Introduction à la Recherche en Laboratoire

Solutions de viscosité et schémas numériques associés

Gaétan Bahl
Encadrant : Emmanuel Maître

Grenoble INP Ensimag
Laboratoire Jean Kuntzmann – Équipe EDP

19 mai 2016

Sommaire

Sujet de l'IRL

- Comprendre la notion de solution de viscosité d'équation aux dérivées partielles
- Trouver dans la littérature un problème à résoudre numériquement
- Implémenter la résolution dans un langage de programmation au choix

Sommaire

Sommaire

Équations aux Dérivées Partielles

Équation aux Dérivées Partielles

Une équation aux dérivées partielles d'ordre k est une équation liant une fonction et ses dérivées jusqu'à l'ordre k. Par exemple :

$$k = 2,\ F(x, u, Du, D^2u) = 0, x \in \Omega \subset \mathbb{R}^n$$

où $F : \Omega \times \mathbb{R} \times \mathbb{R}^n \times \mathcal{M}_{n,n} \to \mathbb{R}$.

Exemple d'EDP : Équation de la chaleur

$$\frac{\partial u}{\partial t} - \Delta u = 0$$

Solution classique

Solution classique d'une EDP

Une fonction $u : \Omega \to \mathbb{R}^n$ est appelée *solution classique* d'une EDP d'ordre k si elle est $\mathcal{C}^k$ et satisfait l'EDP $\forall x \in \Omega$.

Problème des solutions classiques : Équation Eikonale

Toutes les EDP n'admettent pas nécessairement une solution classique, par exemple :

$$|Du| = 1$$

Sommaire

Sous-Différentielle

Sous-gradient

Un vecteur p est un sous-gradient de u en un point x si la droite orientée par p est tangente sous la courbe de u en x. On note l'ensemble des sous-gradients de u en x : $D^{-}u(x)$.

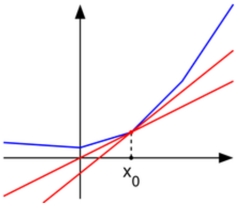

Figure – Exemple de sous-gradients

Solution de viscosité

Sous-solution de viscosité

Une fonction u est une *sous-solution de viscosité* d'une EDP si

$$F(x, u(x), p) \leq 0, \forall x \in \Omega, p \in D^+u(x)$$

Sur-solution de viscosité

Une fonction u est une *sur-solution de viscosité* d'une EDP si

$$F(x, u(x), p) \geq 0, \forall x \in \Omega, p \in D^-u(x)$$

Solution de viscosité

Une fonction u est une *solution de viscosité* si elle est à la fois une *sur-solution* et une *sous-solution*.

Sommaire

Exemple en 1D

Équation eikonale

$$|Du| = 1$$

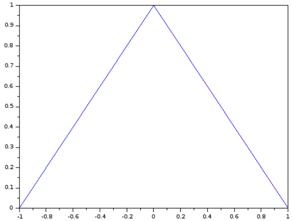

Figure – Solution de viscosité de l'équation eikonale

Sommaire

Sommaire

Problème choisi

Problème de transport optimal en 2D
Déplacement d'une mesure de probabilité
D'après un article de Benamou, Froese, Oberman.

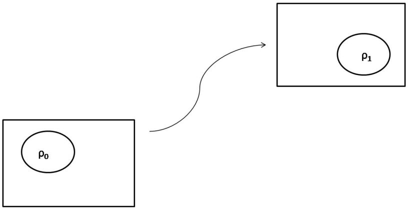

Figure – Transport optimal

Exemple

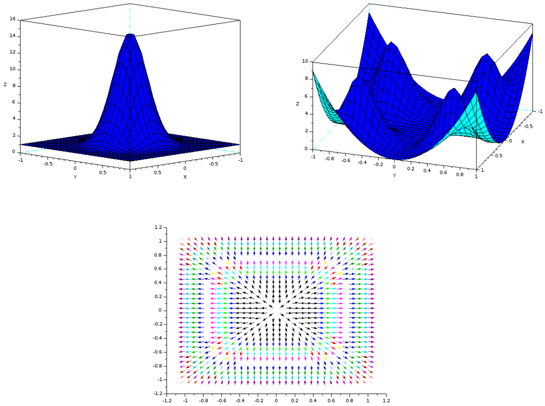

Figure – Exemple de déplacement de mesure de probabilité

Équation à résoudre

Équation de Monge-Ampère

$$\det(D^2 u) = \frac{\rho_X}{\rho_Y(\nabla u)}, \ x \in \Omega$$

Objectif : Trouver numériquement une solution de viscosité de l'équation de Monge-Ampère

Méthode utilisée

- Intérieur du domaine : Itération explicite sur l'équation de Monge-Ampère à l'aide de deux schémas aux différences finies (un schéma précis et un schéma monotone)
- Bord du domaine : Équation de Hamilton-Jacobi pour s'assurer que les bords des domaines se transportent de l'un sur l'autre

Sommaire

Travail réalisé

- Implémentation du schéma précis
- Implémentation du schéma au bord

Résultats

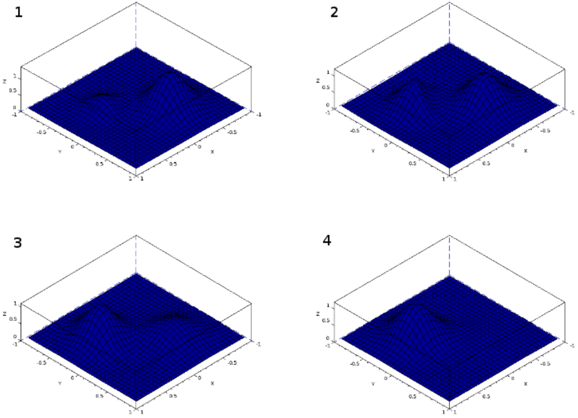

Figure – Déplacement d'une Gaussienne

Résultats

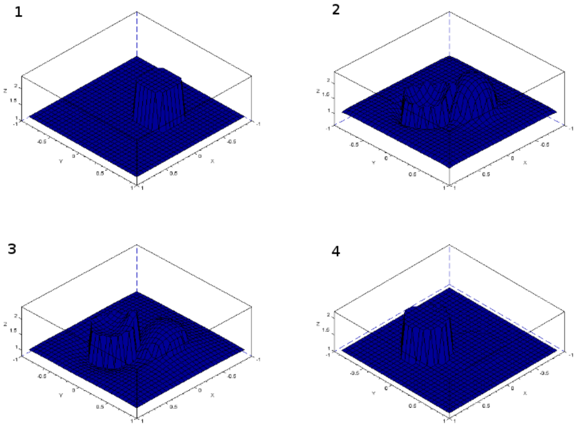

Figure – Déplacement d'un disque

Problème

Le calcul est lent :

- Utilisation d'une itération explicite
- Condition CFL très restreinte
- Donc nombre d'itérations nécessaires très élevé

Sommaire

GPU et OpenCL

- Constitué de milliers de petits processeurs
- Peuvent être programmés avec OpenCL
- Permet d'exécuter des opérations simples en parallèle

GPU et OpenCL

- Constitué de milliers de petits processeurs
- Peuvent être programmés avec OpenCL
- Permet d'exécuter des opérations simples en parallèle
- Parfait pour les différences finies !

Travail réalisé

- Implémentation du schéma précis en OpenCL
- Tests d'exécution sur un GPU intégré de marque Intel et une carte graphique AMD

Résultats

Gain de performances considérable par rapport à Scilab :

Scilab		OpenCL
1m10	$\longrightarrow$ 1000 itérations sur 30x30	0.15s
8 semaines	$\longrightarrow$ 200000 itérations sur 256x256	45s

L'itération explicite sur GPU est plus rapide que la méthode de Newton sur CPU.

Sommaire

Bilan

Travail réalisé :

- Implémentation d'une méthode trouvée dans la littérature
- Accélération considérable en faisant les calculs sur une carte graphique

Travail restant :

- Débogage du second schéma aux différences finies
- Portage du second schéma sous OpenCL
- Implémentation de la méthode de Newton

Apport personnel

- Découverte du monde de la recherche
- Application d'une démarche scientifique