

Problème Inverse : Transfert Radiatif et Apprentissage

Roussel Desmond NZOYEM

Université de Strasbourg
UFR de mathématiques et d'informatique
Master 1 CSMI

24 août 2020

Simulation 2D de l'équation du transfert radiatif et reconstruction de la densité par un réseau de neurones

Roussel Desmond NZOYEM

Enseignant referent

Christophe PRUD'HOMME

Maitre de stage

Emmanuel FRANCK

Laurent NAVORET

Vincent VIGON

Annee Academique 2019/2020

L'équipe MOCO compte plusieurs membres parmi lesquels MM. :

- Emmanuel FRANCK
- Laurent NAVORET

Responsables des séminaires en EDP

L'équipe MOCO compte plusieurs membres parmi lesquels MM. :

- Emmanuel FRANCK
- Laurent NAVORET

Responsables des séminaires en EDP

- Partenariats internationaux (Portugal, Allemagne, USA, etc.)
- Partenariats industriels
- Modélisation des plasmas

L'équipe Probabilités compte plusieurs membres parmi lesquels
M. :

- Vincent VIGON

L'équipe Probabilités compte plusieurs membres parmi lesquels
M. :

- Vincent VIGON

Des activités diverses :

- Partenariats internationaux (Allemagne, Australie, Chine, etc)
- Séminaire (de calcul) stochastique.

Probleme direct

(Resolution de l'EDP du transfer radiatif)

Probleme direct

(Resolution de l'EDP du transfert radiatif)

Probleme inverse

(Reconstruction de la densite par un ANN)

Trois point cles pour situer le stage :

- 1 Explosion du deep learning
- 2 APplications dans le secteur medical (Imagerie medicale)
- 3 Reevaluation des methode de resolution de problemes inverse

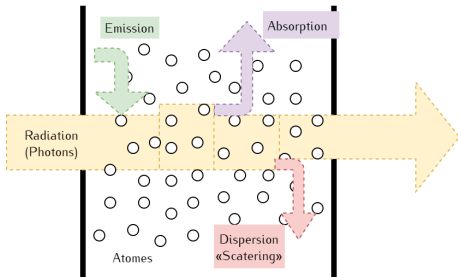
Sommaire

- 1 Introduction
 - L'IRMA
 - Le sujet du stage
- 2 Principe
 - Simulation de l'ETR
 - Réseau de neurones
- 3 Resultats 1D
 - Simulation
 - Apprentissage
- 4 Resultats 2D
 - Simulation
 - Apprentissage
- 5 Conclusion
 - Sur l'apprentissage
 - Sur le stage

Le transfer radiatif

Lorsque la photons se trouvent en presence de la matière, Trois phenomènes majeures (caratistes par leurs opacites) se produisent :

- Emission (σ_e) : Plus la temperature matiere est elevee, plus l'emission est importante
- Absorption (σ_a) : Lorsqu'on est a l'equilibre thermique, $\sigma_a = \sigma_e$
- Scattering (σ_c) : Il faut aussi tenir compte de la fonction de distribution angulaire de « scattering » $p(\Omega' \rightarrow \Omega)$.



L'ETR

L'équation du transfert radiatif est bilan d'énergie lié au rayonnement au niveau mesoscopique.

$$\begin{aligned} \frac{1}{c} \frac{\partial}{\partial t} I(t, \mathbf{x}, \boldsymbol{\Omega}, \nu) + \boldsymbol{\Omega} \cdot \nabla_{\mathbf{x}} I(t, \mathbf{x}, \boldsymbol{\Omega}, \nu) \\ = \sigma_a(\rho, \boldsymbol{\Omega}, \nu) (B(\nu, T) - I(t, \mathbf{x}, \boldsymbol{\Omega}, \nu)) \\ + \frac{1}{4\pi} \int_0^\infty \int_{S^2} \sigma_c(\rho, \boldsymbol{\Omega}, \nu) p(\boldsymbol{\Omega}' \rightarrow \boldsymbol{\Omega}) \left(I(t, \mathbf{x}, \boldsymbol{\Omega}', \nu) - I(t, \mathbf{x}, \boldsymbol{\Omega}, \nu) \right) d\boldsymbol{\Omega}' d\nu \end{aligned}$$

Où

- $I(t, \mathbf{x}, \boldsymbol{\Omega}, \nu)$ désigne l'intensité radiative spécifique ;
- $B(\nu, T)$ la fonction de Planck ;
- $\oint p(\boldsymbol{\Omega}' \rightarrow \boldsymbol{\Omega}) d\boldsymbol{\Omega}' = 1$

Le modele P1

$$\begin{cases} \partial_t E + c \operatorname{div} \mathbf{F} = c\sigma_a (aT^4 - E) \\ \partial_t \mathbf{F} + c \nabla E = -c\sigma_c \mathbf{F} \\ \rho C_v \partial_t T = c\sigma_a (E - aT^4) \end{cases}$$

Ou :

$$E(t, \mathbf{x}) = \frac{4\pi}{c} \int_0^\infty \int_{S^2} I(t, \mathbf{x}, \boldsymbol{\Omega}, \nu) d\boldsymbol{\Omega} d\nu$$

$$\mathbf{F}(t, \mathbf{x}) = \frac{4\pi}{c} \int_0^\infty \int_{S^2} \boldsymbol{\Omega} I(t, \mathbf{x}, \boldsymbol{\Omega}, \nu) d\boldsymbol{\Omega} d\nu$$

Le schema de « splitting » : Etape 1

On pose $\Theta = aT^4$

$$\begin{cases} E_j^{q+1} = \frac{\alpha E_j^n + \beta \gamma \Theta_j^n}{1 - \beta \delta} \\ \Theta_j^{q+1} = \frac{\gamma \Theta_j^n + \alpha \delta E_j^n}{1 - \beta \delta} \end{cases}$$

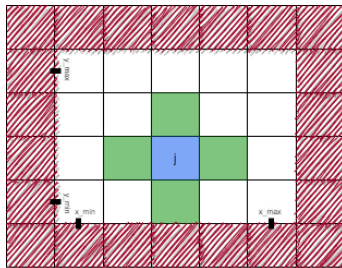
En posant

$$\mu_q = \frac{1}{T^{3,n} + T^n T^{2,q} + T^q T^{2,n} + T^{3,q}}$$

On a

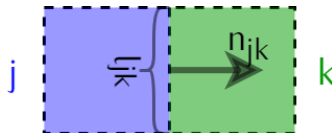
$$\alpha = \frac{1}{\Delta t \left(\frac{1}{\Delta t} + c\sigma_a \right)}, \quad \beta = \frac{c\sigma_a}{\frac{1}{\Delta t} + c\sigma_a}, \quad \gamma = \frac{\rho_j C_v \mu_q}{\Delta t \left(\frac{\rho_j C_v \mu_q}{\Delta t} + c\sigma_a \right)} \quad \text{et} \quad \delta = \frac{c\sigma_a}{\frac{\rho_j C_v \mu_q}{\Delta t} + c\sigma_a}.$$

Convergence vers E_j^* et Θ_j^* . F_j reste constant égale à F_j^* .



Le schema de « splitting » : Etape 2

$$\begin{cases} E_j^{n+1} = E_j^* + \alpha \sum_k (\mathbf{F}_{jk}, \mathbf{n}_{jk}) \\ \mathbf{F}_j^{n+1} = \beta \mathbf{F}_j^* + \gamma E_j^n + \delta \sum_k E_{jk} \mathbf{n}_{jk} \end{cases}$$



Avec :

$$\alpha = -\frac{c\Delta t}{|\Omega_j|},$$

$$\beta = \frac{1}{\Delta t} \left(\frac{1}{\Delta t} + c \sum_k M_{jk} \sigma_{jk} \right)^{-1},$$

$$\gamma = \frac{c}{|\Omega_j|} \left(\frac{1}{\Delta t} + c \sum_k M_{jk} \sigma_{jk} \right)^{-1} \left(\sum_k l_{jk} M_{jk} \mathbf{n}_{jk} \right)$$

$$\delta = -\frac{c}{|\Omega_j|} \left(\frac{1}{\Delta t} + c \sum_k M_{jk} \sigma_{jk} \right)^{-1}$$

$$(\mathbf{F}_{jk}, \mathbf{n}_{jk}) = l_{jk} M_{jk} \left(\frac{\mathbf{F}_j^n \cdot \mathbf{n}_{jk} + \mathbf{F}_k^n \cdot \mathbf{n}_{jk}}{2} - \frac{E_k^n - E_j^n}{2} \right)$$

$$E_{jk} \mathbf{n}_{jk} = l_{jk} M_{jk} \left(\frac{E_j^n + E_k^n}{2} - \frac{\mathbf{F}_k^n \cdot \mathbf{n}_{jk} - \mathbf{F}_j^n \cdot \mathbf{n}_{jk}}{2} \right) \mathbf{n}_{jk}$$

$$M_{jk} = \frac{2}{2 + \Delta x \sigma_{jk}}$$

$$\sigma_{jk} = \frac{1}{2} \left(\sigma_c(\rho_j, T_j^n) + \sigma_c(\rho_k, T_k^n) \right)$$

Implementation C++

- Temps final = 0.01 sh
- $c = 299$ [cm/sh]
- $a = 0.01372$ [g/cm/sh²/keV]
- $C_v = 0.14361$ [Jerk/g/keV]
- La densité ρ est un signal créneau [g cm⁻³]
- $\sigma_a = \rho T$ [cm⁻¹]
- $\sigma_c = \rho T$ [cm⁻¹]
- $T_0, T_{gauche} = 5$ [keV]
- $E_0 = aT_0^4$ [g/cm/sh²]
- $E_{gauche*} = aT_0^4 + 5 \sin(2k\pi t)$ [g/cm/sh²]
- $F_0, F_{gauche} = 0$ [g/sh²]
- Sorties libres sur les autres bords

```
x_min 0
x_max 1
y_min 0
y_max 1
N 90

c 299
a 0.01372
C_v 0.14361

CFL 0.5
precision 1e-6
t_0 0
t_f 0.01

rho crenau(0.5,0.5,0.1,10)
sigma_a rho*T
sigma_c rho*T

E_0 0.01372*(5^4)
F_0_x 0
F_0_y 0
T_0 5
```

```
E_u neumann
F_u_x neumann
F_u_y neumann
T_u neumann

E_d neumann
F_d_x neumann
F_d_y neumann
T_d neumann

E_l ponctuel(0.4,0.6)
F_l_x 0
F_l_y 0
T_l 5

E_r neumann
F_r_x neumann
F_r_y neumann
T_r neumann

export_file data/df_simu.csv
export_mode dataframe
write_mode truncate
```


L'architecture sous Keras

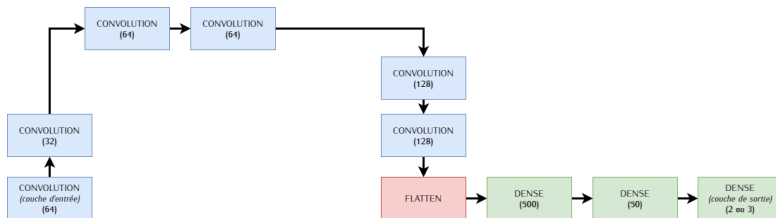


FIGURE – Architecture generale utilisee

Les couches utilisees : Convolutions (Cross-correlation)

$$S(i) = (I * K)(i) = \sum_m I(i+m)K(m)$$

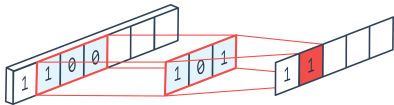


FIGURE – En 1D (GANESH, 2019)

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i+m, j+n) K(m, n)$$

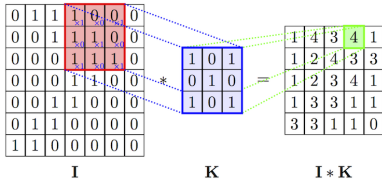


FIGURE – En 2D (PACKT, s.d)

Les couches utilisees : Flatten et Dense

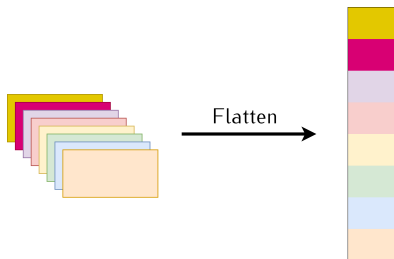


FIGURE – Flatten

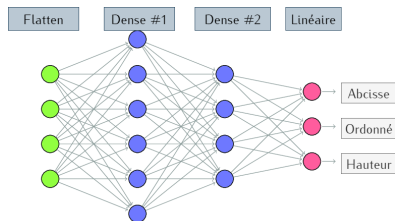


FIGURE – Des couches denses

Les metriques

Coefficient de determination

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Avec

$$SS_{res} = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$SS_{tot} = \sum_{i=1}^n (y_i - \bar{y})^2$$

Ou $\bar{y} = \sum_{i=1}^n y_i$ représente la moyenne des valeurs observées.

Score personlise

Pourcentage des prédiction correcte si la prediction et le label sont suffsament proche :

- **au dixième près** pour la position (suivant x ou y)
- **à l'unité près** pour la hauteur

Les hyper-parametres

TABLE – Liste des paramètres les plus influents pour l'entrainement

Hyper-paramètre	Définition	Valeur 1D / 2D
learning rate	taux d'apprentissage	1e-4 / 1e-5
batch size	taille d'un batch a chaque epoque	32
optimizer	algorithme d'optimisation	Adam
activation	type de fonction d'activation	relu, linear, sigmoid
patience	patience pour l'early stopping	10
epochs	nombre d'époques	100
kernel size	taille du noyau de convolution	3 / (6,2)

Le probleme en 1D

Le probleme en 2D

(La 1D est facile si on veut etudier juste la hauteur) la classification permet de savoir s'il y a un obstacle a un endroit specifique La regression 2D permet de tout savoir



GANESH, Prakhar (oct. 2019). « Types of Convolution Kernels : Simplified ». In :
URL : <https://towardsdatascience.com/types-of-convolution-kernels-simplified-f040cb307c37>.



PACKT (s.d). « Convolutional neural networks ». In : URL :
https://subscription.packtpub.com/book/game_development/9781789138139/4/ch04lvl1sec31/convolutional-neural-networks.