

# TER : Analyse numérique et simulation de systèmes de réaction-diffusion

Hélène BLOCH - Thomas LENNE

Encadrant : Claire CHAINAIS

28 mai 2017

# Table des matières

<b>Introduction</b>	<b>2</b>
<b>1 Mise en place globale et application au cas à une dimension</b>	<b>4</b>
1.1 Résolution du système simplifié . . . . .	4
1.1.1 Discrétisation et schéma différences finies . . . . .	4
1.1.2 Validation, calculs d'erreurs . . . . .	6
1.2 Résolution du système complet . . . . .	9
1.3 Validation, calculs d'erreurs . . . . .	10
1.3.1 Système découplé . . . . .	11
1.3.2 Système couplé . . . . .	12
<b>2 Résolution du système en deux dimensions</b>	<b>14</b>
2.1 Condition aux bords de Dirichlet homogène . . . . .	14
2.1.1 Discrétisation et schéma différences finies avec condition aux bords de Dirichlet homogène . . . . .	14
2.1.2 Validation, calculs d'erreurs . . . . .	15
2.2 Condition aux bords périodique . . . . .	18
2.2.1 Discrétisation et schéma différences finies avec condition aux bords périodique . . . . .	18
2.2.2 Validation, calculs d'erreurs . . . . .	19
<b>3 Recherche des taches</b>	<b>24</b>
3.1 Recherche de motifs à partir de conditions initiales spécifiques . . . . .	25
3.2 Recherche de motifs à partir du régime permanent . . . . .	26
<b>4 Documentation du code</b>	<b>28</b>
<b>5 Conclusion</b>	<b>30</b>
<b>A Schéma différences finies du système à une équation et à une dimension</b>	<b>31</b>
<b>B Jacobienne du schéma du système à une équation et à une dimension</b>	<b>31</b>
<b>C Schéma différences finies du système à deux équations et à une dimension</b>	<b>32</b>
<b>D Jacobienne du schéma du système à deux équations et à une dimension</b>	<b>32</b>
<b>E Schéma différences finies à deux dimensions, condition aux bords de Dirichlet homogène</b>	<b>34</b>
<b>F Jacobienne du schéma du système à deux dimensions, condition aux limites de Dirichlet homogène</b>	<b>35</b>
<b>G Schéma différences finies à deux dimensions, condition aux limites périodique</b>	<b>37</b>
<b>H Jacobienne du schéma du système à deux dimensions, condition aux limites périodique</b>	<b>38</b>

# Introduction



## Modélisation des motifs

Les mammifères exhibent des motifs de fourrure extrêmement variés et complexes. Ces motifs ont été le sujet de moult études zoologiques qui ont pu noter l'incroyable variété de formes et figures présentes dans le monde animal.

Murray, dans son livre *Mathematical biology*, a étudié la formation de ces motifs et d'après lui la plupart des formes observées pourraient être causées par un unique phénomène. La pigmentation de la peau et des poils est due à la mélanine produite par les mélanocytes, des cellules dédiées à la production de cette mélanine. Cette production est liée au patrimoine génétique (différentes espèces animales) et à l'environnement (exposition aux UV chez l'homme par exemple). Le modèle se base sur l'hypothèse que les motifs de fourrure des mammifères ne font que refléter les disparités spatiales et temporelles des concentrations en morphogènes dans l'épiderme (pour la couleur de la peau et des poils). Les morphogènes sont des molécules fondamentales dans le développement cellulaire car elles permettent la spécification des différents types de cellules d'un organisme, leur localisation dans l'espace ainsi que leur orientation, il s'agit aussi d'une forme de repérage spatial biologique.

Murray utilise un système de réaction-diffusion (qui peut être instable) et explique l'écartement des taches dans les pelages par les pré-schémas existants de répartitions des morphogènes. C'est-à-dire que les prédispositions génétiques créent une forme d'esquisse de la forme des taches durant la période embryonnaire ; puis, la mélanine, dont la concentration est soumise aux phénomènes de réaction-diffusion, se disperse et diffuse pour obtenir les motifs que l'on observe dans la nature.

Pour la formation des motifs dans la fourrure, on applique ce système de réaction-diffusion à deux morphogènes qui entrent en jeu dans la production (ou non production) de mélanine au niveau des mélanocytes : les substrats d'oxygène et les enzymes uricases. Elles jouent respectivement le rôle d'activateur et d'inhibiteur dans les réactions chimiques qui régissent la production de mélanine.

Ce système de réaction-diffusion adimensionnel s'écrit alors de la manière suivante :

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u - f(u, v) = 0 \\ \frac{\partial v}{\partial t} - d\Delta v - g(u, v) = 0 \end{cases} \quad (1)$$

avec

$$f(u, v) = \gamma \left( a - u - \frac{\rho uv}{1 + u + Ku^2} \right)$$

$$g(u, v) = \gamma \left( \alpha(b - v) - \frac{\rho uv}{1 + u + Ku^2} \right)$$

- $u$  est proportionnel à la concentration en substrat d'oxygène et  $v$  à la concentration en enzyme uricase
- $\alpha, a, b, \rho$  sont des paramètres positifs
- $d > 0$  est le terme de diffusion « relatif »
- $K > 0$  est un terme positif décrivant la force du caractère inhibiteur de  $v$
- $\gamma$  est un paramètre positif proportionnel au carré de l'aire du domaine, c'est le terme qui permet de dimensionner le domaine d'étude

## Simulation numérique et objectifs

Dans leur article, *Convergence of the Peaceman-Rashford approximation for reaction diffusion system*, 2003, Stéphane Descombes et Magali Ribot présentent une méthode pour résoudre numériquement ces équations sans utiliser les schémas d'Euler pour éviter les contraintes qu'ils imposent :

- $\Delta t = \mathcal{O}(\Delta x^2)$  pour assurer la stabilité du schéma d'Euler explicite
- la résolution de systèmes non linéaires pour le schéma d'Euler implicite

Le schéma d'Euler implicite étant inconditionnellement stable, nous allons nous intéresser à la résolution de (1) par la méthode des différences finies.

Afin de faciliter l'implémentation dans le code `Fortran 90`, nous allons procéder par étapes pour augmenter progressivement la complexité du problème. D'abord, nous allons résoudre numériquement le problème à une dimension avec la condition aux limites de Dirichlet homogène. Puis, nous résoudrons le système de réaction-diffusion en 2 dimensions avec la condition aux bords de Dirichlet homogène dans un second temps. Enfin, nous implémenterons la condition aux limites périodique à nos schémas numériques.

L'objectif de notre TER est de résoudre numériquement le système de réaction-diffusion en une dimension, puis deux dimensions et finalement de parvenir à représenter graphiquement des motifs d'animaux.

# 1 Mise en place globale et application au cas à une dimension

## 1.1 Résolution du système simplifié

### 1.1.1 Discrétisation et schéma différences finies

Le système (1) étant compliqué par rapport aux systèmes que nous avons jusqu'à présent résolus, il nous a fallu procéder par étapes successives afin de mettre en place les différentes méthodes itératives et bien les appréhender avant de les appliquer dans les cas plus compliqués qui nous permettront d'obtenir des résultats expliquant les taches de mammifères. Dans un premier temps, nous allons donc nous limiter à un problème 1D sur l'intervalle  $\Omega = [0, 1]$  sans couplage des équations et avec la conditions aux limites de Dirichlet homogène. Pour découpler les équations tout en conservant la non-linéarité du problème, nous avons modifié le problème de façon arbitraire de la manière suivante :

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - f(u) = 0 \quad (2)$$

avec

$$f(u) = u(1 - u)$$

Nous avons choisi  $f$  sous cette forme pour qu'elle soit nulle sur les bords de  $\Omega$  et non linéaire.

On discrétise  $\Omega = [0, 1]$  en  $I + 1$  intervalles avec le pas de discrétisation  $\Delta x = \frac{1}{I+1}$  et le temps avec un pas  $\Delta t$ . Ainsi, on note  $u_i^n$ ,  $i = \{1, \dots, I\}$ ,  $n \in \mathbb{N}$  l'approximation de  $u(i\Delta x, n\Delta t)$ .

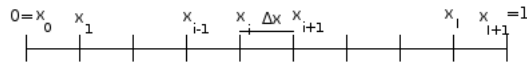


FIGURE 1 – Discrétisation de l'espace en différences finies

Ainsi, le problème simplifié précédent discrétisé sous le schéma d'Euler implicite est écrit en annexe A.

Nous avons donc un système non linéaire de  $I$  équations à  $I$  inconnues à résoudre à chaque pas de temps.

Nous effectuerons par la suite toutes les résolutions de systèmes non linéaires par l'algorithme de Newton.

Soit  $F : \mathbb{R}^I \rightarrow \mathbb{R}^I$ , on veut résoudre  $F(x) = 0$ ,  $x \in \mathbb{R}^I$ .

$F \in \mathcal{C}^2(\mathbb{R}^I, \mathbb{R}^I)$ , on note  $DF(x) \in \mathcal{M}_{\mathcal{I}}(\mathbb{R})$  la jacobienne de  $F$  évaluée en  $x$  et  $\bar{x} \in \mathbb{R}^I$  tel que  $F(\bar{x}) = 0$ .

L'algorithme de Newton s'écrit alors de la manière suivante :

---

**Algorithm 1** Algorithme de Newton

---

$\mathbf{x}_0$  donné ,  $\epsilon$  la tolérance donnée,  $N$  le nombre maximum d'itérations donné  
 $k = 0$   
**while**  $k \leq N$  et  $\epsilon < \|\mathbf{x}_{k+1} - \mathbf{x}_k\|$  **do**  
     $k = k + 1$   
    résoudre  $DF(\mathbf{x}_k)\delta\mathbf{x}_k = F(\mathbf{x}_k)$   
    poser  $\mathbf{x}_{k+1} = \mathbf{x}_k - \delta\mathbf{x}_k$   
**end while**

---

Si  $DF(\bar{x})$  est inversible alors il existe  $\delta > 0$  tel que  $\forall x_0 \in B(\bar{x}, \delta)$ , la suite  $\{x_k\}_{k \in \mathbb{N}}$  converge vers  $\bar{x}$ .

En contrepartie, à chaque itération, elle nécessite le calcul de la jacobienne et la résolution d'un système linéaire. De plus, il faut que  $x_0$  soit proche de la solution du système pour avoir la convergence quadratique. En pratique, on atteint la solution avec une erreur inférieure à  $10^{-6}$  en moins de 5 itérations.

Dans ce cas, la jacobienne donnée en annexe B est symétrique et creuse. Nous avons choisi un stockage adapté à la structure de la matrice : un stockage profil.

À chaque itération de l'algorithme de Newton, il faut résoudre un système linéaire. Ici, le système étant symétrique, nous avons choisi d'utiliser la décomposition de Cholesky  $LDL^T$  où  $L$  est une matrice triangulaire inférieure avec des 1 sur la diagonale.

---

**Algorithm 2** Résolution d'un système linéaire  $A\mathbf{x} = \mathbf{b}$  avec la décomposition de Cholesky

---

calculer la décomposition de Cholesky de  $A$  :  $A = LDL^T$   
résoudre  $L\mathbf{z} = \mathbf{b}$   
résoudre  $D\mathbf{y} = \mathbf{z}$   
résoudre  $L^T\mathbf{x} = \mathbf{y}$   
**return**  $\mathbf{x}$

---

Nous avons retenu cette méthode car elle est directe et son coût algorithmique  $\frac{1}{3}I^3 + \mathcal{O}(I^2)$  est acceptable dans la mesure où nous souhaitons essentiellement nous concentrer sur la méthode de Newton dans un premier temps. De plus, nous n'allons pas utiliser un maillage très raffiné, car nous ne cherchons pas de résultats avec ce modèle.

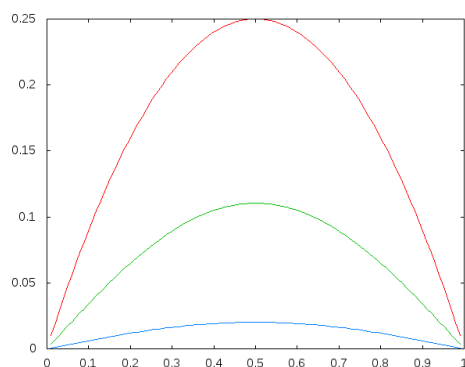


FIGURE 2 – Solution numérique  $u$  du système (2) à  $t = 0$ ,  $t = 0.1$  et  $t = 0.3$

Nous pouvons observer que la condition de Dirichlet homogène et le caractère diffusif du système sont bien respectés.

### 1.1.2 Validation, calculs d'erreurs

#### Ordres de convergence

Nous allons tester les ordres de convergence de la discrétisation différences finies en comparant la solution calculée numériquement à la solution exacte d'une équation aux dérivées partielles de forme similaire. Nous cherchons alors à résoudre numériquement le problème suivant :

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - f(u) = 0$$

avec

$$f(u) = u$$

dont la solution exacte est :

$$u_{ex}(x, t) = \exp(-(\pi^2 - 1)t) \sin(\pi x)$$

Nous utilisons une formule d'erreur cherchant à représenter la norme  $L^2$  pour notre problème discrétisé. Pour cela, on calcule la norme  $L^2$  d'une fonction  $u^n$  constante par morceaux associée à un vecteur  $(U_i^n)_{0 \leq i \leq I}$  telle que  $u^n(x_i) = U_i^n$  avec  $x_i = i\Delta x$ . On définit alors la norme  $L^2$  de  $U^n$  par :

$$\|U^n\|_{L^2} = \|u^n\|_{L^2} = \sqrt{\Delta x \sum_{i=1}^I (u_i^n)^2}$$

Ainsi, on peut calculer l'erreur d'approximation en évaluant la norme de l'erreur entre la solution approchée et la solution exacte. Donc, en notant, pour l'instant  $n\Delta t$ ,  $U^n$  le vecteur solution approchée et  $U_{ex}^n$  le vecteur solution exacte, l'erreur est alors donnée par  $\|U^n - U_{ex}^n\|_{L^2}$ .

Nous allons vérifier les ordres de convergence en espace et en temps. Ici, nous nous attendons à observer une convergence d'ordre 2 en espace et d'ordre 1 en temps.

L'erreur étant un  $\mathcal{O}(\Delta t + \Delta x^2)$ , pour tester l'un des deux ordres de convergence, il nous faut nous assurer que l'erreur associée à l'autre pas de discrétisation soit inférieure, afin qu'il n'y ait pas de phénomène de saturation. Ainsi, pour tester la convergence en temps, nous nous assurerons que  $\Delta x^2 < \Delta t$ . De même, pour tester la convergence en espace, nous nous assurerons que  $\Delta t < \Delta x^2$ .

Pour vérifier l'ordre de convergence en temps, nous calculons l'erreur à un temps final donné, puis nous divisons par 2 le pas de temps  $\Delta t$  avec un pas d'espace  $\Delta x$  fixé vérifiant  $\Delta x^2$  inférieur à tous les  $\Delta t$  testés. En traçant en échelle logarithmique sur la figure 3, l'erreur en fonction du pas de temps ainsi que la courbe d'équation  $y = x$ , nous observons bien la convergence d'ordre 1 en temps.

Pour vérifier l'ordre de convergence en espace, nous calculons l'erreur à un temps final donné, puis nous divisons par 2 le pas d'espace  $\Delta x$  avec un pas d'espace  $\Delta t$  fixé vérifiant  $\Delta t$  inférieur à tous les  $\Delta x^2$  testés. En traçant en échelle logarithmique sur la figure 4, l'erreur en fonction du pas d'espace ainsi que la courbe d'équation  $y = x^2$ , nous observons bien la convergence d'ordre 2 en espace.

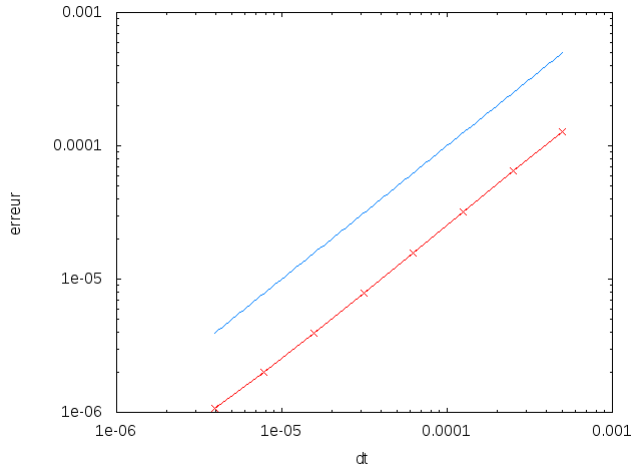


FIGURE 3 – Taux de convergence en temps en échelle logarithmique de  $u$  et droite  $y = x$

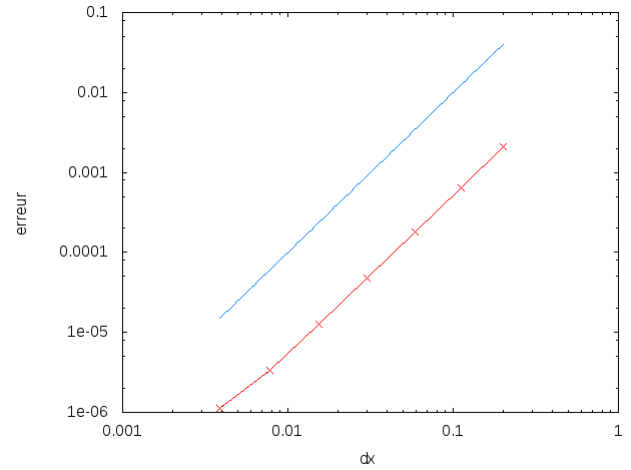


FIGURE 4 – Taux de convergence en espace en échelle logarithmique de  $u$  et droite  $y = x^2$

### Vérification de l'algorithme de Newton

Nous allons tester notre implémentation de l'algorithme de Newton. Nous cherchons alors à résoudre numériquement le problème suivant :

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - f(u) = 0$$

avec

$$f(u) = u - u^3$$

dont la solution exacte sur un domaine non borné est :

$$u(x, t) = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{x + \frac{3t}{\sqrt{2}}}{2\sqrt{2}}\right)$$

Pour vérifier la résolution de système non linéaire, nous adaptons le code pour discrétiser l'intervalle  $\Omega = [-10, 10]$  et résolvons le système précédent sur  $\Omega$ . Étant donné que nous avons pour l'instant seulement implémenté une condition limite de Dirichlet homogène, nous devrions pouvoir obtenir une bonne solution sauf sur les bords du domaine  $\Omega$ .

Sur la figure 5, nous observons la solution numérique ainsi que la solution exacte, nous pouvons noter que les deux solutions sont quasiment confondues tant que les bords de l'intervalle  $\Omega$  sont suffisamment éloignées.



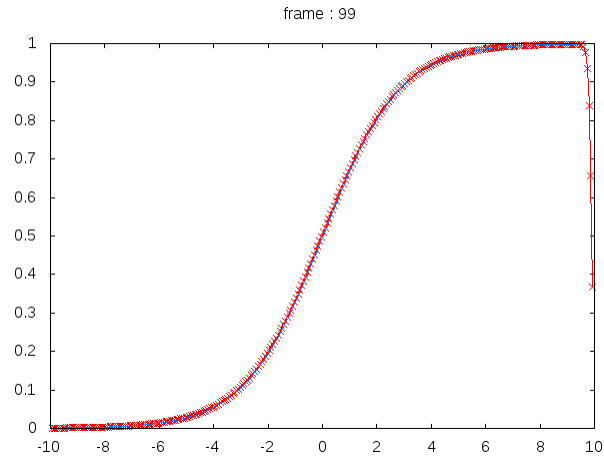


FIGURE 5 – Solution numérique et solution exacte à  $t = 0.01$

Nous pouvons tester aussi l'ordre de convergence de la méthode de Newton :

$\forall k \in \mathbb{N}$ , on note  $u^k$  la  $k$ -ième itération de l'algorithme de Newton et  $\bar{u}$  la discrétisation de  $u(\cdot, t)$ . On vérifie la convergence d'ordre 2 de la méthode en s'assurant que  $\exists C > 0$  tel que

$$\forall k \in \mathbb{N}, \frac{|u^{k+1} - \bar{u}|}{|u^k - \bar{u}|^2} \leq C$$

Pour s'affranchir du problème de la condition limite, nous calculons l'erreur uniquement sur l'intervalle  $[-5, 5]$ . Sur la figure 6, nous traçons le rapport  $\frac{|u^{k+1} - \bar{u}|}{|u^k - \bar{u}|^2}$  en fonction du nombre d'itérations de la méthode de Newton.

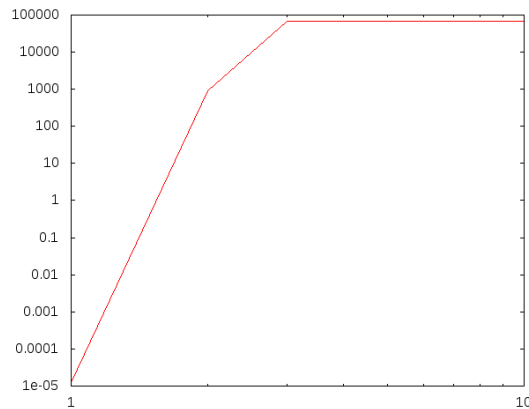


FIGURE 6 –  $\frac{|u^{k+1} - \bar{u}|}{|u^k - \bar{u}|^2}$  en fonction du nombre d'itérations

Le rapport  $\frac{|u^{k+1} - \bar{u}|}{|u^k - \bar{u}|^2}$  est donc bien borné, nous observons alors bien la convergence d'ordre 2 de la méthode. Nous pouvons donc conclure que notre algorithme de Newton est correctement implémenté.

## 1.2 Résolution du système complet

Nous allons maintenant implémenter le couplage des deux équations, toujours en 1D.

$$\begin{cases} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - f(u, v) = 0 \\ \frac{\partial v}{\partial t} - d \frac{\partial^2 v}{\partial x^2} - g(u, v) = 0 \end{cases} \quad (3)$$

avec

$$f(u, v) = \gamma \left( a - u - \frac{\rho uv}{1 + u + Ku^2} \right)$$
$$g(u, v) = \gamma \left( \alpha(b - v) - \frac{\rho uv}{1 + u + Ku^2} \right)$$

En gardant les mêmes discrétisations en temps et en espace, les schémas de différences finies sont écrits en annexe C

Par soucis de simplicité dans le repérage des inconnues, nous avons décidé de les écrire dans l'ordre suivant :

$$(u_1, \dots, u_I, v_1, \dots, v_I)^T$$

Nous avons à nouveau un système non linéaire de  $2I$  inconnues à  $2I$  équations à résoudre. Pour appliquer l'algorithme de Newton à ce système, il faut calculer sa jacobienne (voir annexe F).

La jacobienne n'étant pas symétrique en général, il nous faut changer de stockage et utiliser une autre méthode de résolution de système linéaire. La matrice restant creuse, nous utilisons un stockage Compress Sparse Row (CSR) qui est adapté à ce type de matrices. En effet, ce stockage réduit la matrice à trois tableaux : le premier contient les coefficients non nuls de la matrice, le second sert à repérer le premier coefficient de chaque ligne de la matrice et le troisième repère la colonne de chaque coefficient.

Une méthode directe pour la résolution du système linéaire comme la factorisation LU ne convient pas car la structure creuse de la matrice n'est pas conservée. Cela complexifierait beaucoup le code par rapport à une méthode itérative.

Pour résoudre le système linéaire  $Ax = b$ , où  $A$  est inversible, nous résolvons le système  $A^T Ax = A^T b$ . La matrice  $A^T A$  étant symétrique définie positive, nous pouvons appliquer l'algorithme du gradient conjugué à ce système. Pour éviter de calculer  $A^T A$ , nous utilisons l'algorithme du gradient conjugué résidu normal.

---

**Algorithm 3** Résolution d'un système linéaire  $Ax = b$  avec l'algorithme du gradient conjugué résidu normal

---

$\mathbf{x}_0$  donné,  $\epsilon$  la tolérance donnée,  $N$  le nombre maximum d'itérations donné  
 $k = 0$   
 $\mathbf{x} = \mathbf{x}_0$   
 $\mathbf{r} = \mathbf{b} - A\mathbf{x}$   
 $\mathbf{p} = A^T \mathbf{r}$   
 $test = \epsilon^2 \|\mathbf{b}\|^2$   
 $\rho = \|\mathbf{p}\|^2$   
**while**  $\rho > test$  et  $k < N$  **do**  
   $k = k + 1$   
   $\mathbf{v} = A\mathbf{p}$   
   $\alpha = \frac{\rho}{\|\mathbf{v}\|^2}$   
   $\mathbf{x} = \mathbf{x} + \alpha\mathbf{p}$   
   $\mathbf{r} = \mathbf{r} - \alpha\mathbf{v}$   
   $\mathbf{w} = A^T \mathbf{r}$   
   $\rho_{new} = \|\mathbf{w}\|^2$   
   $\beta = \frac{\rho_{new}}{\rho}$   
   $\mathbf{p} = \mathbf{w} + \beta\mathbf{p}$   
   $\rho = \rho_{new}$   
**end while**

---

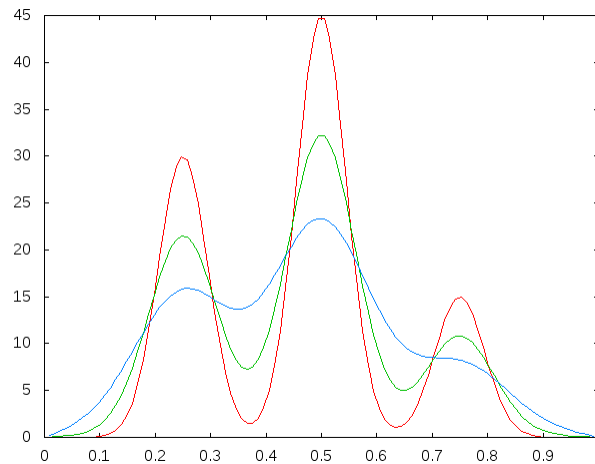


FIGURE 7 – Solution numérique  $u$  du système (1) à  $t = 0$ ,  $t = 0.001$  et  $t = 0.003$

### 1.3 Validation, calculs d'erreurs

Pour valider le code à cette étape de l'implémentation, nous allons le confronter à des conditions initiales et un second membre du problème dont nous pouvons trouver une solution analytique. Nous pourrions alors estimer l'erreur entre la solution numérique et la solution exacte et décider de valider ou non le code. Pour les calculs d'erreurs, nous posons le paramètre  $d$  égal à 2.

### 1.3.1 Système découplé

Testons le code en utilisant un second membre sans couplage entre  $u$  et  $v$  :

$$\begin{cases} \frac{\partial u}{\partial t} - \partial_{xx}u - f(u, v) = 0 \\ \frac{\partial v}{\partial t} - d \partial_{xx}v - g(u, v) = 0 \end{cases} \quad (4)$$

avec

$$f(u, v) = u$$

$$g(u, v) = v$$

Fixons la condition initiale  $u_0(x) = v_0(x) = \sin(\pi x)$ ,  $\forall x \in \Omega$  et cherchons  $u$  et  $v$  sous la forme :

$$\begin{cases} u(x, t) = F(t) \sin(\pi x) \\ v(x, t) = G(t) \sin(\pi x) \end{cases}$$

En injectant cette solution dans le système (4), nous aboutissons aux deux équations différentielles ordinaires suivantes :

$$\begin{cases} F'(t) + (\pi^2 - 1)F(t) = 0 \\ G'(t) + (d\pi^2 - 1)G(t) = 0 \end{cases}$$

dont les solutions sont :

$$\begin{cases} u(x, t) = \exp(-(\pi^2 - 1)t) \sin(\pi t) \\ v(x, t) = \exp(-(d\pi^2 - 1)t) \sin(\pi t) \end{cases}$$

Nous disposons donc de la solution exacte du problème. Nous pouvons alors la comparer avec la solution numérique.

Pour cela, nous mesurons l'erreur en norme  $L^2$ . Nous vérifions les ordres de convergence en espace et en temps. Ici, nous nous attendons à observer une convergence d'ordre 1 en temps et d'ordre 2 en espace que ce soit pour  $u$  ou pour  $v$ .

Sur la figure 8, nous traçons, en échelle logarithmique, l'erreur en fonction du pas de temps ainsi que la droite d'équation  $y = x$ . Nous observons bien l'ordre 1 en temps pour  $u$  et pour  $v$ .

Sur la figure 9, nous traçons, en échelle logarithmique, l'erreur en fonction du pas d'espace ainsi que la courbe d'équation  $y = x^2$ . Nous observons bien l'ordre 2 en espace pour  $u$  et pour  $v$ .

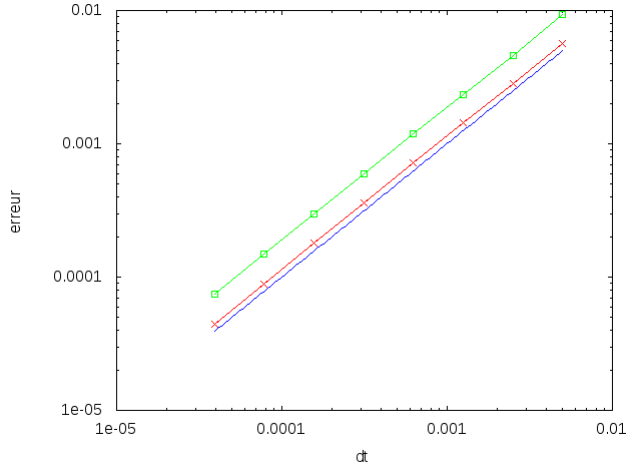


FIGURE 8 – Taux de convergence en temps en échelle logarithmique de  $u$ , de  $v$  et droite  $y = x$

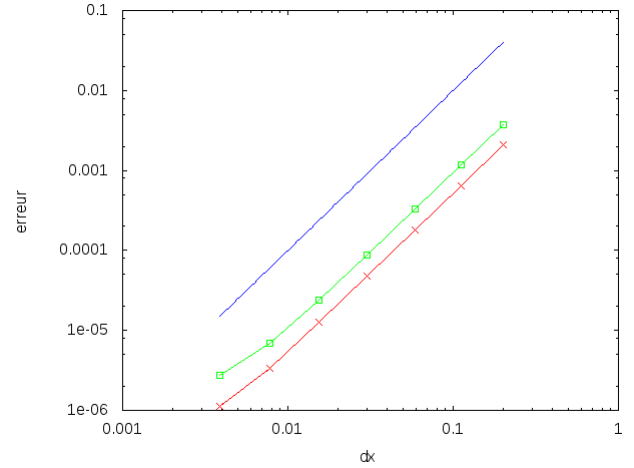


FIGURE 9 – Taux de convergence en espace en échelle logarithmique de  $u$ , de  $v$  et droite  $y = x^2$

### 1.3.2 Système couplé

Utilisons un second membre couplant les équations :

$$\begin{cases} \frac{\partial u}{\partial t} - \partial_{xx}u - f(u, v) = 0 \\ \frac{\partial v}{\partial t} - d \partial_{xx}v - g(u, v) = 0 \end{cases} \quad (5)$$

avec

$$f(u, v) = u + v$$

$$g(u, v) = u + v$$

Fixons la condition initiale  $u_0(x) = v_0(x) = \sin(\pi x)$ ,  $\forall x \in \Omega$  et cherchons  $u$  et  $v$  sous la forme :

$$\begin{cases} u(x, t) = F(t) \sin(\pi x) \\ v(x, t) = G(t) \sin(\pi x) \end{cases}$$

En injectant dans le système, nous obtenons le système différentiel :

$$\begin{pmatrix} F'(t) \\ G'(t) \end{pmatrix} = \begin{pmatrix} -\pi^2 + 1 & 1 \\ 1 & -d\pi^2 + 1 \end{pmatrix} \begin{pmatrix} F(t) \\ G(t) \end{pmatrix}$$

En notant  $\lambda_1$  et  $\lambda_2$  les valeurs propres de la matrice  $A = \begin{pmatrix} -\pi^2 + 1 & 1 \\ 1 & -d\pi^2 + 1 \end{pmatrix}$  et  $v_1 = (x_1, y_1)$  et  $v_2 = (x_2, y_2)$  les vecteurs propres associés, la solution du système différentiel est :

$$\begin{pmatrix} F(t) \\ G(t) \end{pmatrix} = \begin{pmatrix} x_1 & x_2 \\ y_1 & y_2 \end{pmatrix} \begin{pmatrix} e^{\lambda_1 t} & 0 \\ 0 & e^{\lambda_2 t} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$$

avec  $\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \in \mathbb{R}^2$

On détermine  $\begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$  en utilisant les conditions  $F(0) = 1$  et  $G(0) = 1$ .

Ici,  $\lambda_1 = -18.8395$ ,  $\lambda_2 = -8.76875$ ,  $v_1 = \begin{pmatrix} -0.0997957 \\ 0.995008 \end{pmatrix}$  et  $v_2 = \begin{pmatrix} -0.995008 \\ -0.099795 \end{pmatrix}$ . On a donc  $\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 0.8952 \\ -1.0948 \end{pmatrix}$ . Finalement,

$$\begin{pmatrix} F(t) \\ G(t) \end{pmatrix} = \begin{pmatrix} -0.08934e^{-18.8395t} - 1.08933e^{-8.76875t} \\ 0.89073e^{-18.8395t} + 0.10925e^{-8.76875t} \end{pmatrix}$$

Nous disposons donc de la solution exacte du problème. Nous pouvons alors la comparer avec la solution numérique.

Pour cela, nous mesurons l'erreur en norme  $L^2$ . Nous vérifions les ordres de convergence en espace et en temps. Ici, nous nous attendons à observer une convergence d'ordre 1 en temps et d'ordre 2 en espace que ce soit pour  $u$  ou pour  $v$ .

Sur la figure 10, nous traçons, en échelle logarithmique, l'erreur en fonction du pas de temps ainsi que la droite d'équation  $y = x$ . Nous observons bien l'ordre 1 en temps pour  $u$  et pour  $v$ .

Sur la figure 11, nous traçons, en échelle logarithmique, l'erreur en fonction du pas d'espace ainsi que la courbe d'équation  $y = x^2$ . Nous observons bien l'ordre 2 en espace pour  $u$  et pour  $v$ .

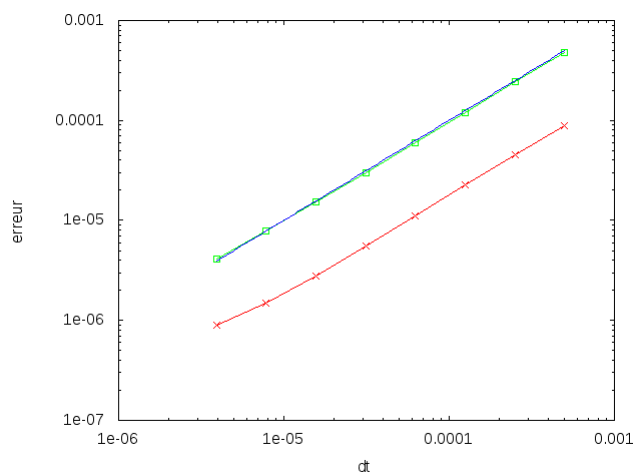


FIGURE 10 – Taux de convergence en temps en échelle logarithmique de  $u$ , de  $v$  et droite  $y = x$

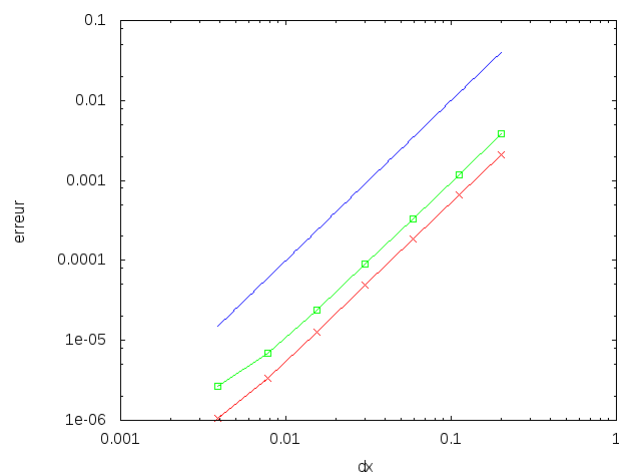


FIGURE 11 – Taux de convergence en espace en échelle logarithmique de  $u$ , de  $v$  et droite  $y = x^2$

## 2 Résolution du système en deux dimensions

### 2.1 Condition aux bords de Dirichlet homogène

#### 2.1.1 Discrétisation et schéma différences finies avec condition aux bords de Dirichlet homogène

Ayant mis en place l'essentiel des méthodes nécessaires à la résolution du système, nous pouvons nous intéresser à la résolution numérique du problème à deux dimensions.

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u - f(u, v) = 0 \\ \frac{\partial v}{\partial t} - d\Delta v - g(u, v) = 0 \end{cases} \quad (6)$$

avec

$$f(u, v) = \gamma \left( a - u - \frac{\rho uv}{1 + u + Ku^2} \right)$$

$$g(u, v) = \gamma \left( \alpha(b - v) - \frac{\rho uv}{1 + u + Ku^2} \right)$$

avec la condition aux limites de Dirichlet homogène.

On discrétise  $\Omega = [0, 1] \times [0, 1]$ . Soient  $I, J \in \mathbb{N}^*$ , on pose  $\Delta x = \frac{1}{I+1}$  et  $\Delta y = \frac{1}{J+1}$ . On discrétise le temps avec un pas  $\Delta t$ . Ainsi, on note  $u_{i,j}^n$ ,  $i = \{1, \dots, I\}$ ,  $j = \{1, \dots, J\}$ ,  $n \in \mathbb{N}$  l'approximation de  $u(i\Delta x, j\Delta y, n\Delta t)$ .

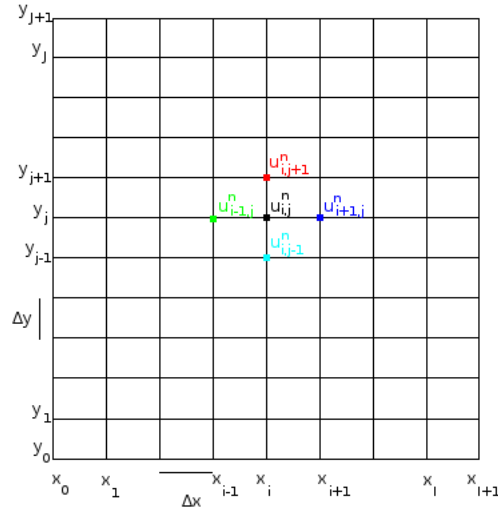


FIGURE 12 – Discrétisation de l'espace en différences finies

Ainsi, le problème simplifié précédent discrétisé sous le schéma d'Euler implicite est écrit en annexe E

Pour l'implémentation, nous rangeons les inconnues  $u_{i,j}^n$  et  $v_{i,j}^n$  pour  $i = \{1, \dots, I\}$ ,  $j = \{1, \dots, J\}$  dans l'ordre lexicographique avec  $u$  avant  $v$  :

$$(u_{1,1}, \dots, u_{I,1}, u_{2,1}, \dots, u_{1,J}, \dots, u_{I,J}, v_{1,1}, \dots, v_{I,1}, v_{2,1}, \dots, v_{1,J}, \dots, v_{I,J})^T$$

De la même manière que précédemment, nous résolvons un système non linéaire de  $2IJ$  équations à  $2IJ$  inconnues dont la jacobienne est donnée en annexe F. La matrice restant creuse et non symétrique, nous utilisons les mêmes méthodes que dans le cas précédent.

### 2.1.2 Validation, calculs d'erreurs

#### Système découplé

Pour valider le code, il nous faut le confronter à une solution exacte.

Dans un premier temps, nous allons chercher une solution particulière du système suivant :

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u - f(u, v) = 0 \\ \frac{\partial v}{\partial t} - d\Delta v - g(u, v) = 0 \end{cases} \quad (7)$$

avec

$$\begin{aligned} f(u, v) &= u \\ g(u, v) &= v \end{aligned}$$

Fixons la condition initiale  $u_0(x, y) = v_0(x, y) = \sin(\pi x) \sin(\pi y) \quad \forall (x, y) \in \Omega$  et cherchons  $u$  et  $v$  sous la forme :

$$\begin{cases} u(x, y, t) = F(t) \sin(\pi x) \sin(\pi y) \\ v(x, y, t) = G(t) \sin(\pi x) \sin(\pi y) \end{cases}$$

En injectant dans le système (7) ces solutions, il faut résoudre :

$$\begin{cases} F'(t) + (2\pi^2 - 1)F(t) = 0 \\ G'(t) + (2d\pi^2 - 1)G(t) = 0 \end{cases}$$

Les solutions sont alors :

$$\begin{cases} u(x, y, t) = \exp((1 - 2\pi^2)t) \sin(\pi x) \sin(\pi y) \\ v(x, y, t) = \exp((1 - 2d\pi^2)t) \sin(\pi x) \sin(\pi y) \end{cases}$$

Nous disposons donc de la solution exacte du problème. Nous pouvons alors la comparer avec la solution numérique.

Nous allons procéder comme précédemment. Toutefois, il nous faut adapter la formule de l'erreur en norme  $L^2$  à notre problème discrétisé à 2 dimensions. Pour cela, on calcule la norme  $L^2$  d'une fonction  $u^n$  constante par morceaux associée à un vecteur  $(U_i^n)_{0 \leq i \leq 2IJ}$  telle que  $u^n(x_i, y_j) = U_{(j-1)I+i}^n$  avec  $x_i = i\Delta x$  et  $y_j = j\Delta y$  (avec utilisation de l'ordre lexicographique). On définit alors la norme  $L^2$  de  $U^n$  par :

$$\|U^n\|_{L^2} = \|u^n\|_{L^2} = \sqrt{\Delta x \Delta y \sum_{i=1}^{IJ} (u_i^n)^2}$$

Ainsi, on peut calculer l'erreur d'approximation en évaluant la norme de l'erreur entre la solution approchée et la solution exacte. Donc, en notant, pour l'instant  $n\Delta t$ ,  $U^n$  le vecteur solution approchée et  $U_{ex}^n$  le vecteur solution exacte, l'erreur est alors donnée par  $\|U^n - U_{ex}^n\|_{L^2}$ .



Nous allons vérifier les ordres de convergence en espace et en temps. Ici, nous nous attendons à observer une convergence d'ordre 2 en espace et d'ordre 1 en temps. Nous traiterons la convergence en espace en une seule fois, c'est-à-dire en les faisant varier simultanément et de la même manière. Nous aurons alors  $\Delta x = \Delta y$ .

L'erreur étant un  $\mathcal{O}(\Delta t + \Delta x^2 + \Delta y^2)$ , pour tester l'un des deux ordres de convergence, il nous faut nous assurer que l'erreur associée à l'autre pas de discrétisation soit inférieure afin qu'il n'y ait pas de phénomène de saturation. Ainsi, pour tester la convergence en temps, nous nous assurerons que  $\Delta x^2 < \Delta t$ . De même, pour tester la convergence en espace, nous nous assurerons que  $\Delta t < \Delta x^2$ .

Pour vérifier l'ordre de convergence en temps, nous calculons l'erreur à un temps final donné puis nous divisons par 2 le pas de temps  $\Delta t$  avec un pas d'espace  $\Delta x$  fixé vérifiant  $\Delta x^2$  inférieur à tous les  $\Delta t$  testés. En traçant en échelle logarithmique sur la figure 13, l'erreur en fonction du pas de temps ainsi que la courbe d'équation  $y = x$ , nous observons bien la convergence d'ordre 1 en temps.

Pour vérifier l'ordre de convergence en espace, nous calculons l'erreur à un temps final donné puis nous divisons par 2 le pas d'espace  $\Delta x$  avec un pas d'espace  $\Delta t$  fixé vérifiant  $\Delta t$  inférieur à tous les  $\Delta x^2$  testés. En traçant en échelle logarithmique sur la figure 14, l'erreur en fonction du pas d'espace ainsi que la courbe d'équation  $y = x^2$ , nous observons bien la convergence d'ordre 2 en espace.

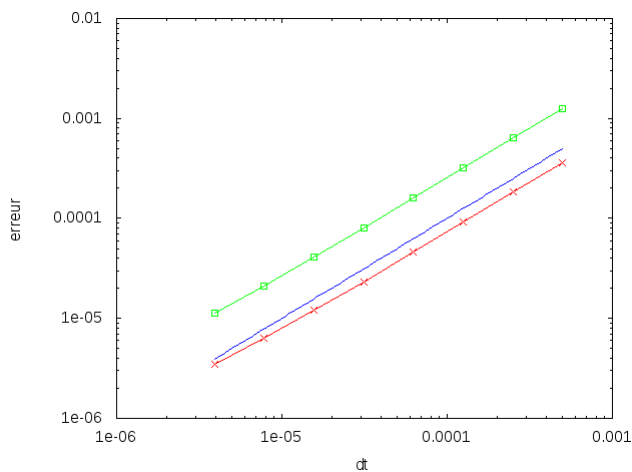


FIGURE 13 – Taux de convergence en temps en échelle logarithmique de  $u$ , de  $v$  et droite  $y = x$

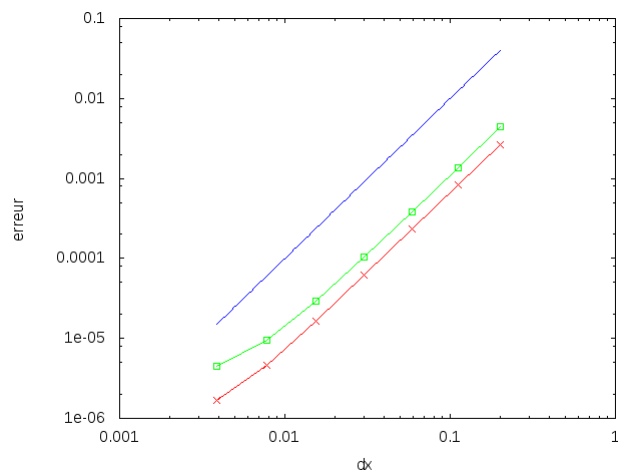


FIGURE 14 – Taux de convergence en espace en échelle logarithmique de  $u$ , de  $v$  et droite  $y = x^2$

## Système couplé

Ensuite, nous voulons vérifier que le couplage de  $u$  et  $v$  fonctionne comme prévu. Pour cela, nous allons résoudre le système suivant :

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u - f(u, v) = 0 \\ \frac{\partial v}{\partial t} - d\Delta v - g(u, v) = 0 \end{cases}$$

avec

$$\begin{aligned} f(u, v) &= u + v \\ g(u, v) &= u + v \end{aligned}$$

Fixons la condition initiale  $u_0(x, y) = v_0(x, y) = \sin(\pi x) \sin(\pi y) \quad \forall (x, y) \in \Omega$  et cherchons  $u$  et  $v$  sous la forme :

$$\begin{cases} u(x, y, t) = F(t) \sin(\pi x) \sin(\pi y) \\ v(x, y, t) = G(t) \sin(\pi x) \sin(\pi y) \end{cases}$$

En procédant de la même manière que précédemment, nous avons  $F$  et  $G$  solutions du système différentiel suivant :

$$\begin{pmatrix} F'(t) \\ G'(t) \end{pmatrix} = \begin{pmatrix} -2\pi^2 + 1 & 1 \\ 1 & -2d\pi^2 + 1 \end{pmatrix} \begin{pmatrix} F(t) \\ G(t) \end{pmatrix}$$

En procédant de la même manière que précédemment, nous obtenons  $\lambda_1 = -38.5289$ ,  $\lambda_2 = -18.6887$ ,  $v_1 = \begin{pmatrix} -0.0504669 \\ 0.998726 \end{pmatrix}$  et  $v_2 = \begin{pmatrix} -0.998726 \\ -0.0504666 \end{pmatrix}$ . On a donc  $\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 0.948259 \\ -1.04919 \end{pmatrix}$ .

Finalement,

$$\begin{pmatrix} F(t) \\ G(t) \end{pmatrix} = \begin{pmatrix} -0.04785571e^{-38.5289t} - 1.08933e^{-18.6887t} \\ 0.947051e^{-38.5289t} + 0.052948e^{-18.6887t} \end{pmatrix}$$

Nous disposons donc de la solution exacte du problème. Nous pouvons alors la comparer avec la solution numérique.

Nous vérifions les ordres de convergence théoriques de la même manière que précédemment. Sur la figure 15, nous retrouvons bien la convergence d'ordre 1 en temps et sur la figure 16, nous avons bien la convergence d'ordre 2 en espace.

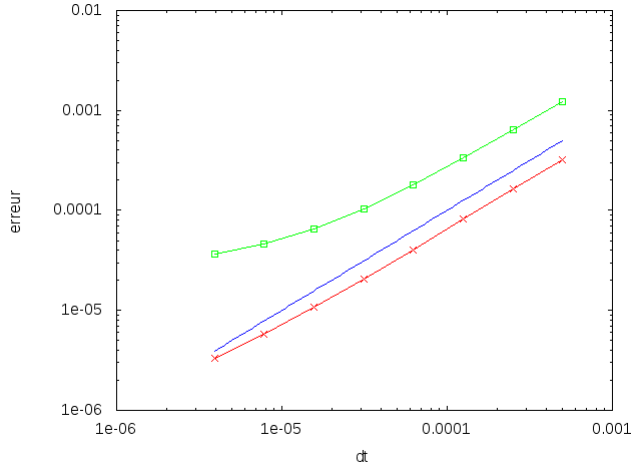


FIGURE 15 – Taux de convergence en temps en échelle logarithmique de  $u$ , de  $v$  et droite  $y = x$

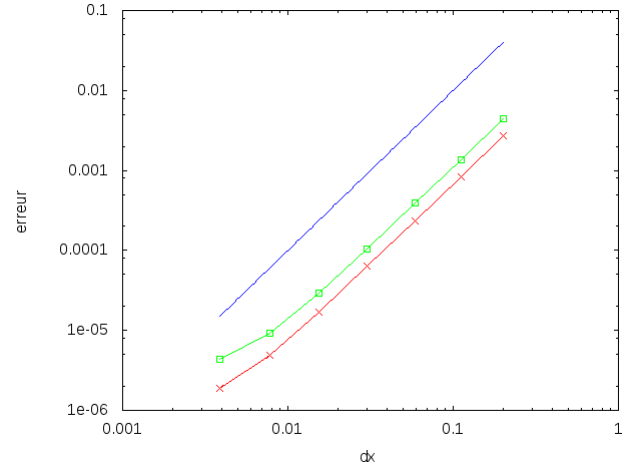


FIGURE 16 – Taux de convergence en espace en échelle logarithmique de  $u$ , de  $v$  et droite  $y = x^2$

## 2.2 Condition aux bords périodique

### 2.2.1 Discrétisation et schéma différences finies avec condition aux bords périodique

Pour permettre une meilleure modélisation du problème physique, nous implémentons la condition aux bords périodique. Cela permet fermer le domaine ou de rendre le domaine infini. De cette manière, nous pouvons modéliser plus correctement la géométrie d'un animal.

Nous réutilisons les notations utilisées précédemment. La condition périodique se traduit par :  $u_{0,j}^n = u_{I+1,j}^n, \forall j \in \{1, \dots, J+1\}$  et  $u_{i,0}^n = u_{i,J+1}^n, \forall i \in \{1, \dots, I+1\}$  et de même pour  $v$ . Nous décidons de considérer  $u_{I+1,j}^n$  et  $u_{i,J+1}^n$  comme inconnues par commodité avec l'indexation des tableaux en **Fortran 90**.

Le problème discrétisé sous le schéma d'Euler implicite est écrit en annexe G.

Ici, le système est constitué de  $2(I+1)(J+1)$  équations à  $2(I+1)(J+1)$  inconnues dont la jacobienne est donnée en annexe H.

La matrice étant creuse et non symétrique, nous avons voulu réutiliser les mêmes méthodes itératives que précédemment. Toutefois, nous constatons que l'algorithme du gradient conjugué résidu normal ne converge pas.

Afin de déterminer l'origine de la non-convergence de la méthode de résolution, nous avons essayé une autre méthode de résolution de système linéaire. Nous avons choisi d'utiliser la factorisation LU (méthode directe mais coûteuse du fait de la structure creuse de la matrice). L'implémentation de la méthode et son application au problème nous ont permis de prouver que la non-convergence du schéma numérique provenait bien du gradient conjugué résidu normal. En effet, nous avons observé graphiquement la périodicité aux bords.

L'hypothèse restante la plus vraisemblable est que la matrice jacobienne du système est mal conditionnée.

Pour continuer à exploiter le caractère creux de la matrice, nous avons ensuite implémenté la méthode du gradient biconjugué stabilisé (BiCGSTAB) :

---

**Algorithm 4** Résolution d'un système linéaire  $Ax = b$  avec l'algorithme BiCGSTAB

---

$\mathbf{x}_0$  donné,  $\epsilon$  la tolérance donnée,  $N$  le nombre maximum d'itérations donné  
 $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$   
 $\hat{\mathbf{r}}_0 = \mathbf{r}_0$   
 $\rho_0 = \alpha = \omega_0 = 1$   
 $\mathbf{v}_0 = \mathbf{p}_0 = \mathbf{0}$   
**for**  $i = 1$  à  $N$  **do**  
   $\rho_i = (\hat{\mathbf{r}}_0, \mathbf{r}_{i-1})$   
   $\beta = \frac{\rho_i \alpha}{\rho_{i-1} \omega_{i-1}}$   
   $\mathbf{p}_i = \mathbf{r}_{i-1} + \beta(\mathbf{p}_{i-1} - \omega_{i-1} \mathbf{v}_{i-1})$   
   $\mathbf{v}_i = A\mathbf{p}_i$   
   $\alpha = \frac{\rho_i}{(\hat{\mathbf{r}}_0, \mathbf{v}_i)}$   
   $\mathbf{h} = \mathbf{x}_{i-1} + \alpha \mathbf{p}_i$   
  **if**  $\mathbf{h}$  est assez précis **then**  
    **return**  $\mathbf{x}_i = \mathbf{h}$   
  **else**  
     $\mathbf{s} = \mathbf{r}_{i-1} - \alpha \mathbf{v}_i$   
     $\mathbf{t} = A\mathbf{s}$   
     $\omega_i = \frac{(\mathbf{t}, \mathbf{s})}{(\mathbf{t}, \mathbf{t})}$   
     $\mathbf{x}_i = \mathbf{h} + \omega_i \mathbf{s}$   
    **if**  $\mathbf{x}_i$  est assez précis **then**  
      **return**  $\mathbf{x}_i$   
    **else**  
       $\mathbf{r}_i = \mathbf{s} - \omega_i \mathbf{t}$   
    **end if**  
  **end if**  
**end for**

---

Contrairement à la méthode itérative précédente, celle-ci converge et donne les mêmes résultats qualitatifs que la factorisation LU tout en étant largement plus rapide. Étant donné la vitesse de convergence de cet algorithme, nous allons l'utiliser à la place de l'algorithme du gradient conjugué résidu normal.

### 2.2.2 Validation, calculs d'erreurs

#### Système découplé

Pour valider le code, il nous faut le confronter à une solution exacte.

Dans un premier temps, nous allons chercher une solution particulière du système suivant :

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u - f(u, v) = 0 \\ \frac{\partial v}{\partial t} - d\Delta v - g(u, v) = 0 \end{cases} \quad (8)$$

avec

$$f(u, v) = u$$

$$g(u, v) = v$$

Fixons la condition initiale  $u_0(x, y) = v_0(x, y) = \cos(2\pi x) \cos(2\pi y) \quad \forall (x, y) \in \Omega$  et cherchons  $u$  et  $v$  sous la forme :

$$\begin{cases} u(x, y, t) = F(t) \cos(2\pi x) \cos(2\pi y) \\ v(x, y, t) = G(t) \cos(2\pi x) \cos(2\pi y) \end{cases}$$

En injectant dans le système (7) ces solutions, il faut résoudre :

$$\begin{cases} F'(t) + (8\pi^2 - 1)F(t) = 0 \\ G'(t) + (8d\pi^2 - 1)G(t) = 0 \end{cases}$$

Les solutions sont alors :

$$\begin{cases} u(x, y, t) = \exp((1 - 8\pi^2)t) \cos(2\pi x) \cos(2\pi y) \\ v(x, y, t) = \exp((1 - 8d\pi^2)t) \cos(2\pi x) \cos(2\pi y) \end{cases}$$

Nous disposons donc de la solution exacte du problème. Nous pouvons alors la comparer avec la solution numérique.

Nous vérifions les ordres de convergence théoriques de la même manière que précédemment. Sur la figure 17, nous retrouvons bien la convergence d'ordre 1 en temps et sur la figure 18, nous avons bien la convergence d'ordre 2 en espace.

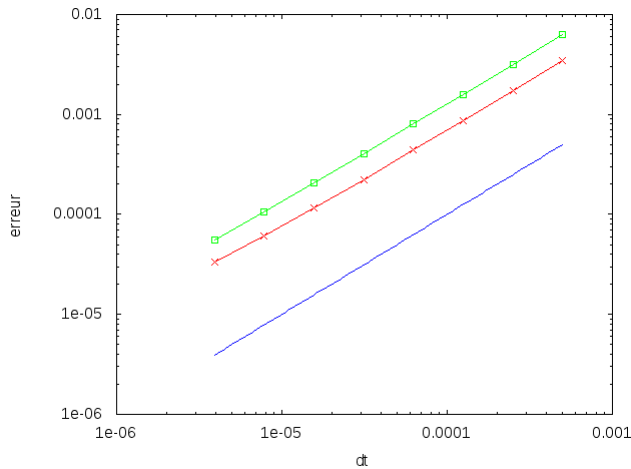


FIGURE 17 – Taux de convergence en temps en échelle logarithmique de  $u$ , de  $v$  et droite  $y = x$

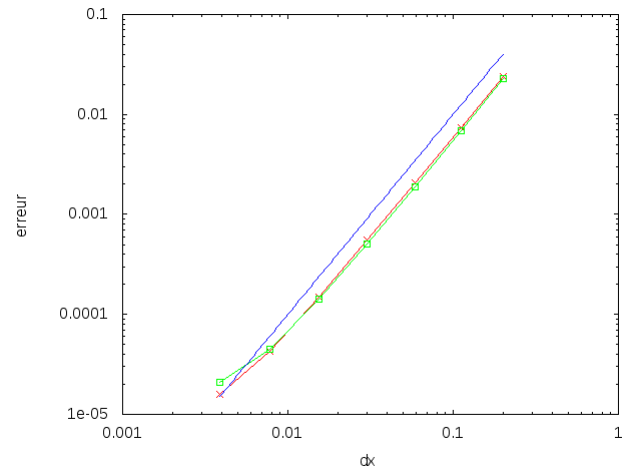


FIGURE 18 – Taux de convergence en espace en échelle logarithmique de  $u$ , de  $v$  et droite  $y = x^2$

## Système couplé

Ensuite, nous voulons vérifier que le couplage de  $u$  et  $v$  fonctionne comme prévu. Pour cela, nous allons résoudre le système suivant :

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u - f(u, v) = 0 \\ \frac{\partial v}{\partial t} - d\Delta v - g(u, v) = 0 \end{cases} \quad (9)$$

avec

$$\begin{aligned} f(u, v) &= u + v \\ g(u, v) &= u + v \end{aligned}$$

Fixons la condition initiale  $u_0(x, y) = v_0(x, y) = \cos(2\pi x) \cos(2\pi y) \quad \forall (x, y) \in \Omega$  et cherchons  $u$  et  $v$  sous la forme :

$$\begin{cases} u(x, y, t) = F(t) \cos(2\pi x) \cos(2\pi y) \\ v(x, y, t) = G(t) \cos(2\pi x) \cos(2\pi y) \end{cases}$$

En procédant de la même manière que précédemment, nous avons  $F$  et  $G$  solutions du système différentiel suivant :

$$\begin{pmatrix} F'(t) \\ G'(t) \end{pmatrix} = \begin{pmatrix} -8\pi^2 + 1 & 1 \\ 1 & -8d\pi^2 + 1 \end{pmatrix} \begin{pmatrix} F(t) \\ G(t) \end{pmatrix}$$

En procédant de la même manière que précédemment, nous obtenons  $\lambda_1 = -156.926$ ,  $\lambda_2 = -77.9742$ ,  $v_1 = \begin{pmatrix} -0.0126669 \\ 0.99992 \end{pmatrix}$  et  $v_2 = \begin{pmatrix} -0.99992 \\ -0.0126669 \end{pmatrix}$ . On a donc  $\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 0.987252 \\ -1.01258 \end{pmatrix}$ . Finalement,

$$\begin{pmatrix} F(t) \\ G(t) \end{pmatrix} = \begin{pmatrix} -0.0125054e^{-156.926t} + 1.0125054e^{-77.9742t} \\ 0.987173e^{-156.926t} + 0.01282633e^{-77.9742t} \end{pmatrix}$$

Nous disposons donc de la solution exacte du problème. Nous pouvons alors la comparer avec la solution numérique.

Nous vérifions les ordres de convergence théoriques de la même manière que précédemment. Sur la figure 19, nous retrouvons bien la convergence d'ordre 1 en temps et sur la figure 20, nous avons bien la convergence d'ordre 2 en espace.

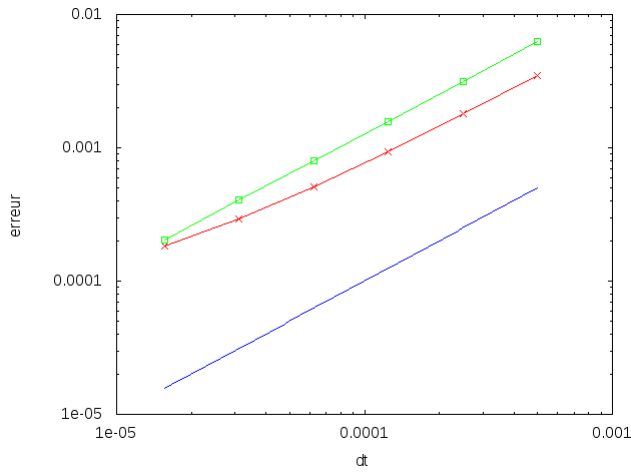


FIGURE 19 – Taux de convergence en temps en échelle logarithmique de  $u$ , de  $v$  et droite  $y = x$

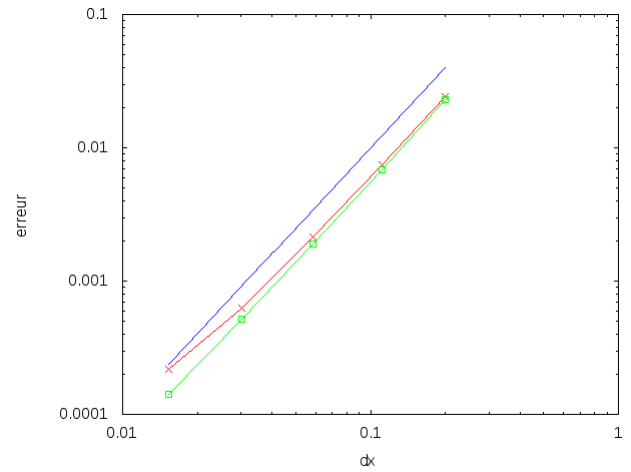


FIGURE 20 – Taux de convergence en espace en échelle logarithmique de  $u$ , de  $v$  et droite  $y = x^2$

### Validation qualitative de la condition aux limites

Pour vérifier que la périodicité aux bords est bien implémentée, nous partons de conditions initiales  $u_0$  et  $v_0$  suivant respectivement les bords gauche et inférieur du domaine (deux demi-cercles). Par la composante diffusive des équations différentielles du problème (1), nous nous attendons à voir la diffusion apparaître autour des demi-cercles et aussi à l'opposé.

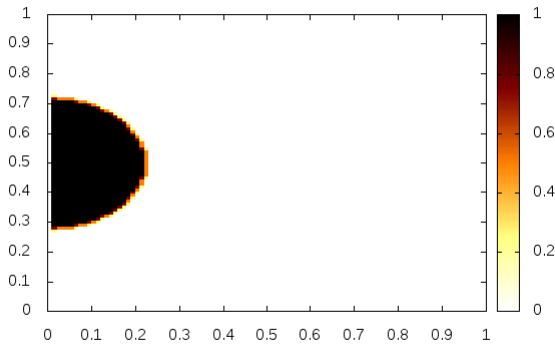


FIGURE 21 – Représentation de  $u_0$  en courbe de niveau

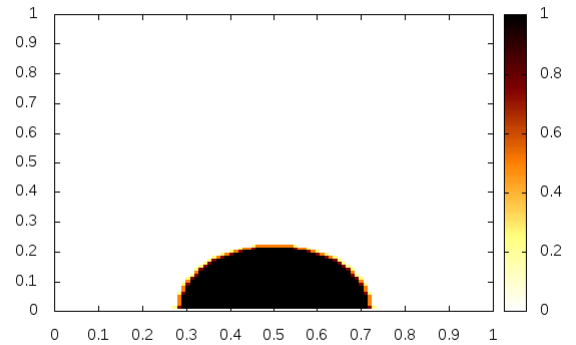


FIGURE 22 – Représentation de  $v_0$  en courbe de niveau

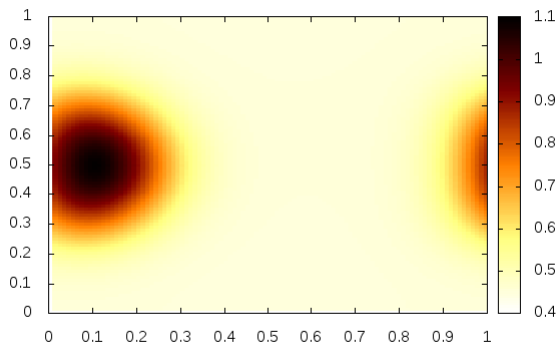


FIGURE 23 – Représentation de  $u$  en courbe de niveau à  $t = 0.005$

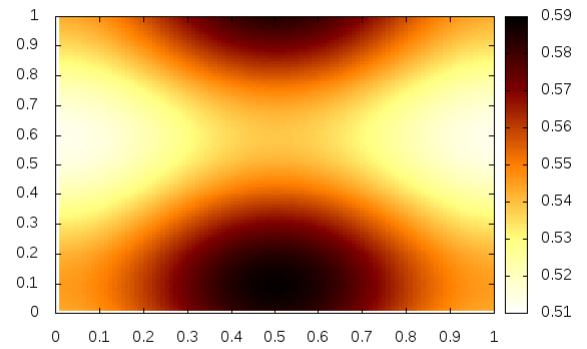


FIGURE 24 – Représentation de  $v$  en courbe de niveau à  $t = 0.005$  avec  $d = 10$

Comme nous nous y attendions, nous observons bien la périodicité aux bords. De plus,  $v$  a beaucoup plus diffusé que  $u$  grâce à son coefficient de diffusion plus élevé.



### 3 Recherche des taches

Nous allons maintenant nous intéresser à la recherche des conditions et paramètres qui permettent de réaliser les taches recherchées. Pour rappel, le système de réaction-diffusion responsable de la formation de motifs est donné par :

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u - f(u, v) = 0 \\ \frac{\partial v}{\partial t} - d\Delta v - g(u, v) = 0 \end{cases} \quad (10)$$

avec

$$f(u, v) = \gamma \left( a - u - \frac{\rho uv}{1 + u + Ku^2} \right)$$

$$g(u, v) = \gamma \left( \alpha(b - v) - \frac{\rho uv}{1 + u + Ku^2} \right)$$

où  $\alpha, a, b, \rho, d, K$  et  $\gamma$  sont des paramètres positifs.

Le problème réside alors dans la définition des paramètres, dans le choix de la condition aux bords et dans le choix des conditions initiales  $u_0$  et  $v_0$ .

Pour les paramètres numériques, on trouve dans le livre de J.D Murray, *Mathematical biology* et dans l'article de Stéphane Descombes et Magali Ribot, *Convergence of the Peaceman-Rashford approximation for reaction diffusion system*, Numer. Math., 2003, plusieurs jeux de paramètres qui permettent d'obtenir des motifs convaincants. Nous utiliserons donc ces valeurs numériques dans nos simulations.

Pour le choix de la condition aux bords, nous avons le choix complet de ce qui nous intéresse selon nos critères de modélisation. Toutefois, nous ne disposons pas de la condition aux limites de Neumann (homogène ou non) ni de conditions mixtes.

En ce qui concerne les conditions initiales, nous disposons de deux approches :

- l'approche de Descombes et Ribot qui consiste à utiliser des conditions initiales spécifiques
- l'approche de Murray qui consiste à perturber l'état stationnaire en utilisant des motifs prédéfinis

Pour modéliser un domaine rectangulaire qui puisse représenter un corps animal allongé comme une queue ou un long cou, nous discrétisons désormais le domaine  $\Omega = [0, L] \times [0, 1]$ . Avec ce type de domaine, nous pouvons à présent résoudre le problème (1) sur tous les domaines rectangulaires.

### 3.1 Recherche de motifs à partir de conditions initiales spécifiques

Dans l'article de Descombes et Ribot, les auteurs obtiennent des motifs de taches et de rayures en utilisant des conditions initiales de la forme :

$$u_0(x, y) = 30 \exp(-250((x - 0.25)^2 + (y - 0.5)^2)) + 15 \exp(-250((x - 0.5)^2 + (y - 0.25)^2)) \\ + 30 \exp(-250((-250(x - 0.5)^2 + (y - 0.75)^2)) + 15 \exp(-250((x - 0.75)^2 + (y - 0.5)^2))$$

et  $v_0(x, y) = 0$

avec les paramètres numériques suivants :

$$a = 92, b = 64, \alpha = 1.5, \rho = 18.5, K = 0.1, d = 10, \gamma = 1$$

Leur simulation numérique utilise la condition de Neumann homogène sur les bords. N'ayant pas implémenté ce type de conditions aux limites, nous pouvons tout de même mettre en place cette condition initiale avec des conditions aux bords périodiques.

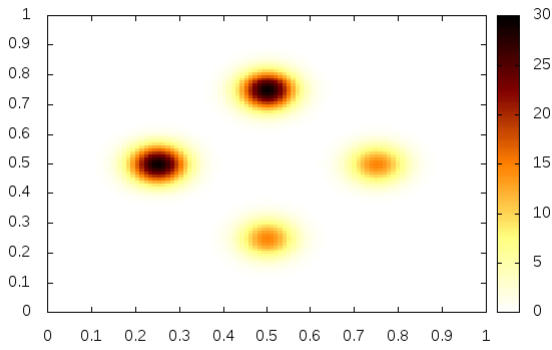


FIGURE 25 – Condition initiale  $u_0$

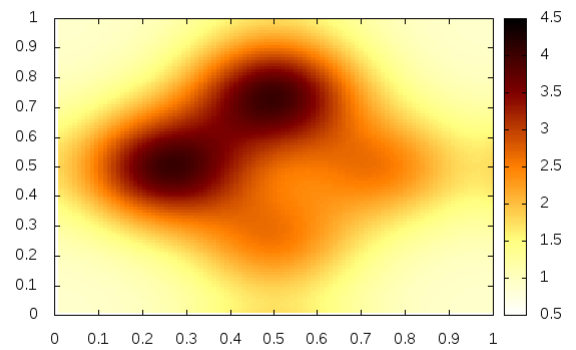


FIGURE 26 – Solution  $u$  à  $t = 0.01$

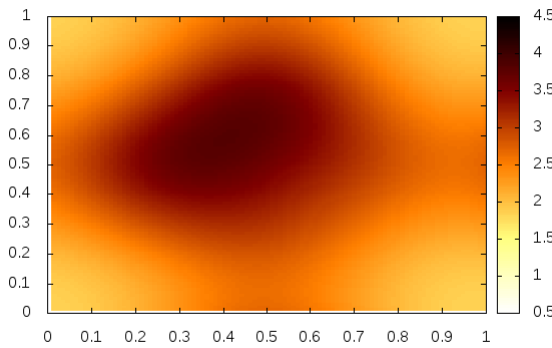


FIGURE 27 – Solution  $u$  à  $t = 0.03$

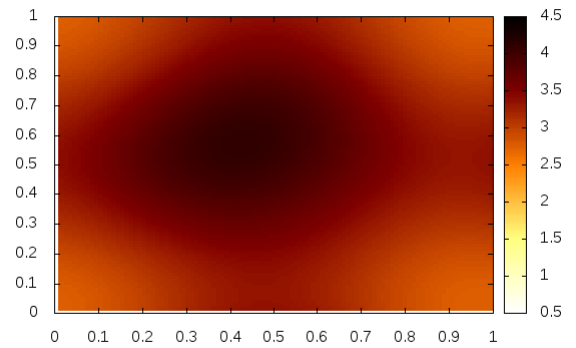


FIGURE 28 – Solution  $u$  à  $t = 0.05$

### 3.2 Recherche de motifs à partir du régime permanent

Nous allons rechercher le régime permanent. Nous disposons de deux méthodes :

- annuler les termes  $\frac{\partial u}{\partial t}$  et  $\frac{\partial v}{\partial t}$  et résoudre le système obtenu
- calculer cet état à partir du système en faisant tendre  $t$  vers l'infini

La première méthode consisterait à appliquer l'algorithme de Newton sur les systèmes précédents modifiés pour supprimer la dérivée temporelle. Après implémentation, nous nous sommes rendu compte que cette méthode ne fonctionnait pas. En effet, selon le point de départ de l'algorithme de Newton, la solution était fautive. Soit elle explosait complètement, soit elle tendait vers  $u = v = 0$  qui n'est pas solution du système différentiel.

La seconde méthode consiste à calculer le régime stationnaire en faisant tendre  $t$  vers l'infini. Avec un  $\Delta t$  assez grand et en partant d'une condition initiale  $u_0 = v_0 = 0$ , nous avons besoin d'un nombre limité d'itérations pour que l'algorithme converge vers cet état stationnaire. Nous utiliserons donc systématiquement cette méthode pour calculer à nouveau le régime stationnaire au début de chaque simulation bien qu'elle requiert quelques itérations supplémentaires.

Dans les deux cas à deux dimensions implémentés, les états stationnaires de  $u$  sont alors :

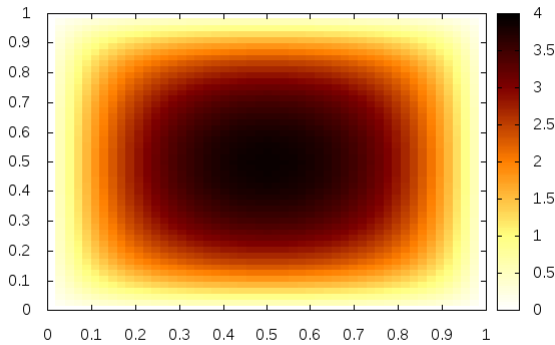


FIGURE 29 – Régime stationnaire avec condition Dirichlet homogène aux bords

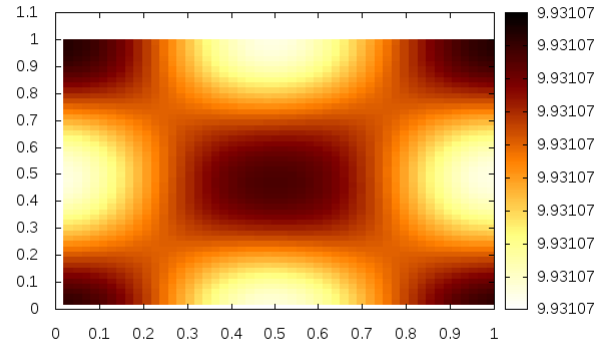


FIGURE 30 – Régime stationnaire avec condition périodique aux bords ( $u$  est constant)

Ensuite, nous perturbons ce régime permanent selon le genre de motifs suggéré dans le livre de Murray. Nous pouvons alors utiliser plusieurs types de perturbations : rayures, gaussiennes pour modéliser des points, aléatoires...

Nous allons essayer la perturbation aléatoire. Pour cela, nous voudrions simuler une loi normale centrée réduite. Nous la simulons à partir de deux tirages  $U_1$  et  $U_2$  de loi uniforme sur  $]0, 1[$ . La méthode de Box-Muller nous offre une méthode de simulation de  $\mathcal{N}(0, 1)$  :  $\sqrt{-2 \log(U_1)} \cos(2\pi U_2)$  et  $\sqrt{-2 \log(U_1)} \sin(2\pi U_2)$  sont des variables aléatoires indépendantes et suivent une loi normale centrée réduite.

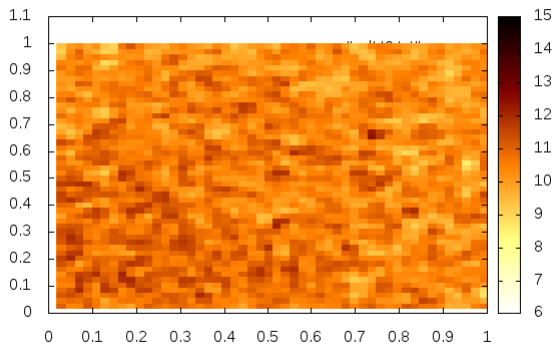


FIGURE 31 –  $u_0$  : régime stationnaire perturbé aléatoirement

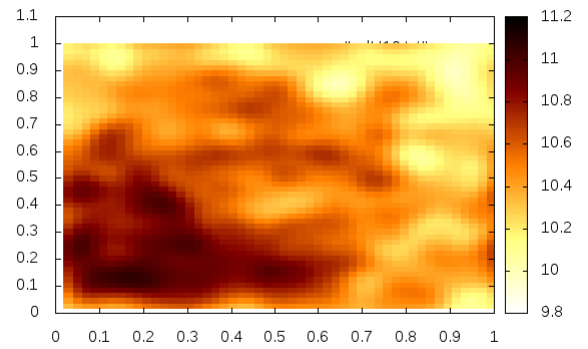


FIGURE 32 – Solution  $u$  à  $t = 0.001$

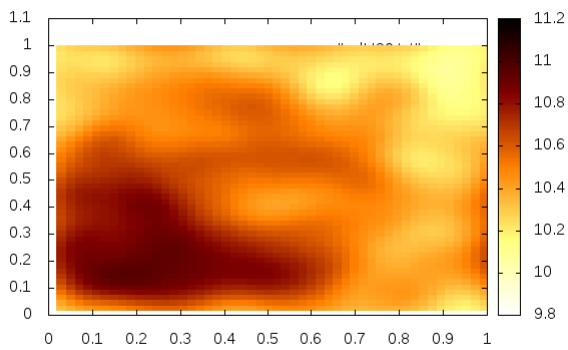


FIGURE 33 – Solution  $u$  à  $t = 0.002$

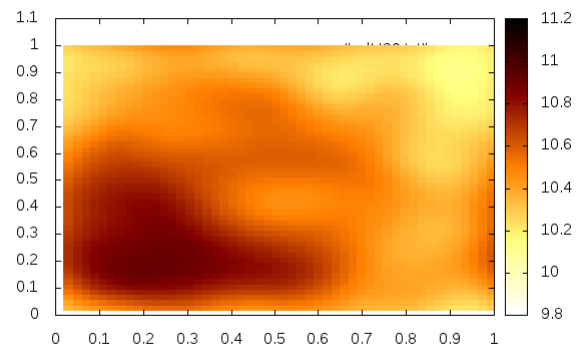


FIGURE 34 – Solution  $u$  à  $t = 0.003$

A l'instant initial, comme voulu,  $u$  est de la forme souhaitée. Par la suite, il y a homogénéisation puis établissement d'un régime stationnaire.

## 4 Documentation du code

Nous avons implémenté les méthodes décrites précédemment en `Fortran 90`. Vous trouverez ici la description des différents modules constituant notre code ainsi que leur contenu et leur fonction.

### Module `resolutions`

La subroutine principale de notre code est `resolutionTemporelle`. Pour un temps `tMax` et un pas `dt` donnés, elle calcule les différentes approximations de  $u_{i,j}^n$  et  $v_{i,j}^n$ . Celles-ci sont alors écrites dans des fichiers, qui sont ensuite lus par `Gnuplot` pour les représenter graphiquement. Si on connaît la solution exacte, la norme de l'erreur entre les solutions numériques et les solutions analytiques sont tracées en fonction du temps.

Cette subroutine fait appel à la subroutine `newton` où est implémenté l'algorithme de Newton.

### Module `resolutionSystemeLineaire`

Pour appliquer l'algorithme de Newton, nous avons besoin de résoudre le système linéaire  $DF(\mathbf{x}_k)\delta\mathbf{x}_k = F(\mathbf{x}_k)$ . Les différents algorithmes de résolution de systèmes linéaires sont implémentés dans ce module et sont appelés en fonction du système de réaction-diffusion que l'on souhaite résoudre.

### Module `secondMembreNewton`

Ce module contient toutes les fonctions de la forme  $F$  pour résoudre le système linéaire  $DF(\mathbf{x}_k)\delta\mathbf{x}_k = F(\mathbf{x}_k)$  dans l'algorithme de Newton.

### Module `assemblageTableaux`

Toujours pour résoudre le système linéaire  $DF(\mathbf{x}_k)\delta\mathbf{x}_k = F(\mathbf{x}_k)$ , les matrices  $DF(\mathbf{x}_k)$  sont creuses et nous utilisons donc différents stockages. Ce module contient les subroutines qui permettent de construire les tableaux associés à ces stockages.

### Module `secondMembre_conditionInitiale`

On résout le système de réaction-diffusion (1) avec différentes fonctions  $f$  et  $g$  et différentes conditions initiales. Ces fonctions sont implémentées dans ce module, ainsi que les dérivées des fonctions  $f$  et  $g$ .

### Module `solutionsExactes`

Avec certains choix de seconds membres pour le système (1) et certaines conditions initiales, on peut calculer les solutions exactes de (1).

## Module fonctionsGenerales

Ce module contient différentes fonctions utilisées dans les autres modules, comme la norme euclidienne, l'affichage des vecteurs, l'écriture des vecteurs dans des fichiers ou le produit matrice vecteur en stockage CSR.

## 5 Conclusion

Nous n'avons pas réussi jusqu'à présent à obtenir des résultats et des représentations graphiques significatifs.

En effet, nous parvenons à observer essentiellement le phénomène diffusif. La composante non linéaire est bien présente comme le montre le calcul de régime permanent. Toutefois, son influence sur le système ne se fait pas sentir sensiblement dans l'évolution temporelle de notre solution numérique. En faisant varier les conditions initiales, nous pouvons constater que les rôles de  $u$  et  $v$ , comme prévus, ne sont pas symétriques.

Nos méthodes sont, malgré tout, correctes et ont été validées expérimentalement : ordres de convergence de la méthode des différences finies corrects, vérification de la méthode de Newton, divers types de conditions aux limites...

Peut-être notre méthode de représentation des résultats n'est-elle pas parfaitement adaptée et nous empêche de visualiser correctement les phénomènes qui nous intéressent. De plus, nous n'avons éventuellement pas cherché les phénomènes aux bons laps de temps. En effet, nous ne conservons que les résultats de 100 itérations réparties de manière homogène entre l'instant initial et l'instant final. Il nous faut donc adapter le  $\Delta t$  en fonction de la plage de temps que nous voulons observer et cela se révèle contraignant. En effet avec  $\Delta t$  petit, nous ne parvenons qu'à visualiser de la diffusion. Et avec un  $\Delta t$  relativement grand, à cause du phénomène diffusif, nous observons très vite une solution numérique constante et le régime permanent. Il serait alors intéressant de permettre la conservation des résultats de davantage d'itérations pour pouvoir approcher les solutions avec un  $\Delta t$  plus petit et à un temps  $t$  plus grand et ainsi visualiser un intervalle de temps plus grand tout en limitant l'erreur d'approximation.

Nous avons également été limités par la puissance de calcul dont nous disposions. En effet, nous recherchions un phénomène assez subtil donc nous ne pouvions pas nous permettre d'utiliser un pas de discrétisation spacial  $\Delta x$  trop grand. Cela causa alors un temps de calcul par itération temporelle vite grand (ordre de grandeur de la taille du système :  $\frac{2}{\Delta x^2}$ ) qui a fortement nuit à la recherche efficace de bons paramètres temporels.

De plus, nous pouvons penser que d'autres conditions aux limites telles que Neumann homogène ou des conditions mixtes (Neumann homogène et conditions périodiques) auraient eu plus d'influence sur l'évolution du système que nous le pensions. Il serait vraisemblablement pertinent de les implémenter par la suite. En parallèle, il serait peut-être aussi intéressant d'essayer une autre méthode d'approximation comme les éléments finis.

## Références

MURRAY, *Mathematical Biology II : Spatial Models and Biomedical Applications*, Springer, 2011, troisième édition

Stéphane DÉSCOMBES, Magali RIBOT, *Convergence of the Peaceman-Rachford approximation for reaction-diffusion systems*, Numer. Math, 2003





## C Schéma différences finies du système à deux équations et à une dimension

$$u_0^n = u_{I+1}^n = v_0^n = v_{I+1}^n = 0 \text{ pour tout } n \in \mathbb{N} \\ 2 \leq i \leq I-1$$

$$\begin{cases} \frac{u_i^{n+1} - u_i^n}{\Delta t} - \frac{u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}}{\Delta x^2} - f(u_i^{n+1}, v_i^{n+1}) = 0 \\ \frac{v_i^{n+1} - v_i^n}{\Delta t} - d \frac{v_{i-1}^{n+1} - 2v_i^{n+1} + v_{i+1}^{n+1}}{\Delta x^2} - g(u_i^{n+1}, v_i^{n+1}) = 0 \end{cases}$$

$$i = 1$$

$$\begin{cases} \frac{u_i^{n+1} - u_i^n}{\Delta t} - \frac{-2u_i^{n+1} + u_{i+1}^{n+1}}{\Delta x^2} - f(u_i^{n+1}, v_i^{n+1}) = 0 \\ \frac{v_i^{n+1} - v_i^n}{\Delta t} - d \frac{-2v_i^{n+1} + v_{i+1}^{n+1}}{\Delta x^2} - g(u_i^{n+1}, v_i^{n+1}) = 0 \end{cases}$$

$$i = I$$

$$\begin{cases} \frac{u_i^{n+1} - u_i^n}{\Delta t} - \frac{u_{i-1}^{n+1} - 2u_i^{n+1}}{\Delta x^2} - f(u_i^{n+1}, v_i^{n+1}) = 0 \\ \frac{v_i^{n+1} - v_i^n}{\Delta t} - d \frac{v_{i-1}^{n+1} - 2v_i^{n+1}}{\Delta x^2} - g(u_i^{n+1}, v_i^{n+1}) = 0 \end{cases}$$

## D Jacobienne du schéma du système à deux équations et à une dimension

La jacobienne du système évaluée en  $(x, y)$  avec  $x = (x_1, \dots, x_I)^T$  et  $y = (y_1, \dots, y_I)^T$  s'écrit

de la manière suivante :  $\begin{pmatrix} M_1 & N_1 \\ N_2 & M_2 \end{pmatrix}$

$M_1, N_1, N_2, M_2 \in \mathcal{M}_I(\mathbb{R})$  avec  $N_1, N_2$  diagonales

$$M_1 = \begin{pmatrix} c_1 - \frac{\partial f}{\partial x}(x_1, y_1) & & & & & & & & & & & \\ & c_2 & & & & & & & & & & \\ & c_2 & c_1 - \frac{\partial f}{\partial x}(x_2, y_2) & & & & & & & & & \\ & & & \ddots & & & & & & & & \\ & & & & \ddots & & & & & & & \\ & & & & & c_2 & c_1 - \frac{\partial f}{\partial x}(x_i, y_i) & & & & & \\ & & & & & & \ddots & & & & & \\ & & & & & & & \ddots & & & & \\ & & & & & & & & c_2 & c_1 - \frac{\partial f}{\partial x}(x_{I-1}, y_{I-1}) & & \\ & & & & & & & & & c_2 & & c_1 - \frac{\partial f}{\partial x}(x_I, y_I) \end{pmatrix}$$

$$\text{avec } c_1 = \frac{1}{\Delta t} + \frac{2}{\Delta x^2} \text{ et } c_2 = -\frac{1}{\Delta x^2}$$

$$M_2 = \begin{pmatrix} c_3 - \frac{\partial g}{\partial y}(x_1, y_1) & & & & & & & & & & & \\ & c_4 & & & & & & & & & & \\ & c_4 & c_3 - \frac{\partial g}{\partial y}(x_2, y_2) & & & & & & & & & \\ & & & \ddots & & & & & & & & \\ & & & & \ddots & & & & & & & \\ & & & & & c_4 & c_3 - \frac{\partial g}{\partial y}(x_i, y_i) & & & & & \\ & & & & & & \ddots & & & & & \\ & & & & & & & \ddots & & & & \\ & & & & & & & & c_4 & c_3 - \frac{\partial g}{\partial y}(x_{I-1}, y_{I-1}) & & \\ & & & & & & & & & c_4 & & c_3 - \frac{\partial g}{\partial y}(x_I, y_I) \end{pmatrix}$$

avec  $c_3 = \frac{1}{\Delta t} + \frac{2d}{\Delta x^2}$  et  $c_4 = -\frac{d}{\Delta x^2}$

$$N_1 = \begin{pmatrix} -\frac{\partial f}{\partial y}(x_1, y_1) & & & \\ & \ddots & & \\ & & -\frac{\partial f}{\partial y}(x_I, y_I) & \\ & & & \end{pmatrix} \text{ et } N_2 = \begin{pmatrix} -\frac{\partial g}{\partial x}(x_1, y_1) & & & \\ & \ddots & & \\ & & -\frac{\partial g}{\partial x}(x_I, y_I) & \\ & & & \end{pmatrix}$$

## E Schéma différences finies à deux dimensions, condition aux bords de Dirichlet homogène

$i = 1$  et  $j = 1$

$$\begin{cases} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{-2u_{i,j}^{n+1} + u_{i+1,j}^{n+1}}{\Delta x^2} - \frac{-2u_{i,j}^{n+1} + u_{i,j+1}^{n+1}}{\Delta y^2} - f(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} - d \frac{-2v_{i,j}^{n+1} + v_{i+1,j}^{n+1}}{\Delta x^2} - d \frac{-2v_{i,j}^{n+1} + v_{i,j+1}^{n+1}}{\Delta y^2} - g(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \end{cases}$$

$i = 1$  et  $j = J$

$$\begin{cases} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{-2u_{i,j}^{n+1} + u_{i+1,j}^{n+1}}{\Delta x^2} - \frac{u_{i,j-1}^{n+1} - 2u_{i,j}^{n+1}}{\Delta y^2} - f(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} - d \frac{-2v_{i,j}^{n+1} + v_{i+1,j}^{n+1}}{\Delta x^2} - d \frac{v_{i,j-1}^{n+1} - 2v_{i,j}^{n+1}}{\Delta y^2} - g(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \end{cases}$$

$i = 1$  et  $2 \leq j \leq J - 1$

$$\begin{cases} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{-2u_{i,j}^{n+1} + u_{i+1,j}^{n+1}}{\Delta x^2} - \frac{u_{i,j-1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j+1}^{n+1}}{\Delta y^2} - f(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} - d \frac{-2v_{i,j}^{n+1} + v_{i+1,j}^{n+1}}{\Delta x^2} - d \frac{v_{i,j-1}^{n+1} - 2v_{i,j}^{n+1} + v_{i,j+1}^{n+1}}{\Delta y^2} - g(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \end{cases}$$

$i = I$  et  $j = 1$

$$\begin{cases} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{u_{i-1,j}^{n+1} - 2u_{i,j}^{n+1}}{\Delta x^2} - \frac{-2u_{i,j}^{n+1} + u_{i,j+1}^{n+1}}{\Delta y^2} - f(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} - d \frac{v_{i-1,j}^{n+1} - 2v_{i,j}^{n+1}}{\Delta x^2} - d \frac{-2v_{i,j}^{n+1} + v_{i,j+1}^{n+1}}{\Delta y^2} - g(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \end{cases}$$

$i = I$  et  $j = J$

$$\begin{cases} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{u_{i-1,j}^{n+1} - 2u_{i,j}^{n+1}}{\Delta x^2} - \frac{u_{i,j-1}^{n+1} - 2u_{i,j}^{n+1}}{\Delta y^2} - f(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} - d \frac{v_{i-1,j}^{n+1} - 2v_{i,j}^{n+1}}{\Delta x^2} - d \frac{v_{i,j-1}^{n+1} - 2v_{i,j}^{n+1}}{\Delta y^2} - g(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \end{cases}$$

$i = I$  et  $2 \leq j \leq J - 1$

$$\begin{cases} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{u_{i-1,j}^{n+1} - 2u_{i,j}^{n+1}}{\Delta x^2} - \frac{u_{i,j-1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j+1}^{n+1}}{\Delta y^2} - f(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} - d \frac{v_{i-1,j}^{n+1} - 2v_{i,j}^{n+1}}{\Delta x^2} - d \frac{v_{i,j-1}^{n+1} - 2v_{i,j}^{n+1} + v_{i,j+1}^{n+1}}{\Delta y^2} - g(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \end{cases}$$

$2 \leq i \leq I - 1$  et  $j = 1$

$$\begin{cases} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{u_{i-1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i+1,j}^{n+1}}{\Delta x^2} - \frac{-2u_{i,j}^{n+1} + u_{i,j+1}^{n+1}}{\Delta y^2} - f(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} - d \frac{v_{i-1,j}^{n+1} - 2v_{i,j}^{n+1} + v_{i+1,j}^{n+1}}{\Delta x^2} - d \frac{-2v_{i,j}^{n+1} + v_{i,j+1}^{n+1}}{\Delta y^2} - g(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \end{cases}$$



$$M_2 = \begin{pmatrix} C_1 & D & & & & & & \\ D & C_2 & D & & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & D & C_j & D & & \\ & & & & \ddots & \ddots & \ddots & \\ & & & & & D & C_{J-1} & D \\ & & & & & & D & C_J \end{pmatrix} \text{ avec } D \in \mathcal{M}_I(\mathbb{R}) \text{ et } D = \begin{pmatrix} -\frac{d}{\Delta y^2} & & & \\ & \ddots & & \\ & & & -\frac{d}{\Delta y^2} \end{pmatrix}$$

et pour  $1 \leq j \leq J$ ,  $C_j \in \mathcal{M}_I(\mathbb{R})$  et  $C_j =$

$$\begin{pmatrix} c_3 - \frac{\partial g}{\partial y}(x_{1,j}, y_{1,j}) & & & & & & & c_4 \\ c_4 & c_3 - \frac{\partial g}{\partial y}(x_{2,j}, y_{2,j}) & c_4 & & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & c_4 & c_3 - \frac{\partial g}{\partial y}(x_{i,j}, y_{i,j}) & c_4 & & \\ & & & & \ddots & \ddots & \ddots & \\ & & & & & c_4 & c_3 - \frac{\partial g}{\partial y}(x_{I-1,j}, y_{I-1,j}) & c_4 \\ & & & & & & c_4 & c_3 - \frac{\partial g}{\partial y}(x_{I,j}, y_{I,j}) \end{pmatrix}$$

avec  $c_3 = \frac{1}{\Delta t} + \frac{2d}{\Delta x^2} + \frac{2d}{\Delta y^2}$  et  $c_4 = -\frac{d}{\Delta x^2}$

$$N_1 = \begin{pmatrix} -\frac{\partial f}{\partial y}((x_{1,1}, y_{1,1})) & & & & & \\ & \ddots & & & & \\ & & & -\frac{\partial f}{\partial y}(x_{I,J}, y_{I,J}) & & \end{pmatrix} \text{ et } N_2 = \begin{pmatrix} -\frac{\partial g}{\partial x}((x_{1,1}, y_{1,1})) & & & & & \\ & \ddots & & & & \\ & & & & & \\ & & & & & -\frac{\partial g}{\partial x}(x_{I,J}, y_{I,J}) \end{pmatrix}$$

## G Schéma différences finies à deux dimensions, condition aux limites périodique

$i = 1$  et  $j = 1$

$$\begin{cases} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{u_{I+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i+1,j}^{n+1}}{\Delta x^2} - \frac{u_{i,J+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j+1}^{n+1}}{\Delta y^2} - f(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} - d \frac{v_{I+1,j}^{n+1} - 2v_{i,j}^{n+1} + v_{i+1,j}^{n+1}}{\Delta x^2} - d \frac{v_{i,J+1}^{n+1} - 2v_{i,j}^{n+1} + v_{i,j+1}^{n+1}}{\Delta y^2} - g(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \end{cases}$$

$i = 1$  et  $j = J + 1$

$$\begin{cases} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{u_{I+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i+1,j}^{n+1}}{\Delta x^2} - \frac{u_{i,j-1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,1}^{n+1}}{\Delta y^2} - f(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} - d \frac{v_{I+1,j}^{n+1} - 2v_{i,j}^{n+1} + v_{i+1,j}^{n+1}}{\Delta x^2} - d \frac{v_{i,j-1}^{n+1} - 2v_{i,j}^{n+1} + v_{i,1}^{n+1}}{\Delta y^2} - g(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \end{cases}$$

$i = 1$  et  $2 \leq j \leq J$

$$\begin{cases} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{u_{I+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i+1,j}^{n+1}}{\Delta x^2} - \frac{u_{i,j-1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j+1}^{n+1}}{\Delta y^2} - f(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} - d \frac{v_{I+1,j}^{n+1} - 2v_{i,j}^{n+1} + v_{i+1,j}^{n+1}}{\Delta x^2} - d \frac{v_{i,j-1}^{n+1} - 2v_{i,j}^{n+1} + v_{i,j+1}^{n+1}}{\Delta y^2} - g(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \end{cases}$$

$i = I + 1$  et  $j = 1$

$$\begin{cases} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{u_{i-1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{1,j}^{n+1}}{\Delta x^2} - \frac{u_{i,J+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j+1}^{n+1}}{\Delta y^2} - f(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} - d \frac{v_{i-1,j}^{n+1} - 2v_{i,j}^{n+1} + v_{1,j}^{n+1}}{\Delta x^2} - d \frac{v_{i,J+1}^{n+1} - 2v_{i,j}^{n+1} + v_{i,j+1}^{n+1}}{\Delta y^2} - g(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \end{cases}$$

$i = I + 1$  et  $j = J + 1$

$$\begin{cases} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{u_{i-1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{1,j}^{n+1}}{\Delta x^2} - \frac{u_{i,j-1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,1}^{n+1}}{\Delta y^2} - f(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} - d \frac{v_{i-1,j}^{n+1} - 2v_{i,j}^{n+1} + v_{1,j}^{n+1}}{\Delta x^2} - d \frac{v_{i,j-1}^{n+1} - 2v_{i,j}^{n+1} + v_{i,1}^{n+1}}{\Delta y^2} - g(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \end{cases}$$

$i = I + 1$  et  $2 \leq j \leq J$

$$\begin{cases} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{u_{i-1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{1,j}^{n+1}}{\Delta x^2} - \frac{u_{i,j-1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j+1}^{n+1}}{\Delta y^2} - f(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} - d \frac{v_{i-1,j}^{n+1} - 2v_{i,j}^{n+1} + v_{i+1,j}^{n+1}}{\Delta x^2} - d \frac{v_{i,j-1}^{n+1} - 2v_{i,j}^{n+1} + v_{i,j+1}^{n+1}}{\Delta y^2} - g(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \end{cases}$$

$2 \leq i \leq I$  et  $j = 1$

$$\begin{cases} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{u_{i-1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i+1,j}^{n+1}}{\Delta x^2} - \frac{u_{i,J+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j+1}^{n+1}}{\Delta y^2} - f(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} - d \frac{v_{i-1,j}^{n+1} - 2v_{i,j}^{n+1} + v_{i+1,j}^{n+1}}{\Delta x^2} - d \frac{v_{i,J+1}^{n+1} - 2v_{i,j}^{n+1} + v_{i,j+1}^{n+1}}{\Delta y^2} - g(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \end{cases}$$

$2 \leq i \leq I$  et  $j = J + 1$

$$\begin{cases} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{u_{i-1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i+1,j}^{n+1}}{\Delta x^2} - \frac{u_{i,j-1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j+1}^{n+1}}{\Delta y^2} - f(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} - d \frac{v_{i-1,j}^{n+1} - 2v_{i,j}^{n+1} + v_{i+1,j}^{n+1}}{\Delta x^2} - d \frac{v_{i,j-1}^{n+1} - 2v_{i,j}^{n+1} + v_{i,j+1}^{n+1}}{\Delta y^2} - g(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \end{cases}$$

$2 \leq i \leq I$  et  $2 \leq j \leq J$

$$\begin{cases} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - \frac{u_{i-1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i+1,j}^{n+1}}{\Delta x^2} - \frac{u_{i,j-1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j+1}^{n+1}}{\Delta y^2} - f(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} - d \frac{v_{i-1,j}^{n+1} - 2v_{i,j}^{n+1} + v_{i+1,j}^{n+1}}{\Delta x^2} - d \frac{v_{i,j-1}^{n+1} - 2v_{i,j}^{n+1} + v_{i,j+1}^{n+1}}{\Delta y^2} - g(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = 0 \end{cases}$$

## H Jacobienne du schéma du système à deux dimensions, condition aux limites périodique

La jacobienne du système évaluée en  $(x, y)$  avec  $x = (x_1, \dots, x_{(I+1)(J+1)})^T$  et  $y = (y_1, \dots, y_{(I+1)(J+1)})^T$  s'écrit de la manière suivante :

$$\begin{pmatrix} M_1 & N_1 \\ N_2 & M_2 \end{pmatrix}$$

$M_1, N_1, N_2, M_2 \in \mathcal{M}_{(I+1)*(J+1)}(\mathbb{R})$  avec  $N_1, N_2$  diagonales

$$M_1 = \begin{pmatrix} A_1 & B & & & & & & & B \\ B & A_2 & B & & & & & & \\ & & \ddots & \ddots & \ddots & & & & \\ & & & B & A_j & B & & & \\ & & & & \ddots & \ddots & \ddots & & \\ & & & & & B & A_{J-1} & B & \\ B & & & & & & B & A_J & \end{pmatrix} \text{ avec } B \in \mathcal{M}_{I+1}(\mathbb{R}) \text{ et } B = \begin{pmatrix} -\frac{1}{\Delta y^2} & & & \\ & \ddots & & \\ & & & -\frac{1}{\Delta y^2} \end{pmatrix}$$

et pour  $1 \leq j \leq J + 1$ ,  $A_j \in \mathcal{M}_{I+1}(\mathbb{R})$  et  $A_j =$

$$\begin{pmatrix} c_1 - \frac{\partial f}{\partial x}(x_{1,j}, y_{1,j}) & & c_2 & & & & & & c_2 \\ c_2 & & c_1 - \frac{\partial f}{\partial x}(x_{2,j}, y_{2,j}^n) & c_2 & & & & & \\ & & \ddots & \ddots & \ddots & & & & \\ & & & c_2 & c_1 - \frac{\partial f}{\partial x}(x_{i,j}, y_{i,j}) & c_2 & & & \\ & & & & \ddots & \ddots & \ddots & & \\ & & & & & c_2 & c_1 - \frac{\partial f}{\partial x}(x_{I-1,j}, y_{I-1,j}) & & c_2 \\ c_2 & & & & & & c_2 & c_1 - \frac{\partial f}{\partial x}(x_{I,j}, y_{I,j}) & \end{pmatrix}$$

avec  $c_1 = \frac{1}{\Delta t} + \frac{2}{\Delta x^2} + \frac{2}{\Delta y^2}$  et  $c_2 = -\frac{1}{\Delta x^2}$

$$M_2 = \begin{pmatrix} C_1 & D & & & & & D \\ D & C_2 & D & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & D & C_j & D & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & D & C_{J-1} & D \\ D & & & & & D & C_J \end{pmatrix} \text{ avec } D \in \mathcal{M}_{I+1}(\mathbb{R}) \text{ et } D = \begin{pmatrix} -\frac{d}{\Delta y^2} & & \\ & \ddots & \\ & & -\frac{d}{\Delta y^2} \end{pmatrix}$$

et pour  $1 \leq j \leq J+1$ ,  $C_j \in \mathcal{M}_{I+1}(\mathbb{R})$  et  $C_j =$

$$\begin{pmatrix} c_3 - \frac{\partial g}{\partial y}(x_{1,j}, y_{1,j}) & & c_4 & & & & & & c_4 \\ c_4 & & c_3 - \frac{\partial g}{\partial y}(x_{2,j}, y_{2,j}) & c_4 & & & & & \\ & & \ddots & \ddots & \ddots & & & & \\ & & & c_4 & c_3 - \frac{\partial g}{\partial y}(x_{i,j}, y_{i,j}) & c_4 & & & \\ & & & & \ddots & \ddots & \ddots & & \\ & & & & & c_4 & c_3 - \frac{\partial g}{\partial y}(x_{I-1,j}, y_{I-1,j}) & & c_4 \\ c_4 & & & & & & c_4 & c_3 - \frac{\partial g}{\partial y}(x_{I,j}, y_{I,j}) & c_4 \end{pmatrix}$$

avec  $c_3 = \frac{1}{\Delta t} + \frac{2d}{\Delta x^2} + \frac{2d}{\Delta y^2}$  et  $c_4 = -\frac{d}{\Delta x^2}$

$$N_1 = \begin{pmatrix} -\frac{\partial f}{\partial y}((x_{1,1}, y_{1,1})) & & & & \\ & \ddots & & & \\ & & -\frac{\partial f}{\partial y}(x_{I,J}, y_{I,J}) & & \end{pmatrix} \text{ et } N_2 = \begin{pmatrix} -\frac{\partial g}{\partial x}((x_{1,1}, y_{1,1})) & & & & \\ & \ddots & & & \\ & & -\frac{\partial g}{\partial x}(x_{I,J}, y_{I,J}) & & \end{pmatrix}$$