\omega = \frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y} \approx (u_y(x+1) - u_y(x-1)) - (u_x(y+1) - u_x(y-1)) \quad (6)$$

5 Conclusion

Ce projet d  montre la puissance des algorithmes modernes (LBM) coupl  s aux capacit  s graphiques des navigateurs web. Il permet une exploration intuitive de la dynamique des fluides, rendant accessibles des concepts complexes comme la turbulence, la couche limite et le d  crochage a  rodynamique.