

# Identification de paramètres pour la méthode de Boltzmann sur réseau

Mohamed-Mahdi TEKITEK

*Université Paris-Sud*

Orsay, 16 Juin 2004

## Plan de l'exposé

1. Méthode de Boltzmann sur réseau.
2. Problème inverse.
3. Résultats numériques.
4. Conclusions.

## Boltzmann sur réseau

Les problèmes de mécanique des fluides :

- Méthodes directes de résolution :  
discrétisation spatiale et temporelle des équations d'évolution macroscopiques. (Euler, Navier-Stokes...)
- Une approche différente :  
simulation direct, au niveau "microscopique" de l'évolution des "molécules".

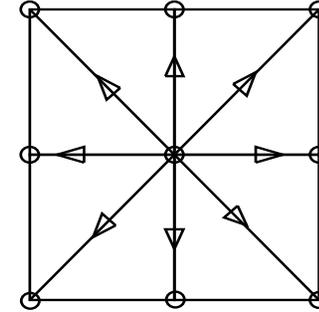
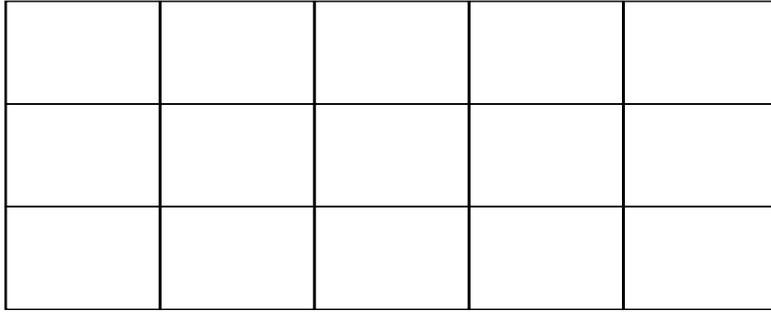
Boltzmann sur réseau :

- Une extension des modèles de gaz booléen sur réseau ( Harddy, Pazzis, Pomeau 1973).
- Raisonner sur des grandeurs moyennées qui constituent des distributions, ou des probabilités de présence de particules.

Modèle particulier de Boltzmann sur réseau en 2-D à 9-vitesses :

P.Lallemand, Li-Shi Luo : *Physical Review E*, 2000

- Espace discrétisé par un maillage des carrés réguliers.
- Espace des phases, discrétisé en 9-vitesses discrètes.



L'équation de Boltzmann discrète sur réseau est donnée par :

$$f_\alpha(\mathbf{x}_i + \mathbf{e}_\alpha \delta t, t + \delta t) = f_\alpha(\mathbf{x}_i, t) + \Omega_\alpha(f)$$

avec

$f_\alpha(\mathbf{x}_i, t) = f(\mathbf{x}_i, \mathbf{e}_\alpha, t)$  fonction de distribution.

$\mathbf{e}_\alpha$  vitesse discrète, avec  $\alpha = 0, 1, \dots, 8$

$\mathbf{x}_i$  sommet numero  $i$  du réseau.

$\delta t = 1$

$\Omega_\alpha(f)$  : collision

L'évolution se compose de deux étapes :

Soit  $V_f \equiv \mathbb{R}^{9*nx*ny}$

- Advection :  $A : V_f \rightarrow V_f$

Déplacement des particules vers les voisins directs

$$f_\alpha(\mathbf{x}_i + \mathbf{e}_\alpha, t + 1) = f_\alpha(\mathbf{x}_i, t)$$

$\mathbf{x}_i + \mathbf{e}_\alpha = \mathbf{x}_j$  un sommet du réseau.

Equation de transport :  $\frac{\partial f}{\partial t} + v \cdot \frac{\partial f}{\partial \mathbf{x}} = 0,$

La méthode des caractéristiques est exacte.

- Collision :  $C : V_f \rightarrow V_f$  Représentée dans l'équation d'évolution par  $\Omega_\alpha$ . Modélisateur les collision entre les particules, noté souvent  $Q(f, f)$  dans l'équation de Boltzmann.  
Cette étape est locale en espace.

On va introduire 9 moments de  $f_\alpha$  :

Ordre	Quantité	Moment
0	Densité	$\rho$
2	Energie	$e$
4	Carré de l'Energie	$\epsilon$
1	$x$ -impulsion	$\dot{j}_x$
3	$x$ -flux de chaleur	$q_x$
1	$y$ -impulsion	$\dot{j}_y$
3	$y$ -flux de chaleur	$q_y$
2	Contraintes tangentielles	$p_{xx}$
2	Contraintes normales	$p_{xy}$

Comme ces moments sont des combinaisons linéaires de  $f_\alpha$ , alors il existe une transformation matricielle  $M$  entre  $\vec{m}$  et  $\vec{f}$  telle que :

$$\vec{m} = M \cdot \vec{f}$$

avec

$$\begin{aligned} \vec{f} &= \vec{f}(\mathbf{x}_i, t) \equiv (f_0(\mathbf{x}_i, t), \dots, f_8(\mathbf{x}_i, t))^T \\ \vec{m} &= \vec{m}(\mathbf{x}_i, t) \equiv (\rho, e, \epsilon, j_x, q_x, j_y, q_y, p_{xx}, p_{xy})^T \end{aligned}$$

A l'aide de cette transformation, il est facile d'exprimer l'opérateur de collision dans l'espace des moments.

- Moments conservés lors de la collision sont masse ( $\rho$ ) et impulsion  $j = (j_x, j_y)$
- On suppose que les moments non conservés relaxent linéairement par rapport à leurs valeurs d'équilibre.

Cela revient à faire une approximation polynomiale d'une EDO satisfaite par les moments : P.Lallemand, D. d'Humière.

$$\frac{dm_i}{dt} + s_i(m_i - m_i^{eq}) = 0 \quad \text{par} \quad m_i^* = m_i - s_i(m_i - m_i^{eq})$$

avec  $m_i^*$  moment après relaxation, et  $s_i$  temps de relaxation.  
par exemple  $e^* = e - s_2(e - e^{eq})$  pour l'énergie.

Les valeurs d'équilibre des moments  $m_i^{eq}$ , sont fonctions des moments conservés (théorie cinétique des molécules de Maxwell).  
par exemple l'énergie d'équilibre  $e^{eq} = \rho + 3(m_1^2 + m_2^2)$

On a l'écriture matricielle :

$$\vec{m}^* = S [\vec{m} - \vec{m}^{eq}]$$

$$\begin{aligned} \vec{\Omega}(f) &= (\Omega_0(f), \Omega_1(f), \dots, \Omega_8(f))^T \\ &= M^{-1}S [\vec{m}(\mathbf{x}_i, t) - \vec{m}^{eq}(\mathbf{x}_i, t)] \end{aligned}$$

avec

$$S = \text{diag}(0, s_2, s_3, 0, s_5, 0, s_5, s_8, s_8)$$

Et l'équation de Boltzmann sur réseau généralisée :

$$\vec{f}(\mathbf{x}_i + \mathbf{e}_\alpha, t + 1) = \vec{f}(\mathbf{x}_i, t) + M^{-1}S [\vec{m}(\mathbf{x}_i, t) - \vec{m}^{eq}(\mathbf{x}_i, t)]$$

Etapes suivantes de l'algorithme :

- Initialisation de  $\vec{f}$
- Projeter  $\vec{f}$  sur les moments à l'aide de  $\vec{m} = M \vec{f}$ , et calculer l'état d'équilibre  $\vec{m}^{eq}$
- Collision dans l'espace des moments
- Projeter  $\vec{m}$  à l'aide de  $\vec{f} = M^{-1} \vec{m}$ .
- Etape d'advection dans l'espace des vitesses.

$$\vec{f}(\mathbf{x}_i + \mathbf{e}_\alpha, t + 1) = \vec{f}(\mathbf{x}_i, t) + M^{-1} \Delta \vec{m}$$

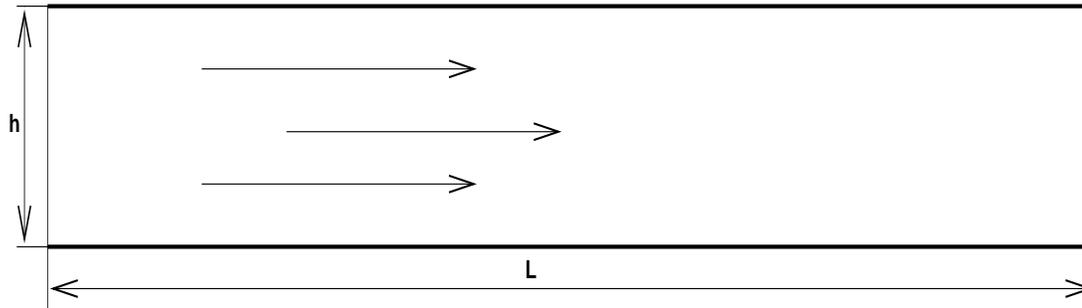
Les paramètres de relaxation  $s_i$  sont fixés par les caractéristiques hydrodynamiques du fluide. Par exemple :

$$\nu = \frac{1}{3} \left( \frac{1}{s_8} - \frac{1}{2} \right)$$

Cet algorithme nous donne l'évolution "microscopique" d'un fluide. Développement asymptotique de Chapman-Enskog, l'échelle macroscopique, on retrouve les équations de Navier-Stokes. Ainsi on va utiliser Boltzmann sur réseau au lieu de résoudre Navier-Stokes par les méthodes de volumes finis ou éléments finis... Boltzmann sur réseau :

- Performant à basse vitesse, et pas coûteux.
- Difficulté : limite Mach  $\rightarrow 1$
- Il existe un code commercial, qui est utilisé par PSA, Renault...

## Application pour un écoulement de Poiseuille :



Soit le domaine  $\Omega = [0, l] \times [0, h]$  maillé par un réseau de carrés.

Avec  $nx * ny$  le nombre de sommets du maillage (sites).

Algorithme en temps (Problème direct) :

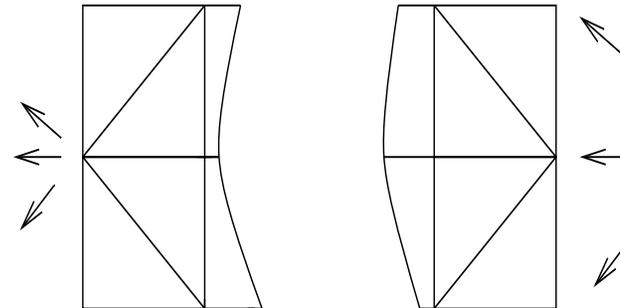
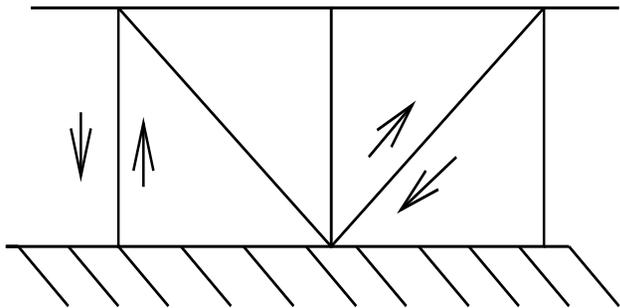
$$\begin{aligned} f^0 &= 0 \\ f^{k+1} &= \Phi(f^k) \text{ avec } k = 0, 1, \dots, N - 1 \end{aligned}$$

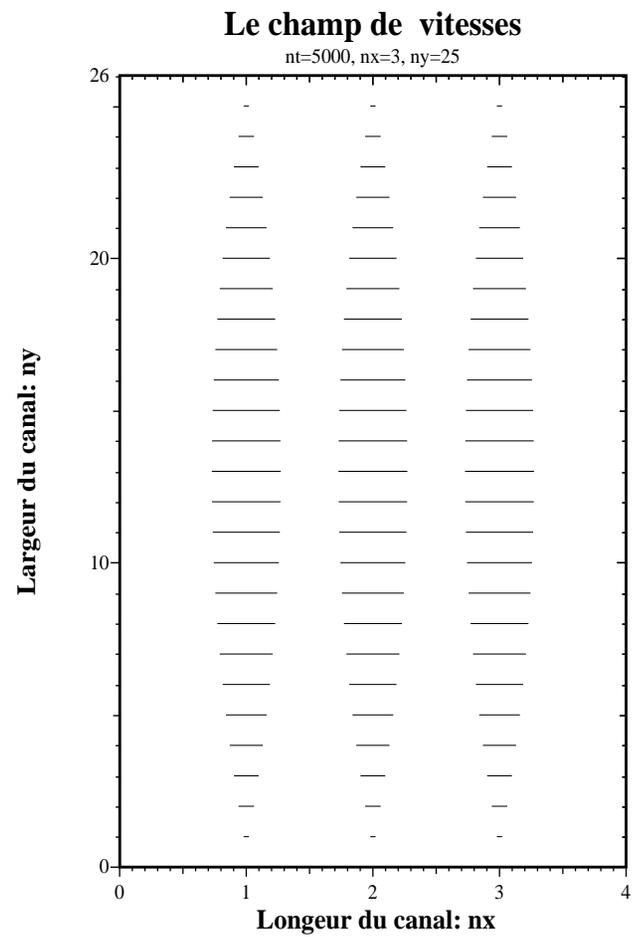
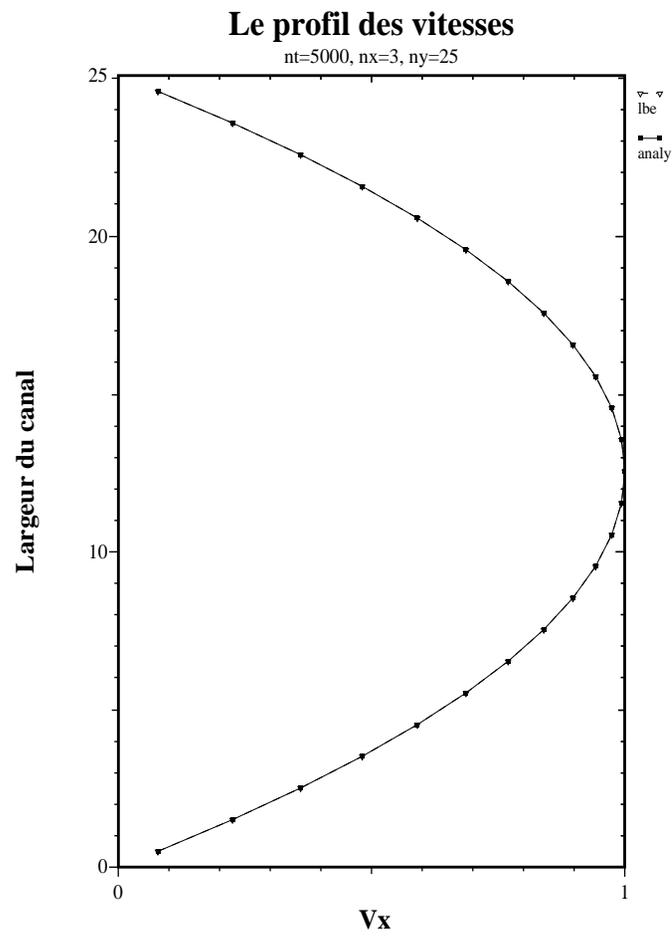
$$f^k = (f_\alpha(\mathbf{x}_i, k\delta t)) \in V_f \equiv \mathbb{R}^{9*nx*ny}$$

$\delta x = 1, \delta t = 1, N\delta t = T$  : temps ou on atteint le régime stationnaire

$$\Phi = A \circ C : V_f \rightarrow V_f$$

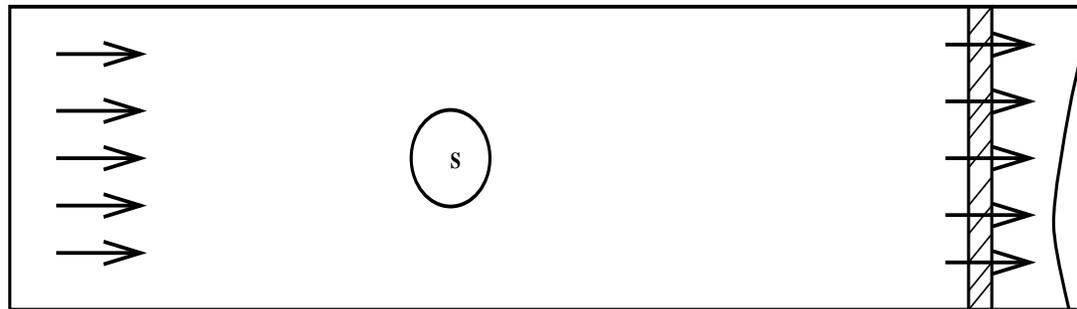
Condition aux limites (réflexion pure et périodique) : Elles sont dans l'opérateur d'advection  $A$ .





## Problème inverse :

- Déterminer des paramètres physiques inaccessibles à l'expérience.
- La détermination de paramètres du système, connaissant (une partie de) son évolution.



On se donne une surface  $S$  dans un écoulement, et on cherche à déterminer la pression autour de  $S$ , à partir des mesures de pression dans le canal.

But : Application aux souffleries.

Cas élémentaire :écoulement de Poiseuille, dont on ne connaît pas la viscosité, mais on connaît la vitesse à la sortie.

La fonctionnelle d'écart (fonction coût) entre

- la vitesse calculée par l'algorithme de Boltzmann sur réseau (pour une viscosité donnée).
- la vitesse exacte(analytique).

$$J(\nu) = \frac{1}{2} |V_{code}(\nu) - V_{ana}|^2$$

Ainsi estimer la viscosité du fluide  $\nu^*$ , revient à minimiser la fonction  $J$ . Pour calculer le bon paramètre  $\nu$ , on va utiliser un méthode de descente :

$$\lambda^{k+1} = \lambda^k - \rho \nabla J^k$$

## Algorithme de l'adjoint pour calculer le gradient :

$V_{code}(\nu)$  est la solution obtenue par la méthode de Boltzmann sur réseau, alors on peut écrire :

$$J(\nu) = \Psi(f^N, \nu)$$

Maintenant on peut voir le problème comme une minimisation sous contrainte :

- minimiser  $J(\nu)$
- sous la contrainte :  $f^{k+1} = \Phi(f^k)$

On introduit alors le Lagrangien :

$$L = J + \sum_{k=0}^{N-1} p^{k+1} (f^{k+1} - \Phi(f^k))$$

avec  $p^k = (p_\alpha(\mathbf{x}_i, k)) \in V_f$  est le multiplicateur de Lagrange.

On a alors par différentiation :

$$dL = dJ + \sum_{k=0}^{N-1} p^{k+1} \left( df^{k+1} - \frac{\partial \Phi}{\partial f} df^k - \frac{\partial \Phi}{\partial \nu} d\nu \right)$$

Comme  $J$  ne dépend pas directement de  $\nu$ , alors :

$$dJ = \frac{\partial \Psi}{\partial f} df^N$$

Intégration par parties discrète donne :

$$dL = \frac{\partial \Psi}{\partial f} df^N + p^N df^N - p^1 \frac{\partial \Phi}{\partial f} df^0 + \sum_{k=1}^{N-1} (p^k - p^{k+1}) \frac{\partial \Phi}{\partial f} df^k - \sum_{k=0}^{N-1} p^{k+1} \frac{\partial \Phi}{\partial \nu} d\nu$$

On obtient :

- Comme on ne change pas la condition initiale, alors  $df^0 = 0$ .
- On choisit  $p^N = -\frac{\partial \Psi}{\partial f}$ , pour éliminer  $df^N$ .
- Comme  $dL = dJ$ , car  $f^k$  sont solution de  $f^{k+1} = \Phi(f^k)$ , alors on impose que  $p^k$  vérifie l'équation adjointe (dynamique rétrograde) :

$$p^N = -\frac{\partial \Psi}{\partial f}$$

$$p^k = p^{k+1} \frac{\partial \Phi}{\partial f} \quad \text{pour} \quad k = N - 1, N - 2, \dots, 1$$

Ainsi le gradient exact de  $J$  est donné par :

$$dL = dJ = - \sum_{k=1}^{N-1} p^{k+1} \frac{\partial \Phi}{\partial \nu}$$

- Algorithme adjoint : Pour calculer  $p^k$ , à partir de  $p^{k+1}$  on a :  

$$p^k = p^{k+1} \frac{\partial \Phi}{\partial f} = C^T A^T p^{k+1}$$
 car  $C$  et  $A$  sont linéaires (première approche) par rapport à  $f$ .

Or l'advection transposée  $A^T$  : Transport avec vitesse opposée (*i.e*  $A^T = A^{-1}$ ), même avec les conditions aux limites.

La collision transposée  $C^T$  est un opérateur qui laisse trois grandeurs conservées.

- Le calcul du gradient : Pour calculer  $dJ$ , on a besoin de résoudre le problème direct et inverse pour une viscosité donnée et de mémoriser **tous** les  $f^k$ ,  $k = 1, N$ , et **tous** les  $p^k$ , **car**

$$dJ = J'(\nu) = - \sum_{k=1}^{N-1} p^{k+1} \frac{\partial \Phi}{\partial \nu} (f^k)$$

## Algorithme :

L'algorithme de minimisation utilisé dans la méthode descente de type gradient nécessite les étapes fondamentales suivantes :

- Etape 0 : Choisir une viscosité pour l'initialisation.
- Etape 1 : Résoudre le problème direct
- Etape 2 : Résoudre le problème inverse
- Etape 3 : Calculer le gradient exact de la fonction coût.
- Etape 4 : Mettre à jour la viscosité par  $\nu^* = \nu + \rho \nabla J$
- Etape 5 : Vérifier le critère de convergence : retourner à l'étape 0 avec la nouvelle viscosité sinon.

## Résultats numériques :

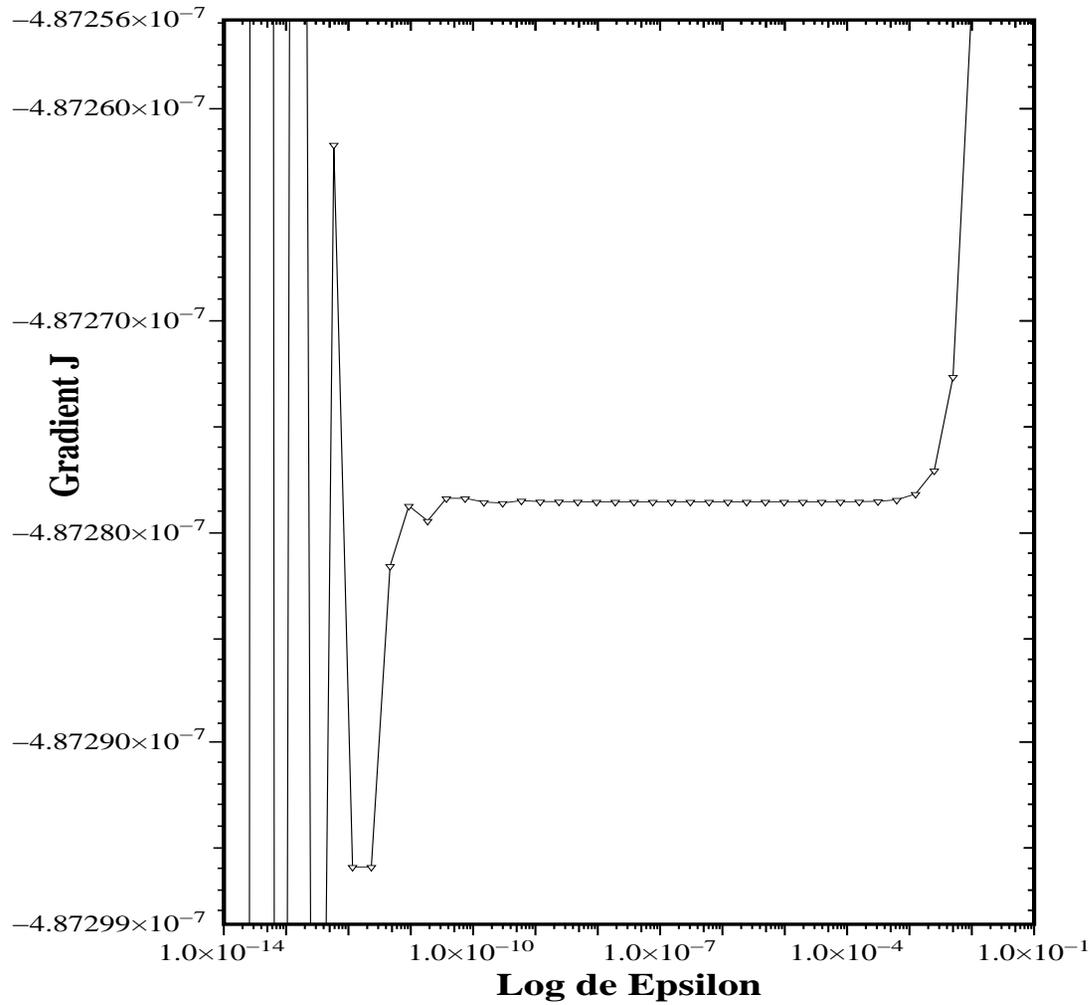
La méthode de l'adjoint, nous donne le gradient exact de la fonction  $J$ .

Comparer le gradient obtenu par le modèle adjoint et la méthode des différences finies.

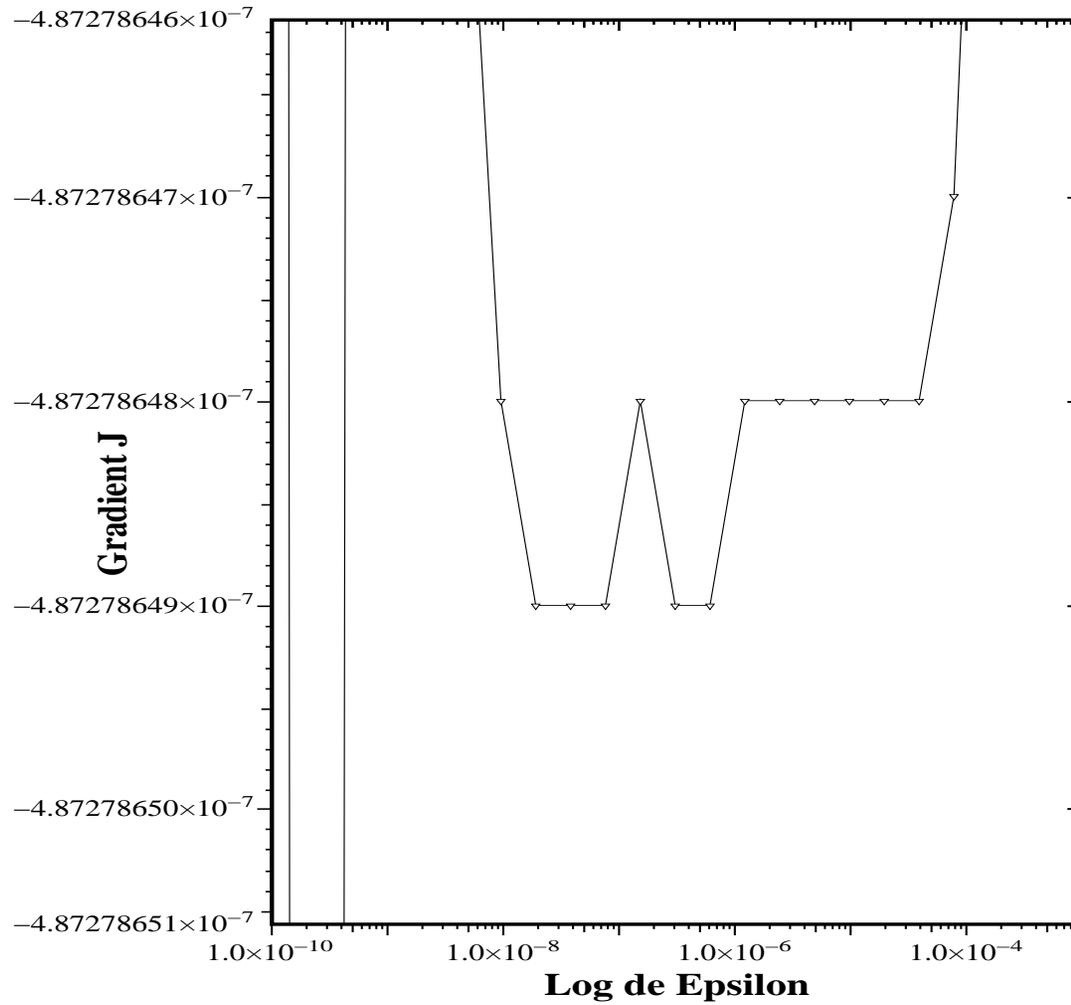
Les quantités :

- $dJ(s_8) = J'(s_8)$
- $\frac{J(s_8+\epsilon) - J(s_8-\epsilon)}{2\epsilon}$

### Gradient J en S\_8=0.3 par differences finis



### Gradient J en S\_8=0.3 par differences finis

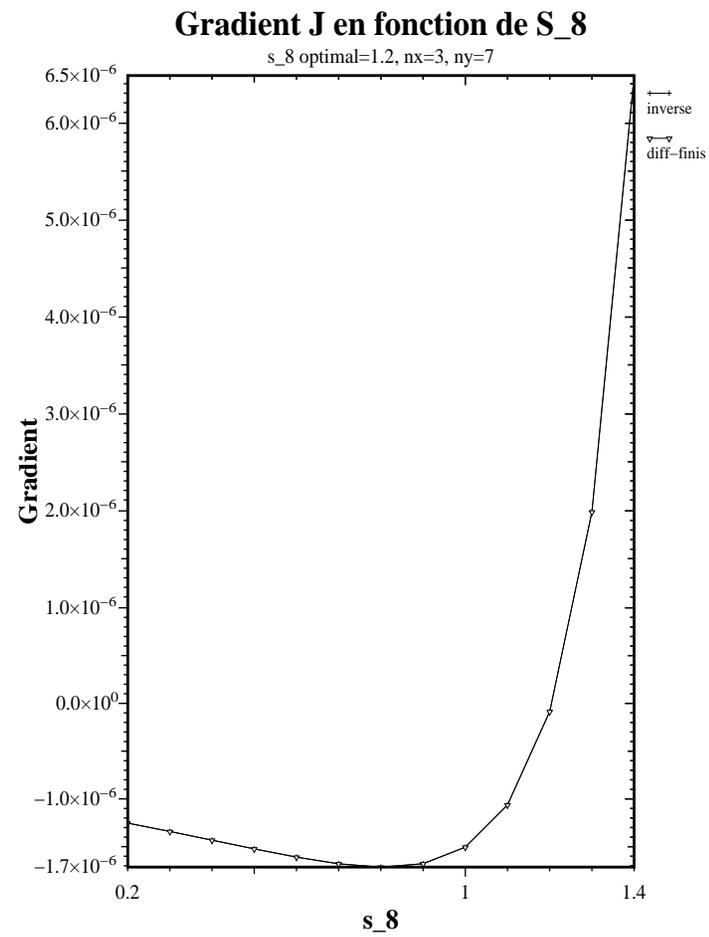
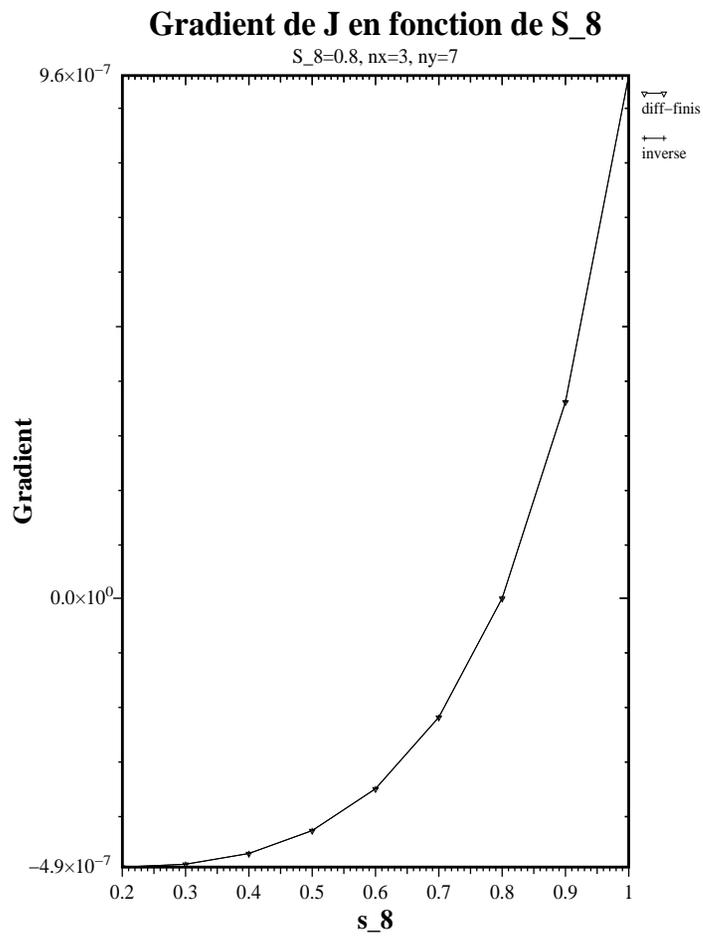


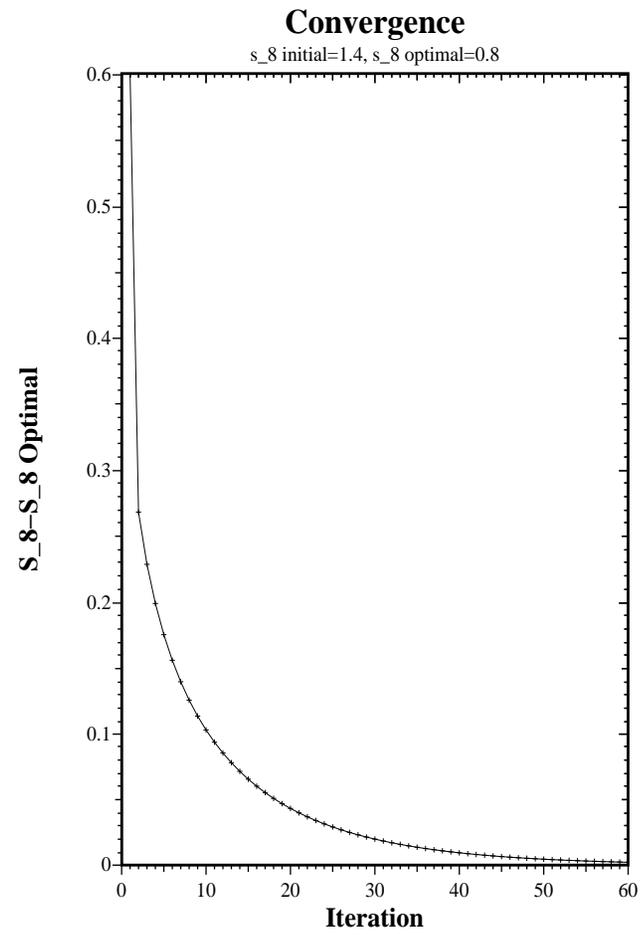
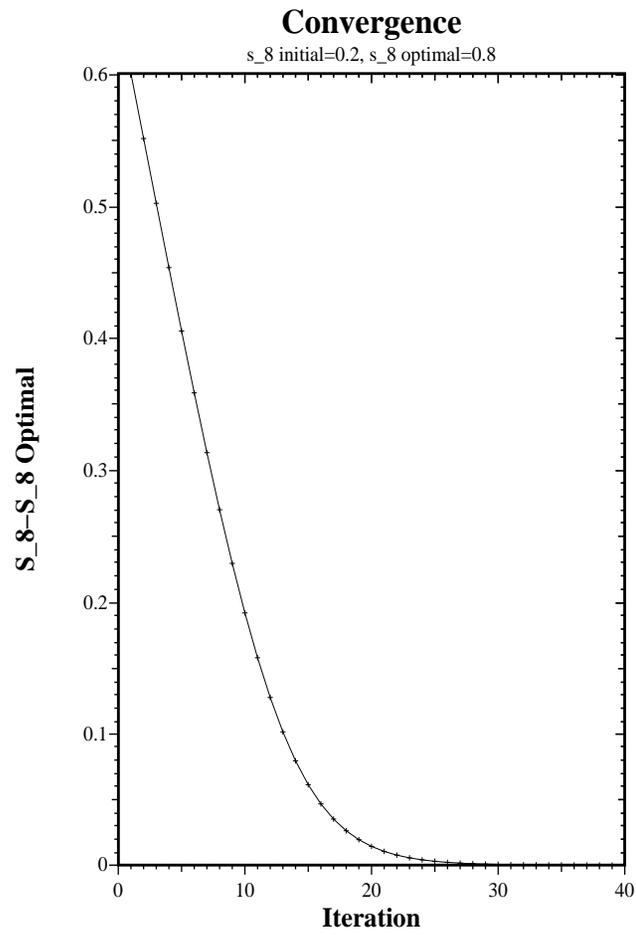
Gradient  $J$  calculé par différences finies, en  $s_8 = 0.3$  :

Epsilon	Gradient J
$1.E - 05$	$-4.87278646E - 07$
$1.E - 06$	$-4.87278648E - 07$
$1.E - 07$	$-4.87278649E - 07$
$1.E - 08$	$-4.8727865E - 07$
$1.E - 09$	$-4.87278626E - 07$

Gradient  $J$  calculé par le modèle adjoint en  $s_8 = 0.3$  :

$$-4.87278648E - 07$$





## Conclusion :

- La modélisation inverse est une bonne approche pour l'estimation des paramètres, fondamentalement basée sur les méthodes de contrôle optimal.
- Algorithme adjoint testé, et en cours de développement.
- Rendre le modèle plus physique, avec des conditions aux limites qui modélisent l'entrée et sortie du fluide.
- Etude de l'algorithme par rapport à plusieurs variables ( $s_5, s_8, \dots$ ).
- Essayer de résoudre un problème plus complexe en vue de la "coopération mesure calcul" pour les souffleries.