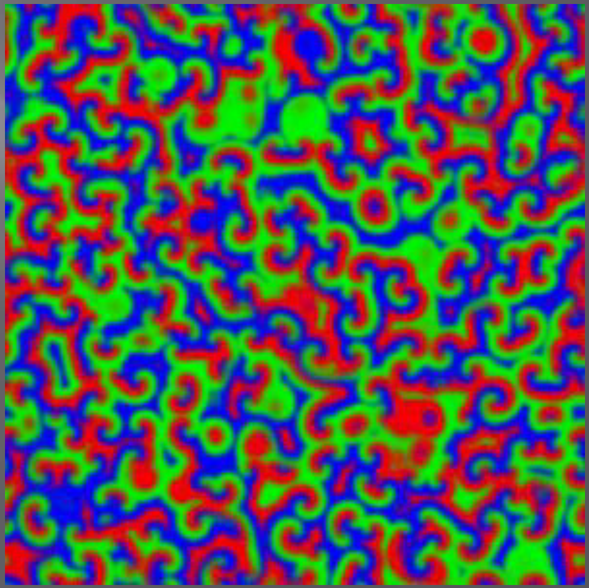
Thème 2018-2019 : Transport

Scei : 7084

ETUDE ET MODÉLISATION DU TRANSPORT DANS UNE RÉACTION OSCILLANTE.



Plan:

I-Réaction oscillante:

- -Caractéristiques.
- -Exemples.

II-La réaction BZ:

1-Equations cinétiques.

2-Système différentiel et modèle retenue.

3-Résolution numérique et simulation.

4-Etude de stabilité.

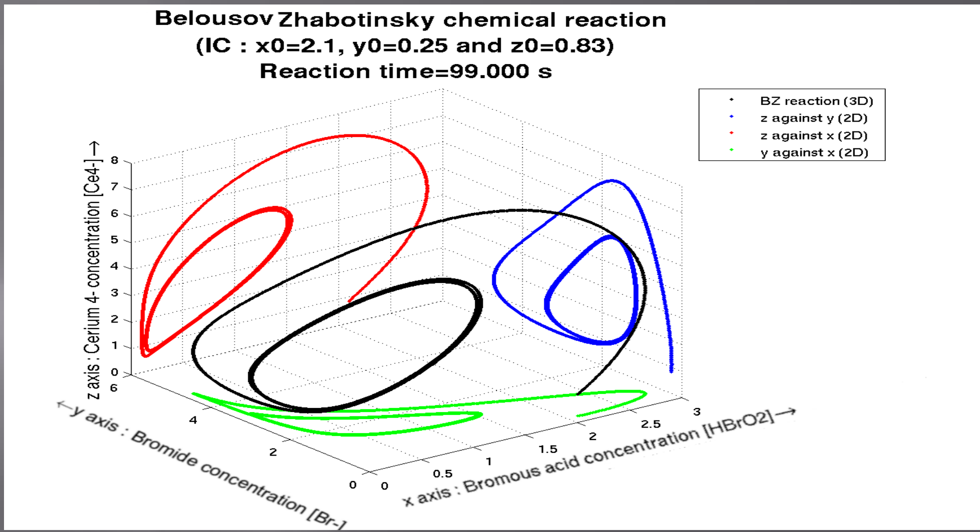
III-Modèle de Lotka Volterra:

IV-Conclusion:

I-Réaction oscillante

Caractéristiques:
-Variation alternative des concentrations.
-système hors équilibre.
-une phase d'oxydation et une phase de réduction s'alternent.

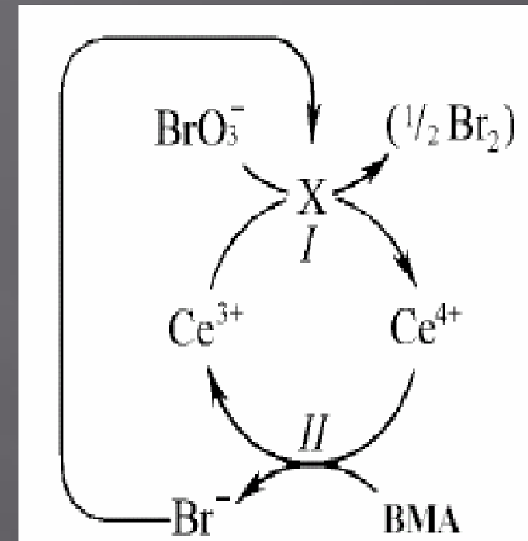
Exemples:
-La réaction BZ
- la réaction de Briggs-Rauscher.
-- la réaction de Bray-Liebhafsky



II-La Réaction de Belousov-Zhabotinsky

1-Mécanisme FKN complet

$$Y \longrightarrow X + P$$
$$X + Y \longrightarrow 2P$$
$$X \longrightarrow 2X + 2Z$$
$$2X \longrightarrow P$$
$$Y + P \longrightarrow W$$
$$W \longrightarrow Y + P$$
$$W + MA \longrightarrow Y$$
$$P + MA \longrightarrow BrMA$$
$$Z + BrMA \longrightarrow Y$$
$$Z + MA \longrightarrow O$$
$$W \longrightarrow U$$
$$U \longrightarrow W$$



composé	Br_2(aq)	Br_2(octane)	HBrO	$HBrO_2$	Br^-	Fe^{3+}
notation	W	U	P	X	Y	Z

Ces équations conduisent à un système différentiel très compliqué .

Mécanisme FKN simplifié:

$$Y \rightarrow X$$
$$X + Y \rightarrow 0$$
$$X \rightarrow 2X + 2Z$$
$$2X \rightarrow 0$$
$$Z \rightarrow hY$$
$$X \rightarrow S$$
$$S \rightarrow X$$
$$Z \rightarrow V$$
$$V \rightarrow U$$
$$U \rightarrow V$$
$$V \rightarrow Z$$
$$U \rightarrow S$$

Modèle simplifié qui permet de décrire la réaction BZ par un nombre réduit d'équations différentielles

notation	composé
S	Br_2O_4
V	Br_2

h: coefficient stoechiométrique

Les équation cinétiques:

$$\frac{d[x]}{dt} = k_1[y] - k_2[x][y] + k_3[x] - k_4[x]^2 - k_6[x] + k_7[s]$$

$$\frac{d[y]}{dt} = -k_1[y] - k_2[x][y] + k_5 h[z]$$

$$\frac{d[z]}{dt} = 2k_3[x] - k_5[z] - k_8[z] + k_{11}[v]$$

$$\frac{d[s]}{dt} = k_6[x] - k_7[s] + k_{12}[u]$$

$$\frac{d[u]}{dt} = k_9[v] - k_{10}[u] - k_{12}[u]$$

$$\frac{d[v]}{dt} = k_8[z] - k_9[v] + k_{10}[u] - k_{11}[v]$$

Les équations cinétiques

$$\frac{d[x]}{dt} = k_1[y] - k_2[x][y] + k_3[x] - k_4[x]^2 - k_6[x] + k_7[s]$$

$$\frac{d[y]}{dt} = -k_1[y] - k_2[x][y] + k_5 h[z]$$

$$\frac{d[z]}{dt} = 2k_3[x] - k_5[z] - k_8[z] + k_{11}[v]$$

$$\frac{d[s]}{dt} = k_6[x] - k_7[s] + k_{12}[u]$$

$$\frac{d[u]}{dt} = k_9[v] - k_{10}[u] - k_{12}[u]$$

$$\frac{d[v]}{dt} = k_8[z] - k_9[v] + k_{10}[u] - k_{11}[v]$$

$\frac{d\xi}{dt} = \frac{1}{\nu_i}\frac{dn_i}{dt}$; $v = \frac{1}{\nu_i}\frac{d(\frac{n_i}{V})}{dt}$. Soit en pratique :

$\forall$ A_i : $\boxed{v = \frac{1}{\nu_i}\frac{d[A_i]}{dt}}$.

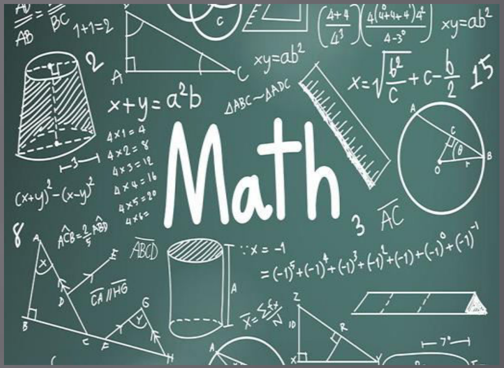
$$[x] = \frac{k_3}{2k_4} \qquad [y] = \frac{k_3 y}{k_2} \qquad [z] = \frac{k_3{}^2 z}{k_4 k_5}$$

$$[s] = \frac{k_3{}^2 s}{2k_4 k_7} \qquad [v] = \frac{k_3{}^2 v}{k_4 k_{11}} \qquad [u] = \frac{k_3{}^2 u}{2k_4 k_{10}}$$

$$t = \frac{\tau}{k_5}$$

Introduction des variables adimensionnés

2-Système différentiel et modèle retenu.



$$\varepsilon \frac{dx}{d\tau} = qy - yx + x - x^2 - \beta x + s$$

$$\varepsilon' \frac{dy}{d\tau} = -qy - yx = fz = 0$$

$$\frac{dz}{d\tau} = x - z - \alpha' z + v$$

$$\varepsilon_2 \frac{ds}{d\tau} = \beta x - s + \chi u$$

$$\varepsilon_3 \frac{du}{d\tau} = Kv - \frac{u}{2} - \frac{\chi u}{2}$$

$$\varepsilon_4 \frac{dv}{d\tau} = \alpha' z - Kv + \frac{u}{2} - v = 0$$

$$f = 2h \qquad \beta = \frac{k_6}{k_3} \qquad \varepsilon = \frac{k_5}{k_3} \qquad \chi = \frac{k_{12}}{k_{10}}$$

$$K = \frac{k_9}{k_{11}} \qquad \alpha' = \frac{k_8}{k_5} \qquad q = \frac{2k_4 k_1}{k_3 k_2} \qquad \varepsilon_2 = \frac{k_5}{k_6}$$

$$\gamma = \frac{1}{2(1+K)} \qquad \alpha = \frac{\alpha' K}{1+K} \qquad \varepsilon_3 = \frac{k_5}{2k_{10}} \prec\prec 1$$

$$\varepsilon' = \frac{2k_4 k_5}{k_3 k_2} \prec\prec 1 \qquad \varepsilon_4 = \frac{k_5}{k_{11}} \prec\prec 1$$

En tenant compte
des conditions:

$$y = \frac{fz}{q+x} \qquad v = \frac{\alpha' z + u/2}{1+K} \qquad \varepsilon' \prec\prec 1 \qquad \varepsilon_4 \prec\prec 1$$

Et en ajoutant les termes de diffusion ,
on obtient le système suivant:

$$\frac{\partial x}{\partial \tau} = \frac{1}{\varepsilon}\left(\frac{fz(q-x)}{q+x} + x - x^2 - \beta x + s\right) + \frac{D_x}{D_u}\Delta x$$

$$\frac{\partial z}{\partial \tau} = x - z - \alpha z + \gamma u + \frac{D_z}{D_u}\Delta z$$

$$\frac{\partial s}{\partial \tau} = \frac{1}{\varepsilon_2}\left(\beta x - s + \chi u\right) + \frac{D_s}{D_u}\Delta s$$

$$\frac{\partial u}{\partial \tau} = \frac{1}{\varepsilon_3}\left(\alpha z - \left(\gamma + \frac{\chi}{2}\right)u\right) + \frac{D_u}{D_u}\Delta u$$

En tenant compte
des conditions:

$$y = \frac{fz}{q+x} \qquad v = \frac{\alpha' z + u/2}{1+K} \qquad \varepsilon' \prec\prec 1 \qquad \varepsilon_4 \prec\prec 1$$

Et en ajoutant les termes de diffusion ,
on obtient le système suivant:

$$\frac{\partial x}{\partial \tau} = \frac{1}{\varepsilon}\left(\frac{fz(q-x)}{q+x} + x - x^2 - \beta x + s\right) + \frac{D_x}{D_u}\Delta x$$

$$\frac{\partial z}{\partial \tau} = x - z - \alpha z + \gamma u + \frac{D_z}{D_u}\Delta z$$

$$\frac{\partial s}{\partial \tau} = \frac{1}{\varepsilon_2}(\beta x - s + \chi u) + \frac{D_s}{D_u}\Delta s$$

$$\frac{\partial u}{\partial \tau} = \frac{1}{\varepsilon_3}\left(\alpha z - \left(\gamma + \frac{\chi}{2}\right)u\right) + \frac{D_u}{D_u}\Delta u$$

La dernière équation peut se simplifier pour : $\chi \prec\prec \gamma$

$$\frac{\partial u}{\partial t} = \frac{1}{\varepsilon_3}(\alpha z - \gamma u) + \Delta u$$

Δ :est le Laplacien

D :coefficient de diffusion

Dx=Dz=10^{-7}cm²/s
Ds=Du=10^{-5}cm²/s

Résolution numérique:

Discrétisation:

Développement de Taylor d'ordre 1 :

$$\frac{\partial X}{\partial t} = \frac{X_{i,j}^{k+1} - X_{i,j}^{k}}{\Delta t}$$

$$\frac{\partial X^{k}}{\partial x} = \frac{X_{i+1,j}^{k} - X_{i,j}^{k}}{\Delta x}$$

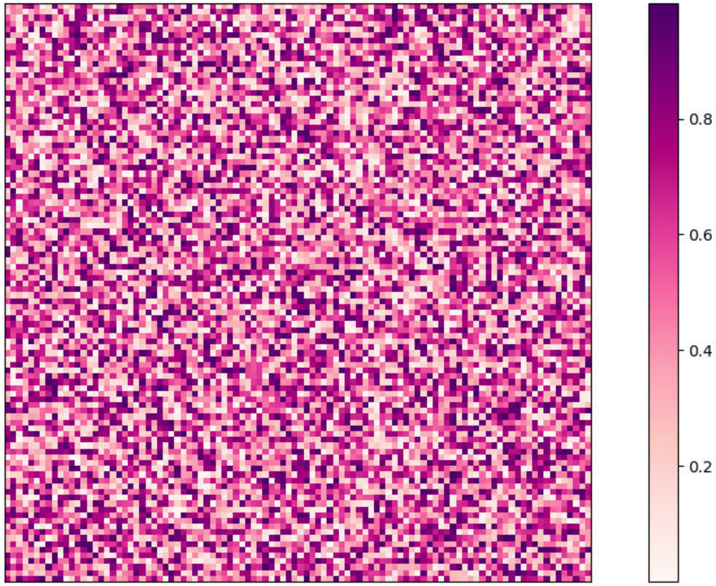
Méthode de Lip Frog:

$$\frac{\partial^2 X^{k}}{\partial x^2} = \frac{X_{i+1,j}^{k} + X_{i-1,j}^{k} - X_{i,j}^{k}}{(\Delta x)^2}$$

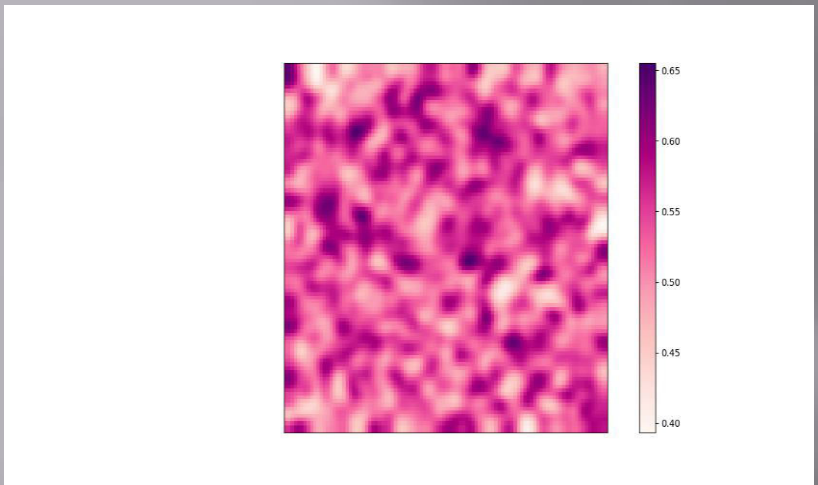
$$\frac{\partial^2 X^{k}}{\partial y^2} = \frac{X_{i,j+1}^{k} + X_{i,j-1}^{k} - X_{i,j}^{k}}{(\Delta y)^2}$$

$$X(x_i, y_j, t_{k+1}) = X(x_i, y_j, t_k) + \Delta t \frac{\partial X(x_i, y_j, t_k)}{\partial t}$$

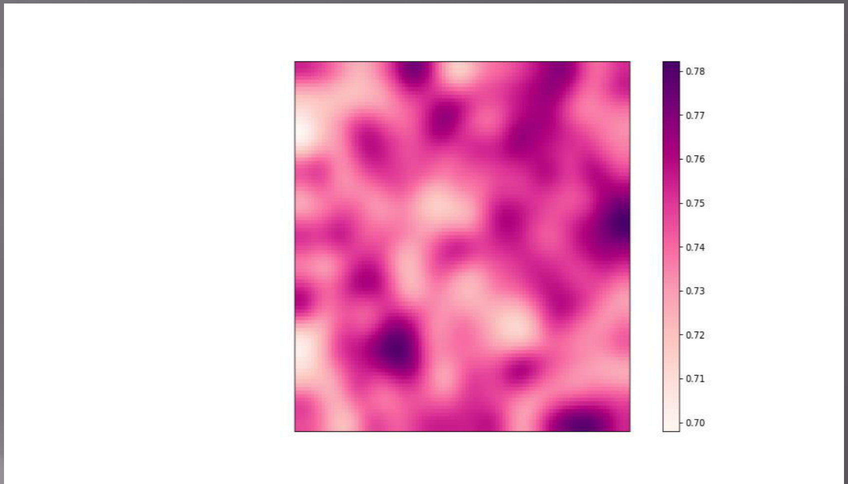
Grille aléatoire initiale représentant X
à l'état initial dans le plan (x,y)



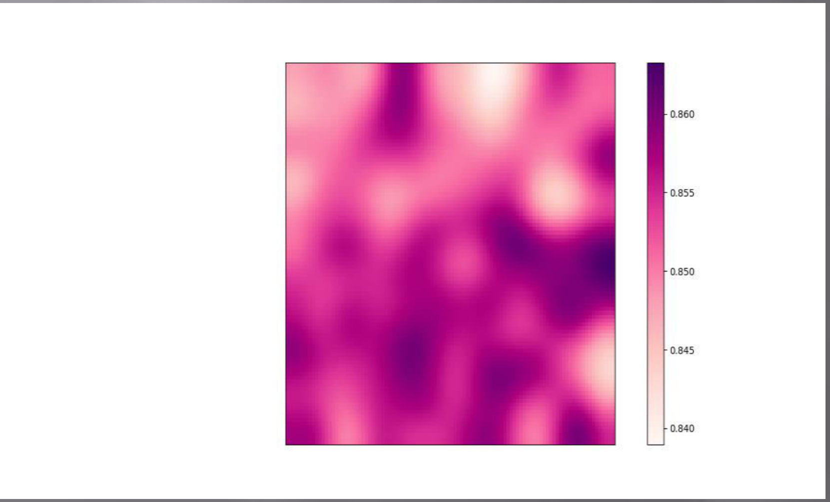
Après 1000 itérations : à t=1000dt



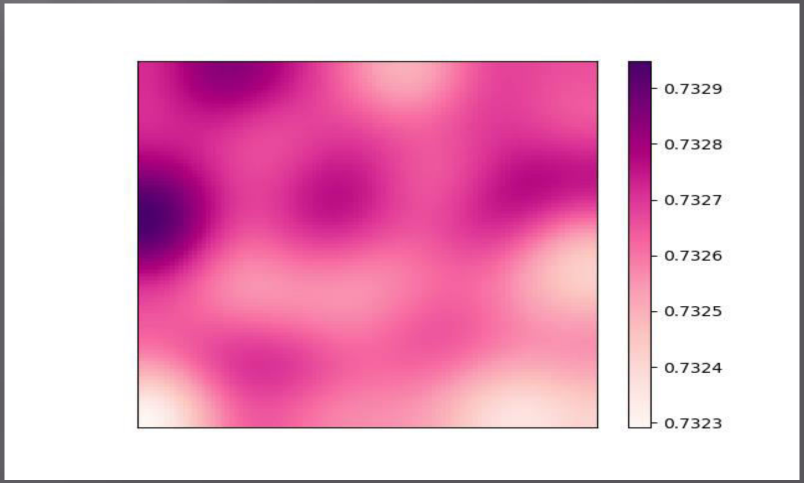
Après 5000 itérations



Après 10000 itérations



Après 30000 itérations



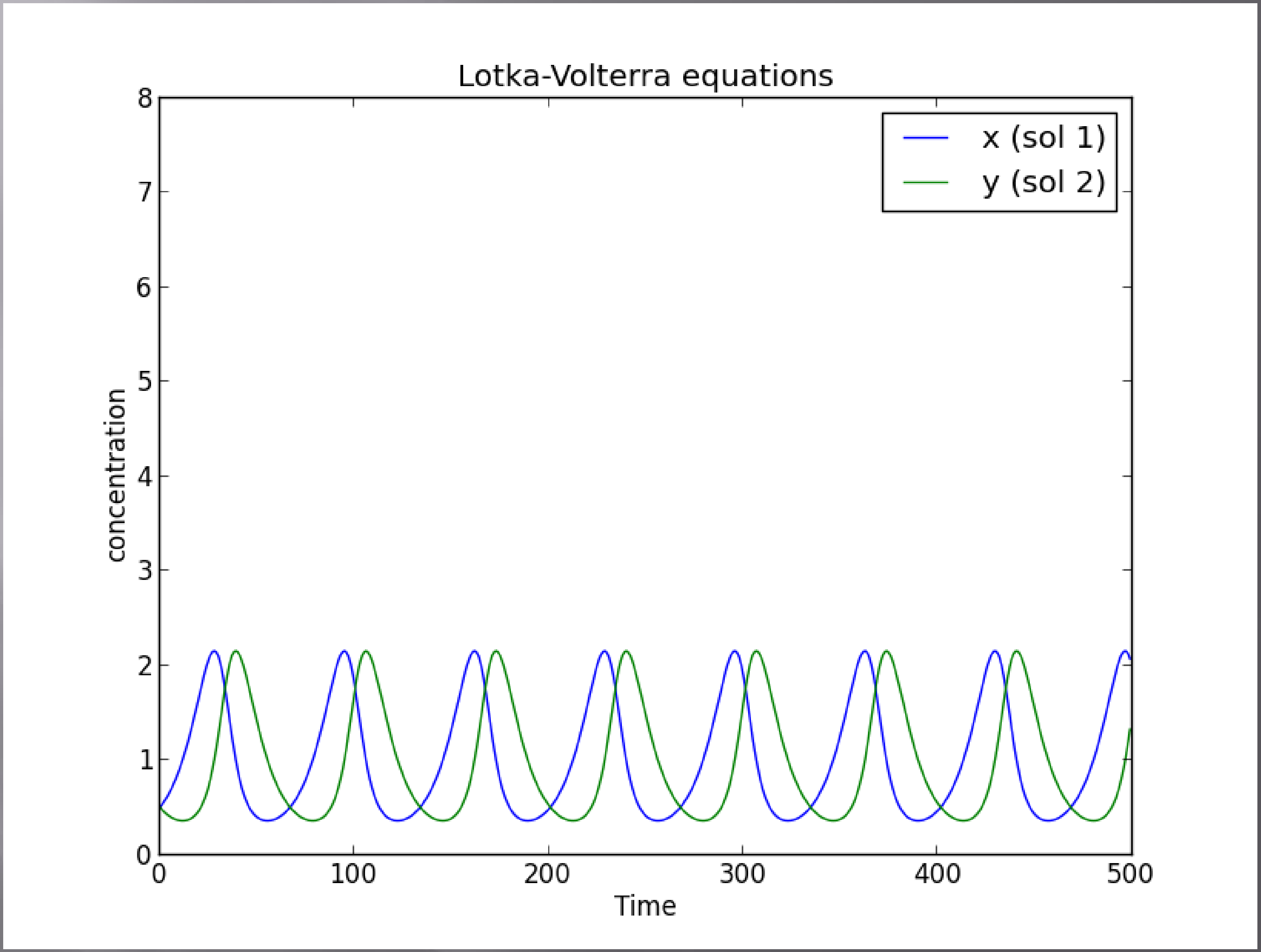
III-Modèle de Lotka Volterra:

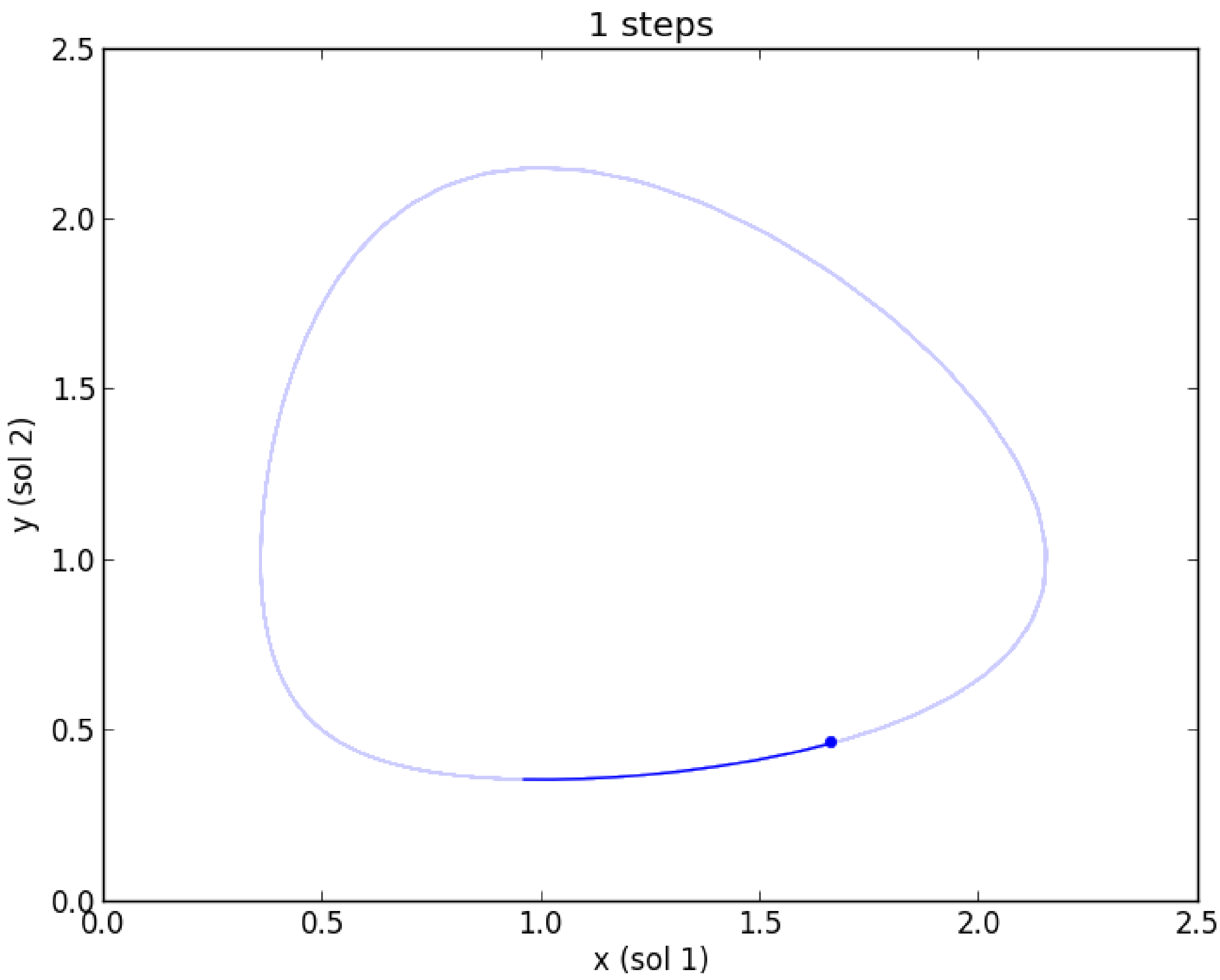
Système différentiel couplé non linéaire .

$$x' = x(\alpha - \beta y)$$
$$y' = -y(\gamma - \sigma x)$$

x : concentration solution 1.
y : concentration solution 2.

$(\alpha, \beta, \gamma, \sigma)$: Des paramètres caractérisant les différentes solutions.


Lotka-Volterra equations
x (sol 1)
y (sol 2)
concentration
Time
0
1
2
3
4
5
6
7
8
0
100
200
300
400
500


1 steps
y (sol 2)
x (sol 1)
0.0
0.5
1.0
1.5
2.0
2.5

IV-Conclusion :

La modélisation de l’équation de Belousov-Zhabotinski a conduit à un système différentiel assez compliqué que l’on a essayé de simplifier pour pouvoir mettre en évidence le caractère Oscillant de cette réaction par une résolution numérique.

Annexe

Code 1:

```
""""""
Crée Janvier 2019
par: Dy El Alem

"""""
Import  numpy  as  np
import  matplotlib.pyplot  as  plt

alpha = 12
beta = 0.1
gama = 0.1
xi = 0
eps=0.5
eps2= 1
eps3= 0.01
q=0.002
f=1.4
Dx=Dz=10**-7
Ds=Du=10**-5
```

```
taille = 100  # taille de la grille
dx = 2./taille  # pas

T = 10.0  # temps total
dt = 0.45 * dx**2/2  # pas de discrétisation
n = int(T/dt)

X = np.random.rand(taille, taille)
Z = np.random.rand(taille, taille)
U = np.random.rand(taille ,taille)
S = np.random.rand(taille, taille)

def  laplacien(Z):
    Zhaut = Z[0:-2,1:-1]
    Zgauche = Z[1:-1,0:-2]
    Zbas = Z[2:,1:-1]
    Zdroite = Z[1:-1,2:]
    Zcentre = Z[1:-1,1:-1]
    return (Zhaut + Zgauche + Zbas + Zdroite - 4 * Zcentre) / dx**2
for i in range(n):
    deltaX = laplacien(X)
    deltaZ = laplacien(Z)
    deltaU = laplacien(U)
    deltaS = laplacien(S)
```

```
# on prend les valeurs à l'intérieur des grilles.
    Xc = X[1:-1,1:-1]
    Zc = Z[1:-1,1:-1]
    Uc = U[1:-1,1:-1]
    Sc = S[1:-1,1:-1]
    # on modifie les variables.
    X[1:-1,1:-1]=Xc+dt*((f*Zc*(q-Xc)/(q+Xc) + (1-beta)*Xc-Xc**2+Sc)/eps+(Dx/Du)*deltaX)
    Z[1:-1,1:-1]=Zc+dt*((Dz/Du)* deltaZ + Xc -(1+alpha)*Zc+gama*Uc)
    S[1:-1,1:-1]=Sc+dt*((beta*Xc+xi*Uc)/eps2+(Ds/Du)*deltaS)
    U[1:-1,1:-1]=Uc+dt*((alpha*Zc-gama*Uc)/eps3+deltaU)

    for  W  in (X,Z,U,S):
        W[0,:] = W[1,:]
        W[-1,:] = W[-2,:]
        W[:,0] = W[:,1]
        W[:,-1] = W[:,-2]

    if  i >=30000:
        break


plt.imshow(X, cmap=plt.cm.RdPu, extent=[-1,1,-1,1]);
plt.colorbar()
plt.xticks([]); plt.yticks
```

Code 2 :

```
from  scipy.integrate  import  odeint
from  numpy  import *
From  matplotlib.pyplot   import *

def  LotkaVolterra(état,t):
        x = état[0]
        y = état[1]
        alpha = 0.1
        beta = 0.1
        sigma = 0.1
        gamma = 0.1
        xd = x*(alpha - beta*y)
        yd = -y*(gamma - sigma*x)
        return [xd,yd]
t = arange(0,500,1)
état0 = [0.5,0.5]
état = odeint(LotkaVolterra,état0,t)
figure()
plot(t,état)
ylim([0,8])
xlabel('Temps')
ylabel('concentration')
legend(('x (sol1)','y  (sol2)'))
title(' équations Lotka-Volterra  ')
Show()
```

animation

```
 figure()
pb, = plot(état[:,0],état[:,1],'b-',alpha=0.2)
xlabel('x (sol1)')
ylabel('y ( sol2)')
 p, = plot(état[0:10,0],état[0:10,1],'b-')
pp, = plot(état[10,0],state[10,1],'b.',markersize=10)
tt = title("%4.2f sec" % 0.00)

step=2
for i in  range(1,shape(état)[0]-10,step):
     p.set_xdata(état[10+i:20+i,0])
     p.set_ydata(état[10+i:20+i,1])
     pp.set_xdata(état[19+i,0])
     pp.set_ydata(état[19+i,1])
     tt.set_text("%d steps" % (i))
     show()
```