

La Chaîne Numérique d'un Modèle de Trafic Routier

Du Problème Physique à la Simulation Numérique

Analyse du code source

November 17, 2025

Le Défi : Modéliser le Trafic Routier

Le monde réel :

- Des millions de véhicules
- Comportements complexes (accélération, freinage)
- Interactions entre différents types de véhicules (voitures, motos)
- Formation d'embouteillages, ondes de choc

La question :

- Comment prédire l'évolution du trafic ?
- Comment comprendre la formation des congestions ?

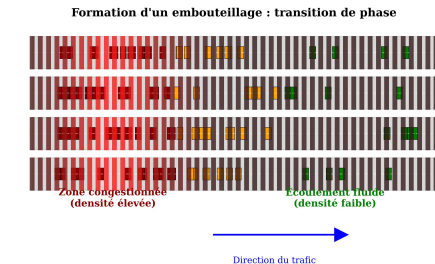


Image conceptuelle

L'Approche : Le Trafic comme un Fluide

Au lieu de suivre chaque véhicule (approche **microscopique**), nous traitons le trafic comme un fluide continu (approche **macroscopique**).

Variables Clés :

- $\rho(x, t)$: Densité de véhicules (veh/km)
- $v(x, t)$: Vitesse moyenne des véhicules (km/h)

L'Objectif :

- Décrire l'évolution de ρ et v dans l'espace (x) et le temps (t) à l'aide d'équations mathématiques.

fluid_flow.jpg

Le Modèle ARZ pour Deux Classes de Véhicules

Le code implémente un modèle de type Aw-Rascle-Zhang (ARZ) pour deux classes : motos (m) et voitures (c).

Le système d'Équations aux Dérivées Partielles (EDP) :

$$\begin{aligned}\frac{\partial \rho_m}{\partial t} + \frac{\partial(\rho_m v_m)}{\partial x} &= 0 \\ \frac{\partial \rho_c}{\partial t} + \frac{\partial(\rho_c v_c)}{\partial x} &= 0 \\ \frac{\partial w_m}{\partial t} + v_m \frac{\partial w_m}{\partial x} &= \frac{V_e(\rho_m, \rho_c) - v_m}{\tau_m} \\ \frac{\partial w_c}{\partial t} + v_c \frac{\partial w_c}{\partial x} &= \frac{V_e(\rho_m, \rho_c) - v_c}{\tau_c}\end{aligned}$$

Variables :

- ρ_m, ρ_c : Densités (motos/voitures)
- w_m, w_c : Variables de momentum. $w = v + p(\rho)$
- v_m, v_c : Vitesses réelles
- V_e : Vitesse à l'équilibre

1. Conservation de la Masse (Densité)

$$\underbrace{\frac{\partial \rho}{\partial t}}_{\text{Variation de la densité dans le temps}} + \underbrace{\frac{\partial(\rho v)}{\partial x}}_{\text{Variation du flux dans l'espace}} = 0$$

- C'est une loi de conservation fondamentale.
- **Intuition** : Le nombre de véhicules ne change pas, ils ne font que se déplacer. Si plus de véhicules entrent dans une zone qu'il n'en sort, la densité augmente.
- Le terme ρv est le **flux** de véhicules.

2. Équation de "Momentum"

$$\frac{\partial w}{\partial t} + v \frac{\partial w}{\partial x} = \frac{V_e - v}{\tau}$$

- Décrit comment la vitesse évolue.
- Le terme de gauche indique que l'information de vitesse w se propage avec la vitesse du trafic v .
- Le terme de droite est une **source de relaxation** : les conducteurs ajustent leur vitesse v pour tendre vers une vitesse désirée V_e sur un temps caractéristique τ .

Qu'est-ce que la variable $w = v + p(\rho)$?

- v est la vitesse physique du véhicule.
- $p(\rho)$ est un terme de "pression" qui modélise l'anticipation des conducteurs.

Variable de moment (suite)

- **Intuition** : Les conducteurs ne réagissent pas seulement à la densité locale, mais anticipent la densité devant eux.
- C'est la caractéristique principale des modèles ARZ.

Qu'est-ce que la vitesse à l'équilibre $V_e(\rho)$?

- C'est la vitesse que les conducteurs adopteraient dans des conditions de trafic stables.
- Elle dépend de la densité totale : plus il y a de monde, plus V_e est faible.
- Une forme typique est :

$$V_e(\rho) = V_{max} \left(1 - \frac{\rho_{total}}{\rho_{jam}} \right)$$

- V_{max} : vitesse maximale (route vide)
- ρ_{jam} : densité maximale (embouteillage complet)

Le Passage du Continu au Discret

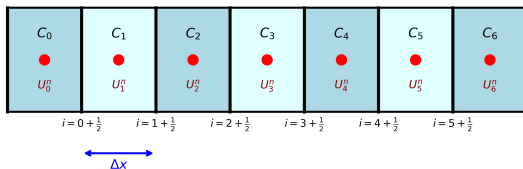
Les équations que nous avons vues sont **continues** (définies en tout point x et t). Un ordinateur ne peut pas les gérer directement.

Solution : La Discrétisation

- 1 **Discrétisation Spatiale** : On découpe la route en segments (cellules) de taille Δx .
- 2 **Discrétisation Temporelle** : On avance dans le temps par petits pas de durée Δt .

Discrétisation Spatiale : Grille de Volumes Finis

Chaque cellule stocke une valeur moyenne



On ne calcule plus la solution exacte partout, mais une **approximation** de

Le cœur de la résolution numérique est la méthode des volumes finis. On transforme l'EDP en une simple mise à jour :

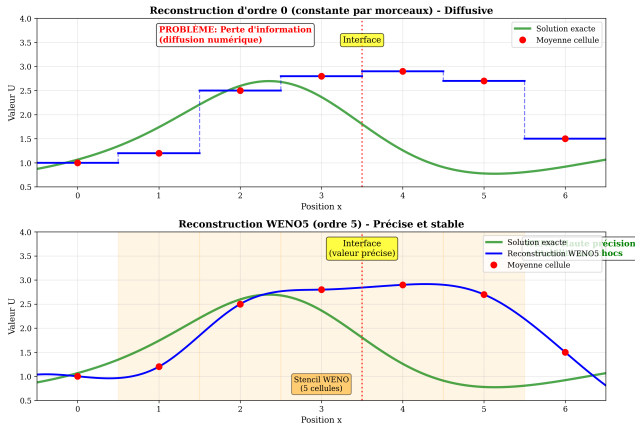
$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x} (F_{i+1/2} - F_{i-1/2})$$

- U_i^n : La valeur moyenne dans la cellule i au temps n .
- U_i^{n+1} : La nouvelle valeur que l'on cherche à calculer.
- $F_{i+1/2}$: Le **flux numérique** à l'interface entre la cellule i et $i + 1$.

Le problème est maintenant réduit à une seule question : Comment calculer le flux F aux interfaces ?

Étape 1 : Reconstruction Spatiale (WENO) - 1/3

À l'intérieur d'une cellule, nous n'avons qu'une valeur moyenne. Mais pour calculer le flux à l'interface, nous avons besoin des valeurs **exactement sur le bord** de la cellule.



Étape 1 : Reconstruction Spatiale (WENO) - 2/3

Le problème :

- Une reconstruction simple (linéaire) crée des oscillations près des chocs (embouteillages).
- Une reconstruction trop simple (constante) "bave" les chocs et les rend flous (diffusion numérique).

Étape 1 : Reconstruction Spatiale (WENO) - 3/3

La solution : WENO (Weighted Essentially Non-Oscillatory)

- Méthode de reconstruction d'ordre élevé (précise).
- Utilise plusieurs "stencils" (groupes de cellules voisines) pour deviner la valeur au bord.
- Attribue des poids à chaque stencil : si un stencil contient un choc, son poids devient quasi nul.
- **Résultat** : Une reconstruction très précise dans les zones fluides, et stable (sans oscillations) près des chocs.

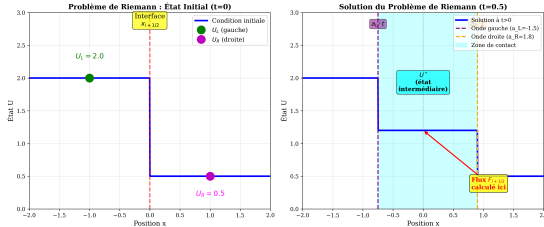
Avantages de WENO5 :

- Ordre 5 de précision en zone lisse
- Pas d'oscillations parasites près des discontinuités
- Préserve la monotonie de la solution

Étape 2 : Le Problème de Riemann à l'Interface - 1/2

Grâce à WENO, nous avons maintenant deux valeurs à chaque interface :

- U_L (reconstruite depuis la cellule de gauche)
- U_R (reconstruite depuis la cellule de droite)



Le Problème de Riemann : C'est un mini-problème mathématique qui consiste à trouver la solution exacte qui résulte de la "collision" de ces deux états U_L et U_R . La solution nous donne le flux $F_{i+1/2}$ que nous cherchions.

La solution du code : Solveur de Riemann "Central-Upwind"

- Un solveur approximatif, mais robuste et efficace.
- Il estime la vitesse de propagation des ondes vers la gauche (a_L) et vers la droite (a_R).
- Il calcule un flux qui est une moyenne pondérée des flux de U_L et U_R , stabilisée par un terme de dissipation basé sur a_L et a_R .

Étape 3 : Intégration Temporelle (SSP-RK3) - 1/2

Maintenant que nous savons calculer la divergence des flux, nous pouvons mettre à jour la solution.

L'équation est devenue une Équation Différentielle Ordinaire (EDO) en temps :

$$\frac{dU_i}{dt} = L(U_i) \quad \text{où} \quad L(U_i) = -\frac{1}{\Delta x} (F_{i+1/2} - F_{i-1/2})$$

Le problème :

- Une simple mise à jour (méthode d'Euler) n'est pas assez stable pour les schémas d'ordre élevé comme WENO.
- Les oscillations peuvent apparaître et détruire la précision.

Étape 3 : Intégration Temporelle (SSP-RK3) - 2/2

La solution : SSP-RK3 (Strong Stability Preserving Runge-Kutta 3ème ordre)

- C'est une méthode de Runge-Kutta qui effectue 3 sous-étapes pour calculer U_i^{n+1} .
- **SSP** signifie qu'elle préserve la stabilité que WENO a établie.
- C'est le partenaire idéal pour WENO.

Les 3 étapes de SSP-RK3 :

$$U^{(1)} = U^n + \Delta t L(U^n)$$

$$U^{(2)} = \frac{3}{4}U^n + \frac{1}{4}(U^{(1)} + \Delta t L(U^{(1)}))$$

$$U^{n+1} = \frac{1}{3}U^n + \frac{2}{3}(U^{(2)} + \Delta t L(U^{(2)}))$$

La Chaîne Numérique Complète

- ➊ **Départ** : On a l'état du trafic U^n à un temps n .
- ➋ **Reconstruction (WENO)** : Pour chaque cellule, on calcule les valeurs U_L et U_R aux bords à partir des moyennes des cellules voisines.
- ➌ **Flux (Solveur de Riemann)** : À chaque interface, on résout le problème de Riemann entre U_L et U_R pour obtenir le flux numérique $F_{i+1/2}$.
- ➍ **Divergence** : Dans chaque cellule, on calcule le terme
$$L(U) = -\frac{1}{\Delta x}(F_{i+1/2} - F_{i-1/2}).$$
- ➎ **Évolution (SSP-RK3)** : On utilise ce terme $L(U)$ dans les 3 étapes du SSP-RK3 pour calculer le nouvel état U^{n+1} .
- ➏ **Répéter** : On recommence pour le pas de temps suivant.

Le tout est orchestré sur GPU pour une performance maximale, en minimisant les transferts de données entre le CPU et le GPU.

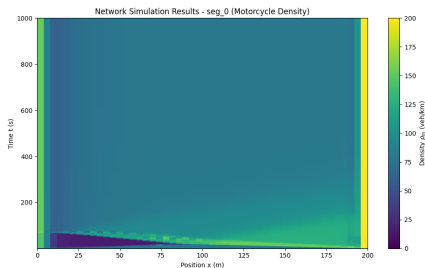
- Nous avons transformé un problème physique complexe (le trafic) en un **modèle mathématique** (système d'EDP hyperboliques).
- Ce modèle continu a été discrétisé pour être résolu par un ordinateur via la **méthode des volumes finis**.
- La précision et la stabilité sont garanties par une combinaison de méthodes numériques avancées :
 - **WENO5** pour la reconstruction spatiale.
 - Un **solveur de Riemann** pour les flux aux interfaces.
 - **SSP-RK3** pour l'intégration temporelle.
- Cette "chaîne numérique" permet de simuler l'évolution du trafic de manière robuste et précise, en capturant des phénomènes complexes comme les ondes de choc.

Scénario de Simulation : Un Embouteillage

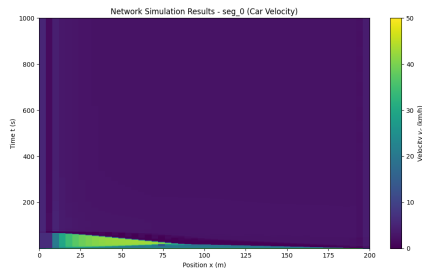
- **Objectif** : Observer la formation et la propagation d'un embouteillage sur un réseau de deux segments de route (0-200m et 200-400m).
- **Cause** : Nous avons simulé une forte densité de véhicules à la fin du premier segment (à 200m), comme un feu rouge ou un rétrécissement.
- **Observation** : Nous allons visualiser la densité des motos et la vitesse des voitures pour voir comment le système réagit.

Résultats : Segment 1 (0-200m)

Densité des Motos (ρ_m)



Vitesse des Voitures (v_c)

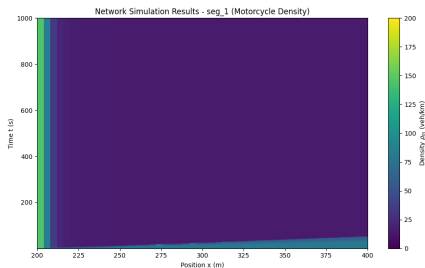


Analyse :

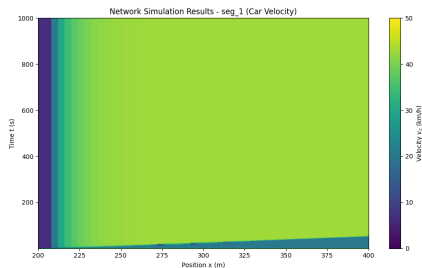
- Une **onde de choc** (embouteillage) se forme à 200m et se propage vers l'arrière (vers $x=0$).
- À *gauche* : La densité augmente (couleurs vertes/jaunes).
- À *droite* : La vitesse des voitures chute drastiquement (couleur violet foncé).

Résultats : Segment 2 (200-400m)

Densité des Motos (ρ_m)



Vitesse des Voitures (v_c)



Analyse :

- Le segment en aval de l'embouteillage reste fluide.
- À gauche : La densité reste très faible (violet), car les véhicules sont bloqués sur le segment précédent.
- À droite : La vitesse des voitures est élevée (vert/jaune), ce qui est cohérent avec un trafic fluide.

La simulation est un succès !

- **Cohérence physique** : Le modèle a correctement reproduit la dynamique d'un embouteillage. Une haute densité correspond bien à une faible vitesse.
- **Propagation d'onde** : On observe clairement la propagation de l'onde de choc vers l'amont, ce qui est un phénomène bien connu en trafic routier.
- **Robustesse du code** : La chaîne numérique, de la discrétisation à la résolution sur GPU, a fonctionné de manière stable et a produit des résultats physiquement réalistes.
- **Validation du modèle** : Ces résultats valident que notre implémentation du modèle ARZ est capable de capturer des phénomènes complexes et pertinents pour l'étude du trafic.