

TP Méthodes Numériques de Bases

Hakim BENKIRANE

Omar BENCHEKROUN

Mars 2018

1 Résolution de l'équation de transport-diffusion

Question 1

— Matrice M

$$M = (m_{ij})_{i,j \in \mathbb{N}}$$

$$= \begin{pmatrix} M_0(0) & M_1(dx) & & \dots & M_{-1}((N_x - 1)dx) \\ M_{-1}(0) & M_0(dx) & M_1(2dx) & & \\ & M_{-1}((k - 1)dx) & M_0(kdx) & M_1((k + 1)dx) & \\ & & M_{-1}((N_x - 3)dx) & M_0((N_x - 2)dx) & M_1((N_x - 1)dx) \\ M_1(0) & \dots & & M_{-1}((N_x - 2)dx) & M_0((N_x - 1)dx) \end{pmatrix}$$

avec pour tout $(i, j) \in \llbracket 1, N_t \rrbracket^2$:

$$m_{ij} = \begin{cases} M_0((j - 1)dx) & \text{si } i = j \\ M_1((j - 1)dx) & \text{si } j = i + 1 \text{ ou } (i, j) = (N_t, 1) \\ M_{-1}((j - 1)dx) & \text{si } i = j + 1 \text{ ou } (i, j) = (1, N_t) \\ 0 & \text{sinon.} \end{cases}$$

et :

$$M_0(x) = 1 - c(x)^2 \frac{dt^2}{dx^2}$$

$$M_1(x) = -c(x) \frac{dt}{2dx} + c(x)^2 \frac{dt^2}{2dx^2}$$

$$M_{-1}(x) = c(x) \frac{dt}{2dx} + c(x)^2 \frac{dt^2}{2dx^2}$$

— Matrice N

$$N = (n_{ij})_{i,j \in \mathbb{N}} = \begin{pmatrix} N_0 & N_1 & & \dots & N_1 \\ N_1 & N_0 & N_1 & & \\ & N_1 & N_0 & N_1 & \\ & & N_1 & N_0 & N_1 \\ N_1 & \dots & & N_1 & N_0 \end{pmatrix}$$

avec pour tout $(i, j) \in \llbracket 1, N_t \rrbracket$:

$$n_{ij} = \begin{cases} N_0 & \text{si } i = j \\ N_1 & \text{si } j = i + 1 \text{ ou } (i, j) = (N_t, 1) \\ & \text{ou } i = j + 1 \text{ ou } (i, j) = (1, N_t) \\ 0 & \text{sinon.} \end{cases}$$

et :

$$N_1 = -\kappa \frac{dt}{dx^2}$$

$$N_0 = 1 + 2\kappa \frac{dt}{dx^2}$$

Question 2

Soit $x = (x_1, x_2, \dots, x_{n-1}, x_n)^T$:

$$\begin{aligned} x^T N x &= \sum_{i=1}^n \sum_{j=1}^n n_{ij} x_i x_j \\ &= (1 + 2\kappa \frac{dt}{dx^2}) \sum_{i=1}^n x_i^2 - 2\kappa \frac{dt}{dx^2} (x_{N_t} x_1 + \sum_{i=1}^{n-1} x_i x_{i+1}) \end{aligned}$$

On pose $y = (y_1, y_2, \dots, y_{n-1}, y_n) = (x_2, x_3, \dots, x_n, x_1)^T$, en appliquant l'inégalité de Cauchy-Schwartz :

$$\begin{aligned} \sum_{i=1}^N x_i y_i &\leq \left(\sum_{i=1}^N x_i^2 \right)^{\frac{1}{2}} \left(\sum_{i=1}^N y_i^2 \right)^{\frac{1}{2}} \\ x_{N_t} x_1 + \sum_{i=1}^{n-1} x_i x_{i+1} &\leq \left(\sum_{i=1}^n x_i^2 \right) \\ &< (1 + \frac{dx^2}{2\kappa dt}) \left(\sum_{i=1}^n x_i^2 \right) \end{aligned}$$

En multipliant par $2\kappa \frac{dt}{dx^2}$, on a $x^T N x > 0$ pour tout $x \in \mathbb{R}^n$, c'est à dire que N est symétrique définie positive.

Question 4

L'objectif de cette question est de vérifier le comportement de l'algorithme de factorisation de Cholesky pour la résolution du système linéaire en présence.

Une fois qu'on l'on a notre fonction ϕ_0 , il s'agit de créer les matrices M et N. On obtient donc notre résultat pour lequel on fait varier la valeur de kappa :

Pour $\kappa = 10^{-2}$

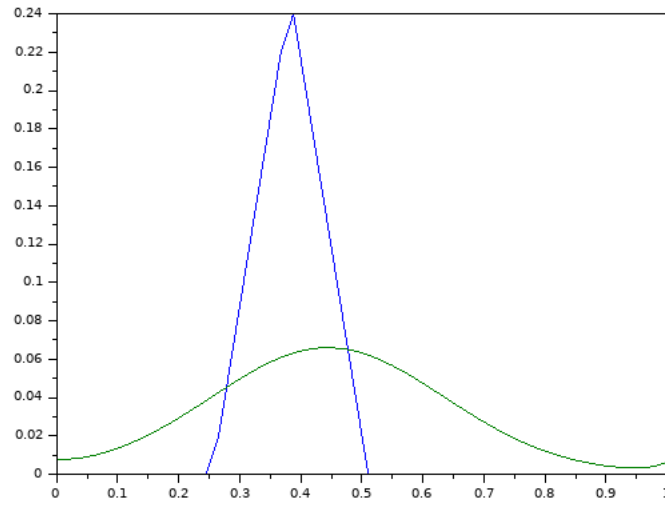


FIGURE 1 – Résolution numérique Cholesky pour $\kappa = 0.01$

On remarque ici que la solution à l'état initial est pentue alors qu'à l'état final son amplitude diminue et prend la forme d'une sinusoïde aplatie.

Pour $\kappa = 10^{-3}$

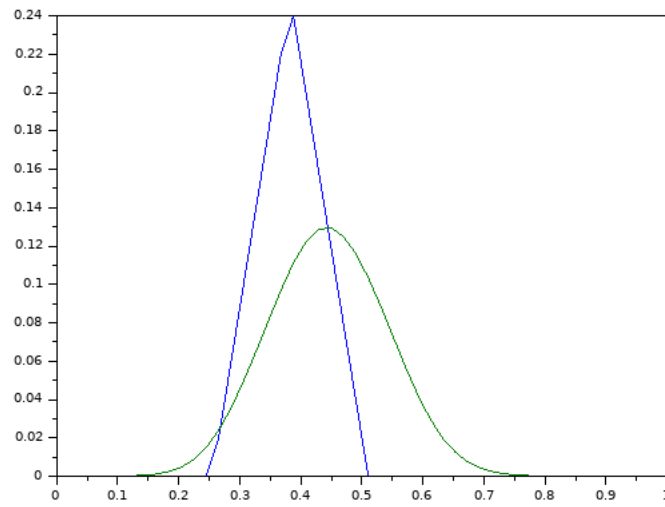


FIGURE 2 – Résolution numérique Cholesky pour $\kappa = 0.001$

Pour $\kappa = 10^{-4}$

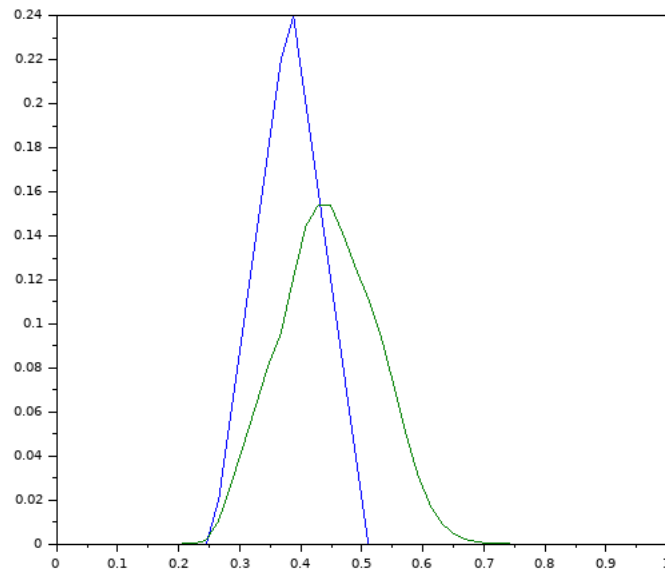


FIGURE 3 – Résolution numérique Cholesky pour $\kappa = 0.0001$

Plus on diminue la valeur de κ , plus la solution finale se redresse pour tendre en amplitude vers la solution initiale. Cependant, elle garde toujours son aspect courbée.

Essayons maintenant la même résolution mais cette fois avec umfpack :

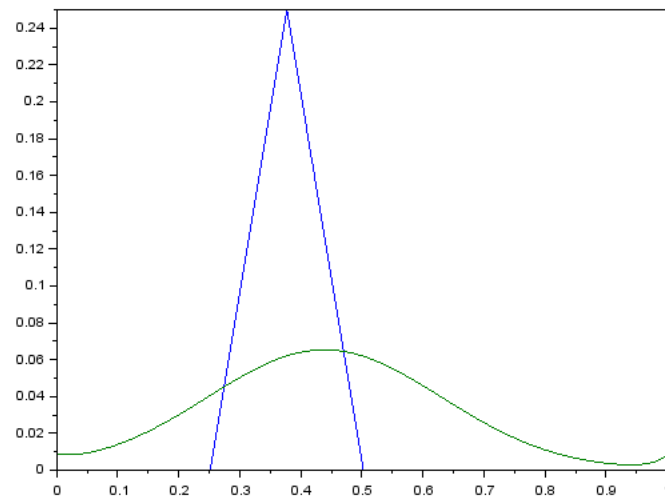


FIGURE 4 – Résolution numérique Umfpack pour $\kappa = 0.01$

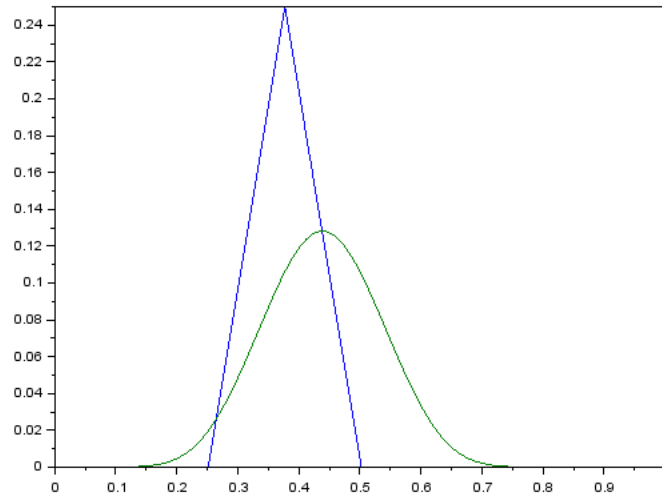


FIGURE 5 – Résolution numérique Umfpack pour $\kappa = 0.001$

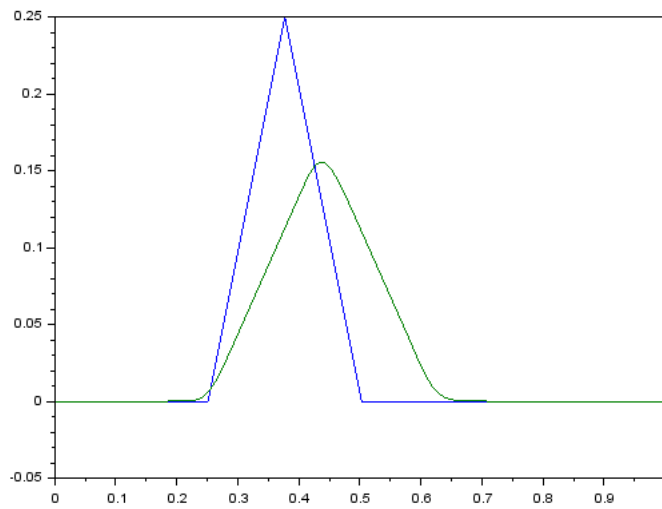


FIGURE 6 – Résolution numérique Umfpack pour $\kappa = 0.0001$

On remarque bien qu'avec umfpack non seulement le temps de simulation est beaucoup plus rapide mais en plus l'aspect des courbes est plus lisse (sûrement à cause du pas spatial qui varie). On privilégiera donc Umfpack pour la suite.

1.1 Question 5

Nous allons désormais nous intéresser à la résolution du problème par le schéma de Splitting. On obtient après simulation le résultat suivant :

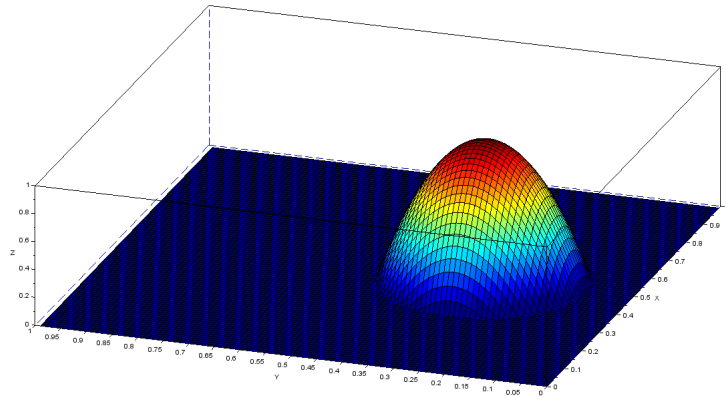


FIGURE 7 – Solution à l'instant initial

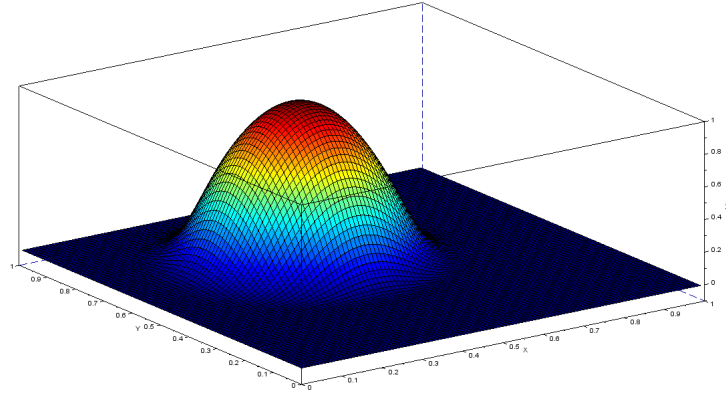


FIGURE 8 – Solution à l'instant final

On remarque dans un premier temps que le résultat concorde bien avec ce que l'on a obtenu en question 4. En effet, on voit l'aspect pentu de la solution initiale en parallèle avec la solution finale qui est plus courbée. Si on intercepte notre plot avec un plan de z constant on obtiendrait la forme obtenue en Question 4 ce qui est déjà rassurant quant à l'implémentation de la représentation 3D.

Vérifions maintenant que le comportement de la solution lorsque l'on fait varier κ concorde avec ce que l'on a obtenu précédemment avec la méthode de la factorisation de Cholesky.

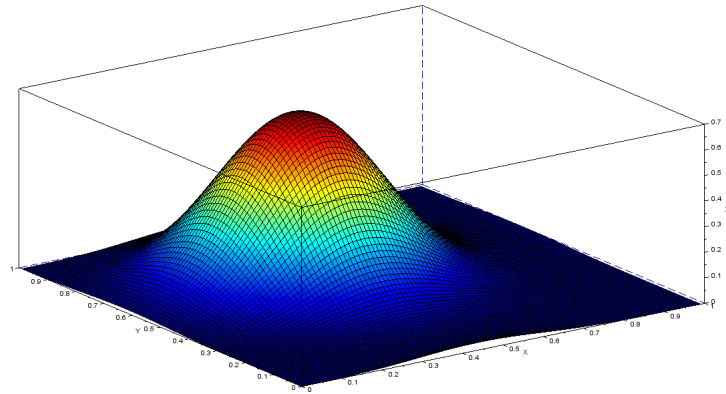


FIGURE 9 – Solution à l’instant final pour $\kappa = 0.01$

L’aspect de la courbe est bien plus aplatie donc plus on augmente κ moins la courbe est pentue et moins l’amplitude est importante.

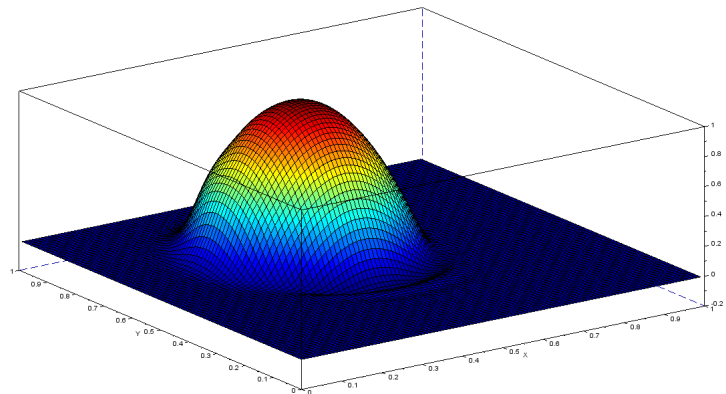


FIGURE 10 – Solution à l’instant final pour $\kappa = 0.0001$

Comme précédemment, on remarque que plus on augmente κ plus la courbe gagne en amplitude et perd de sa courbure.

On peut donc conclure que le résultat obtenu concorde avec la méthode de Cholesky.

2 Résolution du problème de Poisson

Question 6

On a :

$$\begin{aligned}\hat{f}_{pq} &= \frac{1}{L_x L_y} \int_{\Omega} f(x, y) e^{-k_x x - k_y y} dx dy \\ &= \frac{1}{L_x L_y} \int_{\Omega} \frac{\partial^2 \psi}{\partial x^2} e^{-k_x x - k_y y} dx dy + \frac{1}{L_x L_y} \int_{\Omega} \frac{\partial^2 \psi}{\partial y^2} e^{-k_x x - k_y y} dy dx\end{aligned}$$

Par 2 intégrations par parties :

$$\hat{f}_{pq} = (k_x^2 + k_y^2) \cdot \hat{\psi}_{pq}$$

Ou bien :

$$\hat{\psi}_{pq} = \frac{1}{k_x^2 + k_y^2} \hat{f}_{pq}$$

2.1 Résolution numérique de l'équation de Poisson 1D

Intéressons nous à la forme de la solution par comparaison à la fonction théorique de référence donnée dans le sujet.

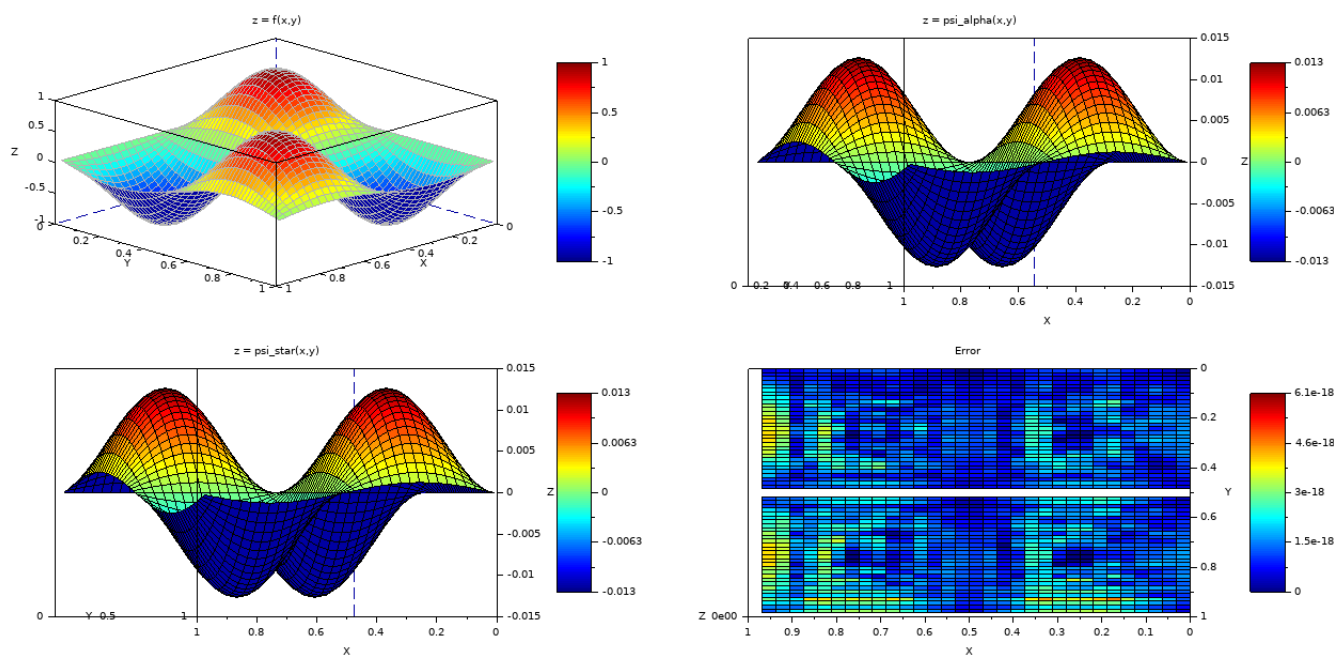


FIGURE 11 – Résolution de l'équation de Poisson par séries de Fourier

A première vue le tracé de la fonction ψ ressemble bien à la fonction de référence. Si on voit de plus près au niveau de l'erreur on voit bien que celle-ci est infime (de l'ordre de 10^{-8}) On peut donc conclure que la méthode de résolution par séries de Fourier est efficace.

2.2 Résolution du problème de Poisson 2D

Nous allons maintenant résoudre par la même méthode le problème de Poisson à 2 dimensions.

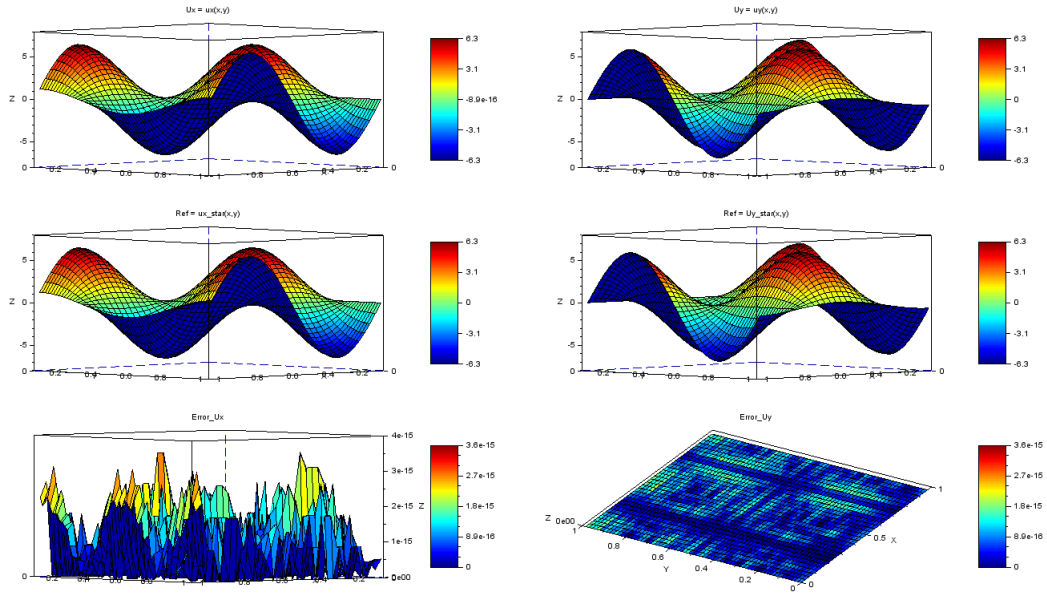


FIGURE 12 – Résolution de Poisson 2D

Le premier point à soulever est la ressemblance entre les courbes au niveau de la forme : On est bien proche de la fonction de référence.

Concernant l'erreur pour y on a une erreur plutôt homogène dans l'espace (et très faible). Par contre pour x les amplitudes de ces erreurs sont très variables même si elles restent très faibles dans l'absolu.

La méthode est donc très satisfaisante pour la résolution du problème de Poisson.

Question 11

$$\Delta \mathbf{u}_x(x, y) = -\frac{\partial w}{\partial y}(x, y)$$

D'après la question 6 :

$$\begin{aligned} (k_x^2 + k_y^2)\hat{u}_x &= -\frac{\widehat{\partial w}}{\partial y}(x, y) \\ &= -\frac{1}{L_x L_y} \int_{\Omega} \frac{\partial w}{\partial y} e^{-k_x x - k_y y} dx dy \\ &= -k_y \hat{w} \end{aligned}$$

D'où :

$$\hat{u}_x = \frac{-k_y}{k_x^2 + k_y^2} \hat{w}$$

Et de même, on a :

$$\hat{u}_y = \frac{k_x}{k_x^2 + k_y^2} \hat{w}$$

Question 12

$$\begin{aligned}\omega^0(x, y) &= \frac{\partial \mathbf{u}_y^0}{\partial x} - \frac{\partial \mathbf{u}_x^0}{\partial y} \\ &= \begin{cases} 2\pi\delta \cos(2\pi x) - \rho(1 - \tanh^2(\rho[y - 0.25])) & \text{pour } y \leq 0.5 \\ 2\pi\delta \cos(2\pi x) + \rho(1 - \tanh^2(\rho[0.75 - y])) & \text{pour } y > 0.5 \end{cases}\end{aligned}$$

2.3 Question 13

On s'intéresse à la résolution de notre système différentiel en prenant pour références $t = 0.8s$ et $t = 1.2s$

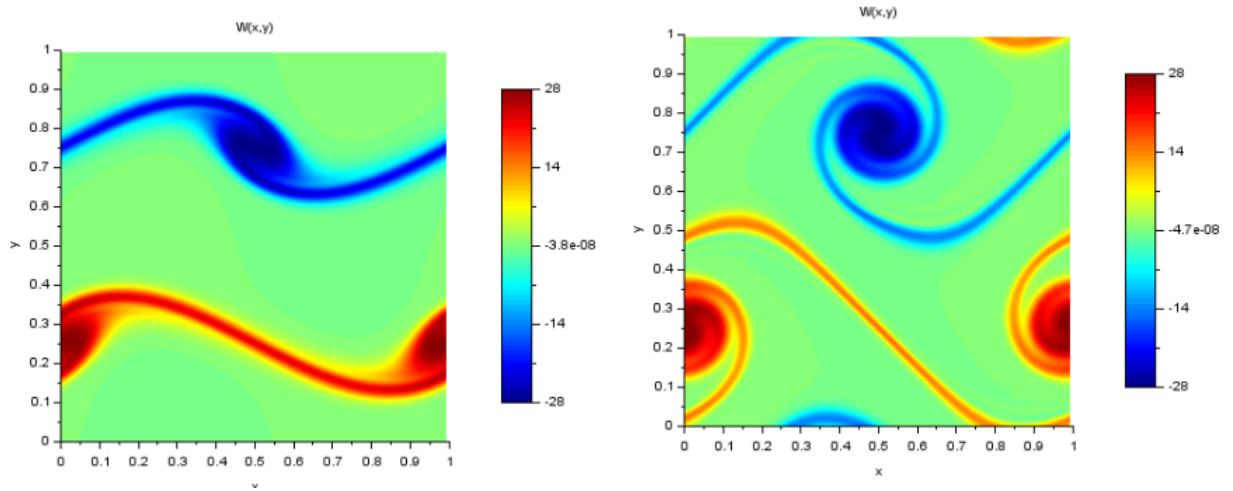


FIGURE 13 – Solution du système à $t=0.8s$ puis à $t=1.2s$

On remarque que plus on avance de le temps, plus la figure de W se distord pour créer une sorte de vortex qui rotate autour du point d'abscisse 0.5 et d'ordonnée 0.75.

Intéressons nous désormais aux isocontours :

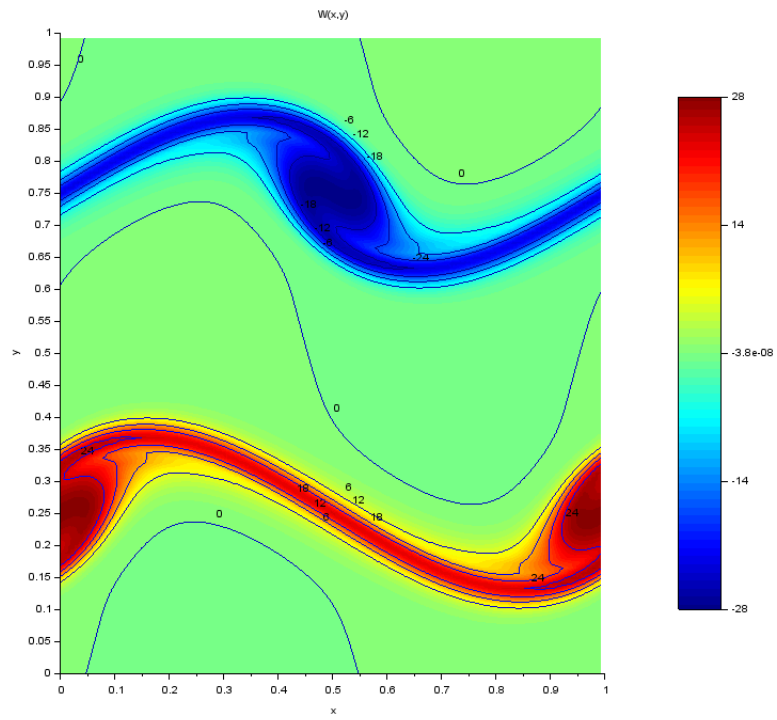


FIGURE 14 – Isocontours à $t=0.8s$

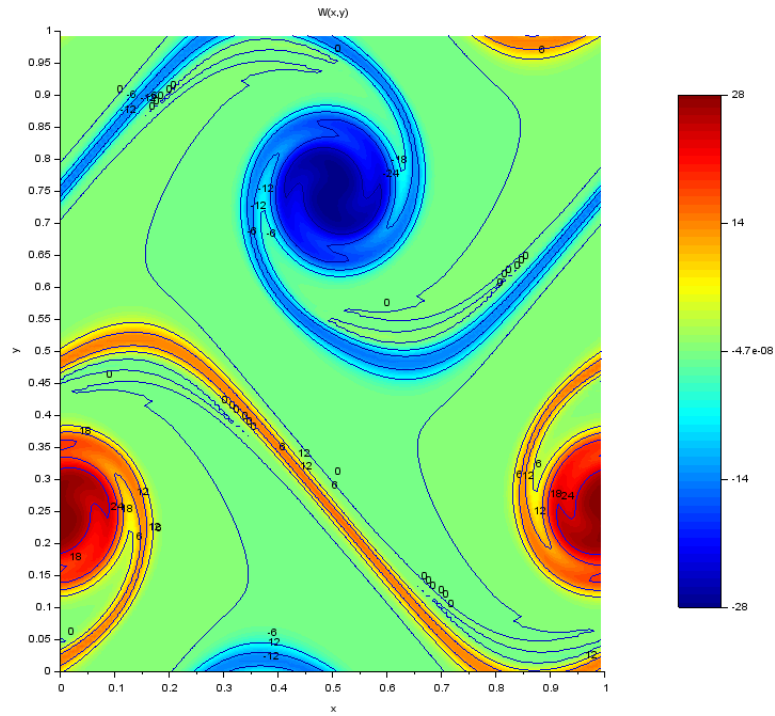


FIGURE 15 – Isocontours à $t=1.2s$

Les résultats obtenus sont bien concordant avec les résultats attendus dans le TP.

2.4 Question 14

On fait varier les paramètres de simulation :

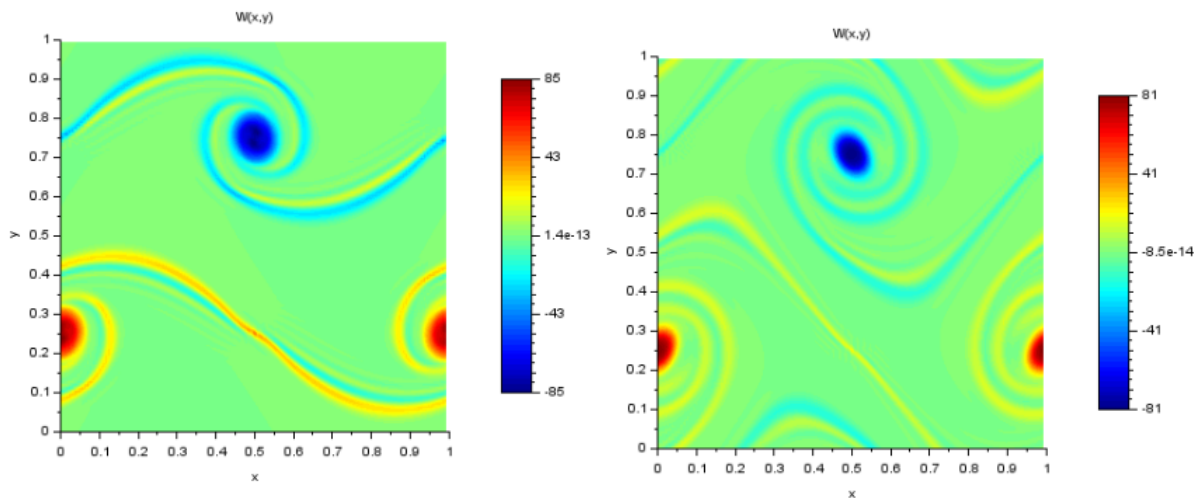


FIGURE 16 – Solution du système à $t=0.8s$ puis à $t=1.2s$

On remarque que avec ces paramètres le vortex se crée plus rapidement mais conserve le même point central de rotation. On remarque que ce point central dépend des conditions initiales puisqu'on retrouve la même valeur $y_0 = 0.75$. On observe une certaine périodicité spatiale qui concorde bien avec notre modèle. Le résultat est donc satisfaisant. Cependant cette nouvelle simulation prend 294 itérations au lieu de 240 pour la première. Il est donc plus difficile de la simuler.

Quant aux isocontours :

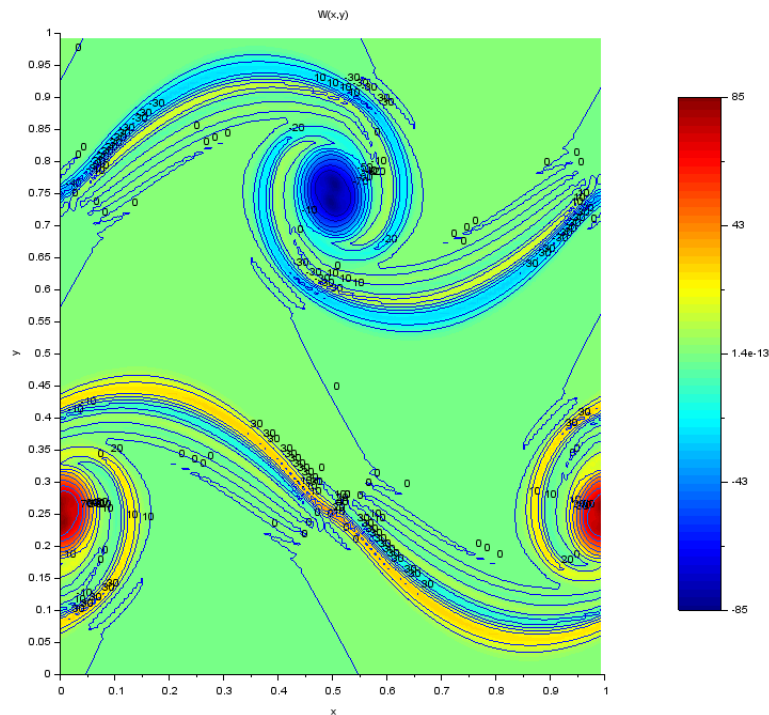


FIGURE 17 – Isocontours à $t=0.8s$

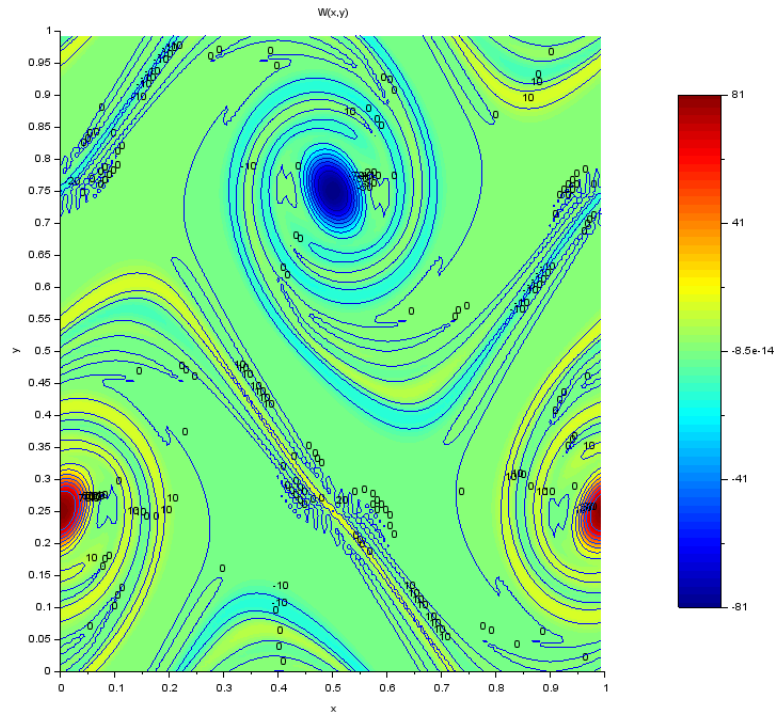


FIGURE 18 – Isocontours à $t=1.2s$

Malheureusement, il n'a pas été possible d'augmenter la résolution spatiale. Après avoir essayé de doubler la résolution, mon ordinateur n'a malheureusement pas survécu à la simulation et le scilab a crashé. La courbe présentée ici provient d'une autre machine beaucoup plus puissante (et qui ne fait pas crasher scilab, est ce à cause de windows ?).

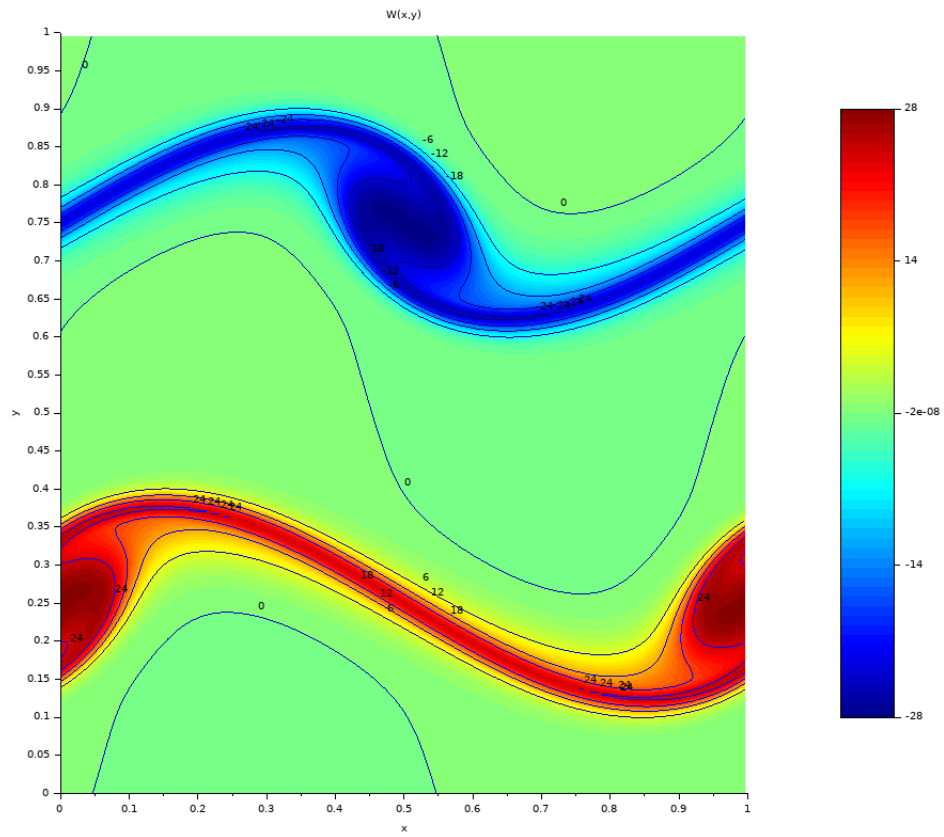


FIGURE 19 – Isocontours à $t=0.8s$

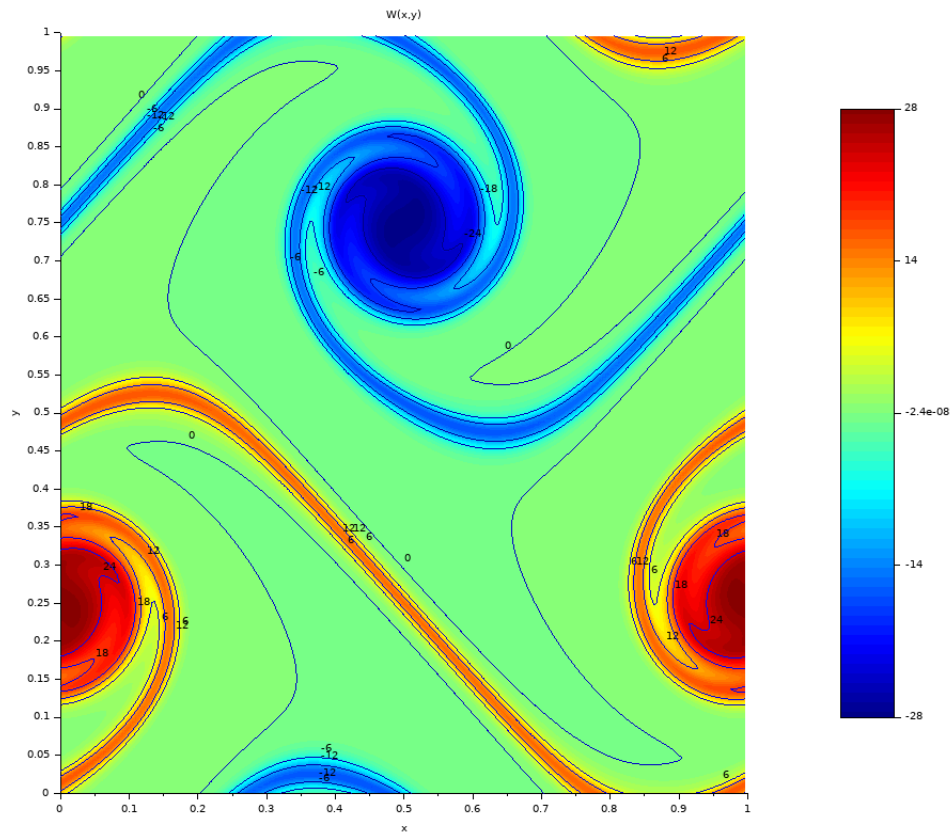


FIGURE 20 – Isocontours à $t=1.2s$

On remarque effectivement qu'avec une résolution plus grande les isocontours sont plus lisses.

2.5 Question 15

Ici nous nous intéresserons au tracé du champ des vitesses. Pour ça on modifie le code pour utiliser `champ1` qui peut renseigner sur les intensités par code couleur.

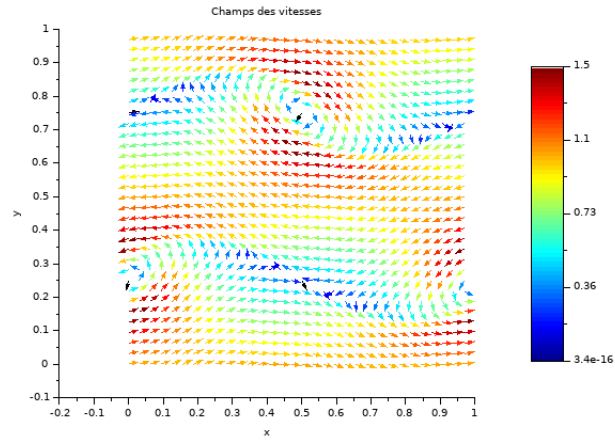


FIGURE 21 – Champ des vitesses à $t=0.8s$

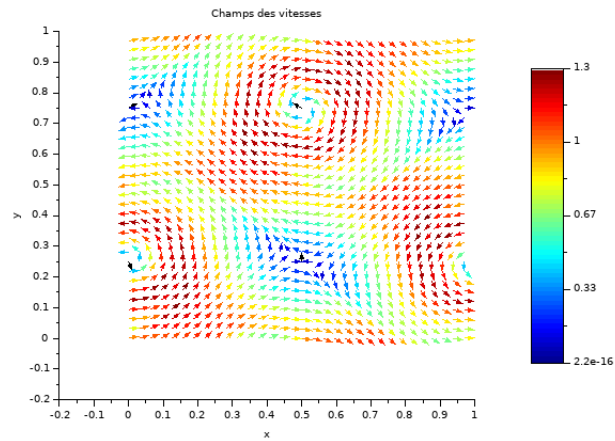


FIGURE 22 – Champ des vitesses à $t=1.2s$

Le champ des vitesses est effectivement en accord avec ce que l'on a obtenu précédemment. IL y a bien rotation autour du point central puisque les vecteurs se dirigent vers ce point là en tangeantant un cercle de rayon de plus en plus petit et plus on s'éloigne du centre, plus la vitesse du fluide est élevée.