Université Claude Bernard Lyon 1 (ucbl)
Villeurbanne
Rhone-Alpes-France

Mathématiques Appliquées

PROJET DE SYSTÈMES DYNAMIQUES

Pour obtenir le diplôme de Master 1 en :
Mathématiques Appliquées : statistique

Présentée et soutenue par :
GAYE Alioune et KALY Bauvary

Étude des schémas numériques

Projet encadré par : PUJO-MENJOUET Laurent

3 mai 2021

Table des matières

1 Un peu de théorie

Problème de Cauchy
Augustin Louis CAUCHY, *au* 19^e siècle, souhaite ne pas séparer la recherche des solutions générales d'une équation différentielle de la recherche des solutions particulière. Dans ce cadre, il posa un problème de système différentiel, connu sous le nom de *Problme de Cauchy*.

Définition 1 *Soit* U *un ouvert de* $\mathbb{R} \times \mathbb{R}^m$ *et une fonction* $f : U \to \mathbb{R}^m$.
Étant donné une équation différentielle du premier ordre sous forme normale

$$x^{'} = f(t, x)$$

pour $(t, x(t)) \in U$ *et un point* $(t_0, x_0) \in U$ *le Problme de Cauchy correspondant est la recherche de solution* x *telles que*

$$x(t_0) = x_0 \, .$$

Notation :
On note le *Problme de Cauchy* de la fonçon suivante :

$$\begin{cases} x^{'} & = f(t, x) \\ x(t_0) & = x_0 \end{cases} \tag{1}$$

2 Étude de la consistance d'ordre au moins p de schéma explicite à un pas

Trouver la solution explicite du *Problme de Cauchy* voir (1) n'étant pas toujours possible, on doit pouvoir trouver des méthodes de résolution numérique qui nous donnes de bonnes approximations numériques.
La méthode générale consiste à découper l'intervalle sur lequel on souhaite résoudre le problème, en découpant l'intervalle en plusieurs pas : ce procédé s'appelle la discrétisation de l'intervalle.
Au lieu d'avoir la solution en toutes les valeurs de l'intervalle, on l'au sur certains points de l'intervalle. cette discrétisation nous donne ensuite les valeurs numériques, par l'intermédiaire d'un schéma défini par une fonction.
On dit qu'un schéma est à un pas si pour trouver la valeur actuelle, on ne s'intéresse qu'à la valeur prise ne le point de discrétisation précédente. On ne s'intéresse icic qu'à des schema à un pas.De plus on considère le pas entre deux point uniforme ; i.e prenant toujours la même valeur.
On supposera toujours que la fonction f définie dans le *Problme de Cauchy* (1) satisfait les conditions du théorème de $Cauchy - Lipschitz$.

Définition 2 *Avec ces précisions, un schéma à un pas est une relation récurrente de la forme!*

$$u_{k+1} = u_k + h * F(t_k, u_k, h) \tag{2}$$

3 Numérique

3.1 Exercice 1

Soit le système suivant :

$$(\mathcal{C}_1) \begin{cases} u'(t) & = f(t, u(t)); \\ u(0) = 1 \end{cases} \tag{3}$$

avec $f : U = [0, 1] \times [0; 1] \to \mathbb{R}$ et $(t, x) \mapsto f(t, x) = -100x + 25$.

1) Solution du du problème 3

2) a) Solution exacte .
f est C^∞, donc il y a existence et unicité de la solution. Et donc par le formule de Duhamel, on trouve la solution ci-dessous :

$$u(t) = \frac{1}{4}(1 + e^{-100t}) \tag{4}$$

b) $\lim_{t\to+\infty} u(t) = \lim_{t\to+\infty} \frac{1}{4}(1 + e^{-100t}) = \frac{1}{4}$

3) Schéma d'Euler

a) Le schéma d'Euler pour le problème $(\mathcal{C}_1)$.
Pour une subdivision $t_1 < t_2 < \cdots < t_n$ à pas constant h, le schéma d'Euler est le suivant :

$$\begin{cases} u_{n+1} & = u_n + h * f(t_n, u_n, h) \\ u_0 & = u(0) = 1 \end{cases} \tag{5}$$

b) Relation de récurrence entre u_{n+1} et u_n.
À partir du schéma d'Euler, on obtient la relation suivante :

$$u_{n+1} = (1 - 100h)u_n + 25h.$$

c) En faisant varier n, on remarque que u_n pourrait avoir la forme suivante :

$$u_n = (1 - 100h)^n u_0 + 25h * \sum_{j=0}^{n-1} (1 - 100h)^j$$

, Il a été montré par récurrence que c'est bien la formule de u_n.

d) Limite du module de $\mid u_n \mid$ pour $h = \frac{1}{25}$

$$\lim_{n\to+\infty} \mid u_n \mid = \lim_{n\to+\infty} \left| \sum_{j=0}^{n} (-3)^j \right|,$$

on a la somme d'une suite géométrique de raison en module supérieur stricte à 1, donc la limite est l'infinie

$$\lim_{n\to+\infty} \mid u_n \mid = +\infty.$$

e) Conclusion :
Le schéma d'Euler explicite ne converge pas.

4) Schéma d'Euler implicite.

a) Avec les mêmes hypothèses que pour le 3)a), le schéma implicite s'écrit :

$$\begin{cases} u_{n+1} & = u_n + h * f(t_{n+1}, u_{n+1}) \\ u_0 & = u(0) = 1 \end{cases} \tag{6}$$

b) Relation de récurrence entre u_{n+1} et u_n.
À partir du schéma d'Euler implicite, on obtient la relation suivante :

$$u_{n+1} = \frac{1}{1 + 100h} u_n + 25h$$

c) En faisant varier n, on remarque que u_n pourrait avoir la forme suivante :

$$u_n = \left(\frac{1}{1+100h}\right)^n u_0 + 25h * \sum_{j=0}^{n-1} \left(\frac{1}{1+100h}\right)^j,$$

Il a été montré par récurrence que c'est bien la formule de u_n.

d) Limite du module de $\mid u_n \mid$ pour $h = \frac{1}{25}$

$$\lim_{n\to+\infty} \mid u_n \mid = \lim_{n\to+\infty} \left| \sum_{j=0}^{n} \left(\frac{1}{5}\right)^j \right|,$$

on a la somme d'une suite géométrique de raison strictement inférieur à 1, donc la limite est finie. On a alors :

$$\lim_{n\to+\infty} \mid u_n \mid = \lim_{n\to+\infty} \frac{1-\left(\frac{1}{5}\right)^n}{1-\frac{1}{5}} = \frac{5}{4}$$

e) Conclusion :
Le schéma converge, mais pas vers la solution.

5) Simulation

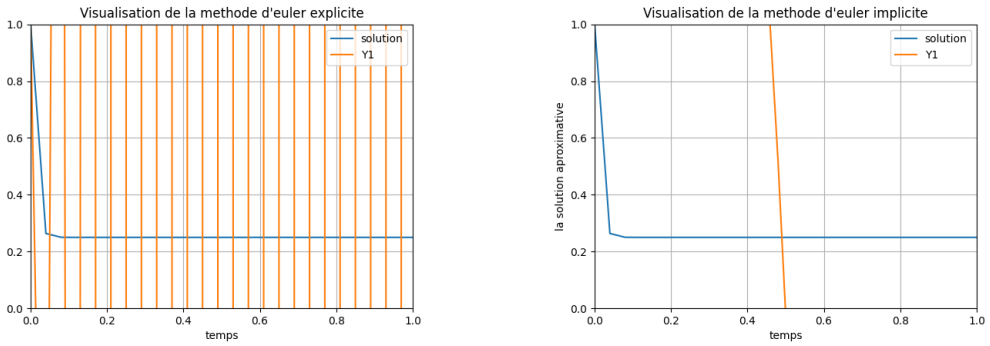
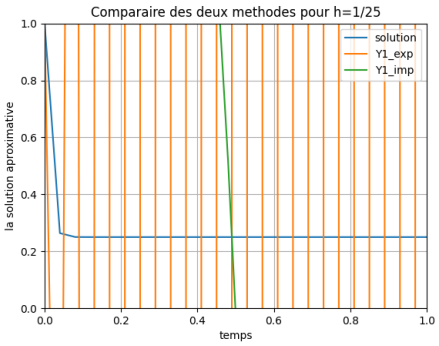


FIGURE 1 – Scémas d'Euler explicite et implicite pour $h = \frac{1}{25}$.



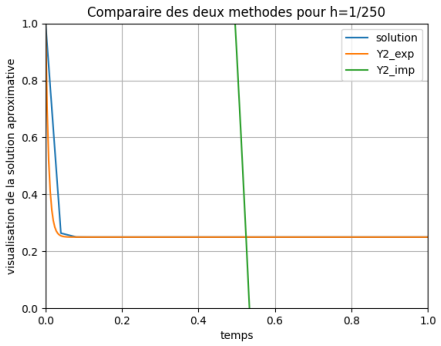


FIGURE 2 – Scémas d'Euler explicite et implicite pour $h = \frac{1}{25}$, et $h = \frac{1}{250}$.

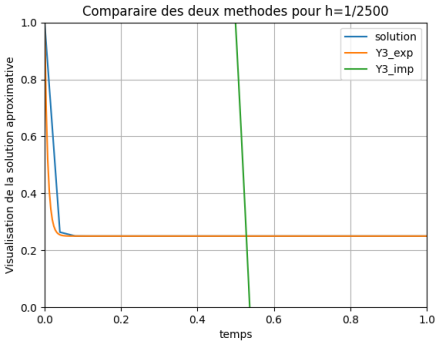


FIGURE 3 – Scémas d'Euler explicite et implicite pour $h = \frac{1}{2500}$.

6) On constate qu'aucun des deux schémas ne converge vers la solution exacte. pour $h = \frac{1}{25}$.
Cependant, on note une convergence de la méthode explicite et non de la méthode implicite lorsque l'on divise h par 10, et 100.
Donc, ici le schéma explicite est plus adapté que le schéma implicite pour une subdivision plus fine.

3.2 Exercice 2

Soit le système suivant :

$$(\mathcal{C}_2) \begin{cases} y'(t) & = f(t, y(t)); \\ y(0) = 1 \end{cases} \tag{7}$$

avec $f : U = [0, 4] \times \mathbb{R} \to \mathbb{R}$ et $(t, x) \mapsto f(t, x) = 3e^{0.5t} - 0.2x$.

1) Solution exacte .
f est C^∞, donc il y a existence et unicité de la solution. Et donc par le formule de Duhamel, on trouve la solution ci-dessous :

$$y(t) = e^{-0.2t}(\frac{23}{7} + \frac{30}{7}e^{0.7t}) \tag{8}$$

2) Convergence : on va utilisé le théorème de Lax

(a) Consistance
Le schéma d'Euler explicite s'écrit :

$$y_{n+1} = y_n + h * \Phi(t_n, y_n, h), \text{ avec } \Phi(t_n, y_n, h) = f(t_n, y_n). \tag{9}$$

On a $\Phi(t, y, 0) = f(t, y)$ donc le schéma est consistant.

(b) Stabilité
Soit $x,\ y\ \in \mathbb{R}$ on veut estimer $||\Phi(t, x, h) - \Phi(t, y, h)||$

$$||\Phi(t, x, h) - \Phi(t, y, h)|| = |f(t, x) - f(t, y)| \tag{10}$$

$$= \frac{1}{5}|x - y|, \qquad \forall x,\ y\ \in \mathbb{R}. \tag{11}$$

Donc le schéma est stable.

On a consistance et stabilité donc le schéma est convergent.

3) Ci-dessous le tableau contenant les solutions exacte, approchée et l'erreur en les points de la discrétisation.

t_i	0	1	2	3	4
$y(t_i)$	[7.571428571428571	9.756063634542489	13.852259416084436	21.010477105757783	33.143749877516505
y_n	1	3.8	7.986163812100385	14.543776535057443	25.080088439060148
e_n	19	51	36	28	24

TABLE 1 – Solutions exacte, approchée et l'erreur en des points de la discrétisation

4) Représentation graphique.

FIGURE 4 – Visualisation des solutions exacte et approchée et l'erreur.
On voit une erreur qui diminue au cours du temps, donc l'on pourrait penser que le schéma pourrait converger

3.3 Exercice 3

dynamic_exercice

May 2, 2021

[24]:
```
% Exercice 4:
```

```
Created file '/home/bauvary.kaly/dydt.m'.
```

[59]:
```
%%file dydt.m
function diffeq=dydt(t,y)
    diffeq=10*exp(-((t-2)^.2)/(2*(0.075)^.2))-0.6*y ;
end
```

```
Created file '/home/bauvary.kaly/dydt.m'.
```

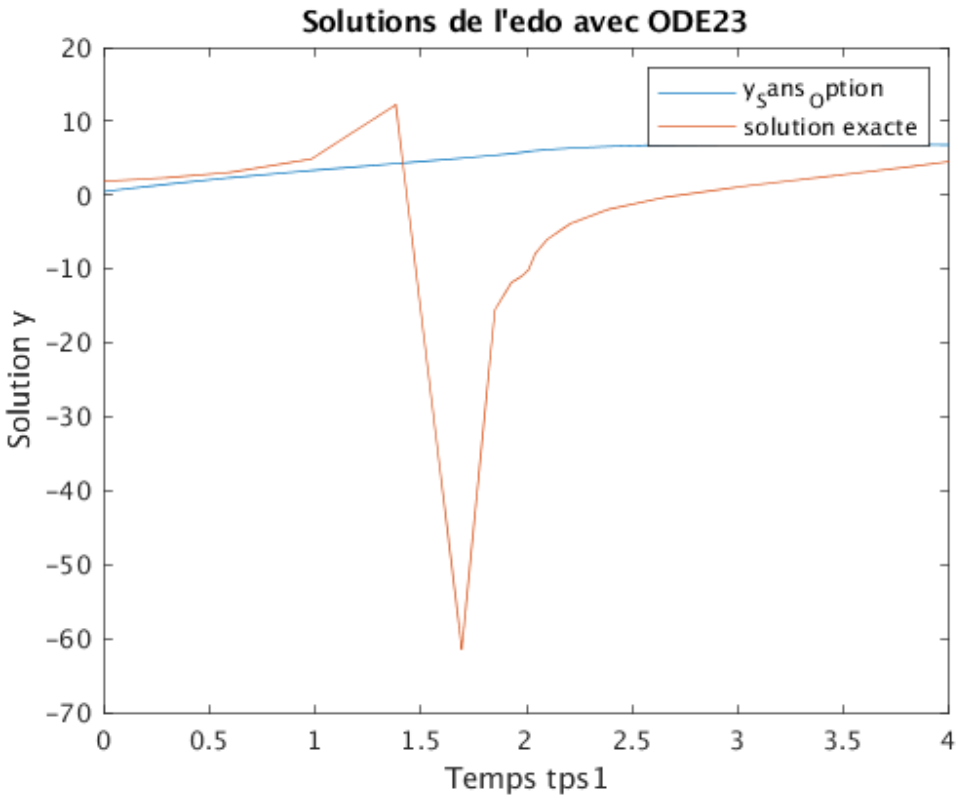
[60]:
```
%%file f.m
function out=f(t)
    out= 0.5*exp(0.6*t) - 10*exp(0.6*t)*exp(-((t-2)^.2)/(2*(0.075)^.2) - 0.6*t)/
 ↪((t-2)/((0.075)^.2) + 0.6);
end
```

```
Created file '/home/bauvary.kaly/f.m'.
```

[61]:
```
y0=0.5;
a=0;
b=4;
tspan=[a,b];
options=odeset('Reltol',1e-4);
[tps1,y_Sans_Option]=ode23(@dydt,tspan,y0);
sol=[];
for i=1:size(tps1)
    sol(i)=f(tps1(i));
end
```

[62]:
```
plot(tps1,y_Sans_Option,tps1,sol);
title("Solutions de l'edo avec ODE23");
xlabel("Temps tps1");
ylabel("Solution y");
legend("y_Sans_Option","solution exacte");
```

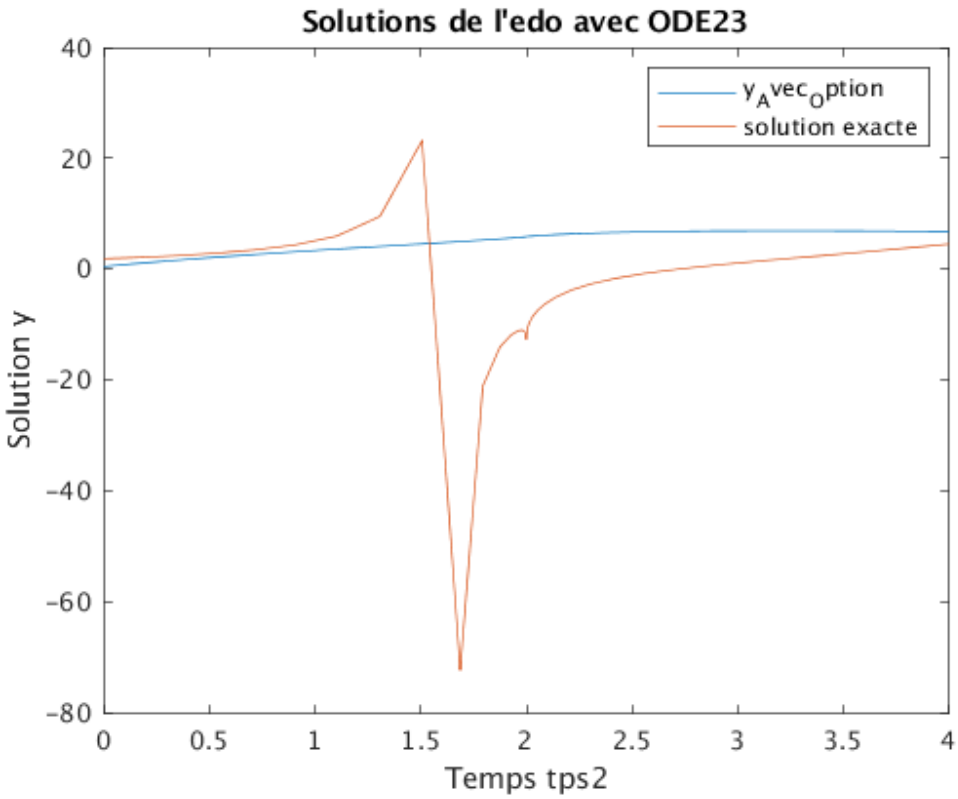
```
Warning: Imaginary parts of complex X and/or Y arguments ignored
```



[63]:
```
[tps2,y_Avec_Option]=ode23(@dydt,tspan,y0,options);
sol2=[];
for i=1:size(tps2)
    sol2(i)=f(tps2(i));
end
```

[64]:
```
plot(tps2,y_Avec_Option,tps2,sol2)
title("Solutions de l'edo avec ODE23");
xlabel("Temps tps2");
ylabel("Solution y");
legend("y_Avec_Option","solution exacte");
```

```
Warning: Imaginary parts of complex X and/or Y arguments ignored
```



[ ]:
```
% Interpretation:
%On constate que l'approximation avec l'option Reltol est meilleur que celle␣
 ↪sans approximation
```

3.4 Exercice 4

Soit le système de Lorentz :

$$(\mathcal{C}_4)\begin{cases} y_1' &= \sigma(y_2 - y_1) \\ y_2' &= ry_2 - y_2 - y_1y_3 \\ y_3' &= y_1y_2 - by_3 \end{cases} \tag{12}$$

1) Schéma et interprétation du système de Lorentz.

2) Recherche d'équilibres.
Les équilibres correspond à l'intersection des isoclines nulles de chaque variable.

$$\begin{cases} y_1' &= 0 \\ y_2' &= 0 \\ y_3' &= 0 \end{cases} \Leftrightarrow \begin{cases} \sigma(y_2 - y_1) &= 0 \\ ry_2 - y_2 - y_1y_3 &= 0 \\ y_1y_2 - by_3 &= 0 \end{cases} \Leftrightarrow \begin{cases} y_2 &= y_1 \\ (r - 1 - y_3)y_1 &= 0 \\ y_1^2 &= by_3 \end{cases} \Leftrightarrow \begin{cases} y_1 &= 0 \\ y_2 &= 0 \end{cases} ou \begin{cases} y_2 &= y_1 \\ r - 1 &= y_3 \\ y_1^2 &= by_3 \end{cases}$$

.
Les deux autres équilibres sont alors donnés par :

$$\begin{cases} y_1^2 &= b(r-1) \\ y_2 &= y_1 \\ y_3 &= r - 1 \end{cases}$$

— Pour $r < 1$ on a un seul équilibre

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

.

— Pour $r > 1$, on a les équilibres suivants :

$$Y_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \; Y_2 = \begin{pmatrix} \sqrt{b(r-1)} \\ \sqrt{b(r-1)} \\ r-1 \end{pmatrix} \text{ et } Y_3 = \begin{pmatrix} -\sqrt{b(r-1)} \\ -\sqrt{b(r-1)} \\ r-1 \end{pmatrix}$$

Les conditions initiales étant toutes strictement positive, donc l'équilibre est le suivant :

$$Y_e = \begin{pmatrix} \sqrt{b(r-1)} \\ \sqrt{b(r-1)} \\ r-1 \end{pmatrix}$$

3) Étude de la stabilité des équilibres.
Pour étudier la stabilité, calculons la matrice jacobienne du système de Lorentz :

$$J = \begin{pmatrix} -\sigma & \sigma & 0 \\ r - y_3 & -1 & -y_1 \\ y_2 & y_1 & -b \end{pmatrix}$$

La stabilité des équilibres dépend du signe des valeurs propres de la jacobienne, qui sont les racines du polynôme :

$$det(\lambda I_3 - J) = \lambda^3 - \lambda^2 tr(J) + \frac{\lambda}{2}[tr^2(J) - tr(J)] - det(J) \tag{13}$$

(a) Stabilité de $Y_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

La jacobienne en Y_1 est :

$$J_1 = \begin{pmatrix} -\sigma & \sigma & 0 \\ r & -1 & 0 \\ 0 & 0 & -b \end{pmatrix}.$$

Par conséquent, on a :

$$tr(J_1) = -(\sigma + 1 + b) \text{ et } det(J_1) = \sigma b(r-1)$$

d'où

$$det(\lambda I_3 - J) = \lambda^3 - (\sigma + 1 + b)\lambda^2 + \frac{\lambda}{2}[\sigma(1-r) - (\sigma + 1 + b)] - \sigma b(r-1) = 0 \qquad (14)$$

ce qui est équivalent à :

$$(\lambda + b)(\lambda^2 + (1+\sigma)\lambda + \sigma(1-r)) = 0.$$

Le terme de droite est un polynôme de degré 2, son discriminant est : $\Delta = (1+\sigma)^2 - 4\sigma(1-r)$
Cependant $\Delta > 0 \Leftrightarrow 1 - r < \frac{(1+\sigma)^2}{4\sigma}$.
Ici $\sigma = 10$ donc le discriminant sera toujours positif si et seulement si $r > \frac{-81}{40}$. Ce qui est toujours le cas, car $r > 0$.
Nous avons donc trois racines réelles :

$$\begin{cases} \lambda_1 = b < 0 \\ \lambda_2 = \frac{-(1+r)-\sqrt{\Delta}}{2} \\ \lambda_3 = \frac{-(1+r)+\sqrt{\Delta}}{2} < 0 \end{cases}$$

On voit que $\lambda_2 < 0$ pour $r < 1$ et $\lambda_2 < 0$ pour $r > 1$.
Par conséquent Y_1 est stable pour $r < 1$ et instable pour $r > 1$.

4) Stabilité de $Y_2 = \begin{pmatrix} \sqrt{b(r-1)} \\ \sqrt{b(r-1)} \\ r-1 \end{pmatrix}$

La jacobienne en Y_1 est :

$$J_2 = \begin{pmatrix} -\sigma & \sigma & 0 \\ 1 & -1 & -\sqrt{b(r-1)} \\ \sqrt{b(r-1)} & \sqrt{b(r-1)} & -b \end{pmatrix}.$$

On a : $tr(J_2) = -(\sigma + 1 + b)$ et $det(J_2) = 2\sigma b(r-1)$
Par conséquent, on a l'équation du polynôme caractéristique suivant :

$$\lambda^3 - (\sigma + 1 + b)\lambda^2 + (br - 3b - 1)\lambda - 2\sigma b(r-1) = 0 \qquad (15)$$

Selon la valeur de r, on peut obtenir comme racines :
— 3 racines réelles toutes négatives
— 1 racine réelle et deux racines complexes conjuguées $\lambda_K = \mu \pm iw$
on cherche s'il existe r_c tels que les équilibres Y_2 et Y_3 devient instables. On s'appuie sur les scénarios présentés dans l'annex (voir A).
Dans la scénario n°1 , une valeur propre réelle change de signe. Cela équivaut à un changement d'une valeur propre λ_K. L'expression du polynôme caratéristique (equation 15) indique que cela n'est possible que si le terme constant $\sigma b(r-1)$ s'annule. Cela signifie que $r = 1$ ce qui est impossible puisque l'équilibre est définit pour $r > 1$. Ce premier scénario n'est donc pas possible.

Dans la scénarion n°2 les deux valeurs propres "traversen" l'axe des ordonnées (bifurcation de Hopf).
C'est-à-dire que $\lambda_K = \pm iw$. En utilisant cette relation dans l'expression de l'équation du polynôme caractéristique, on obtient deux équations suivant :

$$\begin{cases} -w^2(\sigma + 1 + b) + 2b(r-1) & = 0 \\ -iw^3 + iwb(\sigma + r) & = 0 \end{cases}$$

Et donc on trouve $r_c = \frac{\sigma(\sigma+b+3)}{\sigma-b-1} \approx 24.73$. Ainsi pour $r \geq r_c$ la trajectoire est chaotique. Elle tourne autour d'un des deux équilibres (devenue instable lorsque r dépasse le seuil r_c), comme si elle convergeait puis elle bascule vers l'autre équilibre.

5) En se basant sur ce qui a été fait précédemment, on a ci-dessus le diagramme de bifurcation de y_1

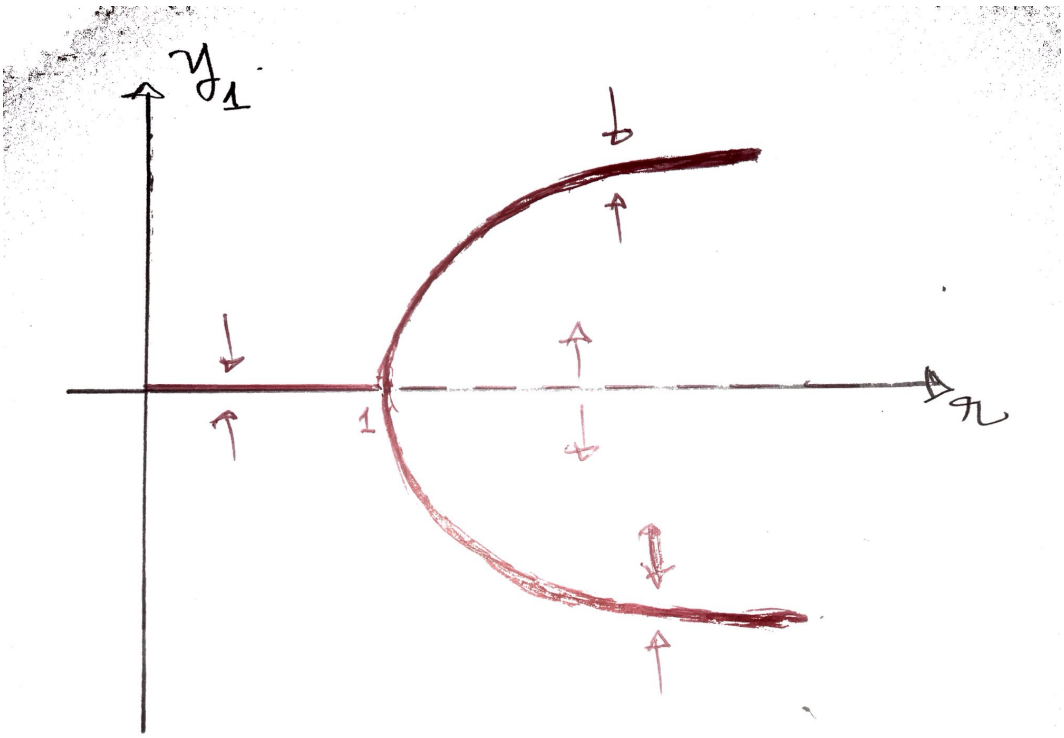


FIGURE 5 – Bifurcation pour y_1.

6) Résolution numérique
   a) Tracé des trajectoire de $y_1\, y_2\, y_3$.
   b) Tracé des orbites dans l'espace des phases de $y_1\, y_2\, y_3$.
   c) Iterpretation
   On constate que pour une perturbation des conditions initiales, on retrouve un autre phénomène.
   Donc on peut dire que le système est chaotique et qu'il n'est pas possible de prédire quelque chose sur une longue durée, car il suffit d'une peite perturbation pour avoir autre chose. Impossible de prévoir à long terme l'état du système.

4 Exercice 5

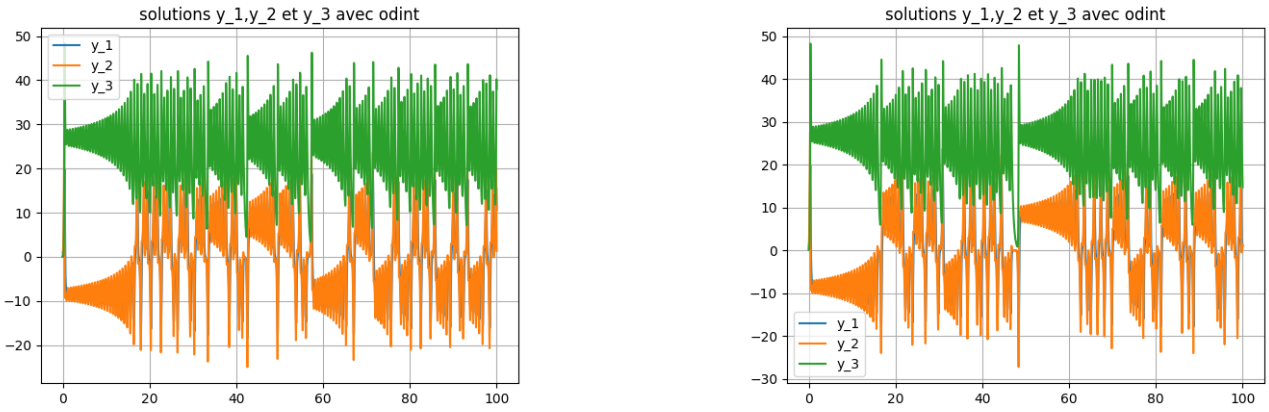


FIGURE 6 – Représentation des solutions.
La figure de gauche est tracée avec comme conditions initiale $(0, 1, 0)$ et $(0.5, 1, 0.1)$.

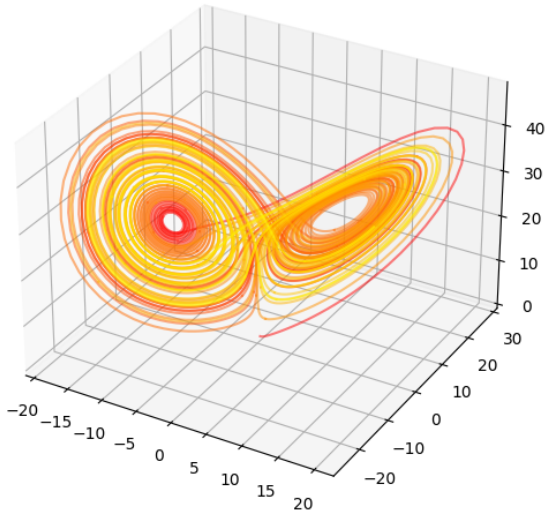


FIGURE 7 – Orbites dans l'espace des phases.

A Scénarios de déstabilisations en dimension 3

Considérons le cas où il y a une valeur propre réelle et deux valeurs propres complexes conjuguées et où le système est initialement dans un état stable(la partie réelle des trois valeurs propres sont strictement négatives). Dans ce cas on observe de deux scénarios de déstabilisations (voir fig. 8) :

- Scénario 1 La valeur propre réelle "traverse" l'axe des ordonnées, un attracteur disparaît. La bifurcation est alors.
  - de type fourche s'il y a une brisure de symétrie
  - de type nœud-col sinon
- Scénario 2 Les deux valeurs propres complexes "traversent" l'axe des ordonnées, un nouveau attracteur apparaît(cycle limite). Dans ce cas il s'agit d'une bifurcation de Hopf.

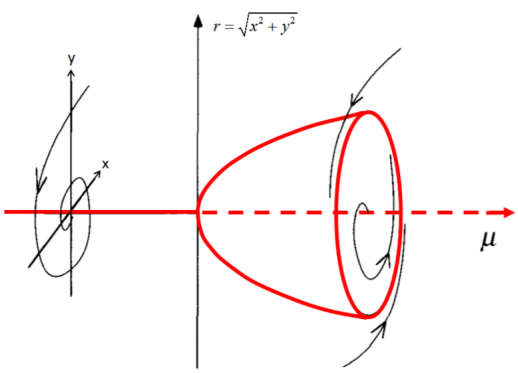


FIGURE 8 – Deux scénarios de déstabilisation. État initialement stable.

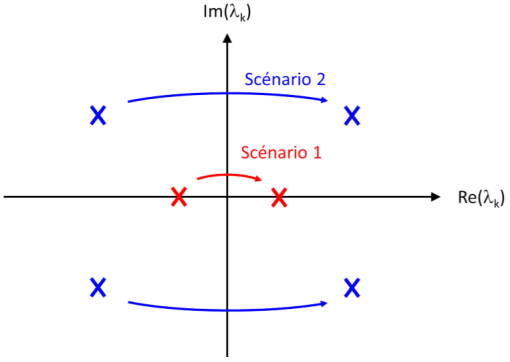


FIGURE 9 – Deux scénarios de déstabilisation. État initialement stable.

B code Python

Ci-dessous vous trouverez les codes qui nous ont permit de faire ce travail.

```
# coding: utf-8

from __future__ import division
from numpy import *
#from math import *
from matplotlib.pyplot import *
from scipy.integrate import *
from scipy.optimize import fsolve
from scipy import *
from pylab import *
from mpl_toolkits.mplot3d import Axes3D



### ********************************************** ###
# definition de la fonction d'euler implicite

def euler_implicite(f,a,b,y0,n):
T,Y,h=[a],[y0],(b-a)/n
for i in range(1,n+1):
T.append(i*h)
```

```
Y.append(fsolve(lambda y:Y[i-1]+h*f(T[i],y)-y,0.01))
return T,Y

### ********************************************** ###

# definition de la fonction euler explicite

def euler(F,t0,tf,y0,n):
"""
Données:
F(y,t) une fonction
t0,t1 deux réels avec t0 < t1
y0 un réel
n un entier
Résultat: le tuple constitué de la liste des temps [t0,...,tn]
et la liste des (n+1) réels [y_0, ...y_n]
qui constituent une approximation de la solution y sur [t0,tf]
de l'ED y'=F(y,t) avec la condition initiale y(t0) = y0

"""
h = (tf-t0)/n
y = y0
t = t0
Y = [y0]
T = [t0]
for k in range(n): # n itérations donc n+1 points
y = y + h*F(y,t)
t = t + h
Y.append(y)
T.append(t)
return T,Y

### ***************fonctions pour l'exercice 1*************** ###

# definition de la fonction definie dans le probleme de cauchy
def F1(u,t):
return -100*u + 25

### ************************************************ ###

def solution1(t,y0):
return (y0+3*exp(-100*t))/4


### *************fonctions pour l'exercice 2************** ###

def F2(y,t):
return 3*exp(0.5*t)-0.2*y

def solution2(t,y0):
```

```
return exp(-0.2*t)*(23+30*exp(0.7*t))/7

### ****************fonction intermediaire************** ###

def solution_vecteur(tvecteur, fonction):
sol=[fonction(t,y0) for t in tvecteur]
return sol

### ************************************************ ###

def erreur(x,y,err):
n=len(err)
for i in range(n):
err[i]=err[i]+abs(x[i]-y[i])
return err

### ************************************************* ###
# Exercice 1
n1=25
n2=250
n3=2500
a=0
b=1
y0 = 1

# Euler explicite

T1,Y1 = euler(F1,a,b,y0,n1)
T2,Y2 = euler(F1,a,b,y0,n2)
T3,Y3 = euler(F1,a,b,y0,n3)
sol=solution_vecteur(T1,solution1)
plot(T1,sol,label="solution")
#plot(T1,Y1, label="Y1")
plot(T2,Y2, label="Y2, h=1/250")
plot(T3,Y3, label="Y3, h=1/2500")
#axis("equal") # repère orthonormé
xlabel('temps')
#ylabel('la solution aproximative')
axis([0,1,0,1])
title("Visualisation de la methode d'euler explicite")
legend(loc='upper right')
grid()
show()

# Euler implicite

T1,Y1 = euler_implicite(F1,a,b,y0,n1)
T2,Y2 = euler_implicite(F1,a,b,y0,n2)
T3,Y3 = euler_implicite(F1,a,b,y0,n3)
sol=solution_vecteur(T1,solution1)
```

```
plot(T1,sol,label="solution")
#plot(T1,Y1, label="Y1")
plot(T2,Y2, label="Y2, h=1/250")
plot(T3,Y3, label="Y3, h=1/2500")
#axis("equal") # repère orthonormé
xlabel('temps')
ylabel('la solution aproximative')
axis([0,1,0,1])
title("Visualisation de la methode d'euler implicite")
legend(loc='upper right')
grid()
show()


# Comparaison des deux methodes d'Euler
T1_exp,Y1_exp = euler(F1,a,b,y0,n1)
T1_imp,Y1_imp = euler_implicite(F1,a,b,y0,n1)
plot(T1,sol,label="solution")
plot(T1_exp,Y1_exp, label="Y1_exp")
plot(T1_imp,Y1_imp, label="Y1_imp")
xlabel('temps')
ylabel('la solution aproximative')
axis([0,1,0,1])
title("Comparaire des deux methodes pour h=1/25")
legend(loc='upper right')
grid()
show()
#************************************************#
T1_exp,Y1_exp = euler(F1,a,b,y0,n2)
T1_imp,Y1_imp = euler_implicite(F1,a,b,y0,n2)
plot(T1,sol,label="solution")
plot(T1_exp,Y1_exp, label="Y2_exp")
plot(T1_imp,Y1_imp, label="Y2_imp")
xlabel('temps')
ylabel('visualisation de la solution aproximative')
axis([0,1,0,1])
title("Comparaire des deux methodes pour h=1/250")
legend(loc='upper right')
grid()
show()
#************************************************#
T1_exp,Y1_exp = euler(F1,a,b,y0,n3)
T1_imp,Y1_imp = euler_implicite(F1,a,b,y0,n3)
plot(T1,sol,label="solution")
plot(T1_exp,Y1_exp, label="Y3_exp")
plot(T1_imp,Y1_imp, label="Y3_imp")
xlabel('temps')
ylabel('Visualisation de la solution aproximative')
axis([0,1,0,1])
title("Comparaire des deux methodes pour h=1/2500")
```

```
legend(loc='upper right')
grid()
show()
### ***************fin exercice 1*****************###


# Exercice 2

a=0
b=4
y0=1
n=4
tab_T,solution_approchée = euler(F2,a,b,y0,n)
solution_exacte=solution_vecteur(tab_T,solution2)
#print("exacte:\n",solution_exacte)
#print("*****************************************\n:")
print("approchée:\n",solution_approchée)
#print("*****************************************\n:")
tab_erreur = array([0,0,0,0,0])
tab_erreur=erreur(solution_exacte,solution_approchée,tab_erreur)
for i in range(len(tab_T)):
tab_erreur[i]=100*(tab_erreur[i]/solution_exacte[i])
print("****************** Tableau *****************\n:")
tableau=zeros([5,3]) #: tableau rempli de flottants zeros ;
tableau[:,0]=tab_T
tableau[:,1]=solution_exacte
tableau[:,2]=tab_erreur
print(tableau)

# Graphe

plot(tab_T,solution_exacte,label="solution exacte")
plot(tab_T,solution_approchée, label="solution approchée")
xlabel('temps')
ylabel('la solution')
axis([0,4,0,100])
title("Comparaison des deux solution pour h=4/5")
legend(loc='upper right')
grid()
show()
#*************************************************#

plot(tab_T,tab_erreur,label='erreur')
xlabel('temps')
ylabel('Erreur')
axis([0,4,0,100])
title("Comparaison des deux solution pour h=4/5")
legend(loc='upper right')
grid()
show()
```

```
### ***********************fin exercice 2********** ###


# Exercice 4

sigma = 10
rho = 28
beta = 8/3
def deriv(syst,t):
    X = syst[0] #Variable 1 : x(t)
    Y = syst[1] #Variable 2 : y(t)
    Z = syst[2] #Variable 3 : z(t)
    dX = sigma*(Y-X)
    dY = rho*X - Y - X*Z
    dZ = -beta*Z + X*Y
    return [dX,dY,dZ]
start = 0
end = 100
numsteps = 10000
t=linspace(start,end,numsteps)
## Conditions initiales
X0 = 0.5
Y0 = 1
Z0 = 0.1
syst_CI = array([X0,Y0,Z0]) #Tableau des Conditions Initiales
sols = odeint(deriv,syst_CI,t)
X = sols[:,0]
Y = sols[:,1]
Z = sols[:,2]

## Graphisme des solutions X, Y et Z
plot(t,X,label="y_1")
plot(t,Y,label="y_2")
plot(t,Z,label="y_3")
title('solutions y_1,y_2 et y_3 avec odint')
grid()
legend()
show()

print(X)
fig = plt.figure(figsize = (10,10))
ax = fig.gca(projection='3d')
nbseg = 100
c = np.linspace(0,1,numsteps)
for i in range(0,numsteps-nbseg,nbseg):
    ax.set_axis_on()
    ax.plot(X[i:i+nbseg+1], Y[i:i+nbseg+1], Z[i:i+nbseg+1], color=(1,c[i],0), alpha=0.5)
    plt.pause(0.1)
plt.show()
```