

Simulation d'un modèle Lotka-Volterra spatialisé

Olivier Bonnefon, BioSP, INRA

6 décembre 2013



Objectif et plan

Objectif : Présenter une méthode de simulation du modèle Lotka-Volterra spatialisé suivant :

$$\left\{ \begin{array}{ll} \partial_t u = \operatorname{div}(\rho \nabla u) + ru(1 - u/K) - \beta pu & \text{dans } \Omega \\ \partial_t p = \operatorname{div}(\rho \nabla p) - \delta p + \gamma pu & \text{dans } \Omega \\ + \text{conditions initiales : } u(x, 0) = u_0(x) \text{ et } p(x, 0) = p_0(x) & \text{dans } \Omega \\ + \text{conditions limites portant sur } u \text{ et } p & \text{sur } \Gamma \end{array} \right. \quad (1)$$

Où :

- Ω est l'intérieur du domaine d'étude, Γ est le bord du domaine.
- u, p sont les densités de populations à déterminer.
- $K, r, \delta, \beta, \gamma$ et ρ sont des fonctions de l'espace et du temps.

Objectif et plan

Objectif : Présenter une méthode de simulation du modèle Lotka-Volterra spatialisé suivant :

$$\left\{ \begin{array}{ll} \partial_t u = \operatorname{div}(\rho \nabla u) + ru(1 - u/K) - \beta pu & \text{dans } \Omega \\ \partial_t p = \operatorname{div}(\rho \nabla p) - \delta p + \gamma pu & \text{dans } \Omega \\ + \text{conditions initiales : } u(x, 0) = u_0(x) \text{ et } p(x, 0) = p_0(x) & \text{dans } \Omega \\ + \text{conditions limites portant sur } u \text{ et } p & \text{sur } \Gamma \end{array} \right. \quad (1)$$

Où :

- Ω est l'intérieur du domaine d'étude, Γ est le bord du domaine.
- u, p sont les densités de populations à déterminer.
- $K, r, \delta, \beta, \gamma$ et ρ sont des fonctions de l'espace et du temps.

Plan

- 1 Schéma d'intégration en temps
- 2 Schéma d'intégration en espace
- 3 Résolution du problème non linéaire
- 4 Simulations
 - du modèle (1)
 - d'un couplage de parcelle et de haie

Schéma d'intégration en temps

On considère le modèle logistique suivant :

$$\begin{cases} \partial_t u(t) = au(t)(b - u(t)) \\ u(0) = 0.5 \end{cases} \quad (2)$$

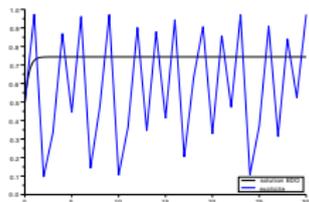
Un schéma d'intégration nous permet de calculer une approximation de la solution, notée $u_{nh} \approx u(nh)$. Ici, h est le pas de temps.

On considère deux schémas :

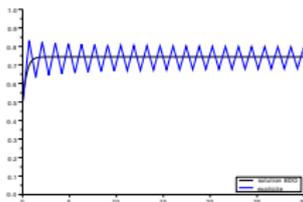
- Le schéma Euler explicite : $(u_{(n+1)h} - u_{nh})/h = au_{nh}(b - u_{nh})$
- Le schéma Euler implicite : $(u_{(n+1)h} - u_{nh})/h = au_{(n+1)h}(b - u_{(n+1)h})$

Comparons ces deux schémas avec $a = 3.9$ et $b = 1 - 1/a$.

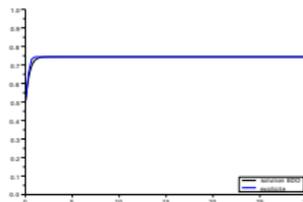
- Euler explicite : $(u_{(n+1)h} - u_{nh})/h = au_{nh}(b - u_{nh})$
- Euler implicite : $(u_{(n+1)h} - u_{nh})/h = au_{(n+1)h}(b - u_{(n+1)h})$



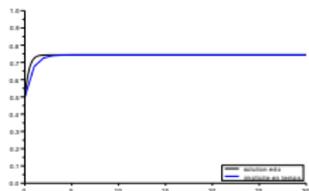
$$u_{n+1} = au_n(b - u_n) + u_n$$



explicite, $h = 0.7$



explicite, $h = 0.33$



$$u_{n+1} = au_{n+1}(b - u_{n+1}) + u_n$$



implicite, $h = 0.7$



implicite, $h = 0.33$

Avec ce schéma explicite, comment choisir h pour ajuster le paramètre a ?

A chaque pas de temps on veut résoudre le problème suivant :

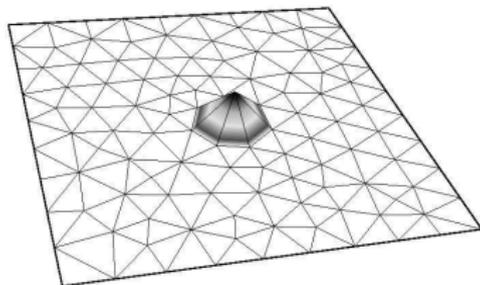
$$\left\{ \begin{array}{l} \frac{1}{h}u = \operatorname{div}(\rho\nabla u) + ru(1 - u/K) - \beta pu + \frac{1}{h}u_{nh} \\ \frac{1}{h}p = \operatorname{div}(\rho\nabla p) - \delta p + \gamma pu + \frac{1}{h}p_{nh} \\ + \text{conditions limites} \end{array} \right. \quad (3)$$

A chaque pas de temps on veut résoudre le problème suivant :

$$\begin{cases} \frac{1}{h} u = \operatorname{div}(\rho \nabla u) + ru(1 - u/K) - \beta pu + \frac{1}{h} u_{nh} \\ \frac{1}{h} p = \operatorname{div}(\rho \nabla p) - \delta p + \gamma pu + \frac{1}{h} p_{nh} \\ + \text{conditions limites} \end{cases} \quad (3)$$

⇒ Méthode des éléments finis¹

On s'appuie sur une discrétisation de l'espace :



On cherche une approximation de la forme :

$$\tilde{u}(x) = \sum_{i=1}^N \alpha_i \phi_i(x)$$

$$\tilde{p}(x) = \sum_{i=1}^N \beta_i \phi_i(x)$$

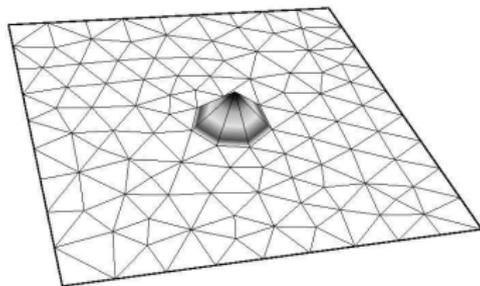
1. *Analyse numérique et optimisation, Grégoire Allaire.*

A chaque pas de temps on veut résoudre le problème suivant :

$$\begin{cases} \frac{1}{h}u = \operatorname{div}(\rho\nabla u) + ru(1 - u/K) - \beta pu + \frac{1}{h}u_{nh} \\ \frac{1}{h}p = \operatorname{div}(\rho\nabla p) - \delta p + \gamma pu + \frac{1}{h}p_{nh} \\ + \text{conditions limites} \end{cases} \quad (3)$$

⇒ Méthode des éléments finis¹

On s'appuie sur une discrétisation de l'espace :



On cherche une approximation de la forme :

$$\tilde{u}(x) = \sum_{i=1}^N \alpha_i \phi_i(x)$$

$$\tilde{p}(x) = \sum_{i=1}^N \beta_i \phi_i(x)$$

$$\forall \phi_i : \int_{\Omega} \left(\left(\frac{1}{h} - 1 \right) \tilde{u} + \tilde{u}^2 + \tilde{u} \tilde{p} \right) \phi_i + \int_{\Omega} \nabla \tilde{u} \nabla \phi_i = \int_{\Omega} \frac{1}{h} u_{nh} \phi_i$$

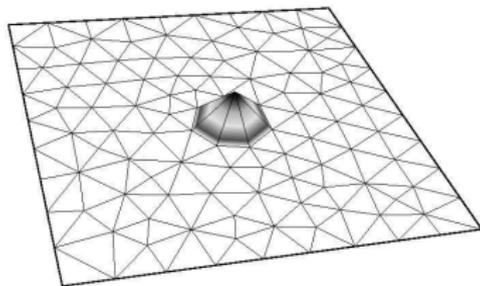
1. *Analyse numérique et optimisation, Grégoire Allaire.*

A chaque pas de temps on veut résoudre le problème suivant :

$$\begin{cases} \frac{1}{h}u = \operatorname{div}(\rho\nabla u) + ru(1 - u/K) - \beta pu + \frac{1}{h}u_{nh} \\ \frac{1}{h}p = \operatorname{div}(\rho\nabla p) - \delta p + \gamma pu + \frac{1}{h}p_{nh} \\ + \text{conditions limites} \end{cases} \quad (3)$$

⇒ Méthode des éléments finis¹

On s'appuie sur une discrétisation de l'espace :



On cherche une approximation de la forme :

$$\tilde{u}(x) = \sum_{i=1}^N \alpha_i \phi_i(x)$$

$$\tilde{p}(x) = \sum_{i=1}^N \beta_i \phi_i(x)$$

$$\forall \phi_i : \int_{\Omega} \left(\left(\frac{1}{h} - 1 \right) \tilde{u} + \tilde{u}^2 + \tilde{u} \tilde{p} \right) \phi_i + \int_{\Omega} \nabla \tilde{u} \nabla \phi_i = \int_{\Omega} \frac{1}{h} u_{nh} \phi_i$$

Cela revient à chercher le zéro d'une fonction non linéaire en $(\alpha, \beta) \Rightarrow$ Newton-Raphson.

1. *Analyse numérique et optimisation, Grégoire Allaire.*

Trouver $(\tilde{u}(x) = \sum_{i=1}^N \alpha_i \phi_i(x), \tilde{p}(x) = \sum_{i=1}^N \beta_i \phi_i(x))$ tel que :

$$\forall \phi_i : \int_{\Omega} \left(\left(\frac{1}{h} - 1 \right) \tilde{u} + \tilde{u}^2 + \tilde{u} \tilde{p} \right) \phi_i + \int_{\Omega} \nabla \tilde{u} \nabla \phi_i = \int_{\Omega} \frac{1}{h} u_{nh} \phi_i$$

Trouver $(\tilde{u}(x) = \sum_{i=1}^N \alpha_i \phi_i(x), \tilde{p}(x) = \sum_{i=1}^N \beta_i \phi_i(x))$ tel que :

$$\forall \phi_i : \int_{\Omega} \left(\left(\frac{1}{h} - 1 \right) \tilde{u} + \tilde{u}^2 + \tilde{u} \tilde{p} \right) \phi_i + \int_{\Omega} \nabla \tilde{u} \nabla \phi_i = \int_{\Omega} \frac{1}{h} u_{nh} \phi_i$$

Algorithme de Newton-Raphson

0) Initialiser l'algorithme : $(u_k, p_k) = (u_{nh}, p_{nh})$

Trouver $(\tilde{u}(x) = \sum_{i=1}^N \alpha_i \phi_i(x), \tilde{p}(x) = \sum_{i=1}^N \beta_i \phi_i(x))$ tel que :

$$\forall \phi_i : \int_{\Omega} \left(\left(\frac{1}{h} - 1 \right) \tilde{u} + \tilde{u}^2 + \tilde{u} \tilde{p} \right) \phi_i + \int_{\Omega} \nabla \tilde{u} \nabla \phi_i = \int_{\Omega} \frac{1}{h} u_{nh} \phi_i$$

Algorithme de Newton-Raphson

- 0) Initialiser l'algorithme : $(u_k, p_k) = (u_{nh}, p_{nh})$
- 1) Linéariser en (u_k, p_k)

Trouver $(\tilde{u}(x) = \sum_{i=1}^N \alpha_i \phi_i(x), \tilde{p}(x) = \sum_{i=1}^N \beta_i \phi_i(x))$ tel que :

$$\forall \phi_i : \int_{\Omega} \left(\left(\frac{1}{h} - 1 \right) \tilde{u} + \tilde{u}^2 + \tilde{u} \tilde{p} \right) \phi_i + \int_{\Omega} \nabla \tilde{u} \nabla \phi_i = \int_{\Omega} \frac{1}{h} u_{nh} \phi_i$$

Algorithme de Newton-Raphson

0) Initialiser l'algorithme : $(u_k, p_k) = (u_{nh}, p_{nh})$

1) Linéariser en (u_k, p_k)

$$u^2 + up \approx 2(u - u_k)u_k + (u - u_k)p_k + (p - p_k)u_k + u_k^2 + u_k p_k$$

Trouver $(\tilde{u}(x) = \sum_{i=1}^N \alpha_i \phi_i(x), \tilde{p}(x) = \sum_{i=1}^N \beta_i \phi_i(x))$ tel que :

$$\forall \phi_i : \int_{\Omega} \left(\left(\frac{1}{h} - 1 \right) \tilde{u} + \tilde{u}^2 + \tilde{u} \tilde{p} \right) \phi_i + \int_{\Omega} \nabla \tilde{u} \nabla \phi_i = \int_{\Omega} \frac{1}{h} u_{nh} \phi_i$$

Algorithme de Newton-Raphson

0) Initialiser l'algorithme : $(u_k, p_k) = (u_{nh}, p_{nh})$

1) Linéariser en (u_k, p_k)

$$u^2 + up \approx 2(u - u_k)u_k + (u - u_k)p_k + (p - p_k)u_k + u_k^2 + u_k p_k$$

$$\forall \phi_i : \int_{\Omega} \left(\left(\frac{1}{h} - 1 + 2u_k + p_k \right) u + p u_k \right) \phi_i + \int_{\Omega} \nabla u \nabla \phi_i = \int_{\Omega} \left(\frac{1}{h} u_{nh} + u_k^2 + u_k p_k \right) \phi_i$$

Trouver $(\tilde{u}(x) = \sum_{i=1}^N \alpha_i \phi_i(x), \tilde{p}(x) = \sum_{i=1}^N \beta_i \phi_i(x))$ tel que :

$$\forall \phi_i : \int_{\Omega} \left(\left(\frac{1}{h} - 1 \right) \tilde{u} + \tilde{u}^2 + \tilde{u} \tilde{p} \right) \phi_i + \int_{\Omega} \nabla \tilde{u} \nabla \phi_i = \int_{\Omega} \frac{1}{h} u_{nh} \phi_i$$

Algorithme de Newton-Raphson

0) Initialiser l'algorithme : $(u_k, p_k) = (u_{nh}, p_{nh})$

1) Linéariser en (u_k, p_k)

$$u^2 + up \approx 2(u - u_k)u_k + (u - u_k)p_k + (p - p_k)u_k + u_k^2 + u_k p_k$$

$$\forall \phi_i : \int_{\Omega} \left(\left(\frac{1}{h} - 1 + 2u_k + p_k \right) u + p u_k \right) \phi_i + \int_{\Omega} \nabla u \nabla \phi_i = \int_{\Omega} \left(\frac{1}{h} u_{nh} + u_k^2 + u_k p_k \right) \phi_i$$

2) Résoudre le problème linéaire $M \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = B$

Trouver $(\tilde{u}(x) = \sum_{i=1}^N \alpha_i \phi_i(x), \tilde{p}(x) = \sum_{i=1}^N \beta_i \phi_i(x))$ tel que :

$$\forall \phi_i : \int_{\Omega} \left(\left(\frac{1}{h} - 1 \right) \tilde{u} + \tilde{u}^2 + \tilde{u} \tilde{p} \right) \phi_i + \int_{\Omega} \nabla \tilde{u} \nabla \phi_i = \int_{\Omega} \frac{1}{h} u_{nh} \phi_i$$

Algorithme de Newton-Raphson

0) Initialiser l'algorithme : $(u_k, p_k) = (u_{nh}, p_{nh})$

1) Linéariser en (u_k, p_k)

$$u^2 + up \approx 2(u - u_k)u_k + (u - u_k)p_k + (p - p_k)u_k + u_k^2 + u_k p_k$$

$$\forall \phi_i : \int_{\Omega} \left(\left(\frac{1}{h} - 1 + 2u_k + p_k \right) u + p u_k \right) \phi_i + \int_{\Omega} \nabla u \nabla \phi_i = \int_{\Omega} \left(\frac{1}{h} u_{nh} + u_k^2 + u_k p_k \right) \phi_i$$

2) Résoudre le problème linéaire $M \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = B$

$$M = \left(\begin{array}{c|c} \int_{\Omega} \left(\frac{1}{h} - 1 + 2u_k + p_k \right) \phi_j \phi_i + \nabla \phi_j \nabla \phi_i & \int_{\Omega} u_k \phi_j \phi_i \\ \hline \int_{\Omega} p_k \phi_j \phi_i & \int_{\Omega} \left(\frac{1}{h} + 1 - u_k \right) \phi_j \phi_i + \nabla \phi_j \nabla \phi_i \end{array} \right)$$

$$B = \begin{pmatrix} \int_{\Omega} \left(\frac{1}{h} u_{nh} + u_k^2 + u_k p_k \right) \phi_i \\ \int_{\Omega} \left(\frac{1}{h} p_{nh} - u_k p_k \right) \phi_i \end{pmatrix}$$

Trouver $(\tilde{u}(x) = \sum_{i=1}^N \alpha_i \phi_i(x), \tilde{p}(x) = \sum_{i=1}^N \beta_i \phi_i(x))$ tel que :

$$\forall \phi_i : \int_{\Omega} \left(\left(\frac{1}{h} - 1 \right) \tilde{u} + \tilde{u}^2 + \tilde{u} \tilde{p} \right) \phi_i + \int_{\Omega} \nabla \tilde{u} \nabla \phi_i = \int_{\Omega} \frac{1}{h} u_{nh} \phi_i$$

Algorithme de Newton-Raphson

0) Initialiser l'algorithme : $(u_k, p_k) = (u_{nh}, p_{nh})$

1) Linéariser en (u_k, p_k)

$$u^2 + up \approx 2(u - u_k)u_k + (u - u_k)p_k + (p - p_k)u_k + u_k^2 + u_k p_k$$

$$\forall \phi_i : \int_{\Omega} \left(\left(\frac{1}{h} - 1 + 2u_k + p_k \right) u + p u_k \right) \phi_i + \int_{\Omega} \nabla u \nabla \phi_i = \int_{\Omega} \left(\frac{1}{h} u_{nh} + u_k^2 + u_k p_k \right) \phi_i$$

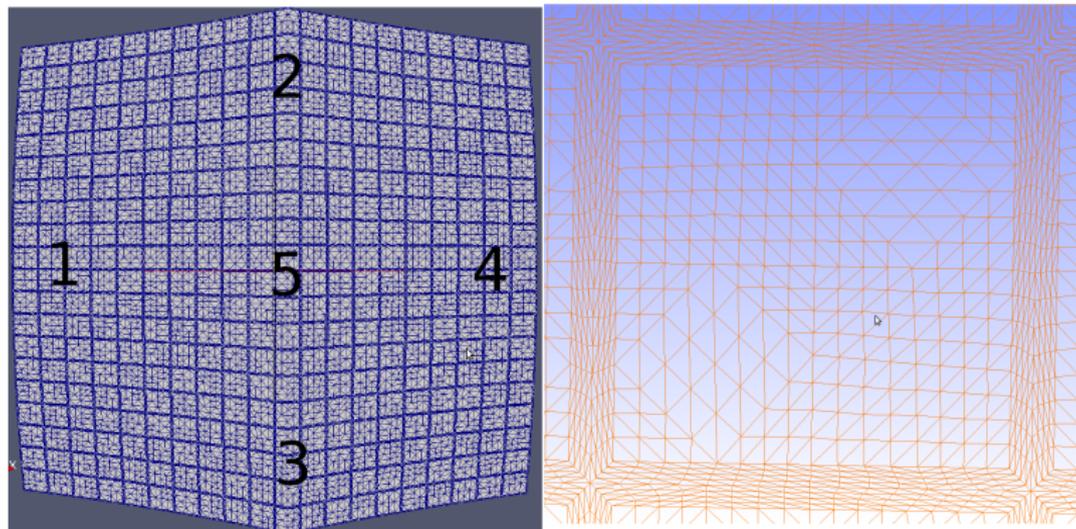
2) Résoudre le problème linéaire $M \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = B$

$$M = \left(\begin{array}{c|c} \int_{\Omega} \left(\frac{1}{h} - 1 + 2u_k + p_k \right) \phi_j \phi_i + \nabla \phi_j \nabla \phi_i & \int_{\Omega} u_k \phi_j \phi_i \\ \hline \int_{\Omega} p_k \phi_j \phi_i & \int_{\Omega} \left(\frac{1}{h} + 1 - u_k \right) \phi_j \phi_i + \nabla \phi_j \nabla \phi_i \end{array} \right)$$

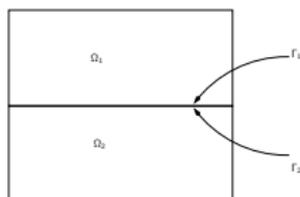
$$B = \begin{pmatrix} \int_{\Omega} \left(\frac{1}{h} u_{nh} + u_k^2 + u_k p_k \right) \phi_i \\ \int_{\Omega} \left(\frac{1}{h} p_{nh} - u_k p_k \right) \phi_i \end{pmatrix}$$

3) Si convergence alors $(\tilde{u}, \tilde{p}) = (u, v)$, sinon $(u_k, p_k) = (u, v)$ et aller à l'étape 1).

Premier exemple



- 300 pas de temps
- 5 types de cultures + une bande entre les cultures
- Les paramètres du modèle dépendent du type de culture et varient dans le temps

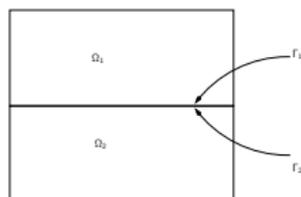
Deuxième exemple : Couplage d'un modèle 2D avec un modèle 1D.²

$$\left\{ \begin{array}{l} \partial_t u_1 = D \partial_{xx} u_1 \\ \partial_t v_1 = d \Delta v_1 + f(v_1) \\ \\ + \text{conditions initiales} \end{array} \right.$$

$$\begin{array}{l} x \in \Gamma_1, t > 0 \\ (x, y) \in \Omega_1, t > 0 \end{array}$$

(4)

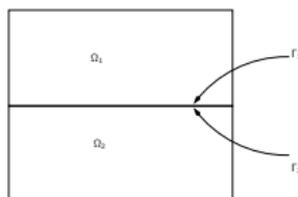
2. The influence of a line with fast diffusion on fisher-KPP propagation. H Berestycki, JM Roquejoffre, L Rossi.

Deuxième exemple : Couplage d'un modèle 2D avec un modèle 1D.²

$$\left\{ \begin{array}{l} \partial_t u_1 = D\partial_{xx} u_1 + \nu v_1(x, 0, t) - \mu u_1 \\ \partial_t v_1 = d\Delta v_1 + f(v_1) \\ -d\partial_y v_1 = \mu u_1(x, t) - \nu v_1(x, 0, t) \end{array} \right. \quad \begin{array}{l} x \in \Gamma_1, t > 0 \\ (x, y) \in \Omega_1, t > 0 \\ x \in \Gamma_1, t > 0 \end{array} \quad (4)$$

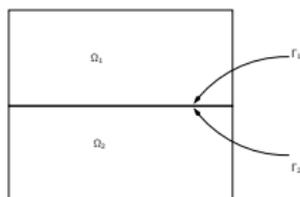
+conditions initiales

2. The influence of a line with fast diffusion on fisher-KPP propagation. H Berestycki, JM Roquejoffre, L Rossi.

Deuxième exemple : Couplage d'un modèle 2D avec un modèle 1D.²

$$\left\{ \begin{array}{l}
 \partial_t u_1 = D\partial_{xx} u_1 + \nu v_1(x, 0, t) - \mu u_1 \\
 \partial_t v_1 = d\Delta v_1 + f(v_1) \\
 -d\partial_y v_1 = \mu u_1(x, t) - \nu v_1(x, 0, t) \\
 \\
 \partial_t u_2 = D\partial_{xx} u_2 + \nu v_2(x, 0, t) - \mu u_2 \\
 \partial_t v_2 = d\Delta v_2 + f(v_2) \\
 d\partial_y v_2 = \mu u_2(x, t) - \nu v_2(x, 0, t) \\
 + \text{conditions initiales}
 \end{array} \right. \quad \begin{array}{l}
 x \in \Gamma_1, t > 0 \\
 (x, y) \in \Omega_1, t > 0 \\
 x \in \Gamma_1, t > 0 \\
 \\
 x \in \Gamma_2, t > 0 \\
 (x, y) \in \Omega_2, t > 0 \\
 x \in \Gamma_2, t > 0
 \end{array} \quad (4)$$

2. The influence of a line with fast diffusion on fisher-KPP propagation. H Berestycki, JM Roquejoffre, L Rossi.

Deuxième exemple : Couplage d'un modèle 2D avec un modèle 1D.²

$$\left\{ \begin{array}{ll}
 \partial_t u_1 = D\partial_{xx} u_1 + \nu v_1(x, 0, t) - \mu u_1 - \alpha_1 u_1 + \alpha_2 u_2 & x \in \Gamma_1, t > 0 \\
 \partial_t v_1 = d\Delta v_1 + f(v_1) & (x, y) \in \Omega_1, t > 0 \\
 -d\partial_y v_1 = \mu u_1(x, t) - \nu v_1(x, 0, t) & x \in \Gamma_1, t > 0 \\
 \\
 \partial_t u_2 = D\partial_{xx} u_2 + \nu v_2(x, 0, t) - \mu u_2 - \alpha_2 u_2 + \alpha_1 u_1 & x \in \Gamma_2, t > 0 \\
 \partial_t v_2 = d\Delta v_2 + f(v_2) & (x, y) \in \Omega_2, t > 0 \\
 d\partial_y v_2 = \mu u_2(x, t) - \nu v_2(x, 0, t) & x \in \Gamma_2, t > 0 \\
 + \text{conditions initiales} &
 \end{array} \right. \quad (4)$$

2. The influence of a line with fast diffusion on fisher-KPP propagation. H Berestycki, JM Roquejoffre, L Rossi.