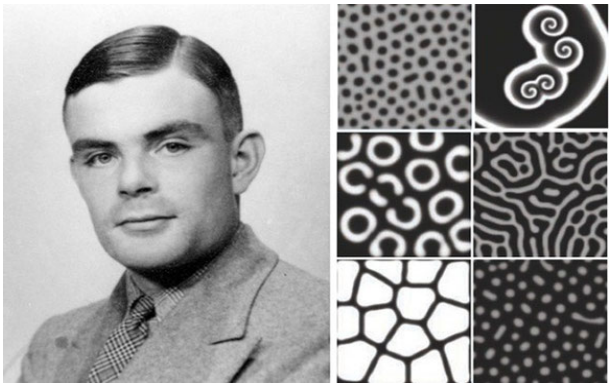
POLYTECH LYON

Rapport du T.P. :
5A. Mathématiques appliquées et Modélisation 2017-2018 (option C.O. et B.D.P.)

Modélisation numérique : Module Biologie



Travail effectué par : ABOUADALLAH Mohamed Anwar,

École Polytechnique Lyon
15 Boulevard Latarjet 69622 Villeurbanne

Table des matières

1 Introduction et présentation de notre modèle :

1.1 Table des notation :

Notation	Signification
[A]	La concentration de la substance A
D	Notre domaine d'étude, ouvert de $\mathbb{R}^n$.
D^*	$D^* \in D$ de frontière ∂D^*.
$u = u(x,t)$	La concentration du morphogène au point x et au temps t
$f = f(u)$	Le taux de création de u
J	Le flux de concentration à travers D^*
D	Matrice de diffusion à coefficients constants.

1.2 Explication de la morphogénèse

Dans ce T.P., nous allons étudié la morphogénèse [3] qui est un processus par lequel se développent les structures d'un organisme. Chez les organismes multicellulaires, une théorie stipule que ce sont des substances appelées morphogènes qui, lors du développement de l'embryonun responsable de la répartition de couleurs sur les poils. En effet cette coloration est due à la présence de la mélanine produits par des mélanocytes. Ainsi d'un point de vue mathématique, poil sera noir s'il est coloré ou blanc s'il n'y a pas de pigment.
Pour expliquer ce qu'on serait amené à faire, nous allons commencé par explicité comment ce phénomène Il y a deux substances chimiques :

- **Activateurs (A) :** Ils ont pour rôle de favoriser l'activité des mélanocytes.
- **Inhibiteurs (B) :** Ils ont pour rôles de retarder l'activité des mélanocytes.

Phase I :
Au début, chacune de ces substances serait répartie de manière homogène dans une certaine couche de l'embryon. .
À un moment, on assiste à une perte de l'homogénéité, cela à cause des réactions chimiques et de la diffusion. Les petites irrégularités sont donc amplifiées et les répartitions de concentrations des morphogènes évoluent vers une autre configuration in-homogène mais encore stable.

Phase II :
Les mélanocytes réagissent aux morphogènes, Le seuil de concentration qui déclenche l'activité des mélanocytes est inconnu mais peut être résolu par notre modèle. On assiste ainsi à une instabilité diffusionnelle.
À partir du principe de conservation nous obtenons

1.3 Étude de la morphogénèse :

Avons de commencer à expliquer le travaille effectué nous allons d'abord présenté les hypothèses sur lesquels est basé notre modèle.
Hypothèses :

- Nous ne distingueront pas les différentes couleurs des poils : Un poil sera noir (s'il y a un pigment) ou blanc (sans pigment).
- Nous ne tiendront pas compte de la densité du pigment dans les poils.

On suppose que notre matrice de diffusion est à coefficients constants, ce qui nous donne : $D = \begin{pmatrix} d_u & 0 \\ 0 & d_v \end{pmatrix}$ puis notre équation en 2D :

$$\boxed{\frac{\partial \underline{u}}{\partial t} = \underline{f}(\underline{u}) + D\nabla^2 \underline{u}}$$

$$\Rightarrow$$

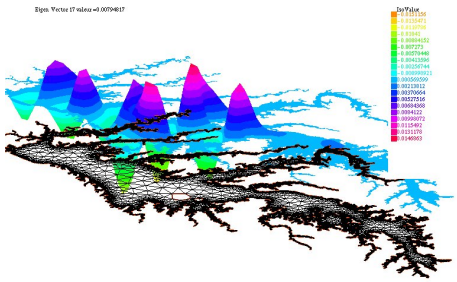
$$\begin{cases} \dfrac{\partial u}{\partial t} = f(u,v) + d_u \nabla^2 u \\ \dfrac{\partial v}{\partial t} = g(u,v) + d_v \nabla^2 v \\ + \text{ conditions initiales} \\ + \text{ conditions au bord} \longrightarrow \frac{\partial u}{\partial n} = \nabla u.n = 0 \text{ et } \frac{\partial v}{\partial n} = \nabla v.n = 0 \end{cases}$$

Pour résoudre ce système nous allons utilisé à la fois l'algorithme d'Euler et la méthode de Gear.

2 Travail effectué :

2.1 Présentation du logiciel utilisé :

FreeFem++ [1] est un logiciel Open Source permettant de résoudre numériquement des équations différentielles par éléments finis1. Il possède son propre langage de script [2], inspiré du C++, pour décrire le type de problème différentiel, les équations aux dérivées partielles et les conditions initiales et aux limites. Il peut ainsi résoudre des problèmes dits multi-physiques, présentant des non-linéarités, en bi- comme en tri-dimensionnel, sur des maillages pouvant aller au million de nœuds (ordinateur de calcul standard) jusqu'à quelques milliards de nœuds (gros système multi-processeurs dédié au calcul).



Ainsi l'une des raison qui nous pousse à utiliser ce logiciel sont : Sa facilité et le fait qu'il soit le plus optimisé pour résoudre les équations différentielles par F.E.M. (comparé aux autres langages open sources utilisés : Python, Fortran et C++). Une deuxième serait qu'il soit open source (à la différence de Matlab et Comsol).

2.2 Présentation de la démarche effectué :

Dans ce t.p., nous allons effectuer la même démarche, simple mais très efficace. Dans la suite de cette sous-section, nous allons mettre les étapes de notre démarche ainsi que les codes utilisées :

- **Définition des bords :**
  On commence par définir nos bords tel cela :

```
border floor(t = 0,5){x = t; y = 0;};
border right(t = 0,2){x = 5; y = t;};
border ceiling(t = 5,0){x = t; y = 2;};
border left(t = 2,0){x = 0; y = t;};

```

Listing 1 – Définition des maillages

- **Paramètres :**
  On définit nos paramètres pour les trois cas tests, le premier sera non commenté tandis que les deux suivants seraient commentés :

```
real gamma = 55.; // // 106.68
real d = 29.3; // // 75.09
real a = .3; // // 0.2
real b = 1.; // // 3
```

```
5       real u0 = a + b;
        real v0 = b / ((a + b)*(a + b));
7       real n = 3.; // 8 // 10
        real m = 2.; // // 0
9       real p = 5.; // 5 // 5
        real q = 2.; // 2 // 2
11      real epsilon = 1.; // 2.2 // 3*1e-3
        // real lambda = (pi*pi) * ( (n*n) / (p*p) + (m*m) / (q*q) );
13      real alpha = 0.985194703; // // 0.9987866914
        real beta = - 0.1714391572; // // - 0.0492457629
15
```

Listing 2 – Définition des paramètres

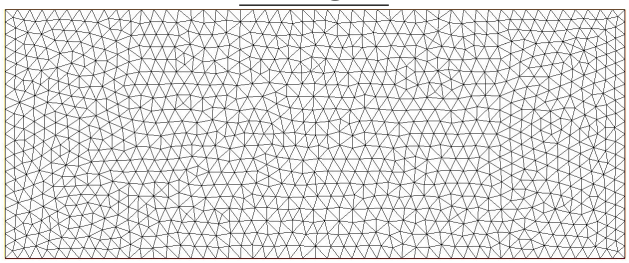
♣ **Construction des maillages :**
Cette étape se fait par le biais du mot clés buildmesh. Cela nous permet d'avoir un maillage de cette forme. Ensuite on use du mot clés adaptmesh qui permet de créer un maillage adapté à notre nombre de sommets maximale choisi (**nbvx**) et en précisons que (à l'aide de **ismetric=1**)

```
    mesh Th = buildmesh(floor(l1) + right(l2) +
2   ceiling(l1) + left(l2));
    Th = adaptmesh(Th, 1./10, IsMetric=1, nbvx=10000);
4   Th = adaptmesh(Th, 1./10, IsMetric=1, nbvx=10000);
```

Listing 3 – Définition des Création de nos maillages :

On obtient des maillages de cette forme :

Maillages :



Résumé de ce qu'on a fait :

```
 --  mesh:  Nb of Triangles =   2194, Nb of Vertices 1168
 number of required edges : 0
 -- adaptmesh Regulary:  Nb triangles 2202 , h  min 0.0659692 , h max 0.145071
    area =  10 , M area = 1000 , M area/( |Khat| nt) 1.04877
    infiny-regulaty:  min 0.516927  max 1.46106
    anisomax  2.47691, beta max = 1.26624 min  0.771199
 --  mesh:  Nb of Triangles =   2202, Nb of Vertices 1172
 number of required edges : 0
 -- adaptmesh Regulary:  Nb triangles 2196 , h  min 0.0677877 , h max 0.146127
    area =  10 , M area = 1000 , M area/( |Khat| nt) 1.05164
    infiny-regulaty:  min 0.527045  max 1.47364
    anisomax  2.35067, beta max = 1.24937 min  0.766768
 --  mesh:  Nb of Triangles =   2196, Nb of Vertices 1169
times: compile 0.02486s, execution 0.081416s,  mpirank:0
 CodeAlloc : nb ptr  2620,  size :339672 mpirank: 0
Ok: Normal End
```

On peut conclure cette étape par dire que nous ne sommes pas trop intéressé à tous les paramètres de buildmesh comme les h_{min} et les h_{max} mais à notre niveau cela ne dérangera car ils ont été effectué automatiquement.

♣ Solution initiale :

```
u = u0 + epsilon * alpha * cos( * pi * ) * cos(m * pi * y / q); // x, y in [0, 1]
v = v0 + epsilon * beta * cos(n * pi * x / p) * cos(m * pi * y / q);
du = 0.;
dv = 0.;
```

Listing 4 – Solution initiale :

♣ Construction du problème variationnel :

✓ Définition des espaces :

```
fespace Uh(Th, P1);
Uh u, v, uu, vv, uvisu, uvisu2 ;
Uh du, dv;
```

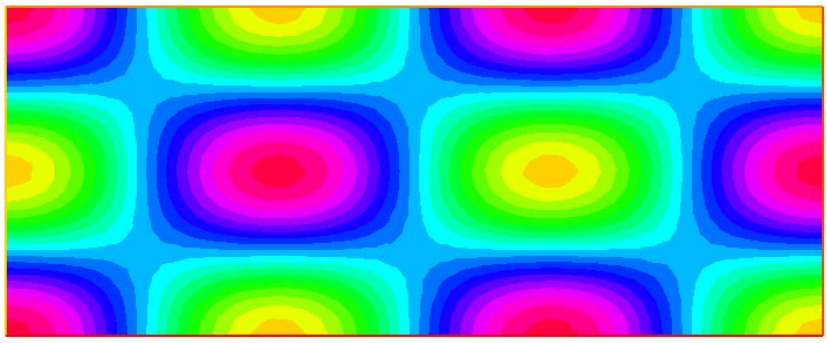
Listing 5 – Définition de nos espaces :

Dans un premier temps nous allons utiliser des éléments de type P_1 ensuite nous allons utiliser éléments de type P_{1nc} puis P_2

- Polynôme de degré 1 P_1 : élément linéaire c'est à dire contiennent deux nœuds par arête.
- Polynôme de degré 2 P_2 : élément parabolique c'est à dire contiennent trois nœuds par arête.

Ainsi, à partir de tout ce qu'on a fait auparavant nous allons sortir la solution initiale :

Solution initiale :



Interprétation : Comme nous avons explicité auparavant durant notre introduction, durant la phase initiale, les concentrations des activateurs et des inhibiteurs sont évoluent dans une phase homogène, c'est ce qu'on peut remarquer dans notre graphique.

✓ Discrétisation en temps et en espace :

On peut intégrer numériquement les systèmes différentiels par des méthodes classiques, mais dans notre cas on utilisera la méthode de gear à deux pas qui semble très stable pour les équations raides.
L'algorithme de cette méthode s'écrit de cette forme :

$$\begin{cases} u_0 \\ u_1(calculerparnewton) \\ u_{n+1} = \frac{4u_n}{3} - \frac{u_{n-1}}{3} + \frac{2hf(u_{n+1},v_{n+1})}{3} \end{cases}$$

Mais ici, l'approximation en temps ne peut pas être indépendante de l'approximation en espace, ce qui nous oblige à mobiliser cette méthode qui dont la formulation variationnel s'écrit sous la forme :

```
1
uold = u0 + epsilon * alpha * cos(n*pi*x/p)* cos(m*pi*y/q) ;
3 vold = v0 + epsilon * beta * cos(n*pi*x/p)* cos(m*pi*y/q) ;

5 uprecprec=uold;
vprecprec=vold;
7 uprec=uold;
vprec=vold;
9 deltau=0.0;
deltav=0.0;
11
problem GEAR(deltau,deltav, uu, vv) =
13 int2d(Th)((-2.0*dt/3.0)*(gamma*((-1.0+2.0*
uprec*vprec)*deltau+uprec*uprec*deltav)*uu
15 + gamma*(-2.0*uprec*vprec*deltau-uprec*uprec*deltav)*vv
-(dx(deltau)*dx(uu)+dy(deltau)*dy(uu))-d*(dx(
17 deltav)*dx(vv)+dy(deltav)*dy(vv)))+deltau*uu+deltav*vv)
+ int2d(Th)((-2.0*dt/3.0)*(gamma*(a-uold+uold*uold*
19 vold)*uu+gamma*(b-uold*uold*vold)*vv
-(dx(uold)*dx(uu)+dy(uold)*dy(uu))-d*(dx(vold)*
21 dx(vv)+dy(vold)*dy(vv)))
+uold*uu+vold*vv-4.0/3.0*uprec*uu-4.0/3.0*vprec*
23 vv+1.0/3.0*uprecprec*uu+1.0/3.0*vprecprec*vv);
```

Listing 6 – Formulation variationnelle :

♣ **Calcul des γ_k :**
Pour cela fera une boucle sur le temps en utilisant cette formule :

$$\begin{cases} \gamma_0 = 57 \\ \gamma_1 = \gamma_{max} = 700 \\ \gamma_j = \gamma_0 + (\gamma_1 - \gamma_0) * (1 - \exp(-x_i j \Delta t)) \end{cases}$$

```
1
for(int j=0;j<371;j++){
3     deltau=0.0;
      deltav=0.0;
5     gamma=gamma0+(gamma1-gamma0)*(1-exp(-xi*j*dt));

7     cout<<" gamma = "<<gamma<<" pour j = "<<j<<endl;
```

Listing 7 – Calcul des gammas :

♣ Enfin, nous allons utiliser l'algorithme de Newton combiné à la méthode de Gear :

Algorithm 1 Algorithme de Newton

$\mathbf{x_0}$ donné , ϵ 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

Ainsi, ici nous calculons les coéfficients u_1 et v_1 par la méthode de Newton puis $(u_2, u_3, ..., u_n)$ et $(v_2, v_3, ..., v_n)$

```
1
for (int i=0; i<5 ; i++){
3     uold = uold + deltau ;
      vold = vold + deltav ;
5     GEAR;
}
7
```

Listing 8 – Itération de Newton :

♣ **Résultat :**
Pour afficher notre résultat on utilise le mot clés plot puis on peut rajouter le titre, enregistré l'image ...

3 Résultats obtenus :

3.1 Présentation des différents cas tests ;

Durant la section précédente nous avons défini la démarche qu'on suivi durant ce tp, maintenant nous allons montrer les résultats qu'on a obtenus =. Mais d'abord nous allons utiliser le tableau ci-dessous à dessein d'expliciter les différents jeux de données qu'on va mobiliser.

Notation	**Données 1 :**	**Données 2 :**	**Données 3 :**
x_i	1.7	1.7	1.7
p	5	5	5
q	2	2	2
n	3	8	10
m	2	4	0
a	0.3	0.2	0.2
b	1.0	3.094	3
d	29.3	94	75.09
u_0	$a+b$	$a+b$	$a+b$
v_0	$\frac{b}{(a+b)^2}$	$\frac{b}{(a+b)^2}$	$\frac{b}{(a+b)^2}$
α	0.9851947093	0.9991631501	0.9987866914
β	-0.17144391572	- 0.04090231744	- 0.04090231744
ε	1.0	2.2	3*1e-3
Δt	0.02	0.02	0.02
γ	57	174	106.68

De plus, afin de raccourcir ce rapport, je vais utiliser un seul élément fini selon le cas, pour cela on choisi :

- ◇ On use d'un élément P_1 pour le premier cas test avec 10000 sommets.
- ◇ On mobilise un élément P_{1nc} pour le second cas test avec 10000 sommets.
- ◇ On utilise un élément P_2 pour le troisième cas test avec 28000 sommets.

Voici les données utilisées pour les cas tests [1] :

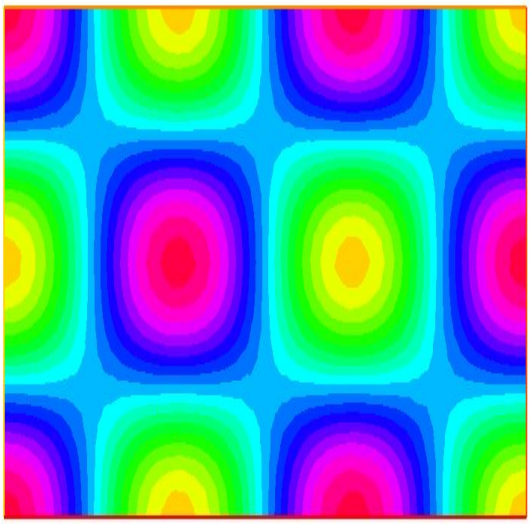
Notation	Cas1	Cas2	Cas3
γ_0	57	57	57
γ_1	700	700	700
p	1	1	1
q	1	1	1
n	1	7	7
m	1	0	1
a	0.1	0.1	0.1
b	0.9	0.9	0.9
d	9.5	0.5	9.5
α	0.1	0.1	0.1
β	0.1	0.1	0.1

1. Par soucis de non répétition nous mettrons que les données non rebandentes.

3.2 Résultats obtenus avec γ constant :

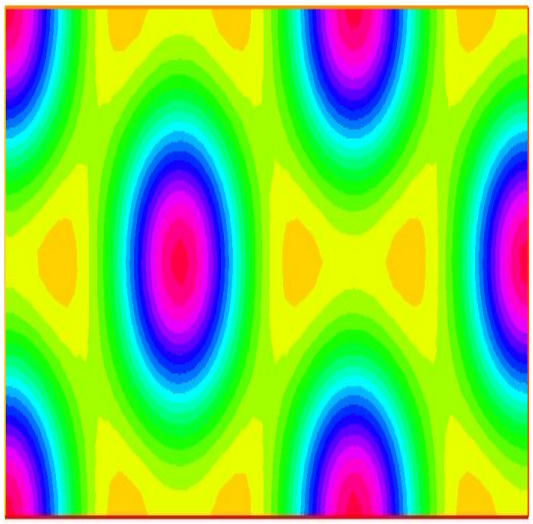
3.2.1 Données 1 :

Solution initiale



Interprétation : Comme nous avons explicité auparavant durant notre introduction, durant la phase initiale, les concentrations des activateurs et des inhibiteurs sont évoluent dans une phase homogène, c'est ce qu'on peut remarquer dans notre graphique.

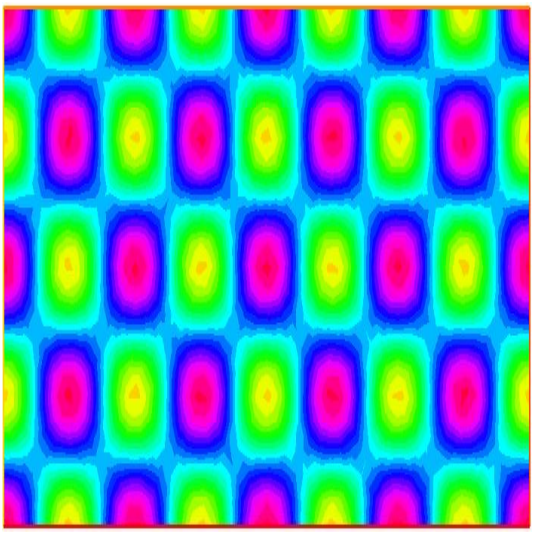
Solution Finale :



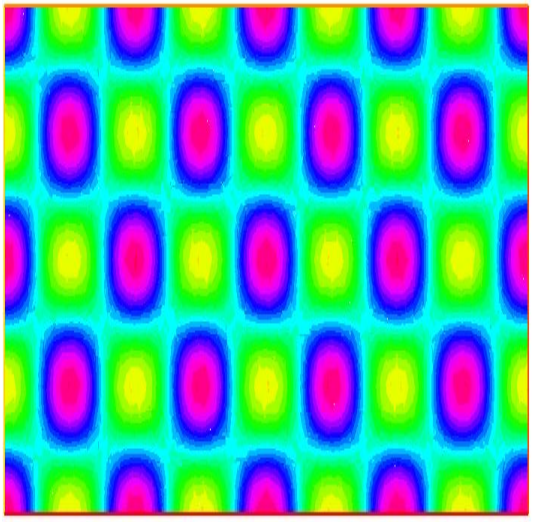
Interprétation : Ici on modélise la seconde phase de la morphogénèse, celle de la perte de l'instabilité diffusionnelle, on peut tout d'abord remarquer un grand changement dans le graphique dus au fait que les les mélanocytes réagissent aux morphogènes. On peut aussi voir le changement des formes des motifs.

3.2.2 Second jeux de données :

Solution initiale



Solution Finale :

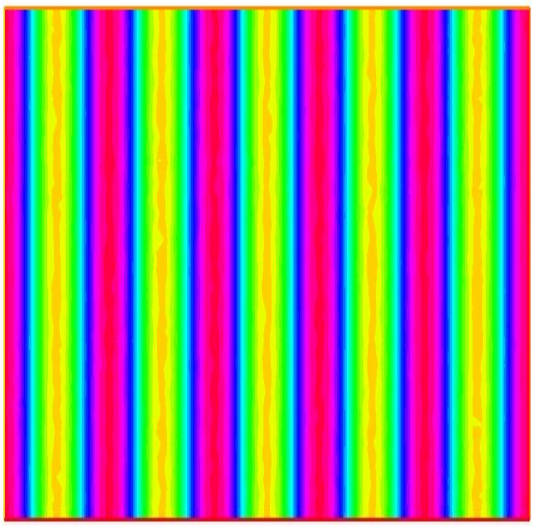


Interprétation : Ici, on est présent face à une situation non similaire au premier jeux de données puisqu'on peut remarquer que notre modèle ne semble pas avoir atteint la phase de l'instabilité diffusionnelle

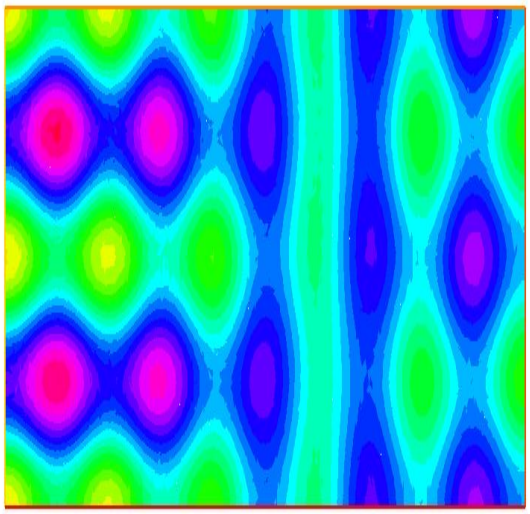
ce qui fait que la solution finale et initiale semblent être assez similaires.

3.2.3 Troisième jeux de données :

Solution initiale



Solution Finale :



Les endroits plus foncés représentent les concentrations de morphogènes favorisant la production de pigments (présence du morphogène u). On remarque que dans la partie rectangulaire du domaine, représentant en deux dimensions le corps, les pigments forment des taches.

3.3 Cas test 1 :

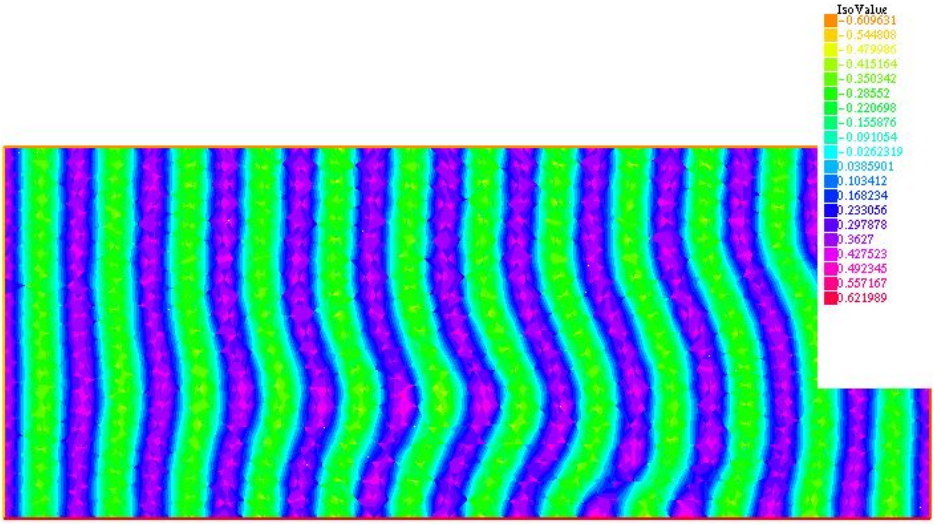
3.3.1 Résultat :

3.3.2 Interprétation :

lklk

3.4 Cas test 2 :

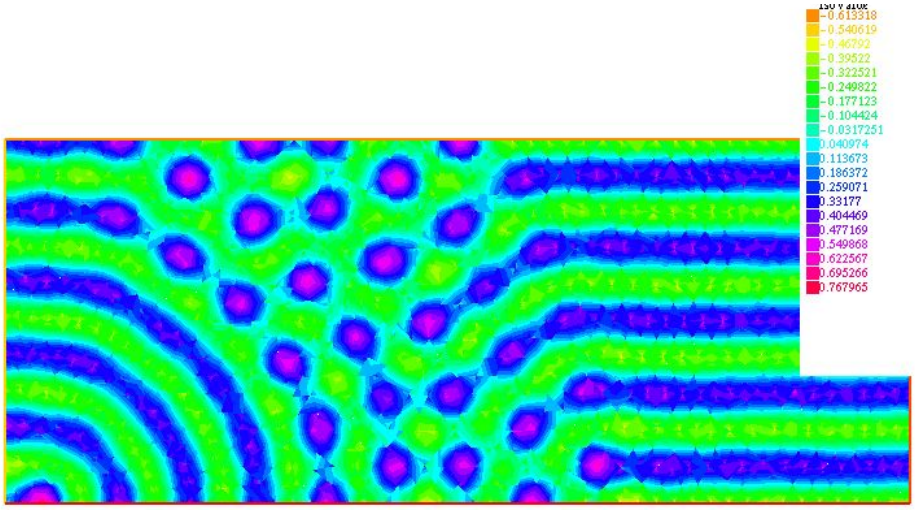
3.4.1 Résultat :

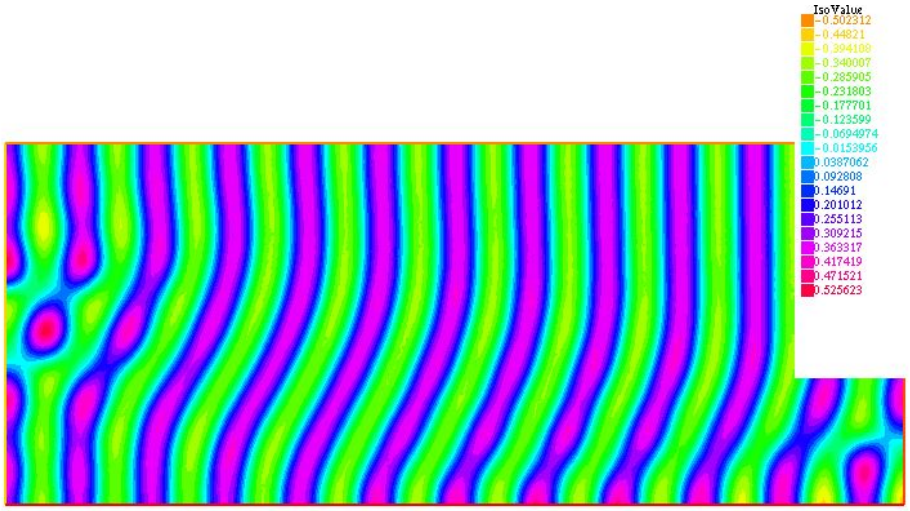


3.4.2 Interprétation :

3.5 Cas test 3 :

3.5.1 Résultat :





3.5.2 Interprétation :

4 Conclusion et bibliographie :

À dessein de conclure ce projet je commencerais tout d'abord par dire que bien qu'au début l'interprétation des résultats m'a semblé une tache difficile et contraignante, cela m'a permis d'effectuer cette analogie cours/réalité sur un domaine dans lequel j'ai déjà effectué un stage.
Ensuite, je tient à parler du fait que bien que les statistiques et le machine learning assouvissent mes prurits et mes envies d'apprendre, des matières comme la modélisation et l'optimisation/analyse numérique me poussent à m'orienter vers un domaine plus déterministe.
Enfin, je parlerai des apports de ce projet :

- Tout d'abord, ce module biologie m'a permis d'agrandir mes connaissances dans un domaines qui me fascine :"**Les bio-mathématiques**" et l'élaboration de ce rapport m'a poussé à lire plusieurs documents dont certains m'ont été confiés durant mon stage qui a porté sur "**Estimation des paramètres de diffusion d'une population dans un paysage hétérogène**" et m'a permis de faire une auto remise en question.
- Ce Tp m'as permis de manipuler le logiciel Freefem++ qui peut être une solution redoutables quand il s'agit de résoudre des E.D.P. surtout pour quelqu'un qui n'est pas très chevronné dans la programmation. De plus, ce langage/logiciel peut être combiné à Python où c++ (via des librairies tels FREEFEM++_MSH_IO), permettant de réduire considérablement la charge de travail.

Références

[1] Page Freefem wiki.

[2] F. Hecht. New development in freefem++. *J. Numer. Math.*, 20(3-4) :251–265, 2012.

[3] Daniel Le Roux. Module Biologie. pages 1–6. Villeurbanne.