Guillaume PARIS
Hélène TRAN

Projet de Maths

La Magnétohydrodynamique

Contexte mathématique

1) A l'aide des équations de la mécanique des fluides et les équations de Maxwell, on obtient le système d'équations de la Magnétohydrodynamique idéale suivant :

$$\partial_t \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ Q \\ \mathbf{B} \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u} \otimes \mathbf{u} + (p + \frac{\mathbf{B}\cdot\mathbf{B}}{2})\mathbf{I} - \mathbf{B} \otimes \mathbf{B} \\ (Q + p + \frac{\mathbf{B}\cdot\mathbf{B}}{2})\mathbf{u} - (\mathbf{B}\cdot\mathbf{u})\mathbf{B} \\ \mathbf{u} \otimes \mathbf{B} - \mathbf{B} \otimes \mathbf{u} \end{pmatrix} = 0$$

2) La condition $\nabla \cdot B = 0$ doit être vérifiée pour qu'il n'y ait pas apparition de charge magnétique. Le modèle précédent étant idéal, la méthode de Dedner propose un modèle plus proche de la réalité. Elle donne ainsi le système suivant avec une inconnue ψ et une équation supplémentaire (on notera qu'en posant comme condition $\nabla \cdot B = 0$ et donc ψ constante, on retrouve le modèle idéal) :

$$\partial_t \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ Q \\ \mathbf{B} \\ \psi \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u} \otimes \mathbf{u} + (p + \frac{\mathbf{B}\cdot\mathbf{B}}{2})\mathbf{I} - \mathbf{B} \otimes \mathbf{B} \\ (Q + p + \frac{\mathbf{B}\cdot\mathbf{B}}{2})\mathbf{u} - (\mathbf{B}\cdot\mathbf{u})\mathbf{B} \\ \mathbf{u} \otimes \mathbf{B} - \mathbf{B} \otimes \mathbf{u} + \psi\mathbf{I} \\ c_h^2\mathbf{B} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

A présent, il nous faut résoudre ce dernier système à l'aide d'un programme informatique.

1) Le système de la MHD idéale peut s'écrire sous la forme d'une équation conservative

$$\partial_t \mathbf{W} + \partial_i \mathbf{F}^i(\mathbf{W}) = 0$$

avec $W = (\rho, \rho u, Q, B, \Psi)^T$ et $F(W) =$

$$\begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\otimes\mathbf{u} + (p + \frac{\mathbf{B}\cdot\mathbf{B}}{2})\mathbf{I} - \mathbf{B}\otimes\mathbf{B} \\ (Q + p + \frac{\mathbf{B}\cdot\mathbf{B}}{2})\mathbf{u} - (\mathbf{B}\cdot\mathbf{u})\mathbf{B} \\ \mathbf{u}\otimes\mathbf{B} - \mathbf{B}\otimes\mathbf{u} + \psi\mathbf{I} \\ c_h^2\mathbf{B} \end{pmatrix}$$

Le vecteur de variables conservatives est $W = (\rho, \rho u, Q, B, \Psi)^T$ et le vecteur de variables primitives $Y = (\rho, u, p, B, \Psi)^T$.

2) Nous avons codé les fonctions *conservatives* et *primitives*, prenant toutes les deux les paramètres Y[_M] et W[_M] (ce sont des vecteurs de réels). La fonction *conservatives* crée le vecteur de variables conservatives W[_M] à partir du vecteur de variables primitives Y[_M]. A l'inverse, la fonction *primitives* crée le vecteur Y[_M] à partir de W[_M].

Pour créer la fonction *primitives*, nous nous sommes aidés des équations suivantes :

$$Y_1 = W_1$$
$$Y_2 = \frac{W_2}{W_1}$$
$$Y_3 = \frac{W_3}{W_1}$$
$$Y_4 = \frac{W_4}{W_1}$$

$$Y_5 = (\gamma - 1)\left(W_5 - \frac{W_1}{2}\left(\left(\frac{W_2}{W_1}\right)^2 + \left(\frac{W_3}{W_1}\right)^2 + \left(\frac{W_4}{W_1}\right)^2 + \right) - \frac{1}{2}\left(W_6^2 + W_7^2 + W_8^2\right)\right)$$

$$Y_6 = W_6$$
$$Y_7 = W_7$$
$$Y_8 = W_8$$
$$Y_9 = W_9$$

De même, pour la fonction *conservatives*, nous avons exprimé les coordonnées W_i en fonction de Y_i.

$$W_1 = Y_1$$

$$W_2 = Y_1 \times Y_2$$

$$W_3 = Y_1 \times Y_3$$

$$W_4 = Y_1 \times Y_4$$

$$W_5 = \frac{Y_5}{\gamma - 1} + \frac{Y_1}{2}[(Y_2)^2 + (Y_3)^2 + (Y_3)^2] + \frac{1}{2}[(Y_6)^2 + (Y_7)^2 + (Y_8)^2]$$

$$W_6 = Y_6$$

$$W_7 = Y_7$$

$$W_8 = Y_8$$

$$W_9 = Y_9$$

3) Le flux numérique est défini par le vecteur **F(W,n) = F(W).n** où n est un vecteur quelconque dans R^3. On obtient alors le vecteur suivant :

$$\mathbf{F}(\mathbf{W}, \mathbf{n}) = \begin{pmatrix} \rho \mathbf{u} \cdot \mathbf{n} \\ \rho(\mathbf{u} \cdot \mathbf{n})\mathbf{u} + (p + \frac{\mathbf{B} \cdot \mathbf{B}}{2})\mathbf{n} - (\mathbf{B} \cdot \mathbf{n})\mathbf{B} \\ (Q + p + \frac{\mathbf{B} \cdot \mathbf{B}}{2})\mathbf{u} \cdot \mathbf{n} - (\mathbf{B} \cdot \mathbf{u})(\mathbf{B} \cdot \mathbf{n}) \\ (\mathbf{u} \cdot \mathbf{n})\mathbf{B} - (\mathbf{B} \cdot \mathbf{n})\mathbf{u} + \psi \mathbf{n} \\ c_h^2 \mathbf{B} \cdot \mathbf{n} \end{pmatrix}$$

La fonction *flux* prend en paramètre le vecteur W défini précédemment et le vecteur n dans R^3 et retourne le vecteur F correspondant.

Autres

1) La fonction *Ref2PhysMap* prend en argument des coordonnées $(xx, yy) \in [0,1]^2$ et retournant les coordonnées $(x, y) \in [XMIN, XMAX] \times [YMIN, YMAX]$. Pour cela, nous avons utilisé la formule suivante :

$$x = xx \times LONGUEURX + XMIN$$

$$y = yy \times LONGUEURY + YMIN$$

où $LONGUEURX = XMAX - XMIN$ et $LONGUEURY = YMAX - YMIN$.

2) La fonction *InitData* prend en paramètre un tableau Wn1 de *_NXTRANSBLOCK*_NYTRANSBLOCK*_M* éléments. Il représente un tableau bidimensionnel de vecteurs W, eux-mêmes représentant l'état du plasma dans chaque case, correspondant à des zones.

Cette fonction initialise ce tableau en utilisant la fonction *Wexact*, donnée par le professeur. Le tableau Wn1 est, en sortie de la fonction, initialisé et prêt à être actualisé dans *TimeStepCPU1D* ou *TimeStepCPU2D* suivant le cas.

3) En une dimension, la méthode des volumes finis nous donne la formule :

$$\mathbf{W}_{i,j}^{n+1} = \mathbf{W}_{i,j}^{n} - \frac{\Delta t}{\Delta x}\left(\mathbf{F}(\mathbf{W}_{i,j}^{n}, \mathbf{W}_{i+1,j}^{n}, \mathbf{n}_x) - \mathbf{F}(\mathbf{W}_{i-1,j}^{n}, \mathbf{W}_{i,j}^{n}, \mathbf{n}_x)\right)$$

Cette formule nous permet de passer à l'étape suivante. Dans cette formule, $\mathbf{F}(\mathbf{W}_L, \mathbf{W}_R, \mathbf{n})$ est défini comme le flux de Rusanov :

$$\mathbf{F}(\mathbf{W}_L, \mathbf{W}_R, \mathbf{n}) = \frac{1}{2}\left(\mathbf{F}(\mathbf{W}_L, \mathbf{n}) + \mathbf{F}(\mathbf{W}_R, \mathbf{n})\right) - \frac{\lambda_{\max}}{2}(\mathbf{W}_R - \mathbf{W}_L).$$

Dans cette formule, λ_{max} est un majorant de la valeur propre maximale du système, qui peut, dans notre cas, être fixée : $c_h = \lambda_{max}$.

Δx représente la taille d'une case, soit $\Delta x = \frac{x_{\max} - x_{\min}}{N_x}$

(Nx = _NXTRANSBLOCK dans notre cas) et Δt peut être, dans notre cas, d'ores et déjà fixé. En effet, pour minimiser le temps de calcul, nous voulons le fixer aussi grand que possible. Mais pour que le calcul soit exact, Δt doit respecter une condition : $\Delta t \leqslant \frac{\Delta x}{\lambda_{\max}}$

On fixe donc Δt à $CFL \times (\frac{\Delta x}{\lambda_{max}})$ où CFL est un réel compris entre 0 et 1 permettant d'assurer la stabilité.

La fonction *TimeStepCPU1D* prend ainsi en argument Δt et Wn1 et met à jour Wn1 d'après la formule vue au-dessus.

4) De la même manière, *TimeStepCPU2D* prend en argument Δt et Wn1 et met à jour Wn1 d'après la formule :

$$\begin{aligned}\mathbf{W}_{i,j}^{n+1} = \mathbf{W}_{i,j}^{n} &- \frac{\Delta t}{\Delta x}\left(\mathbf{F}(\mathbf{W}_{i,j}^{n}, \mathbf{W}_{i+1,j}^{n}, \mathbf{n}_x) - \mathbf{F}(\mathbf{W}_{i-1,j}^{n}, \mathbf{W}_{i,j}^{n}, \mathbf{n}_x)\right) \\ &- \frac{\Delta t}{\Delta y}\left(\mathbf{F}(\mathbf{W}_{i,j}^{n}, \mathbf{W}_{i,j+1}^{n}, \mathbf{n}_y) - \mathbf{F}(\mathbf{W}_{i,j-1}^{n}, \mathbf{W}_{i,j}^{n}, \mathbf{n}_y)\right).\end{aligned}$$

où les fonctions et variables sont définies comme à la question précédente.

5) Lorsqu'on remplace le flux de Rusanov par un schéma décentré, on prend pour F la formule :

$$\mathbf{F}(\mathbf{W}_L, \mathbf{W}_R, \mathbf{n}) = \frac{1}{2}\left(\mathbf{F}(\mathbf{W}_L, \mathbf{n}) + \mathbf{F}(\mathbf{W}_R, \mathbf{n})\right) - \frac{1}{2} V\left(\mathbf{A}(\mathbf{W}_{\text{mil}})\right)(\mathbf{w}_R - \mathbf{W}_L).$$

Où : $$\mathbf{W}_{\text{mil}} = \frac{\mathbf{W}_L + \mathbf{W}_R}{2}.$$

Et, pour une fonction polynomiale :

$$V(\mathbf{A}(\frac{\mathbf{W}_L + \mathbf{W}_R}{2}))(\mathbf{W}_R - \mathbf{W}_L)$$
$$= \frac{5}{16}(\mathbf{W}_R - \mathbf{W}_L) + \mathbf{A}(0 + \mathbf{A}(\frac{15}{16}(\mathbf{W}_R - \mathbf{W}_L) + \mathbf{A}(0$$
$$+ \mathbf{A}(-\frac{5}{16}(\mathbf{W}_R - \mathbf{A}_L) + \mathbf{A}(0 + \mathbf{A}(\mathbf{W}_R - \mathbf{W}_L)))))).$$

Où A est :

$$\mathbf{A}(\mathbf{Y}, \mathbf{n}) = \begin{pmatrix} \mathbf{u}\cdot\mathbf{n} & \rho n_1 & \rho n_2 & \rho n_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{u}\cdot\mathbf{n} & 0 & 0 & \frac{n_1}{\rho} & -\frac{\mathbf{B}\cdot\mathbf{n}}{\rho} & \tau_{2,1} & \tau_{3,1} & 0 \\ 0 & 0 & \mathbf{u}\cdot\mathbf{n} & 0 & \frac{n_2}{\rho} & \tau_{1,2} & -\frac{\mathbf{B}\cdot\mathbf{n}}{\rho} & \tau_{3,2} & 0 \\ 0 & 0 & 0 & \mathbf{u}\cdot\mathbf{n} & \frac{n_3}{\rho} & \tau_{1,3} & \tau_{2,3} & -\frac{\mathbf{B}\cdot\mathbf{n}}{\rho} & 0 \\ 0 & \gamma p n_1 & \gamma p n_2 & \gamma p n_3 & \mathbf{u}\cdot\mathbf{n} & (\gamma-1)\mathbf{B}\cdot\mathbf{u}n_1 & (\gamma-1)\mathbf{B}\cdot\mathbf{u}n_2 & (\gamma-1)\mathbf{B}\cdot\mathbf{u}n_3 & (1-\gamma)\mathbf{B}\cdot\mathbf{n} \\ 0 & \kappa_{2,3} & B^1 n_2 & B^1 n_3 & 0 & u^2 n_2 + u^3 n_3 & -u^1 n_2 & -u^1 n_3 & n_1 \\ 0 & B^2 n_1 & \kappa_{1,3} & B^2 n_3 & 0 & -u^2 n_1 & u^1 n_1 + u^3 n_3 & -u^2 n_3 & n_2 \\ 0 & B^3 n_1 & B^3 n_2 & \kappa_{1,2} & 0 & -u^3 n_1 & -u^3 n_2 & u^1 n_1 + u^2 n_2 & n_3 \\ 0 & 0 & 0 & 0 & 0 & c h^2 n_1 & c h^2 n_2 & c h^2 n_3 & 0 \end{pmatrix}$$

Où : Y peut être calculé à partir de W_{mil} et n provient de $\mathbf{F}(\mathbf{W}_L, \mathbf{W}_R, \mathbf{n})$

Pour trouver les κ (kappa) et τ (tau), il « suffit » de dériver les fonctions de F(W,n) en fonction des variables de Y. Méthode qui fonctionne très bien pour la première et les 4 dernières lignes. En revanche, les résultats des lignes 2,3,4,5 ne correspondent absolument pas. N'ayant pas accès à l'astuce permettant de trouver ces résultats, nous ne pouvons pas mettre en œuvre cette méthode. Il en va de même pour la méthode suivante.

6)

Pour une fraction rationnelle, on utilise la fraction : $\frac{1 + 3\mathbf{A}^2}{3 + \mathbf{A}^2}$

On obtient donc $$\mathbf{X} = \frac{1}{c_h} V(\mathbf{A}(\frac{\mathbf{W}_L + \mathbf{W}_R}{2}))(\mathbf{W}_R - \mathbf{W}_L).$$

Où X est solution de : $\mathbf{MX} = \mathbf{B}$

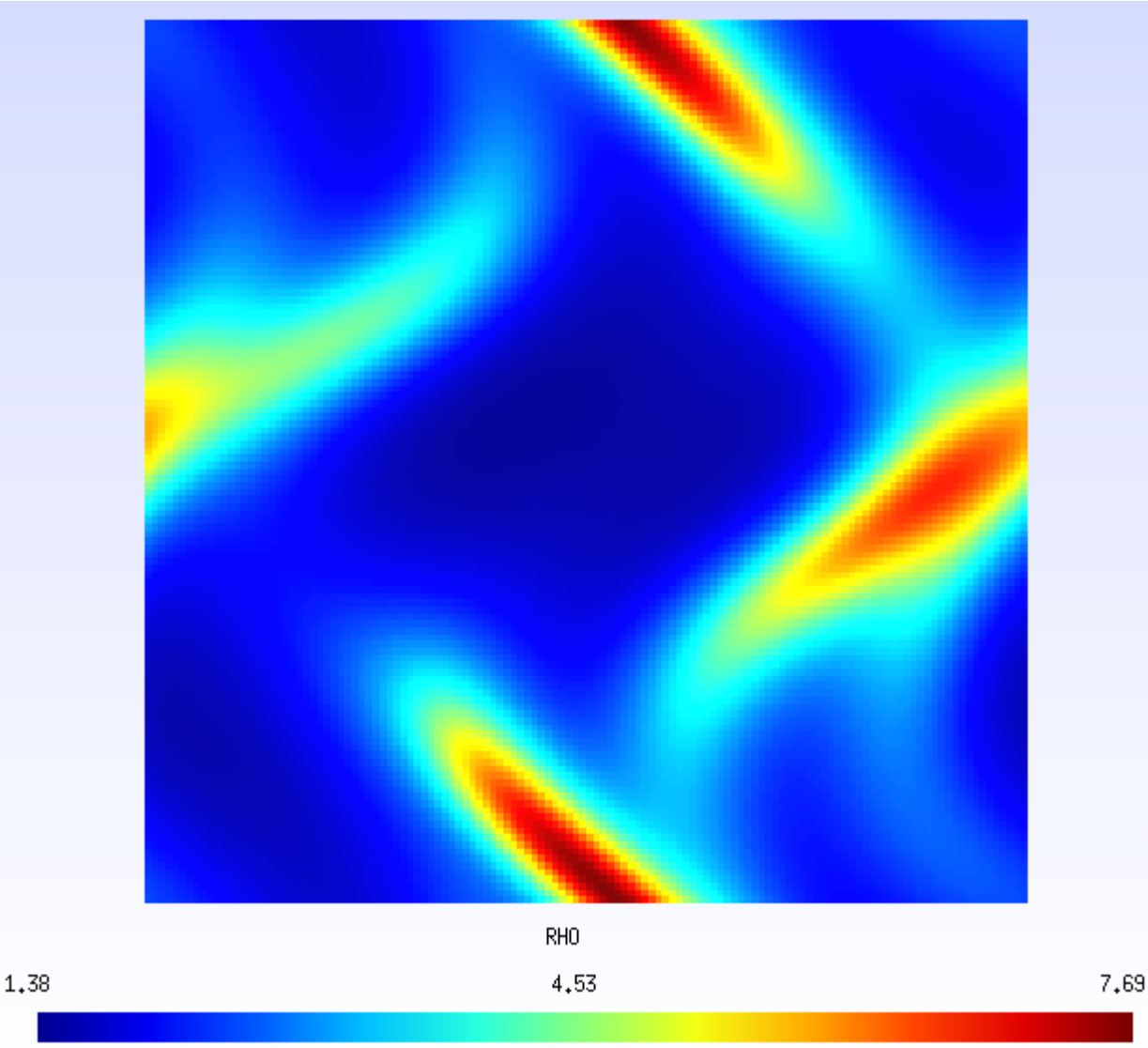
Avec : $\mathbf{M} = 3I + A^2 \left(\frac{\mathbf{W}_L + \mathbf{W}_R}{2c_h} \right)$

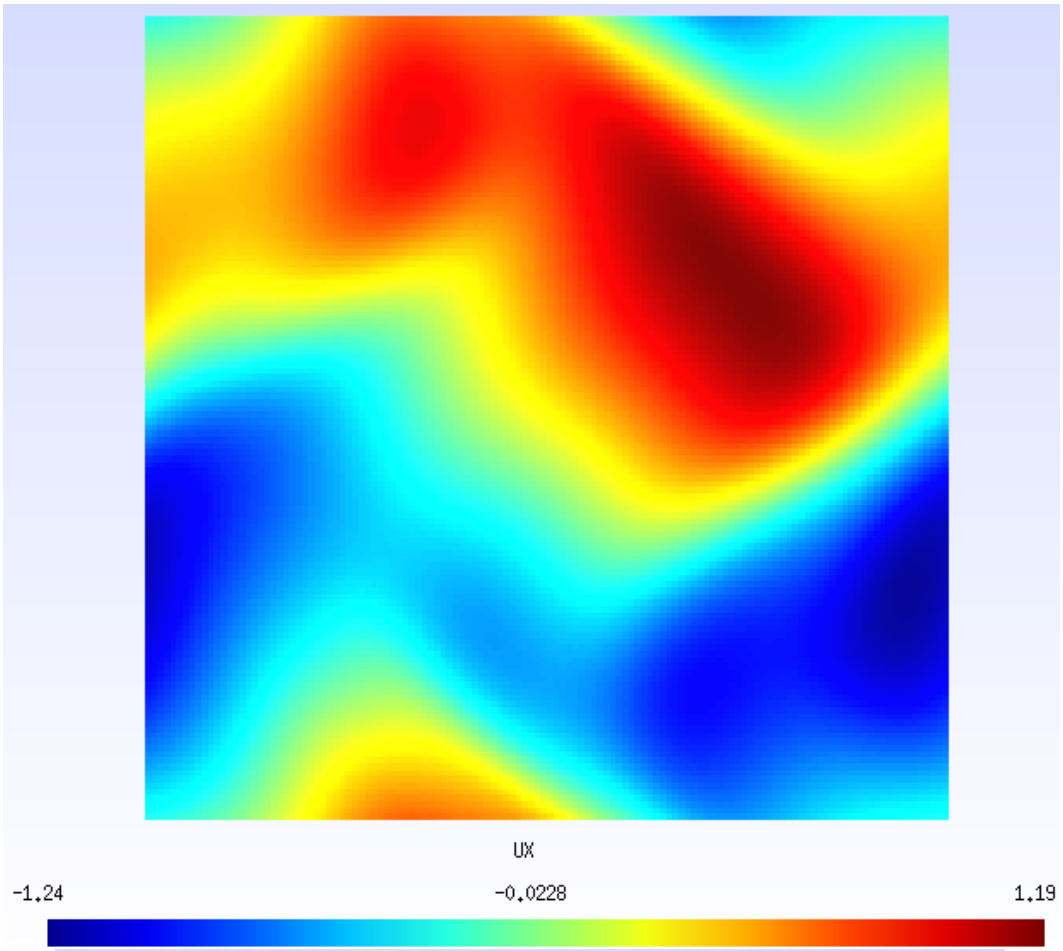
Et : $\mathbf{B} = \left(I + 3\mathbf{A}^2 \left(\frac{\mathbf{W}_L + \mathbf{W}_R}{2c_h} \right) \right) (\mathbf{W}_R - \mathbf{W}_L).$

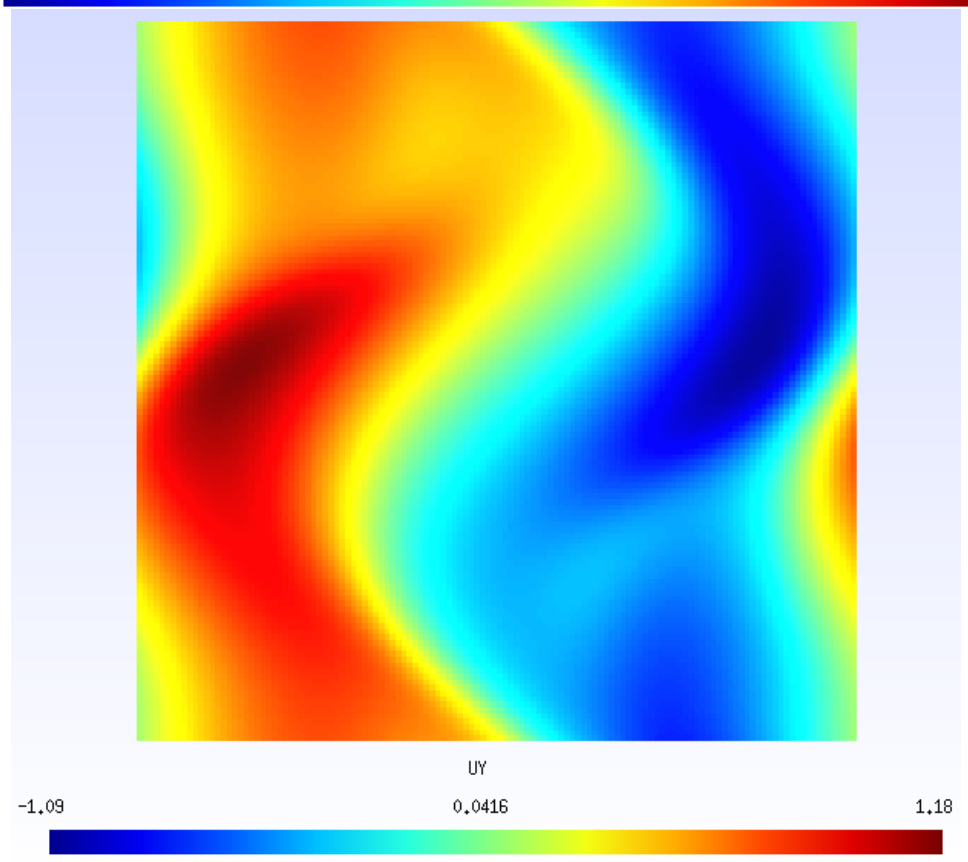
Pensant pouvoir réussir les questions bonus, nous avons programmé en parallèle la méthode de résolution d'un système linéaire de type AX=B avec la méthode LU. Le module correspondant est *matrice.cpp*.

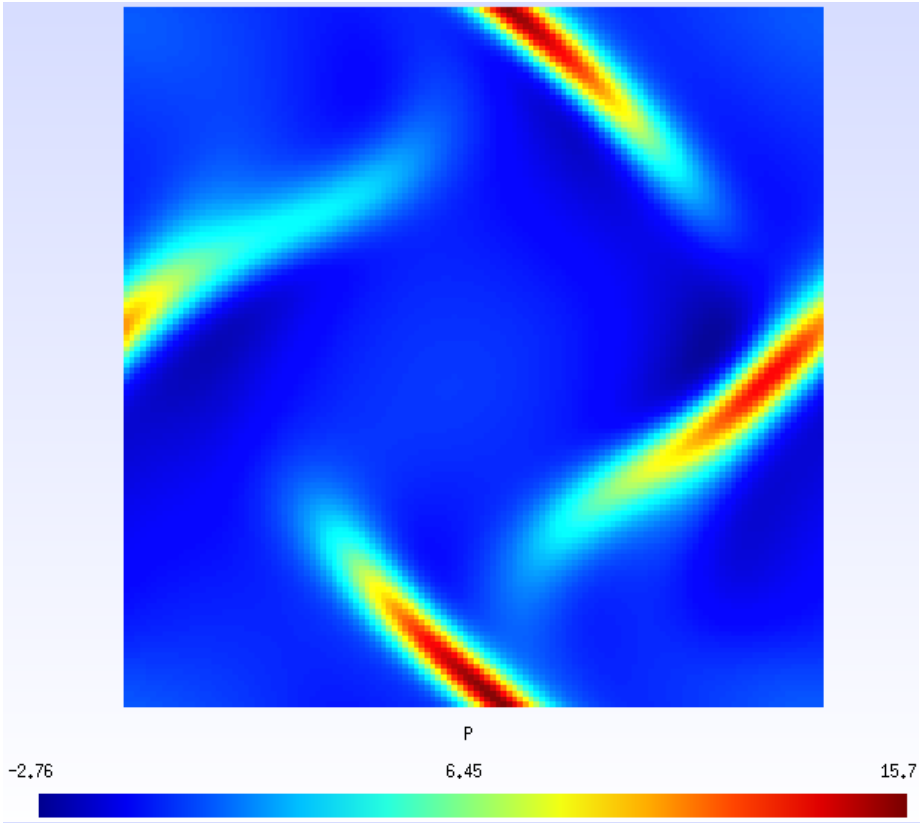
Résultats :

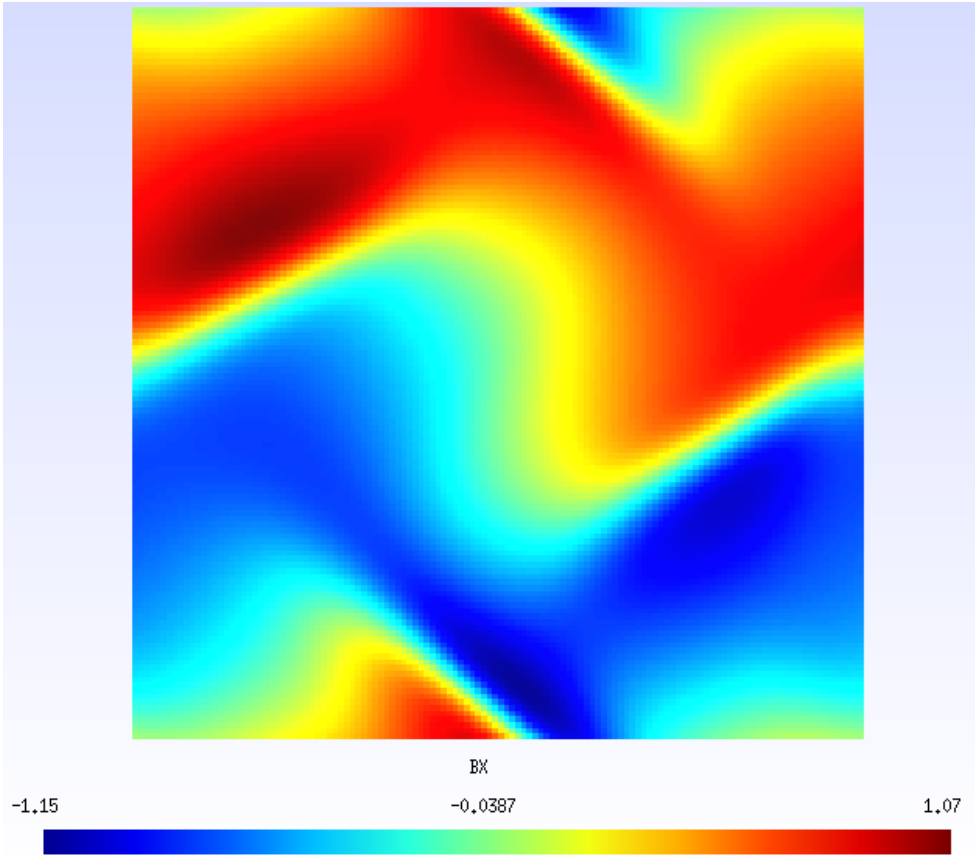
En 2D, Orzag-Tang :

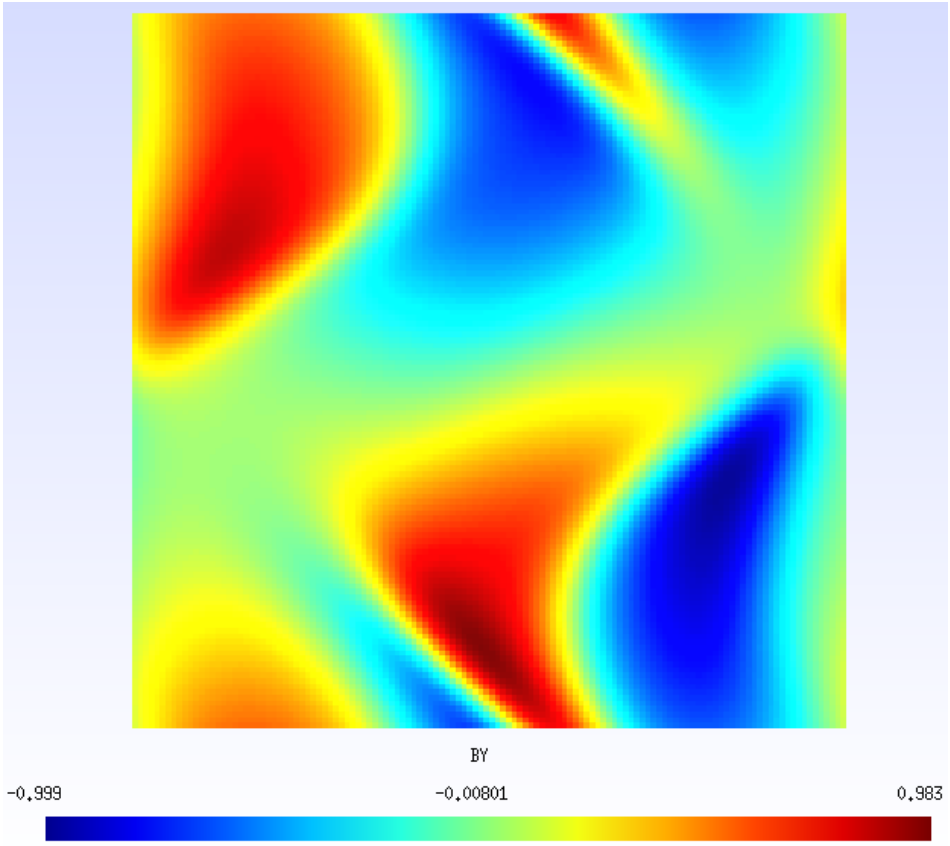



UX
-1.24
-0.0228
1.19


UY
-1.09
0.0416
1.18


P
-2.76
6.45
15.7

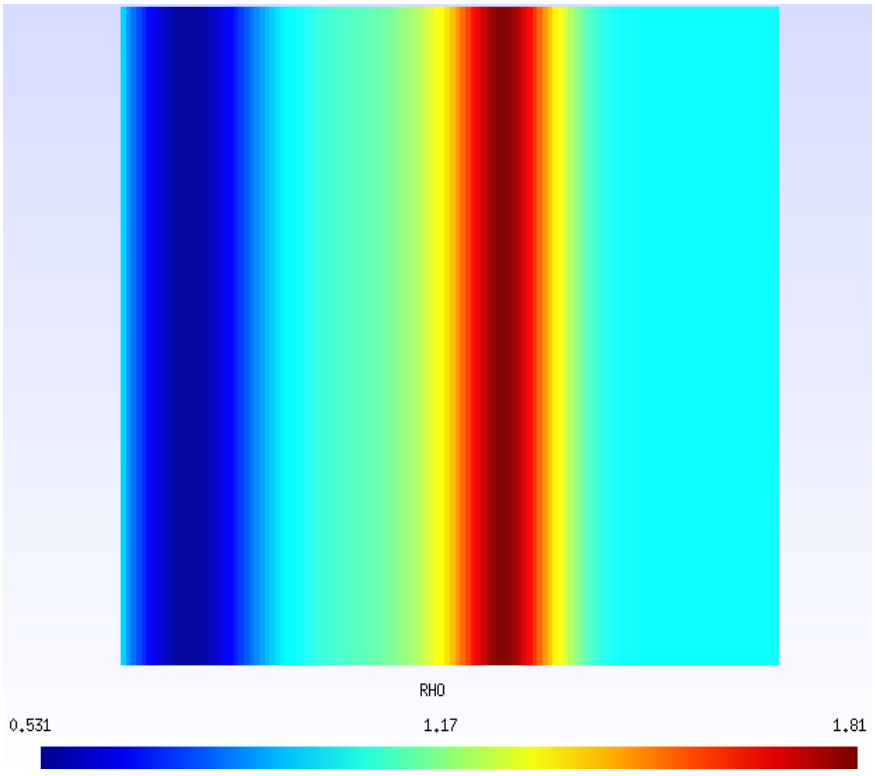

BX
-1.15
-0.0387
1.07

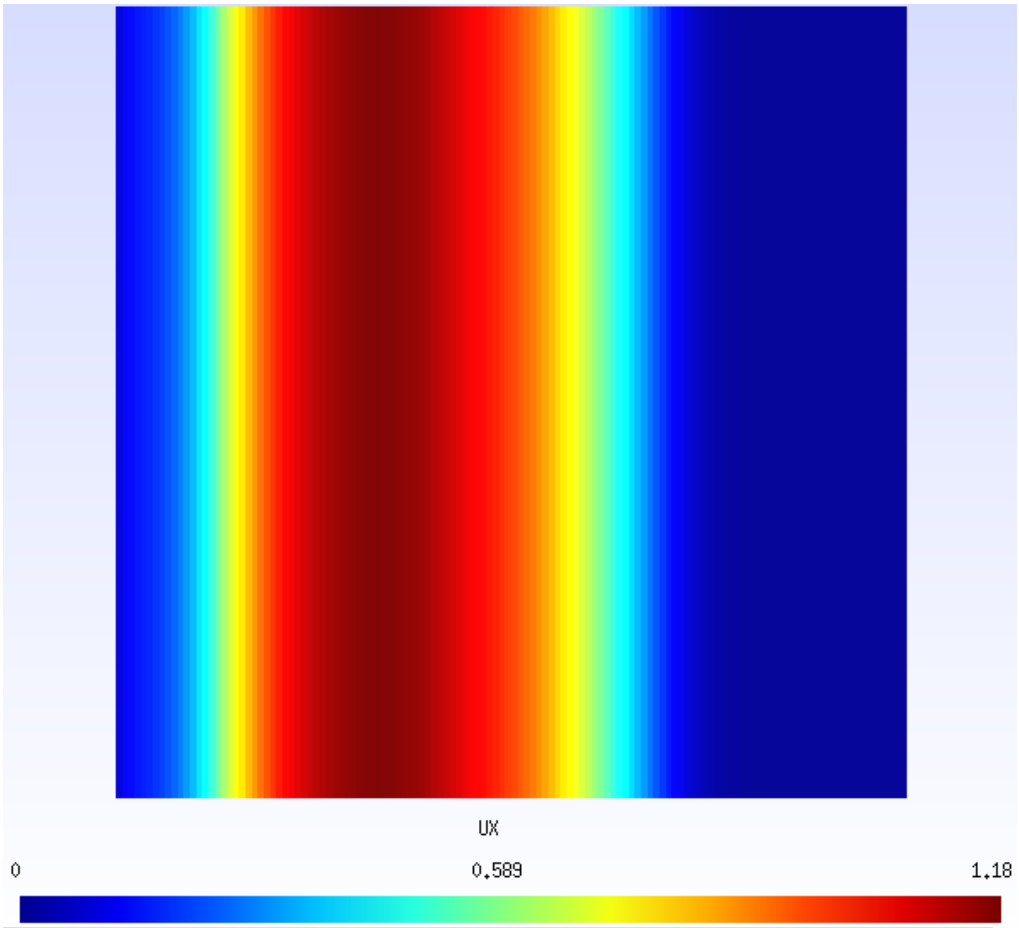


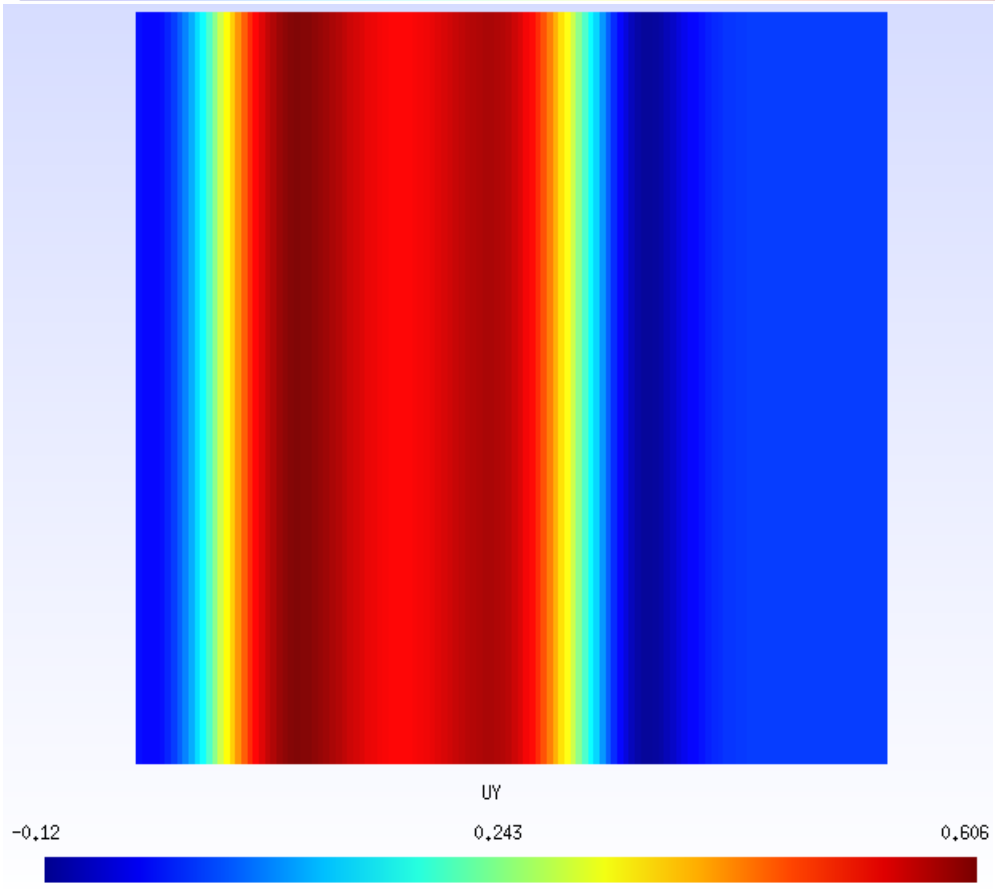
Pour Bz, Uz et PSI, on a des valeurs nulles (à des erreurs de précision près) partout.

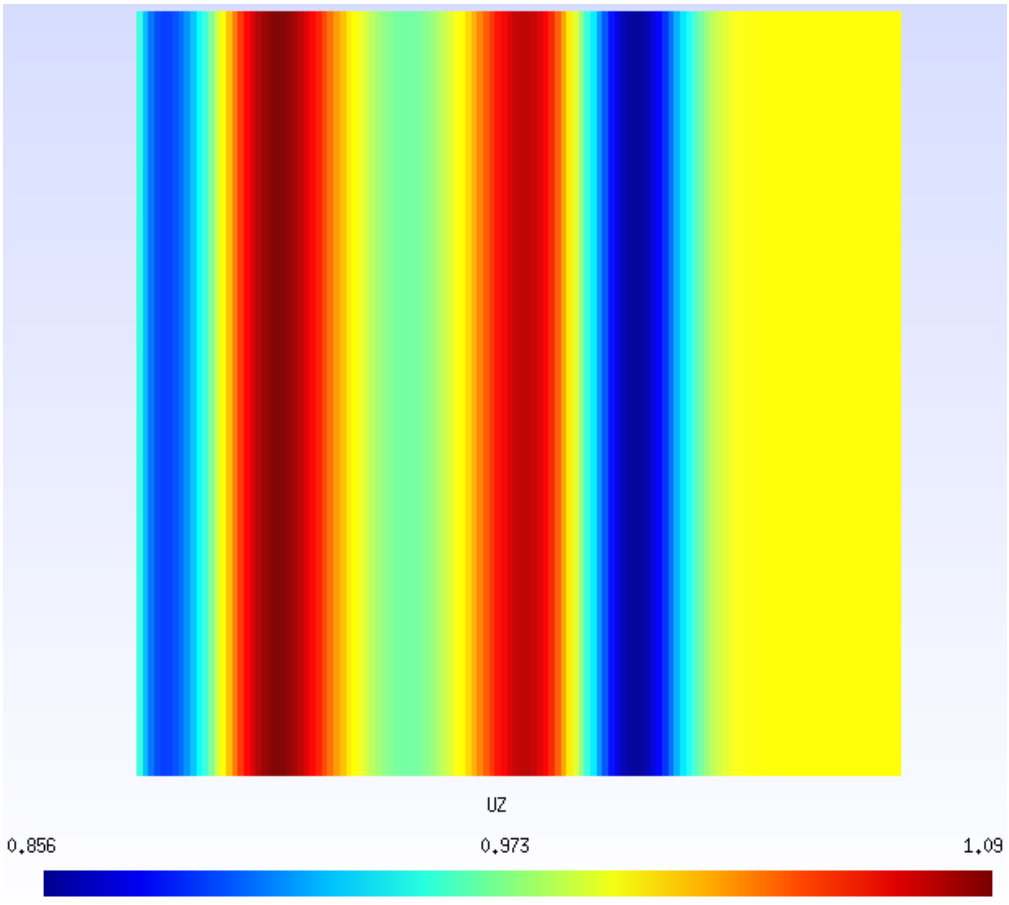
En 1D, Dai et Woodward :

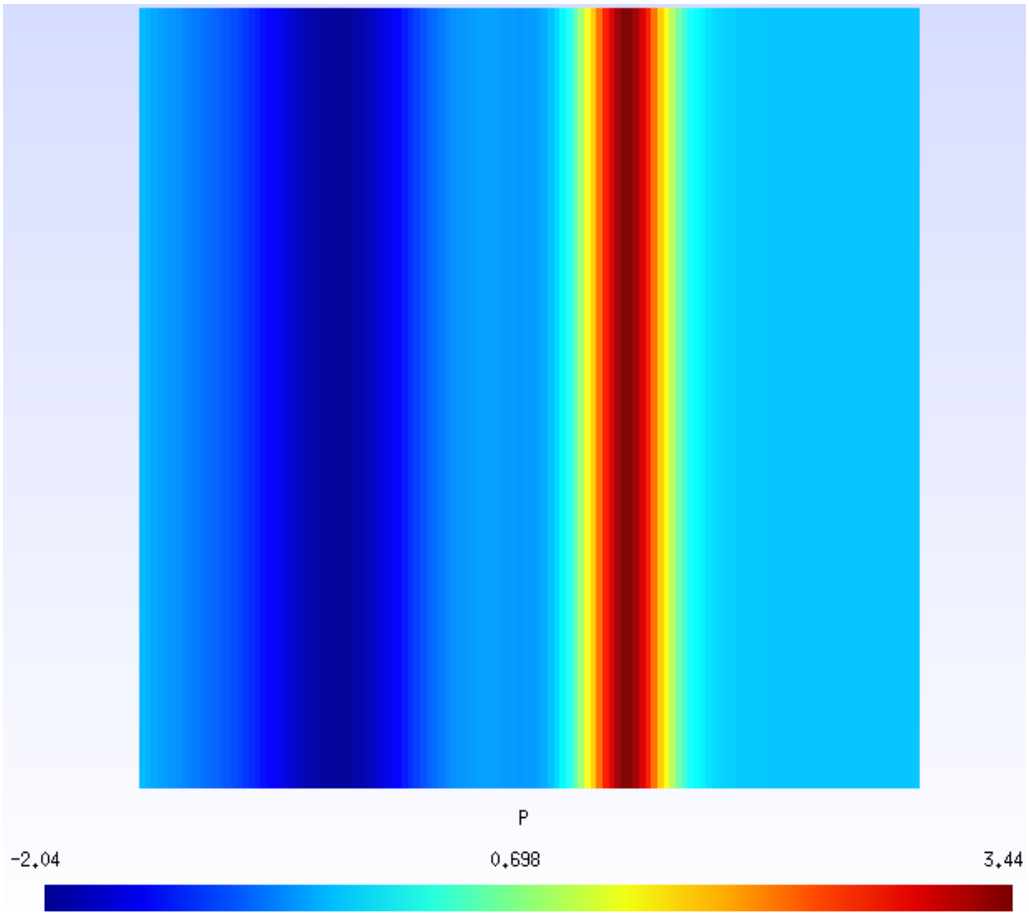
Pour PSI et Bx, on a des valeurs nulles.

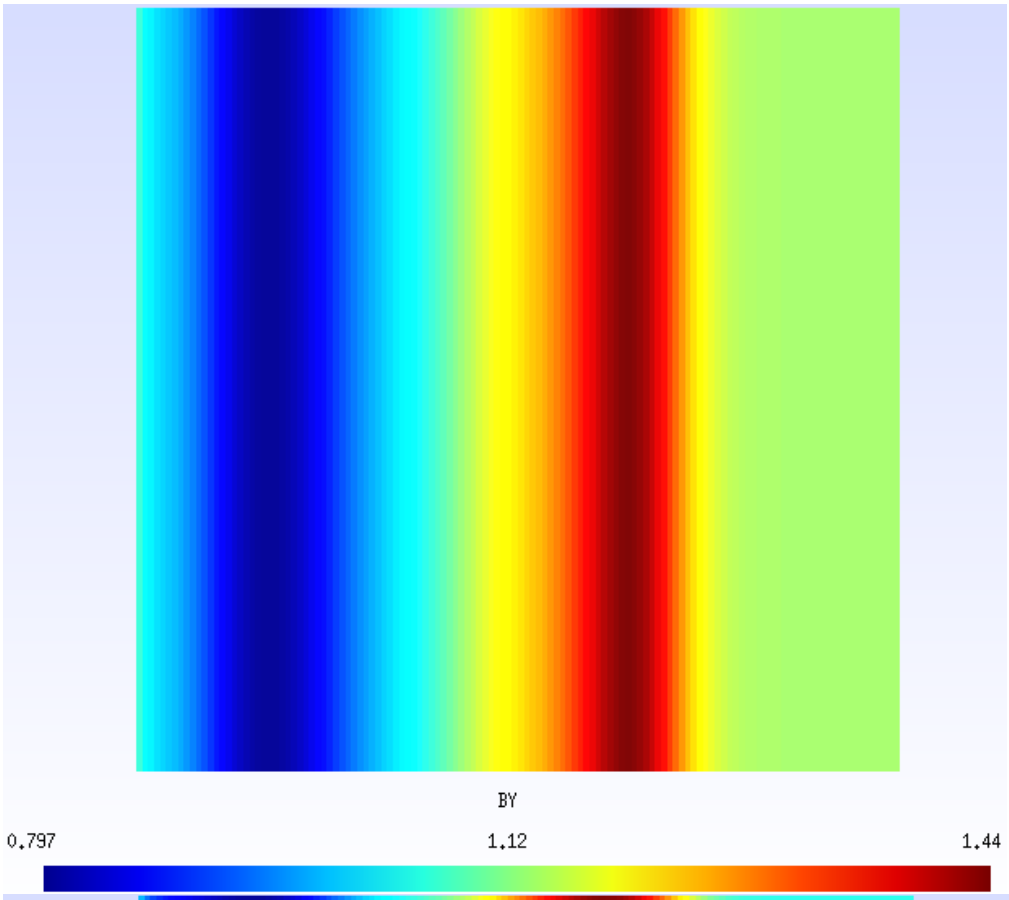


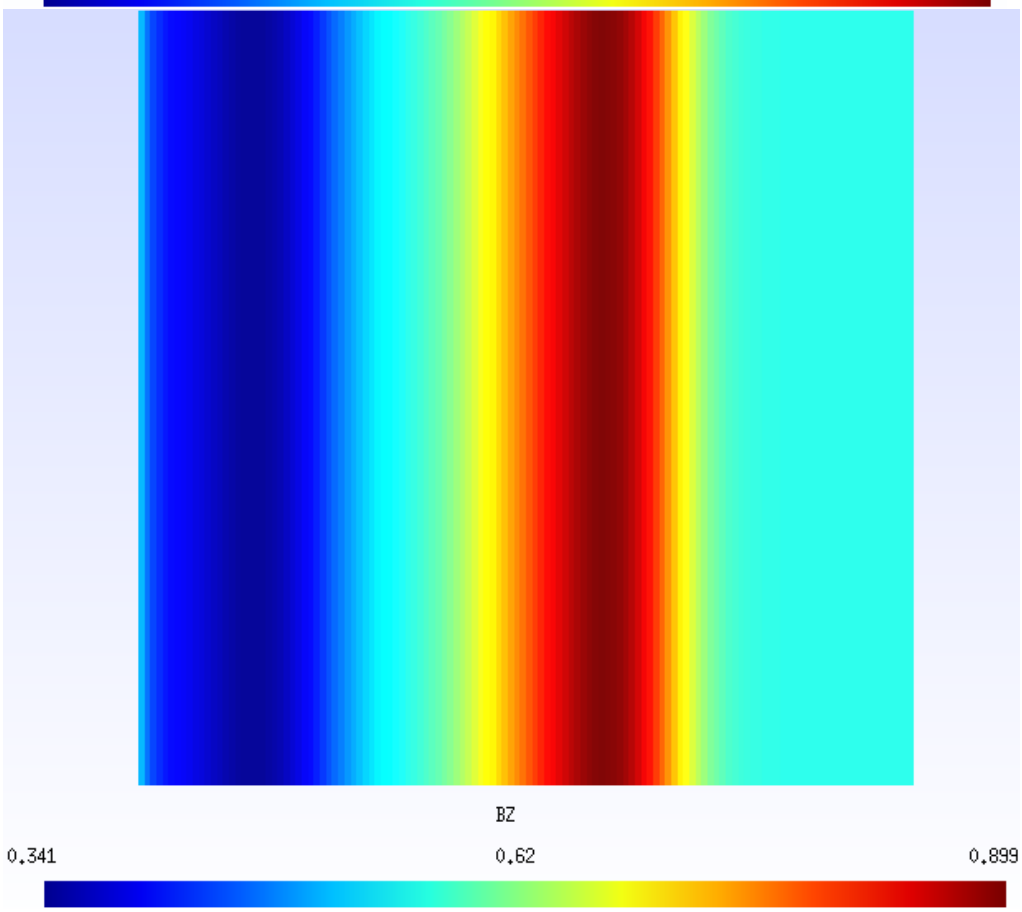

UX
0
0.589
1.18


UY
-0.12
0.243
0.606


UZ
0.856
0.973
1.09

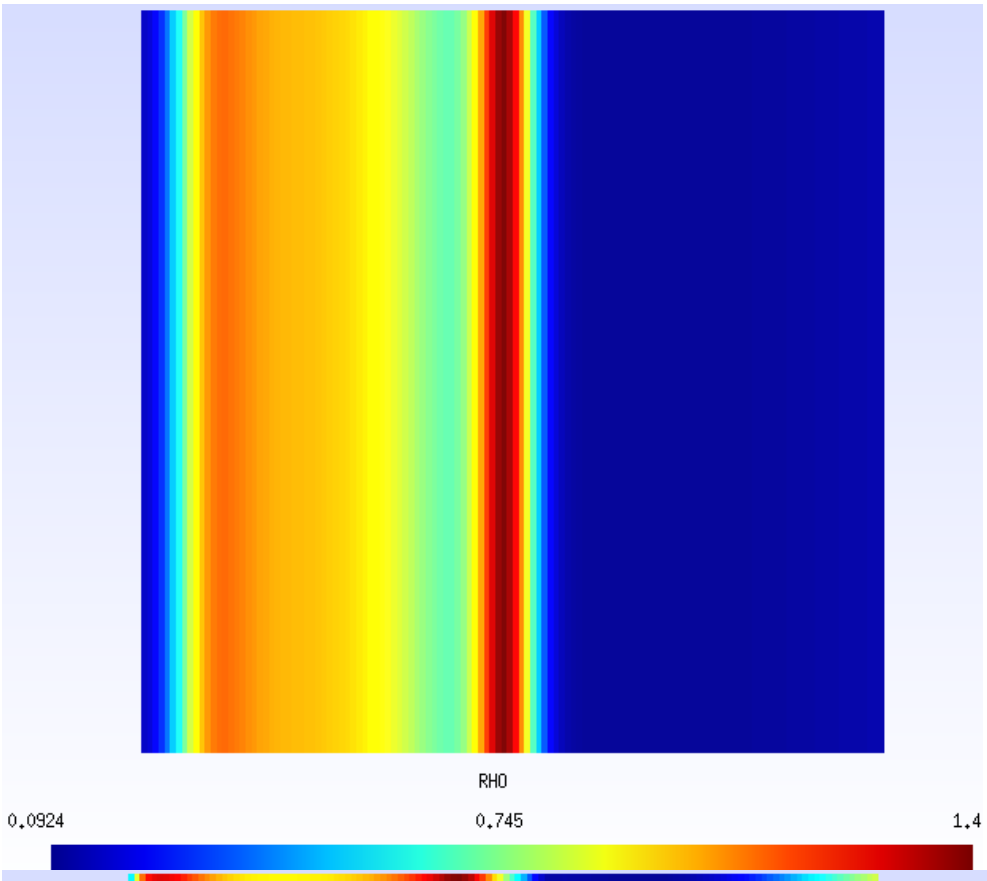

P
-2.04
0.698
3.44

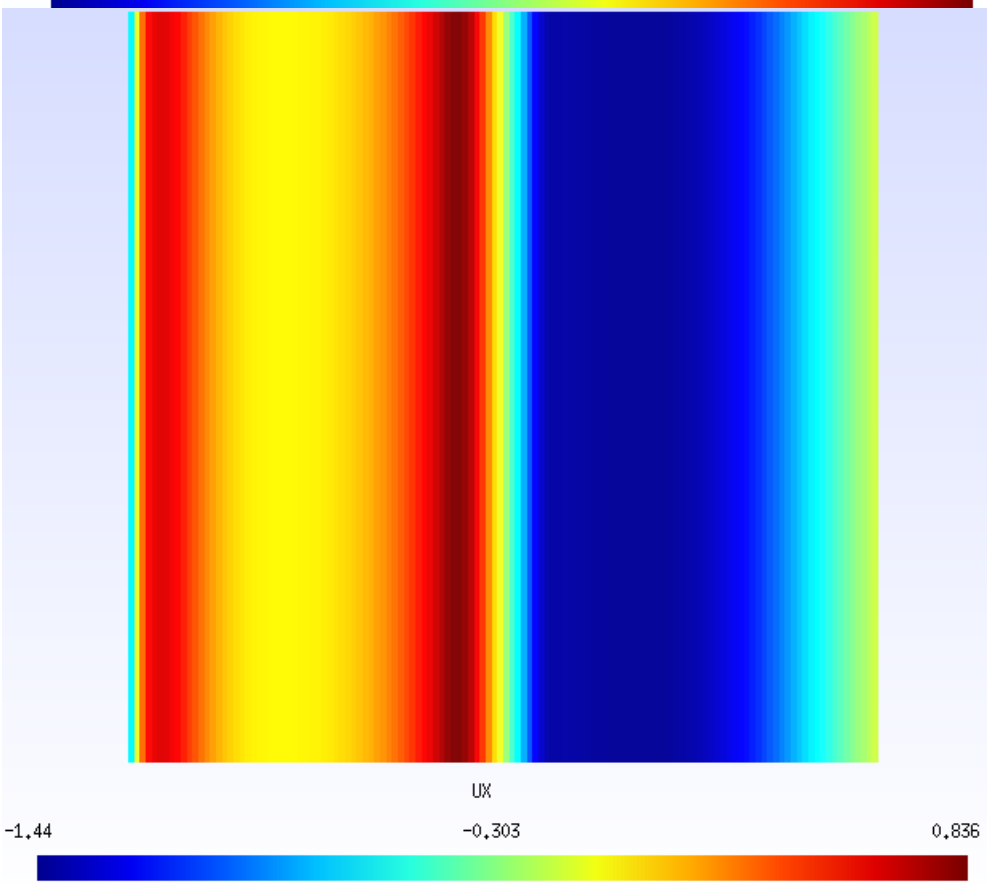

BY
0.797
1.12
1.44

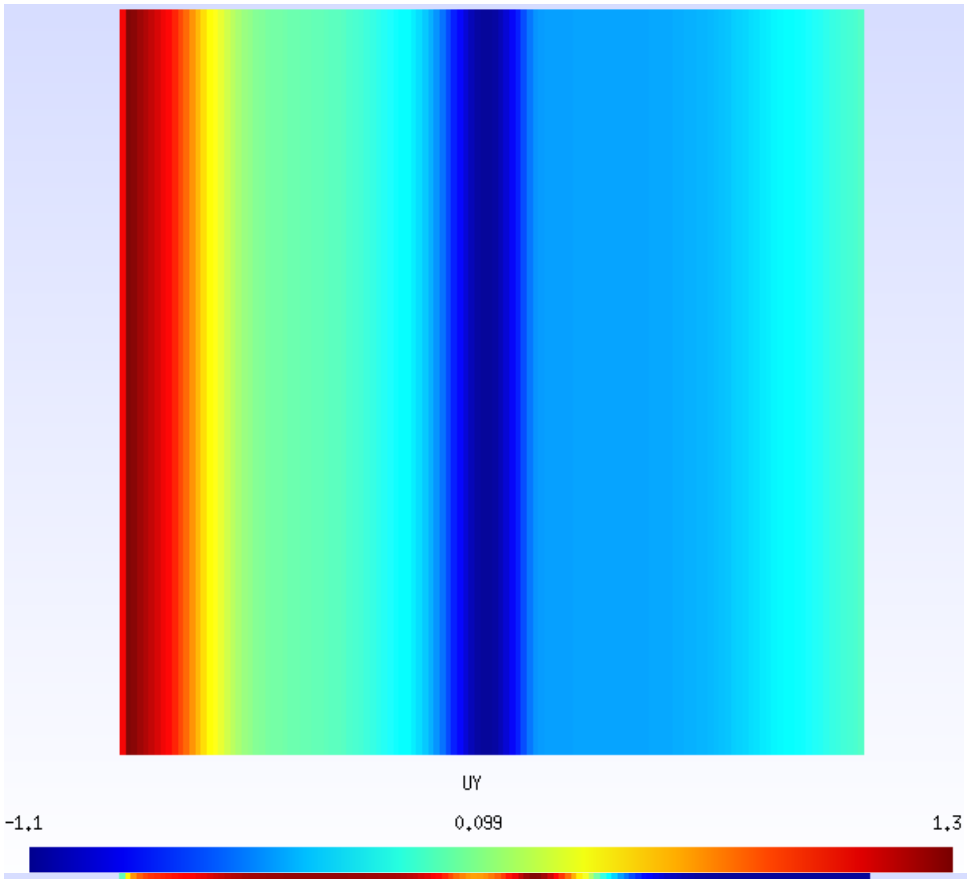

BZ
0.341
0.62
0.899

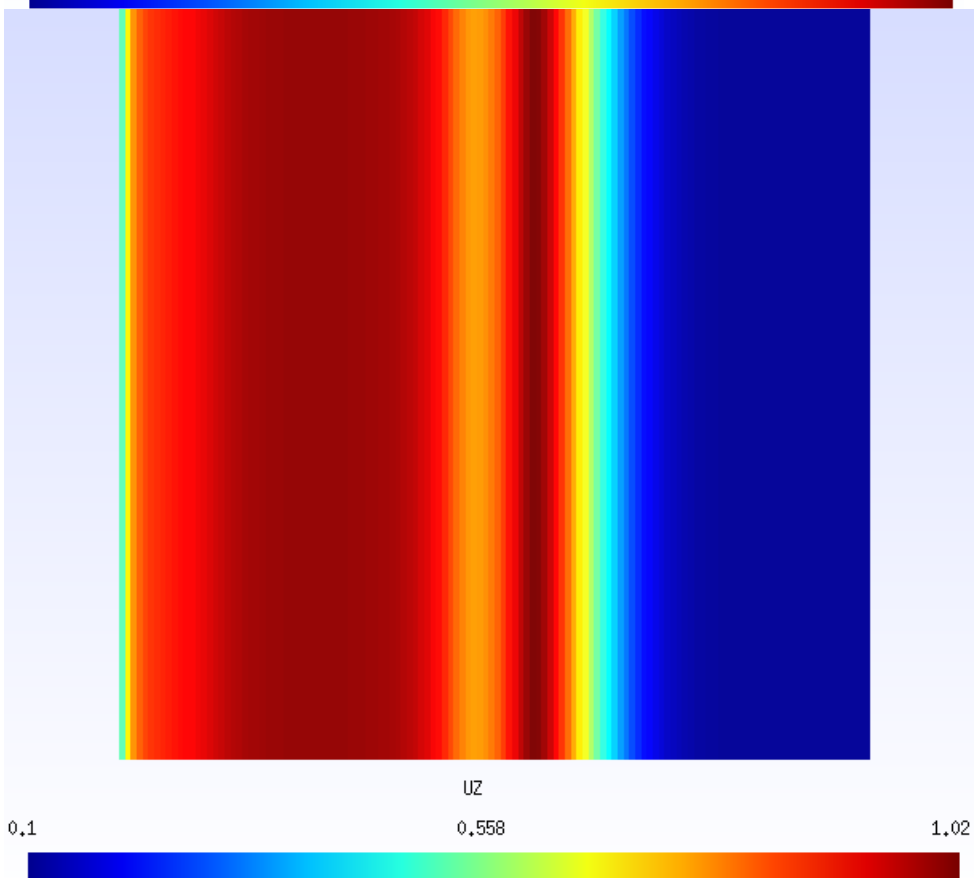
En 1D, Brio et Wu :

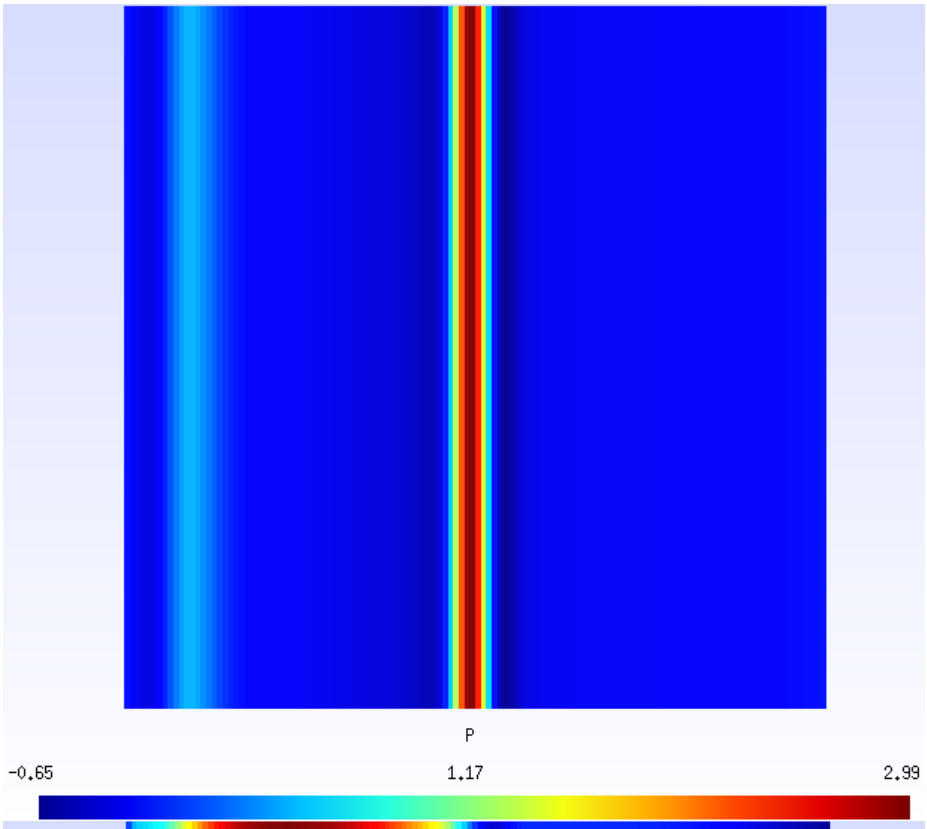
Pour PSI et Bx, on a des valeurs nulles.

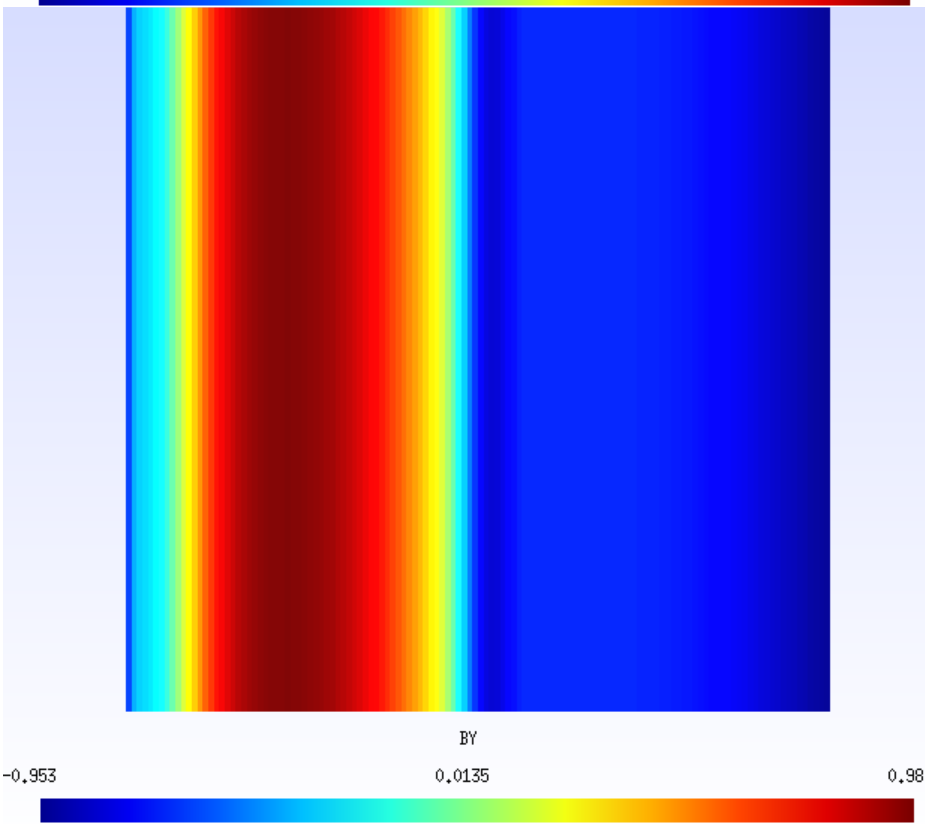


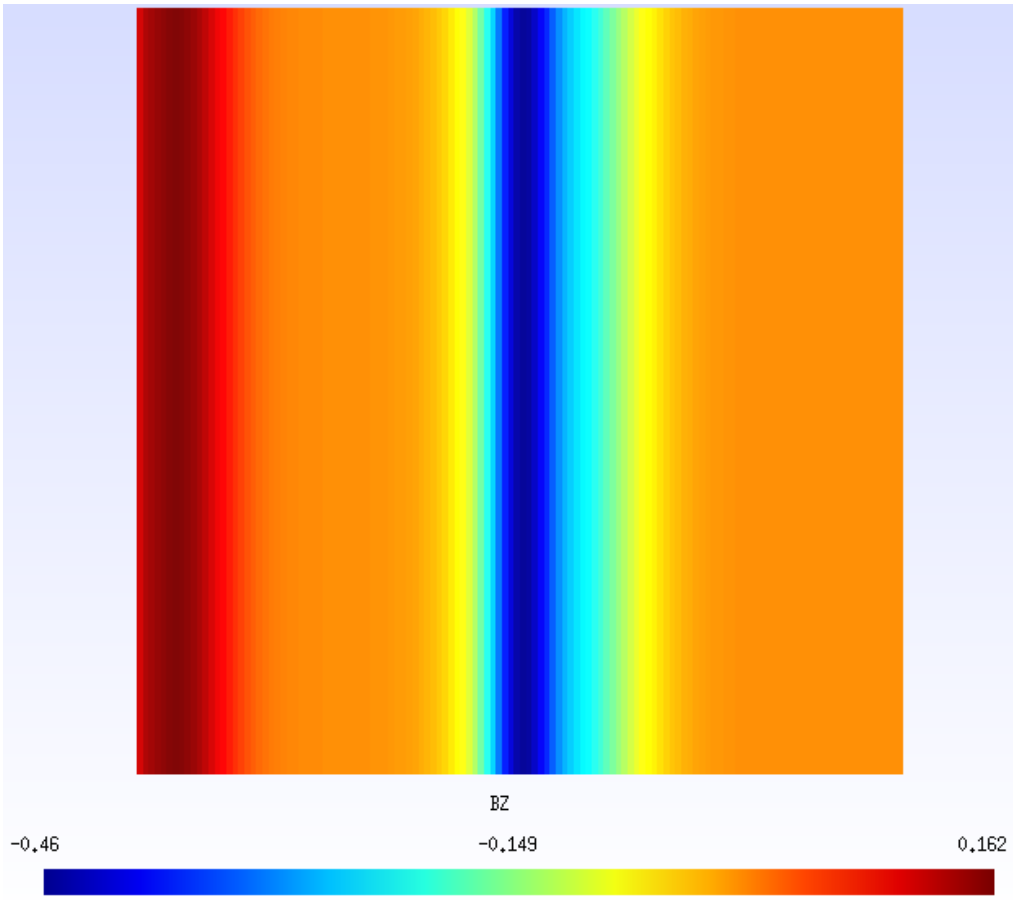



UY
-1.1
0.099
1.3


UZ
0.1
0.558
1.02

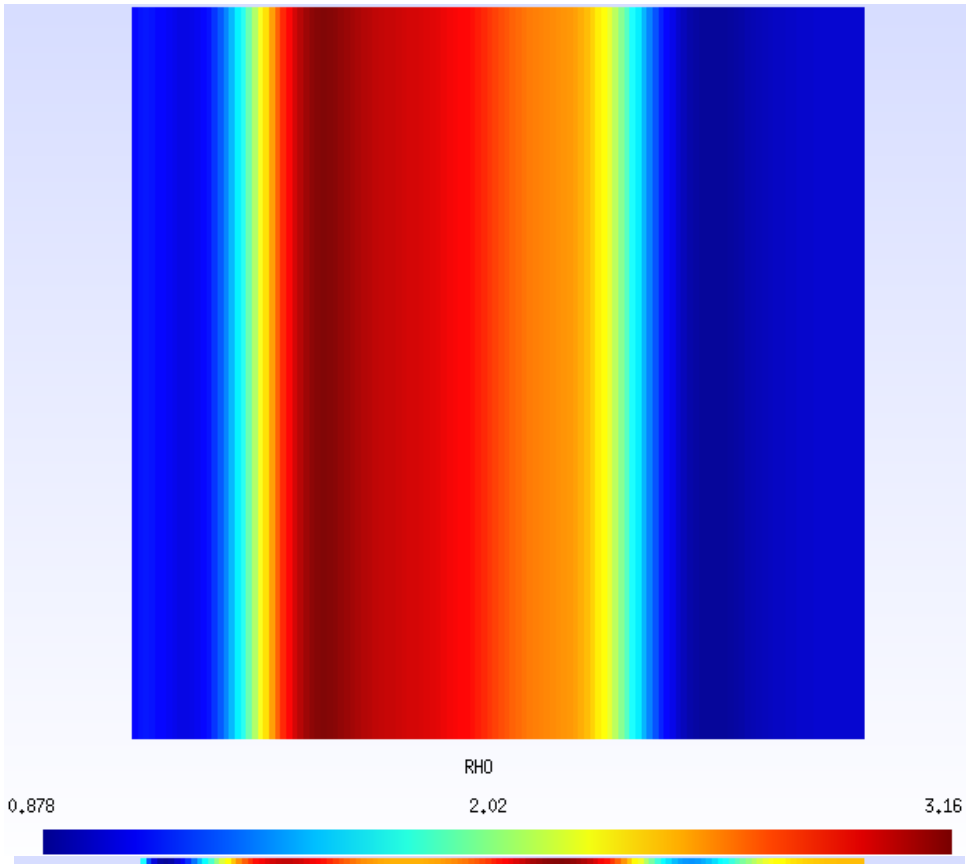

P
-0.65
1.17
2.99

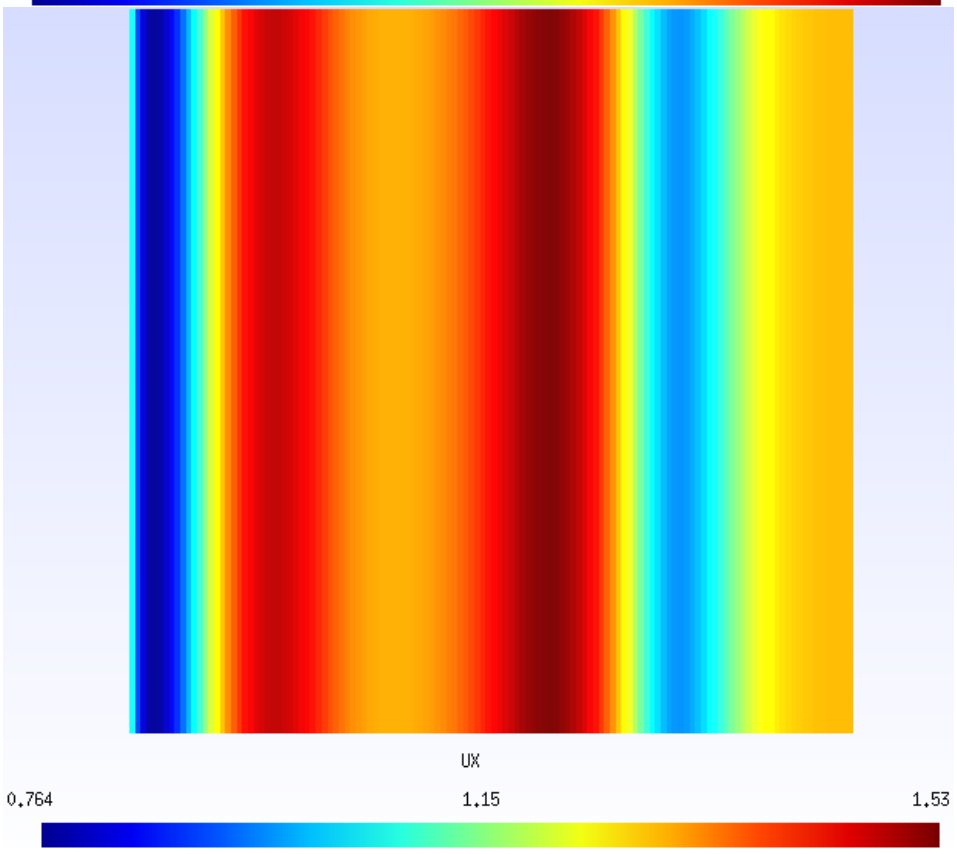

BY
-0.953
0.0135
0.98

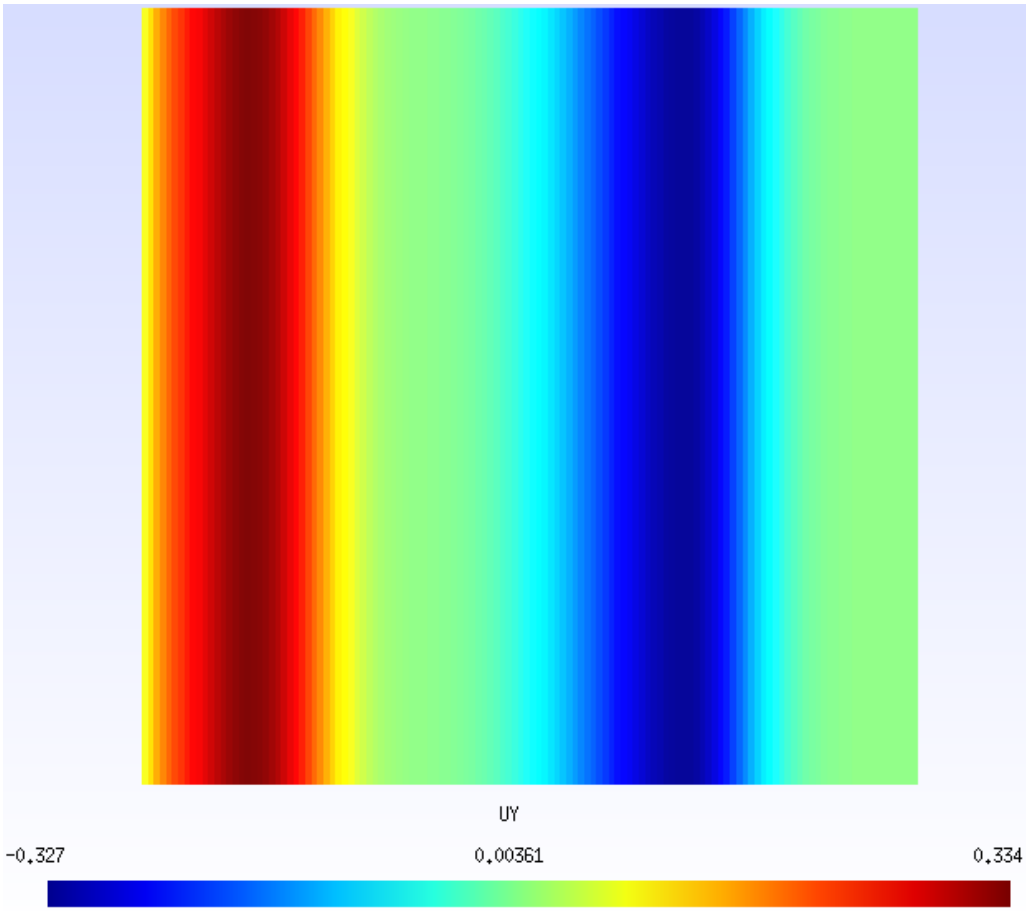

BZ
-0.46
-0.149
0.162

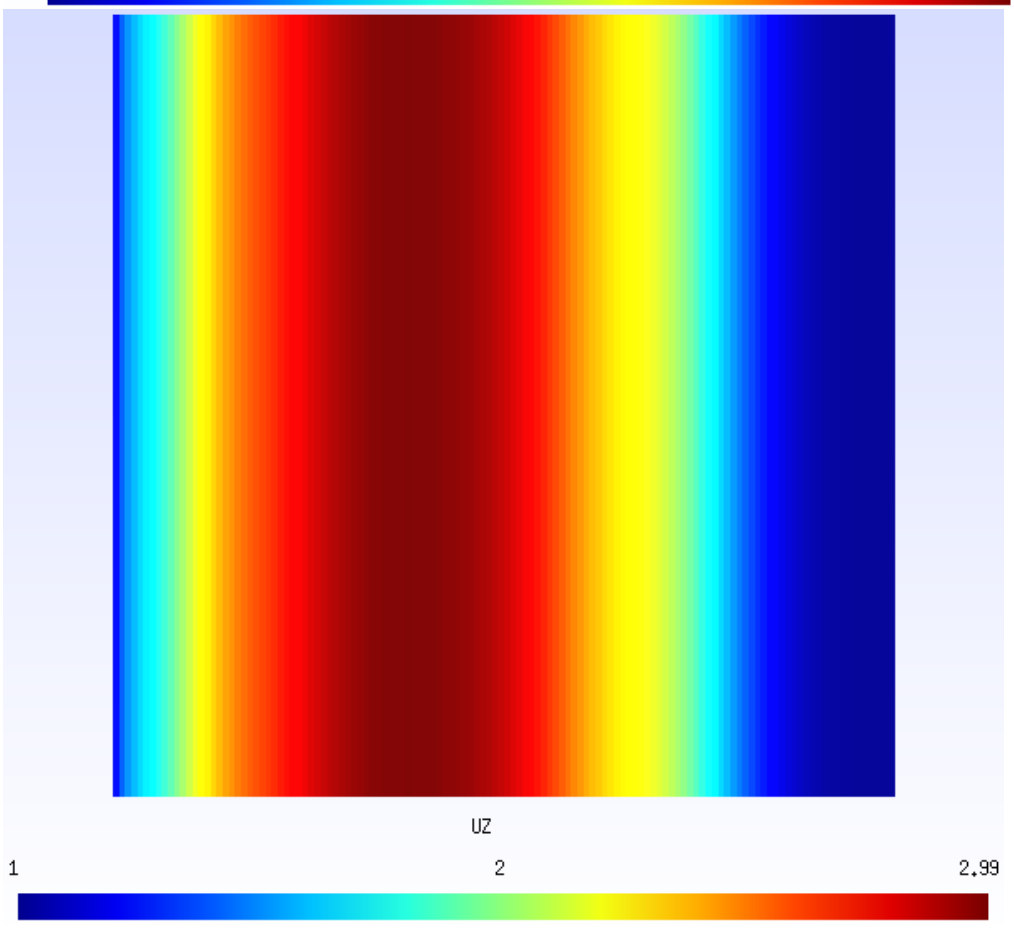
En 1D, Choc fort :

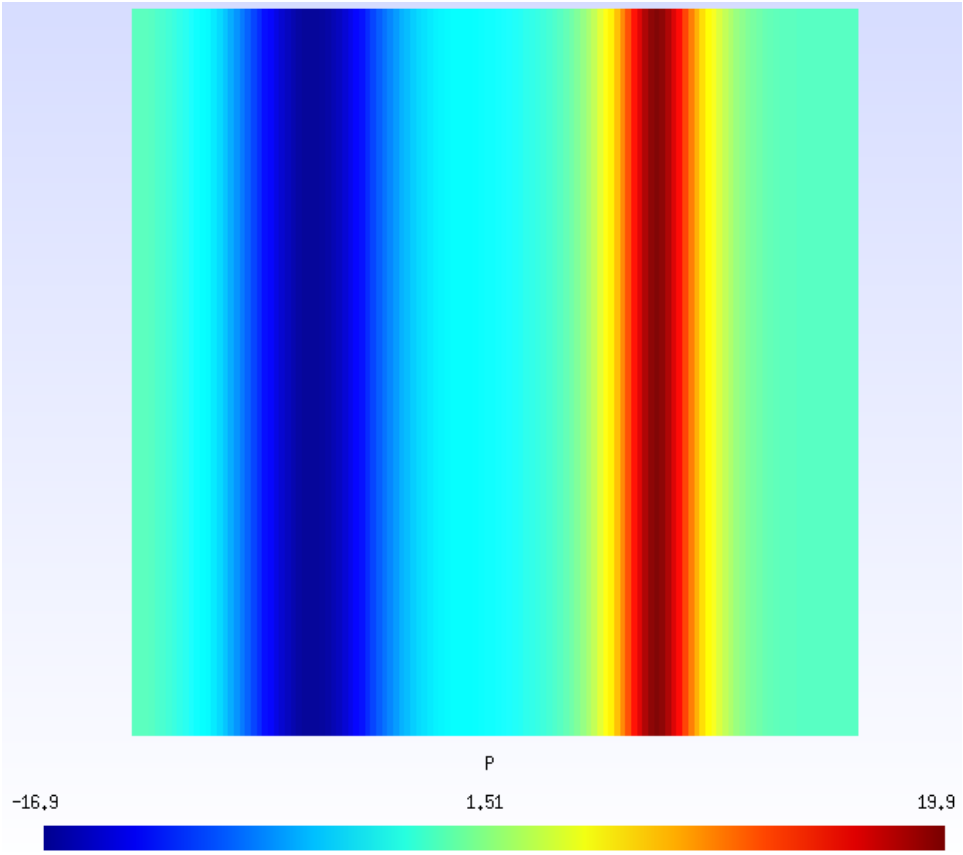
Pour PSI et Bx, on a des valeurs nulles.

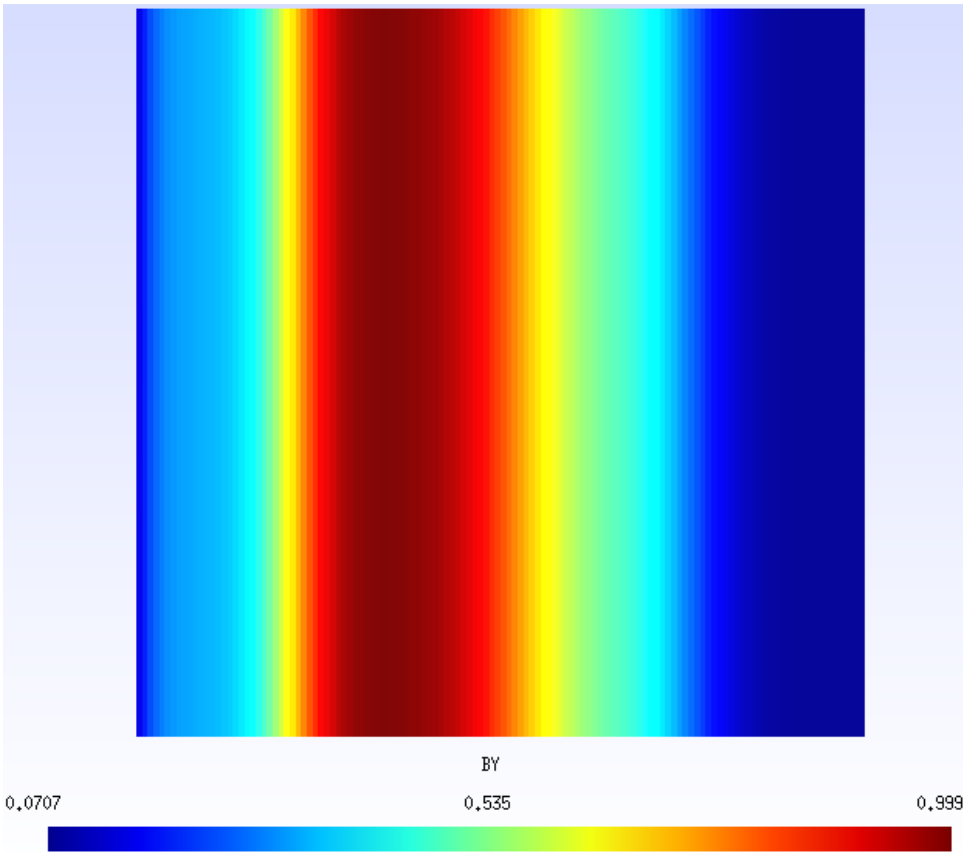


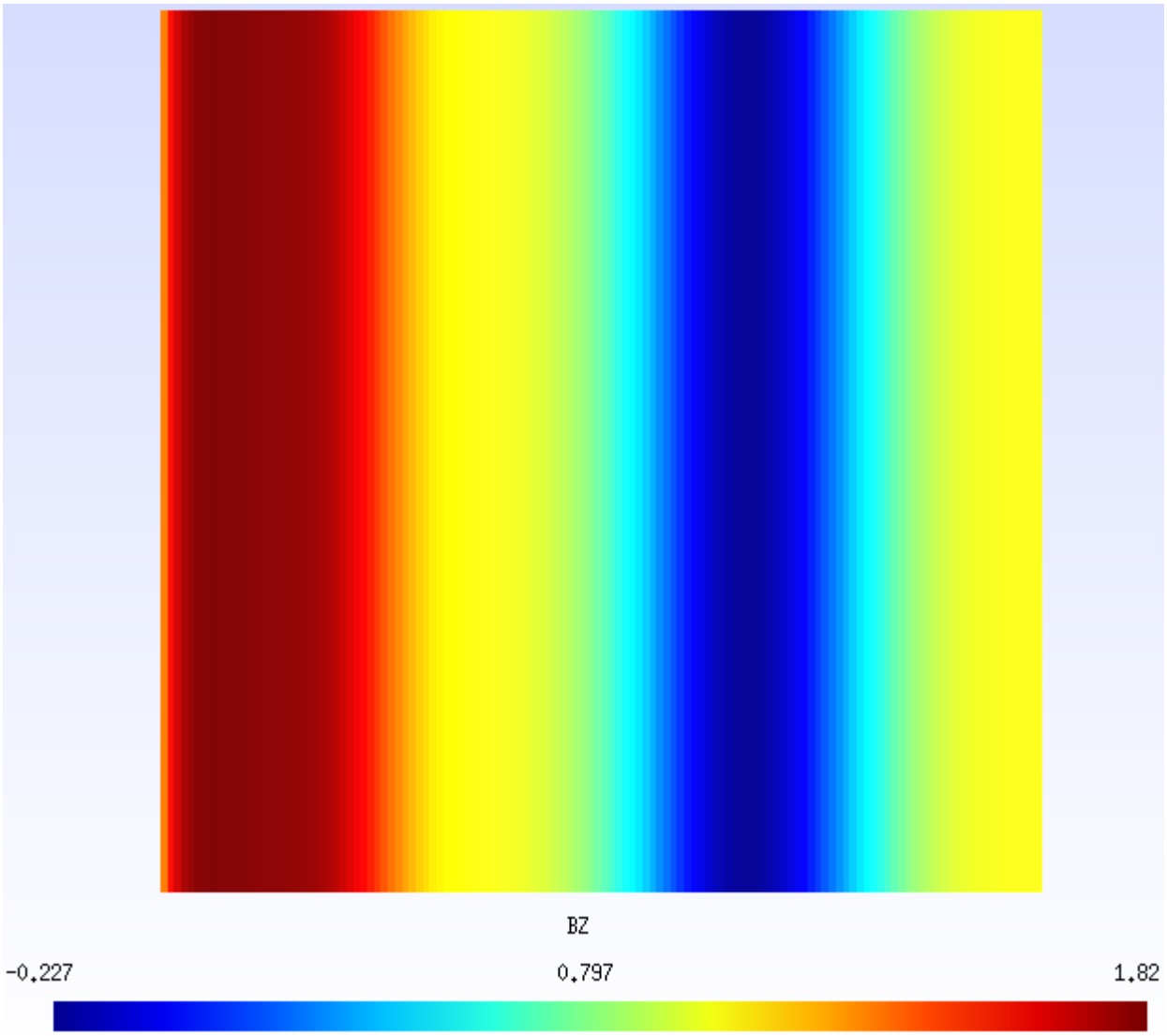



UY
-0.327
0.00361
0.334


UZ
1
2
2.99


P
-16.9
1.51
19.9


BY
0.0707
0.535
0.999


BZ
-0.227
0.797
1.82