

Electrocardiogramme

Rémi HERNANDEZ
Romuald MAKOSSO NOMBO
Amandine CAUT
Célia BOULTABI

Tuteur : Y.Courdiere

Université de Bordeaux

9 décembre 2017

Table des matières

1	Étude théorique	4
2	programmation dans un cas simplifié	6
2.1	Schémas numériques pour la résolution	6
2.1.1	Euler explicite	6
2.1.2	Euler implicite	10
2.2	Méthodes de travail	10
2.3	Modèle	11

Chapitre 1

Étude théorique

L'électrophysiologie permet de modéliser la propagation de différentes ondes dans le cœur.

Pour cela nous allons utiliser cette équation :

$$\partial_t u(x, t) - \partial_x(a(x)\partial_x u(x, t)) = f(t, x)$$

Afin de résoudre cette équation il faut certaines conditions initiales et aux bords. Nos recherches dans différents livres et articles nous permettent alors de déduire les conditions suivantes :

$$\begin{cases} \partial_t u(x, t) - \partial_x(a(x)\partial_x u(x, t)) = f(t, x) \\ a(x)\partial_x u(x, t) = 0 \text{ sur } \partial\Omega \\ u(0, x) = u_0(x) \text{ } t \in [0, T] \end{cases}$$

Il est nécessaire que $a(x) > 0$, afin que le phénomène ne soit pas réversible ! En effet, on ne peut inverser la diffusion de la chaleur.

On remarque par ailleurs que l'on a l'équation chaleur pour le cas $a(x) = 1$. On résoudra cette équation dans le chapitre 2.

Situation où l'on peut calculer des solutions analytiques :
dans le cas ($a(x) = 1$) on a l'équation de la chaleur. De plus avec nos conditions aux bords, ie les dérivées sont nulles, on a deux cas, $f(x) \neq 0$ ou $f(x) = 0$. Ici, on prend le cas où $f(x) = 0$:

la solution est la fonction u définie par :

$$u(x, t) = \cos(k\pi x) \exp(-k^2 \pi^2 t)$$

Résolution du cas :

Nous allons résoudre le système par la méthode de séparation des variables.

On pose $u(x,t) = f(x)g(t)$

de ce fait :

$$\partial_t u(x,t) = f(x)g'(t)$$

et

$$\partial_{xx}^2 u(x,t) = f''(x)g(t)$$

On obtient :

$$\frac{g'(t)}{g(t)} = \frac{f''(x)}{f(x)} = c$$

Et donc : $g(t) = \lambda \exp(ct)$ avec $c < 0$ sinon la solution explose ce qui est impossible.

$$\frac{f''(x)}{f(x)} = c \Leftrightarrow f''(x) - cf(x) = 0$$

$$\Leftrightarrow f(x) = P \cos(x\sqrt{|c|}) + Q \sin(x\sqrt{|c|})$$

par compilation on a alors,

$$u(x,t) = \lambda \exp(ct) (P \cos(x\sqrt{|c|}) + Q \sin(x\sqrt{|c|}))$$

or $\partial_x u(x,t) = 0$ sur $\partial\Omega$, donc on dérive l'expression et on remplace :

$$-P\sqrt{|c|} \sin(x\sqrt{|c|}) + Q\sqrt{|c|} \cos(x\sqrt{|c|}) = 0$$

On obtient le système de résolution suivant :

$$\begin{cases} x = 0 \Leftrightarrow Q\sqrt{|c|} \cos(0) = 0 \\ x = 1 \Leftrightarrow -P\sqrt{|c|} \sin(\sqrt{|c|}) = 0 \end{cases}$$
$$\Leftrightarrow \begin{cases} x = 0 \Leftrightarrow Q = 0 \text{ car } c < 0 \\ x = 1 \Leftrightarrow \sin(\sqrt{|c|}) = 0 \text{ car } P \neq 0 \end{cases}$$

On obtient ainsi : $c = -(k\pi)^2 \forall k \in \mathbb{Z}$

La solution s'écrit donc $u(x,t) = \lambda \exp(-(k\pi)^2 t)$, avec la condition initiale $u(0,x) = u_0(x)$ on a alors $u_0(x) = \cos(xk\pi)$ avec $\lambda = 1$.

Chapitre 2

programmation dans un cas simplifié

2.1 Schémas numériques pour la résolution

Pour la résolution de cette équation, on utilise un maillage uniforme avec un pas de Δt en temps et Δx en espace.

La dérivée seconde en espace est approchée par :

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j)}{\Delta x^2} \quad (2.1)$$

2.1.1 Euler explicite

On approche $\frac{\partial u}{\partial t}(x_i, t_j)$ par :

$$\frac{u(x_i, t_j + 1) - u(x_i, t_j)}{\Delta t} \quad (2.2)$$

On obtient le schéma suivant :

$$\frac{u(x_i, t_j + 1) - u(x_i, t_j)}{\Delta t} - \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j)}{\Delta x^2} = f(x_i, t_j) \quad (2.3)$$

Pour la suite on écrira u_i^j à la place de $u(x_i, t_j)$.

Aussi on utilisera la notation : $U^j = \begin{pmatrix} u_0^j \\ u_1^j \\ \vdots \\ u_N^j \end{pmatrix}$

Le schéma s'écrit sous forme matricielle :

$$\frac{U^{j+1} - U^j}{\Delta t} + A_{\Delta x} U^j = F^j \quad (2.4)$$

Avec :

$$A_{\Delta x} = \frac{1}{\Delta x^2} \begin{pmatrix} 2 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & -1 & 2 & -1 \\ 0 & \dots & \dots & 0 & -1 & 2 \end{pmatrix} \quad (2.5)$$

$$\text{Et on a } F^j = \begin{pmatrix} f(x_1, t_j) \\ f(x_2, t_j) \\ \vdots \\ f(x_N, t_j) \end{pmatrix}$$

Par les formules de Taylor, on discrétise la dérivée seconde en espace :

$$f(x + \Delta x, t) = f(x, t) + \Delta x \partial_x f(x, t) + \frac{(\Delta x)^2}{2} \partial_{xx}^2 f(x, t) + \theta(\Delta x^3)$$

$$f(x - \Delta x, t) = f(x, t) - \Delta x \partial_x f(x, t) + \frac{(\Delta x)^2}{2} \partial_{xx}^2 f(x, t) + \theta(\Delta x^3)$$

On soustrait les deux équations :

$$f(x + \Delta x, t) - f(x - \Delta x, t) = 2\Delta x \partial_x f(x, t) + \theta(\Delta x^3)$$

$$\frac{f(x + \Delta x, t) - f(x - \Delta x, t)}{2\Delta x} = \partial_x f(x, t) + \theta(\Delta x^2)$$

$$\partial_x f_k^n = \frac{f_{k+1}^n - f_{k-1}^n}{2\Delta x}$$

On effectue le même raisonnement que précédemment pour trouver la dérivée seconde :

$$\partial_x(a(x) \partial_x u(x, t)) = \frac{1}{\Delta x} (a_{k+\frac{1}{2}} \partial_x u_{k+\frac{1}{2}}^n - a_{k-\frac{1}{2}} \partial_x u_{k-\frac{1}{2}}^n)$$

On remplace dans l'équation de la dérivée seconde ce que l'on a trouvée pour la dérivée première :

$$\partial_x(a(x)\partial u(x,t))_k^n = \frac{1}{\Delta x}(a_{k+\frac{1}{2}}(\frac{u_{k+1}^n - u_k^n}{\Delta x}) - a_{k-\frac{1}{2}}(\frac{u_k^n - u_{k-1}^n}{\Delta x}))$$

On factorise par $\frac{1}{\Delta x}$, ce qui donne :

$$\partial_x(a(x)\partial u(x,t)) = \frac{1}{(\Delta x)^2}(a_{k+\frac{1}{2}}u_{k+1}^n - (a_{k+\frac{1}{2}} + a_{k-\frac{1}{2}})u_k^n + a_{k-\frac{1}{2}}u_{k-1}^n)$$

On discrétise la dérivée en temps par $\frac{u_k^{n+1} - u_k^n}{\Delta t}$, de ce fait on a le système suivant :

Pour $n \in [0, J]$ et $k \in [1, N-1]$, avec $J = \frac{T}{\Delta t}$ et $N = \frac{1}{\Delta x}$, on a :

$$\begin{cases} \frac{u_k^{n+1} - u_k^n}{\Delta t} - \frac{1}{(\Delta x)^2}(a_{k+\frac{1}{2}}u_{k+1}^n - (a_{k+\frac{1}{2}} + a_{k-\frac{1}{2}})u_k^n + a_{k-\frac{1}{2}}u_{k-1}^n) = f_k^n \\ a(x)\partial_x u_0^n = 0 \leftrightarrow \partial_x u_0^n = 0 \\ \partial_x u_N^n = 0 \\ a_N \partial_x u_N^n = 0 \end{cases}$$

On isole le u_k^{n+1} :

$$u_k^{n+1} = \frac{\Delta t}{\Delta x^2}(a_{k+\frac{1}{2}}u_{k+1}^n - (a_{k+\frac{1}{2}} + a_{k-\frac{1}{2}})u_k^n + a_{k-\frac{1}{2}}u_{k-1}^n) + f_k^n + u_k^n$$

On regroupe les termes u_k^n , u_{k-1}^n et u_{k+1}^n :

$$u_k^{n+1} = \frac{\Delta t}{\Delta x^2}a_{k-\frac{1}{2}}u_{k-1}^n + (1 - \frac{\Delta t}{\Delta x^2}(a_{k-\frac{1}{2}} + a_{k+\frac{1}{2}}))u_k^n + \frac{\Delta t}{\Delta x^2}a_{k+\frac{1}{2}}u_{k+1}^n + f_k^n$$

D'après les conditions au bord, on a :

$$\begin{cases} \partial_x(a(x)\partial u(x,t))_0^n = \frac{2}{\Delta x}a_{\frac{1}{2}}\partial_x u_{\frac{1}{2}}^n = \frac{2}{\Delta x}a_{\frac{1}{2}}\frac{u_1^n - u_0^n}{\Delta x} \\ \partial_x(a(x)\partial u(x,t))_N^n = 0 - \frac{a_{N-\frac{1}{2}}\partial_x u_{N-\frac{1}{2}}^n}{\frac{\Delta x}{2}} = -\frac{2}{\Delta x}a_{N-\frac{1}{2}}\frac{u_N^n - u_{N-1}^n}{\Delta x} \end{cases}$$

De ce fait l'équation discrétisée au bord donne :

$$\frac{u_k^{n+1} - u_k^n}{\Delta x} - \frac{2}{(\Delta x)^2}a_{N-\frac{1}{2}}(\frac{u_N^n - u_{N-1}^n}{\Delta x}) = f_k^n$$

pour $k=0$: $\frac{u_0^{n+1}-u_0^n}{\Delta x} - \frac{2}{(\Delta x)^2}a_{\frac{1}{2}}(u_1^n - u_0^n) = f_0^n$

On isole u_0^{n+1} : $u_0^{n+1} = \frac{2\Delta t}{\Delta x^2}a_{\frac{1}{2}}(u_1^n - u_0^n) + f_0^n + u_0^n$

Puis, $u_0^{n+1} = u_0^n - \frac{2\Delta t}{\Delta x^2}a_{\frac{1}{2}}u_0^n + \frac{2\Delta t}{\Delta x^2}a_{\frac{1}{2}}u_1^n + f_0^n$

Finalement, l'équation s'écrit pour $k=0, \dots, N$ et $n=1, \dots, J$:

$$\begin{cases} u_k^{n+1} = u_k^n + \frac{\Delta t}{\Delta x^2}a_{k-\frac{1}{2}}u_{k-1}^n - \frac{\Delta t}{\Delta x^2}(a_{k+\frac{1}{2}} + a_{k-\frac{1}{2}})u_k^n + \frac{\Delta t}{\Delta x^2}a_{k+\frac{1}{2}}u_{k+1}^n + f_k^n \\ u_k^{n+1} = u_k^n + \frac{2\Delta t}{\Delta x}a_{n-\frac{1}{2}}u_{N-1}^n - \frac{2\Delta t}{\Delta x}a_{n-\frac{1}{2}}u_N^n + f_N^n \end{cases}$$

avec $u^n = (u_0^n, u_1^n, \dots, u_N^n)$ et $F^n = (f_0^n, f_1^n, \dots, f_N^n)$

On écrit cela sous forme matricielle :

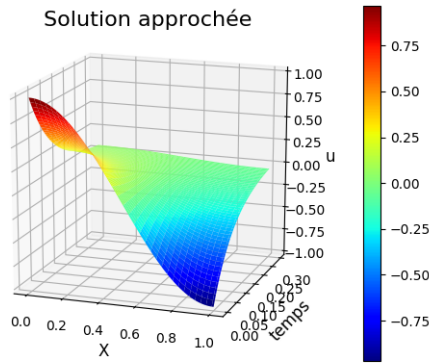
$$U^{n+1} = U^n - \frac{\Delta t}{\Delta x^2}AU^n + F^n\Delta t$$

On obtient alors la matrice suivante :

$$A = \begin{pmatrix} 2a_{\frac{1}{2}} & -2a_{\frac{1}{2}} & 0 & \dots & \dots & 0 \\ -a_{\frac{1}{2}} & -(a_{\frac{1}{2}} + a_{\frac{3}{2}}) & -a_{\frac{3}{2}} & 0 & \ddots & \vdots \\ 0 & -a_{\frac{3}{2}} & -(a_{\frac{5}{2}} + a_{\frac{3}{2}}) & -a_{\frac{5}{2}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & -a_{N-\frac{3}{2}} & -(a_{N-\frac{1}{2}} + a_{N+\frac{3}{2}}) & a_{N-\frac{1}{2}} \\ 0 & \dots & \dots & 0 & -2a_{N-\frac{1}{2}} & -2a_{N-\frac{1}{2}} \end{pmatrix} \quad (2.6)$$

$$U^{n+1} = (I_N - \frac{\Delta t}{\Delta x^2}A_N)U^n + F^n\Delta t$$

On obtient par transformée de Fourier en espace, la condition CFL pour la stabilité L^2 , $\Delta t \leq \frac{\Delta x^2}{2}$.



2.1.2 Euler implicite

Par le même raisonnement que pour Euler explicite, en prenant pour la dérivée en espace :

$$\frac{u_k^n - u_k^{n-1}}{\Delta t} - \frac{1}{(\Delta x)^2} (a_{k+\frac{1}{2}} u_{k+1}^n - (a_{k+\frac{1}{2}} + a_{k-\frac{1}{2}}) u_k^n + a_{k-\frac{1}{2}} u_{k-1}^n) = f_k^n$$

Enfin, l'écriture matricielle obtenue est :

$$(Id - A)U^n = U^{n-1} + \Delta t F^n$$

2.2 Méthodes de travail

Nous allons présenter dans cette partie comment nous avons travaillé en groupe afin de résoudre le projet.

Le problème rencontré est la différence de niveau en informatique au sein du groupe. Afin de surmonter ce problème, il a été décidé que nous résoudrions le projet par compétences. La partie programmation a été résolue par ceux qui avaient des compétences en informatiques. La partie théorique, comme la résolution de l'équation de la chaleur, et l'écriture du rapport a été élaboré par ceux qui ne savaient pas programmer. Il nous a paru clair que cette méthode était la plus efficace au niveau gain de temps. Le problème pouvant être la répartition de la charge de travail qui était affecté par les lacunes en informatique de certains membres du

groupe.

2.3 Modèle

Le système linéaire du minimal model, en dimension 1 :

$$\frac{\partial u}{\partial t} = \nabla(D \nabla U) - (J_{fi} + J_{so} + J_{si}) \quad (2.7)$$

$$\frac{\partial v}{\partial t} = \frac{(1 - H(u - \theta_v))(v_\infty - v)}{\tau_v^-} - \frac{H(u - \theta_v)v}{\theta_v^+} \quad (2.8)$$

$$\frac{\partial w}{\partial t} = \frac{(1 - H(u - \theta_w))(w_\infty - w)}{\tau_w^-} - \frac{H(u - \theta_w)w}{\theta_w^+} \quad (2.9)$$

$$\frac{\partial w}{\partial t} = \frac{\frac{1 + \tanh(k_s(u - u_s))}{2 - s}}{\tau_s} \quad (2.10)$$

Avec :

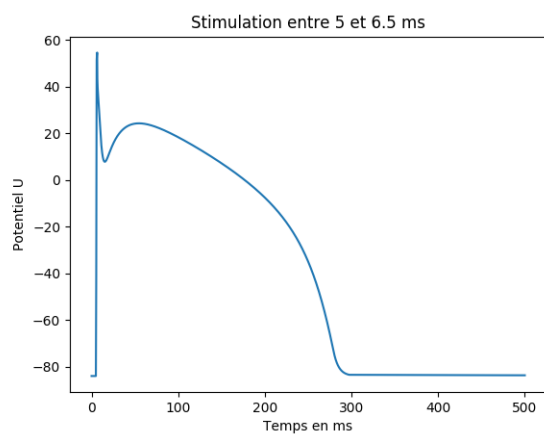
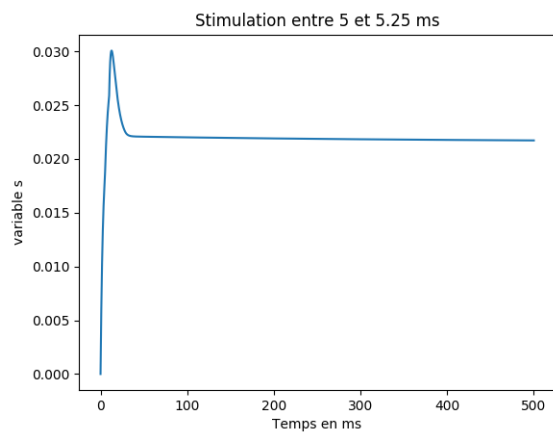
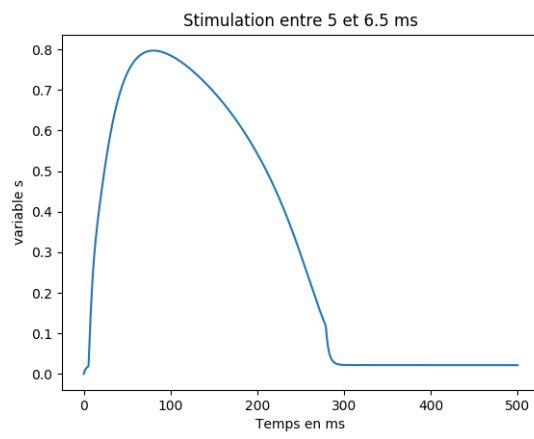
$$J_{si} = \frac{-vH(u - \theta_v)(u - \theta_v)(u_u - u)}{\tau_{fi}}$$

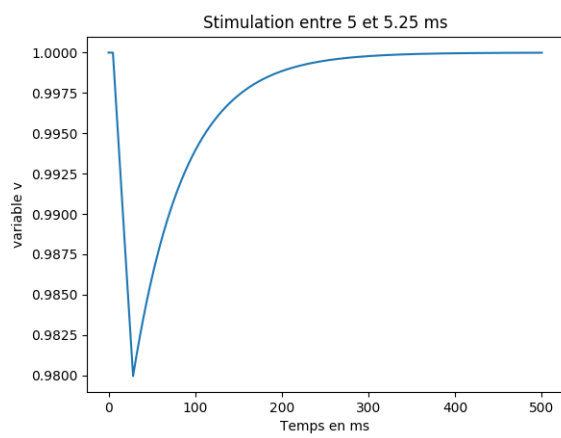
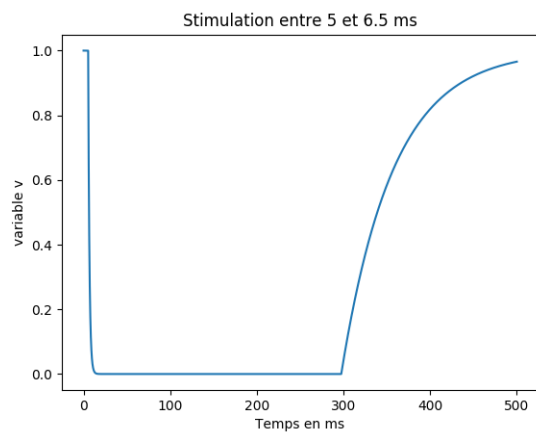
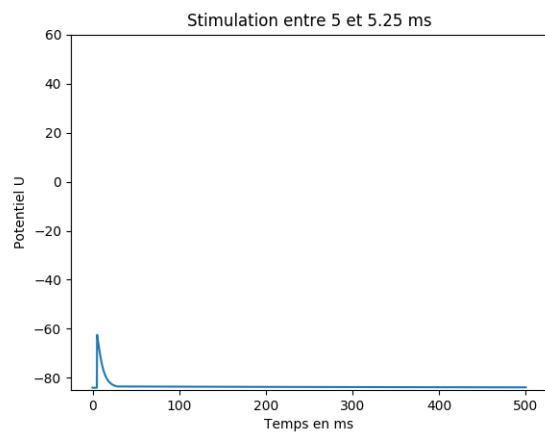
$$J_{so} = \frac{\frac{u - u_0}{1 - H(u - \theta_w)}}{\tau_{so}}$$

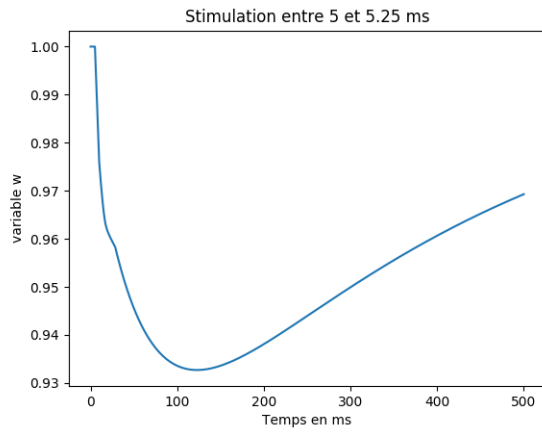
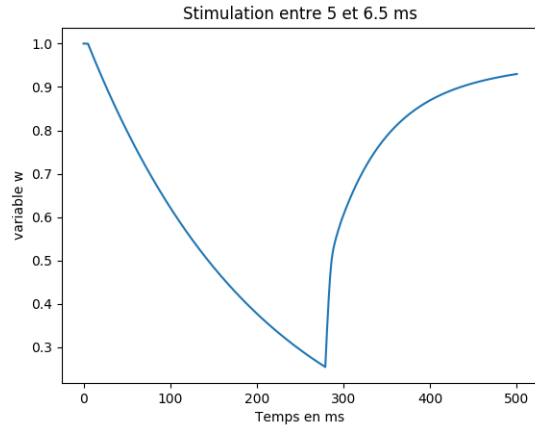
$$J_{si} = \frac{-H(u - \theta_w)ws}{\tau_{si}}$$

Nous avons des représentations graphiques du minimal model après un codage en python :

Sur les quatres premières images nous avons choisit une simulation sur l'intervalle de temps : $[5; 6.5]$ ms







Passage en dimension 2 :

Dans le passage en dimension 2 on se retrouve avec (l'équation) , la variable $a(x,y)$ devient une matrice $2 * 2$.

Nous choisirons la matrice nommée A , diagonale composée de variables mesurables et positives :

$$\alpha(x, y) \text{ et } \beta(x, y)$$

le choix de ces propriétés vient du fait que nous modélisons un phénomène de propagation (du flux électrique qui travers les ventricule du cœur)

Notre matrice A sera ainsi de cette forme là :

$$A = \begin{pmatrix} \alpha(x, y) & 0 \\ 0 & \beta(x, y) \end{pmatrix} \quad (2.11)$$

Nous allons essayer de trouver les solutions analytiques de cette equation de dimension

2 :

$$\partial_t u - \begin{pmatrix} \partial_x & \partial_y \end{pmatrix} \begin{pmatrix} \alpha(x, y) & 0 \\ 0 & \beta(x, y) \end{pmatrix} \begin{pmatrix} \partial_x u \\ \partial_y u \end{pmatrix} = f(x, y) \quad (2.12)$$

On développe :

$$\partial_t u - \begin{pmatrix} \partial_x & \partial_y \end{pmatrix} \begin{pmatrix} \alpha \partial_x U & \beta \partial_y U \end{pmatrix} = f(x, y) \quad (2.13)$$

Et on se retrouve avec cette equation :

$$\partial_t u - \partial_x(\alpha \partial_x U) - \partial_y(\beta \partial_y U) = f(x, y) \quad (2.14)$$

Avec les conditions aux bords :

$$\partial_x U = \partial_y U = 0$$

Nous allons maintenant appliquer le schéma d'Euler explicite

$$\partial_t (U_{i,j})^n = \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\delta t} \quad (2.15)$$

On applique les différences finies :

$$(\partial_x(\alpha \partial_x U))_{i,j}^n = \frac{\alpha_{i+1/2}^j \partial_x U_{i+1/2}^n - \alpha_{i-1/2}^j \partial_x U_{i-1/2}^n}{\delta x} = \alpha_{i+1/2}^j \left(\frac{U_{i+1,j}^n - U_{i,j}^n}{\delta^2} \right) - \alpha_{i-1/2}^j \left(\frac{U_{i,j}^n - U_{i-1,j}^n}{\delta^2} \right) \quad (2.16)$$

$$\partial_x(\alpha \partial_x U)_{i,j}^n = \frac{\alpha_{i-1/2}^j}{\delta x^2} U_{i,j-1}^n + \frac{\alpha_{i+1/2}^j - \alpha_{i-1/2}^j}{\delta x^2} U_{i,j}^n + \frac{\alpha_{i+1/2}^j}{\delta x^2} U_{i+1,j}^n \quad (2.17)$$

Idem pour les dérivées partielles par rapport à y :

$$\partial_y(\beta \partial_y U)_{i,j}^n = \frac{\beta_i^{j-1/2}}{\delta y^2} U_{i,j-1}^n + \frac{\beta_i^{j+1/2} - \beta_i^{j-1/2}}{\delta y^2} U_{i,j}^n + \frac{\beta_i^{j+1/2}}{\delta y^2} U_{i,j+1}^n \quad (2.18)$$

$$i = 1, \dots, N - 1$$

$$j = 1, \dots, N - 1$$

Du coup ,

$$\partial t(U_{i,j})^n - \partial y(\beta \partial y U)_{i,j}^n - \partial x(\alpha \partial x U)_{i,j}^n = f_i^j \quad (2.19)$$

Donc on a comme schéma :

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\delta t} - \frac{\alpha_{i-1/2}^j u_{i-1,j}^n + \beta_i^{j-1/2} u_{i,j-1}^n + (\alpha_{i+1/2}^j - \alpha_{i-1/2}^j + \beta_i^{j+1/2} - \beta_i^{j-1/2}) u_{i,j}^n + \alpha_{i+1/2}^j u_{i+1,j}^n + \beta_i^{j+1/2} u_{i,j+1}^n}{h^2} \quad (2.20)$$

Avec les conditions aux bords :

$$\partial x(\alpha \partial x u)_{0,j}^n = 0 \quad (2.21)$$

$$u_{0,j}^{n+1} = 2 \frac{\delta t}{\delta x^2} \alpha_{1/2}^j (u_{1,j}^n - u_{0,j}^n) f_0^j + u_{0,j}^n \quad (2.22)$$

$$u_{N,j}^{n+1} = u_{N,j}^n + 2 \frac{\delta t}{\delta x} \alpha_{N-1/2}^j U_{N-1,j}^n - 2 \frac{\delta t}{\delta x} \alpha_{N-1/2}^j U_{N,j}^n + f_N^j \quad (2.23)$$

$$u_{i,N}^{n+1} = u_{i,N}^n + 2 \frac{\delta t}{\delta x} \beta_i^{N-1/2} U_{i,N-1}^n - 2 \frac{\delta t}{\delta x} \beta_i^{N-1/2} U_{i,N}^n + f_i^N \quad (2.24)$$

$$u_{i,0}^{n+1} = \frac{\delta t}{\delta x} \beta_i^{1/2} (U_{j,1}^n - U_{i,0}^n) j_j^0 + U_{i,0}^n \quad (2.25)$$

Pour coder, on pose A comme étant la matrice identité, et on fait une discrétisation uniforme en espace $\delta x = \delta y = h$