Projet Equations différentielles
L'attracteur de Lorenz

Valentin KRAEMER et Félix HUSSON
15 septembre 2022

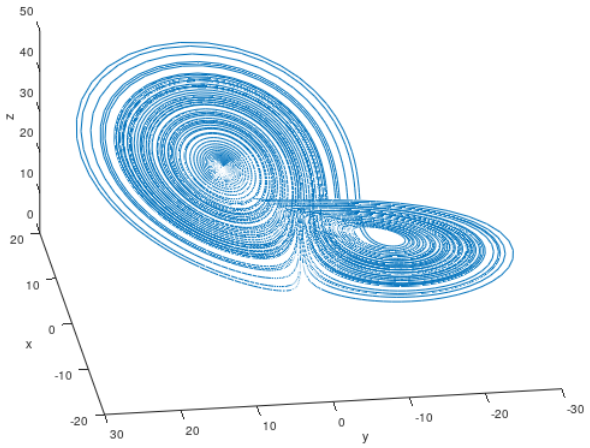


Table des matières

1 Introduction **2**
- 1.1 Idée générale . . . 2
- 1.2 Equation . . . 2
- 1.3 Analyse théorique de l'équation . . . 2
  - 1.3.1 Points critiques . . . 2
  - 1.3.2 Nature des points critiques . . . 3

2 Schémas numériques **6**
- 2.1 Euler explicite . . . 6
  - 2.1.1 Algorithme . . . 6
  - 2.1.2 Stabilité absolue . . . 6
- 2.2 Euler implicite . . . 7
  - 2.2.1 Algorithme . . . 7
- 2.3 Runge Kutta 4 . . . 8
  - 2.3.1 Algorithme . . . 8

3 Représentation graphique et interprétation **9**
- 3.1 Au voisinage du point (0,0,0) . . . 9
- 3.2 Au voisinage d'un autre point critique . . . 12

4 Limite des schémas numériques **14**
- 4.1 Euler implicite : dépendance du nombre de points N . . . 14
- 4.2 Stabilité . . . 14
  - 4.2.1 Euler explicite . . . 15
  - 4.2.2 Euler implicite . . . 16
  - 4.2.3 Runge Kutta 4 . . . 17
  - 4.2.4 Résultats de la stabilité numérique . . . 17
  - 4.2.5 Superposition des graphes implicites . . . 17

5 Conclusion rapide **22**

6 Annexe **23**
- 6.1 Euler explicite . . . 23
- 6.2 Euler implicite . . . 24
- 6.3 Runge Kutta 4 . . . 26

7 Bibliographie **27**

1 Introduction

Edward Lorenz est un mathématicien et météorologue américain du XXe sciècle. Il est à l'origine d'un modèle météorologique (1963), portant son nom. Celui-ci sert à décrire des mouvements de convection et découle des équations de Navier-Stokes ainsi que du couplage de l'atmosphère terrestre et des Océans.

1.1 Idée générale

Ce système est une simplification des modèles de l'époque afin de résoudre le problème numériquement après une centaine d'itérations. En effet, la capacité de calcul des ordinateurs de l'époque était sensiblement plus faible par rapport à celle d'aujourd'hui. Ce système est connu pour son comportement chaotique et ce phénomène de double attracteur appelé "ailes de papillon".

1.2 Equation

Ce modèle, tiré des équations différentielles de la mécanique des fluides s'exprime de cette manière :
Posons x, y, z des variables représentant respectivement l'intensité de mouvement de convection, la différence de température entre les courants ascendants et descendants et l'écart de température vertical par rapport à un profil linéaire (cf littérature sur le modèle de Lorenz).

On obtient alors le système différentiel ci-contre :

$$\begin{cases} x' = \sigma(y - x) \\ y' = x(\rho - z) - y \\ z' = xy - \beta z \end{cases} \tag{1}$$

Avec σ, β, ρ trois constantes (nombre de Prandt, nombre de Rayleigh et paramètre lié à l'épaisseur du système étudié) et $\beta \in]0, 4]$, $\sigma, \rho > 0$.
Ces conditions physiques nous serviront par la suite pour écarter des sous-cas mathématiquement plausibles.

1.3 Analyse théorique de l'équation

1.3.1 Points critiques

Nous commençons cette étude par une analyse des points critiques du système. Ceux-ci se déterminent en résolvant ce système :

$$\begin{cases} \sigma(y - x) = 0 \\ x(\rho - z) - y = 0 \\ xy - \beta z = 0 \end{cases} \tag{2}$$

On trouve alors les trois points critiques du système :
- l'origine (0,0,0)
- deux points opposés ($\pm\sqrt{\beta(\rho - 1)}, \pm\sqrt{\beta(\rho - 1)}$,$\rho$ -1), que l'on appellera "points conjugués" dans la suite de notre analyse.

Il s'agit maintenant d'étudier la stabilité de ces points critiques.

1.3.2 Nature des points critiques

Commençons par étudier la nature du point critique (0,0,0). Pour ce faire, on étudie la jacobienne du système au point (0,0,0) et on cherche ses valeurs propres. Celle-ci s'écrit :

$$\begin{pmatrix} -\sigma & \sigma & 0 \\ \rho & -1 & 0 \\ 0 & 0 & -\beta \end{pmatrix}$$

Le point d'origine est attractif si et seulement si les valeurs propres de la jacobienne sont négatives.
On détermine le polynôme caractéristique : $P(X) = (X+\beta)(X^2 + (\sigma+1)X + \sigma(1-\rho))$
Le discriminant s'écrit : $\Delta = (\sigma+1)^2 - 4\sigma(1-\rho) = (\sigma-1)^2 + \rho$. Il est positif car $\rho > 0$. On en déduit les racines réelles suivantes :

$$X_{0,1} = \frac{-(\sigma+1)-\sqrt{\Delta}}{2},\ X_{0,2} = \frac{-(\sigma+1)+\sqrt{\Delta}}{2}\ et\ -\beta.$$

On a alors stabilité si $X_{0,1} < 0, X_{0,2} < 0$ et $\beta > 0$. La dernière condition est toujours vérifiée d'après les données du problème.
Pour avoir $X_{0,2} < 0$, il faut avoir $\rho < 1$ On en déduit finalement que le point critique (0,0,0) est attracteur (stable) dans le cas où $\rho < 1$.

A contrario, si $\rho > 1$, les racines ne sont pas toutes négatives, le point est répulsif (instable).

On s'intéresse maintenant au point critique ($\sqrt{\beta(\rho-1)}$, $\sqrt{\beta(\rho-1)}$,ρ -1).

De même, sa jacobienne s'écrit :

$$\begin{pmatrix} -\sigma & \sigma & 0 \\ 1 & -1 & -\sqrt{\beta(\rho-1)} \\ \sqrt{\beta(\rho-1)} & \sqrt{\beta(\rho-1)} & -\beta \end{pmatrix}$$

On a le polynôme caractéristique suivant : $P(X) = X^3 + (1+\beta+\sigma)X^2 + (\beta.\rho + \beta.\sigma)X + 2\sigma\beta(\rho-1)$

Remarquons que, pour le point, $(-\sqrt{\beta(\rho-1)}, -\sqrt{\beta(\rho-1)}, \rho-1)$, on obtient le même polynôme caractéristique. En effet, cela se retrouve par sa jacobienne très ressemblante :

$$\begin{pmatrix} -\sigma & \sigma & 0 \\ 1 & -1 & \sqrt{\beta(\rho-1)} \\ -\sqrt{\beta(\rho-1)} & -\sqrt{\beta(\rho-1)} & -\beta \end{pmatrix}$$

Il est difficile de trouver les valeurs propres du polynôme caractéristique. Nous avons donc utilisé un logiciel de calcul formel (MAPLE) pour les déterminer. Cependant la forme des valeurs propres en fonction des paramètres α, β, ρ est très complexe, et difficile à manipuler analytiquement. Nous notons néanmoins le résultat suivant : la première valeur propre est toujours réelle et négative, et les deux autres sont conjuguées.

En fixant σ et β et en faisant varier ρ (de 2 à 26), on observe la présence d'une valeur $\rho_{critique}$, pour laquelle la partie réelle de la valeur propre problématique change de signe.

En notant ces valeurs dans Matlab, on obtient le graphe présent en figure 1 qui suit :

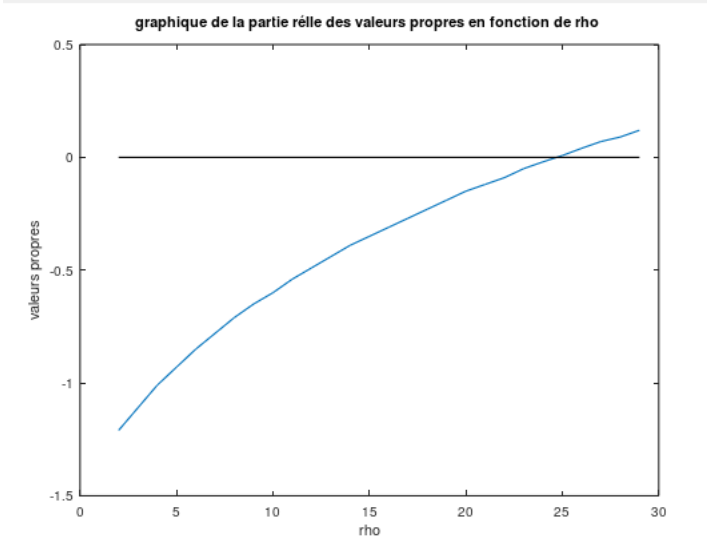


FIGURE 1 – Partie réelle des valeurs propres conjuguées en fonction de ρ

Numériquement, nous constatons la présence d'une valeur $\rho_{critique}$ entre 24 et 25 pour $\sigma = 10$ et $\beta = 8/3$ (valeurs très courantes dans la littérature).

Dans un second temps, nous avons souhaité déterminer théoriquement la valeur de $\rho_{critique}$. Pour ce faire, on revient à la forme du polynôme caractéristique P :

$$P(X) = X^3 + (1 + \beta + \sigma)X^2 + (\beta.\rho + \beta.\sigma)X + 2\sigma\beta(\rho - 1)$$

Cette fonction est polynomiale selon X et ρ donc continue. De plus, on sait qu'il existe des valeurs de ρ pour lesquelles on a stabilité (pour $\rho = 13$ par exemple). Par continuité de la valeur de la racine (en fonction de ρ), on en déduit que ce changement de signe se passe lorsque la racine est complexe. En effet, pour X=0, on a $\rho = 1$, qui n'est pas possible compte tenu de la condition d'existence du point critique.

De même, par continuité de la partie réelle de cette racine, on en déduit que le $\rho_{critique}$ induit une racine imaginaire pure pour P que le polynôme P.

On injecte alors une racine $i\omega$ dans le polynôme caractéristique, pour ω réel.Il vient, en isolant partie réelle et imaginaire, le système suivant :

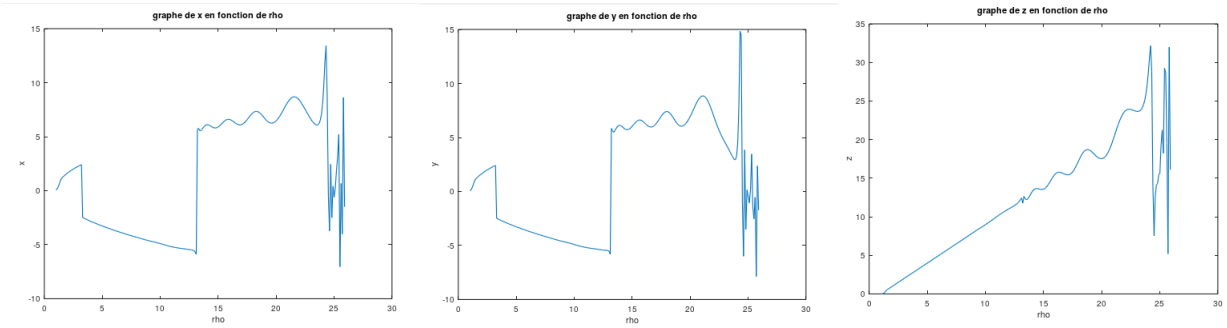
$$\begin{cases} -\omega^2(\sigma + \beta + 1) + 2\beta(\rho - 1)\sigma = 0 \\ -i\omega^3 + i\omega\beta(\sigma + \rho) = 0 \end{cases} \tag{3}$$

Compte tenu du fait que $\omega \neq 0$, on peut diviser dans la deuxième ligne pour obtenir : $\rho_{critique} = \frac{\sigma(\sigma+\beta+3)}{(\sigma-\beta-1)}$

L'application numérique avec des valeurs courantes dans la littérature ($\sigma = 10$ et $\beta = 8/3$) nous donne $\rho_{critique} \simeq 24,73$.Cela confirme notre hypothèse numérique.

Nous venons alors de déterminer la valeur de ρ pour laquelle les deux points critiques basculent d'un état stable à instable (stable pour $1<\rho < \rho_{critique}$ et instable pour $\rho > \rho_{critique}$).

Pour confirmer notre analyse théorique, nous étudions la valeur finale des coordonnées (x, y, z) au bout de 10 000 itérations pour ρ variant de 2 à 26. On exhibe alors un changement de comportement à partir du $\rho_{critique}$. Ci-dessous les graphes de la valeur de (x, y, z) en fonction de ρ avec pour valeurs initiales (10,10,20).

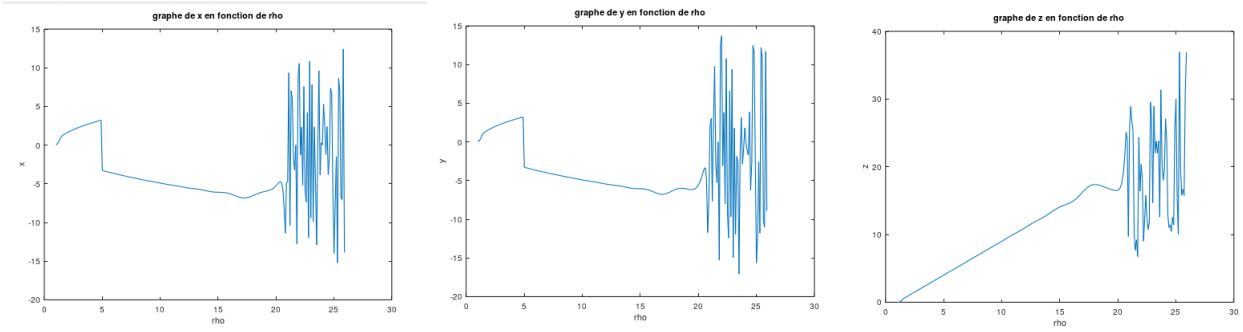


(a) Graphe de x(ρ) (b) Graphe de y(ρ) (c) Graphe de z(ρ)

FIGURE 2 – Détermination du $\rho_{critique}$ avec (10,10,20) comme point initial

De plus, on remarque que x et y ont un comportement similaire. Cela vient du fait que l'on a, proche des points critiques l'égalité x=y. En outre, on remarque des sauts au niveau de $\rho = 3$ et $\rho = 14$. Cependant, il ne s'agit pas du comportement chaotique présent dans les trois graphes au niveau de $\rho = 24$.

Changeons alors de coordonnées initiales. Prenons les valeurs (5,-5,20).



(a) Graphe de x(ρ) (b) Graphe de y(ρ) (c) Graphe de z(ρ)

FIGURE 3 – Détermination du $\rho_{critique}$ avec (5,-5,20) comme point initial

L'emplacement initial a beaucoup d'importance dans le comportement du système. En effet, on remarque qu'il n'y a plus de saut pour $\rho = 13$ et que le phénomène chaotique que l'on interprétait comme l'expression du $\rho_{critique}$ se produit à $rho = 22$. Ces sauts de discontinuité et ce comportement chaotique pour une valeur anormale de ρ sont mentionnés dans la littérature sous le nom de bifurcation de Hopf.

Précisons néanmoins que les derniers graphes ont été obtenus en utilisant le schéma d'Euler explicite que l'on développe dans la partie suivante.

2 Schémas numériques

Trois méthodes numériques ont été mises en place pour tenter de résoudre le problème : la méthode d'Euler explicite, implicite, et Runge Kutta 4. Ces méthodes ont été implémentées dans Matlab dans un premier temps pour vérifier notre code, puis en C pour augmenter la capacité de calcul.

2.1 Euler explicite

Commençons par présenter le schéma d'Euler explicite utilisé dans la résolution de ce système.

2.1.1 Algorithme

En reprenant le système différentiel $X' = F(X,t)$, on obtient le schéma $X_{n+1} = X_n + hF(X_n,t)$ qui s'écrit :

$$\begin{cases} x_{n+1} = x_n + dt.\sigma(y_n - x_n) \\ y_{n+1} = y_n + dt.(x_n(\rho - z_n) - y_n) \\ z_{n+1} = z_n + dt.(x_n y_n - \beta z_n) \end{cases} \tag{4}$$

Précisons que le h choisit dans notre méthode est de la forme (dt, dt, dt) par souci de simplification.

2.1.2 Stabilité absolue

Nous allons étudier la stabilité absolue c'est à dire au bout d'un grand nombre d'itérations. Précisons que la non linéarité de l'équation différentielle nous rend le travail plus difficile. Cette partie est alors une tentative d'explication non rigoureuse et inachevée.

$$\begin{cases} x_{n+1} = (1 - dt.\sigma)x_n + dt.\sigma.y_n \\ y_{n+1} = (1 + dt)y_n + dt.x_n(\rho - z_n) \\ z_{n+1} = (1 - \beta.dt)z_n + dt.x_n.y_n \end{cases} \tag{5}$$

Par récurrence :

$$\begin{cases} x_{n+1} = x_0(1 - dt.\sigma)^n + dt.\sigma.y_n \\ y_{n+1} = y_0(1 + dt)^n + dt.x_n(\rho - z_n) \\ z_{n+1} = z_0(1 - \beta.dt)^n + dt.x_n.y_n \end{cases} \tag{6}$$

Ce sont des suites géométriques ainsi elles convergent si et seulement si $|1 - dt.\sigma| < 1$ pour la première équation, ssi $|1 - h| < 1$ pour la deuxième et ssi $|1 - \beta dt| < 1$ pour la troisième.
Heureusement les contraintes sont respectées en pratique à condition de prendre un pas de temps relativement petit devant σ et β.
Les paramètres σ et β sont positifs donc il faut seulement que le produit de $\sigma.dt$ soit inférieur à 2 comme celui de $\beta.dt$. Pour notre cas il nous faut un pas de temps inférieur à 0.2 pour être stable au long terme avec la méthode d'Euler explicite.Mais ici nous avons ignoré la non linéarité rendant peut être le résultat caduc.

Après quelques recherches, nous avons trouvé que le raisonnement sur la stabilité des points critiques demandait une utilisation du théorème de stabilité de Liapounov. Nous ne nous sommes pas engagés dans cette voie.

2.2 Euler implicite

Dans la même idée que le schéma explicite, nous avons souhaité utiliser un schéma implicite, potentiellement plus stable sur le long terme. C'est la raison pour laquelle nous avons choisi d'utiliser la méthode d'Euler implicite. Voici le schéma numérique par la méthode d'Euler implicite :

2.2.1 Algorithme

A nouveau, pour le système différentiel $X' = F(X,t)$, on a le schéma $X_{n+1} = X_n + hF(X_{n+1}, t)$, s'écrivant sous la forme de système :

$$\begin{cases} x_{n+1} = x_n + dt.\sigma(y_{n+1} - x_{n+1}) \\ y_{n+1} = y_n + dt.(x_{n+1}(\rho - z_{n+1}) - y_{n+1}) \\ z_{n+1} = z_n + dt.(x_{n+1}y_{n+1} - \beta z_{n+1}) \end{cases} \tag{7}$$

Il peut se réécrire le système de la manière suivante :

$$\begin{cases} x_{n+1}(1 + dt.\sigma) = x_n + dt.\sigma y_{n+1} \\ y_{n+1}(1 + dt) = y_n + dt.x_{n+1}(\rho - z_{n+1}) \\ z_{n+1}(1 + \beta dt) = z_n + dt.x_{n+1}y_{n+1} \end{cases} \tag{8}$$

En remplaçant x_{n+1} par son expression dans les deux dernières lignes, puis z_{n+1} dans la deuxième, on obtient une équation avec pour unique inconnue y_{n+1} que l'on résout numériquement alors par la méthode de Newton. En effet, cette équation polynomiale de degré 3 n'est pas aisée à résoudre à la main (même en considérant les formules pour les racines du troisième degré). La fonction f qui est utilisée pour déterminer y_{n+1} est la suivante :

$$f := y_{n+1} \mapsto A^2Cy_n + ACh\rho x_n - Ahz_nx_n + (ACh^2\rho\sigma - Ah^2z_n - x_n^2h^2 - A^2BC)y_{n+1} - 2x_n\sigma h^3y_{n+1}^2 - h^4\sigma^2y_{n+1}^3$$

$$df := y_{n+1} \mapsto A^2Cy_n + ACh\rho x_n - Ahz_nx_n + (ACh^2\rho\sigma - Ah^2z_n - x_n^2h^2 - A^2BC)y_{n+1} - 2x_n\sigma h^3y_{n+1}^2 - h^4\sigma^2y_{n+1}^3$$

Avec $A = 1 + h\sigma$, $B = 1 + h$, et $C = 1 + \beta h$

2.3 Runge Kutta 4

Pour aller plus loin dans la résolution numérique, nous allons de prendre une méthode numérique d'ordre supérieur (ordre 4) : Runge Kutta 4. Cette méthode explicite étant théoriquement plus efficace, nous nous attendons à de meilleurs résultats numériques.
Nous utiliserons le terme "RK4" pour nommer ce schéma.

2.3.1 Algorithme

Le schéma RK4 est de la forme suivante : pour résoudre le système différentiel $X' = F(X,t)$.

$$X_{n+1} = X_n + \frac{k_1 + 2k_2 + 2k_3 + k_k}{6} h$$

Avec : $k_1 = F(X_n, t_n)$, $k_2 = F(X_n + hk_1/2, t_n + h/2)$, $k_3 = F(X_n + hk_2/2, t_n + h/2)$, $k_4 = F(X_3 + h, t_n + h)$

En remplaçant par le système (1) on obtient le schéma numérique suivant :

$$\begin{cases} k_{11} = \sigma(y_n - x_n) \\ k_{21} = \sigma(y_n - (x_n + h.k_{11}/2)) \\ k_{31} = \sigma(y_n - (x_n + h.k_{21}/2)) \\ k_{41} = \sigma(y_n - (x_n + h.k_{31})) \\ x_{n+1} = x_n + h(k_{11} + 2k_{21} + 2k_{31} + k_{41})/6 \\ \\ k_{12} = \rho.x_n - y_n - x_n.z_n \\ k_{22} = \rho.x_n - (y_n + h.k_{12}/2) - x_n.z_n \\ k_{32} = \rho.x_n - (y_n + h.k_{22}/2) - x_n.z_n \\ k_{42} = \rho.x_n - (y_n + h.k_{32}) - x_n.z_n \\ y_{n+1} = y_n + h.(k_{12} + 2k_{22} + 2k_{32} + k_{42})/6 \\ \\ k_{13} = x_n.y_n - \beta.z_n \\ k_{23} = x_n.y_n - \beta(z_n + h.k_{13}/2) \\ k_{33} = x_n.y_n - \beta(z_n + h.k_{23}/2) \\ k_{43} = x_n.y_n - \beta(z_n + h.k_{33}) \\ z_{n+1} = z_n + h.(k_{13} + 2k_{23} + 2k_{33} + k_{43})/6 \end{cases} \tag{9}$$

3 Représentation graphique et interprétation

Ci-dessous ce trouve le schéma dit à "ailes de papillon", caractéristique de l'attracteur de Lorenz. Dans celui-ci, nous observons que la solution tournoie autour de deux attracteurs qui sont en fait les deux points critiques "conjugués" du système.

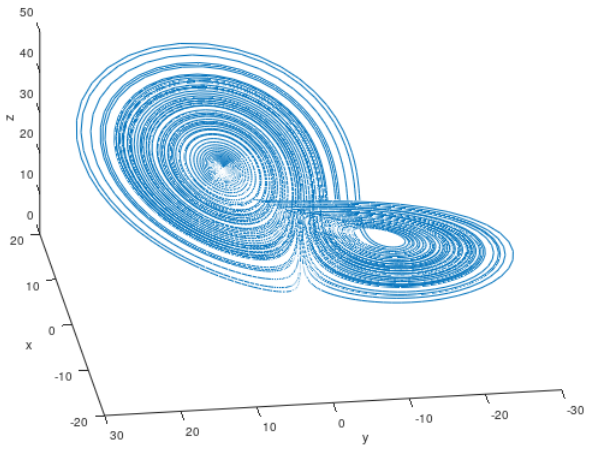


FIGURE 4 – "Ailes de papillon" pour les paramètres $\rho = 28, \sigma = 10, \beta = \dfrac{8}{3}$

Ici, ceux-ci sont présents aux coordonnées (8.48,8.48,27) et (-8.48,-8.48,27) pour des valeurs de $\sigma = 10, \rho = 28, \beta = 8/3$. Dans la suite de notre étude, on se placera proche des points critiques (l'origine et les points "conjugués") : le plus souvent à (0.001, 0.001, 0.001) et (10, 10, 20).

Dans la partie précédente, nous avons vu que la stabilité du point critique dépendait du paramètre ρ à σ, β fixés. Nous avions également remarqué que la stabilité du schéma était altérée après avoir franchi la valeur de $\rho_{critique} \simeq 24.7$. Nous nous intéressons alors à l'évolution de la solution au voisinage de ses points critiques, pour des valeurs de ρ différentes.

Par souci de non redondance, nous choisissons un unique graphe d'une des trois méthodes numériques, ceux-ci étant extrêmement similaires et suffisants pour illustrer nos propos dans cette partie.

3.1 Au voisinage du point (0,0,0)

Commençons par étudier le point critique nul. On rappelle que celui-ci est attractif pour $\rho < 1$ et répulsif pour $\rho > 1$. Nous nous intéressons néanmoins aux deux sous cas $\rho < 24$ et $\rho > 25$.
Nous lançons alors la méthode numérique (ici implicite) au point (0.001,0.001,0.001). Il est nécessaire de se placer proche du point critique sans pour autant y être initialement. En effet, que le point soit attractif ou répulsif, le programme nous donnera une valeur constante nulle si l'on se place initialement en (0,0,0).

Pour $\rho < 1$:
Commençons par le cas où $\rho < 1$. En exécutant le programme avec la méthode d'Euler implicite, on obient le graphe ci-dessous :

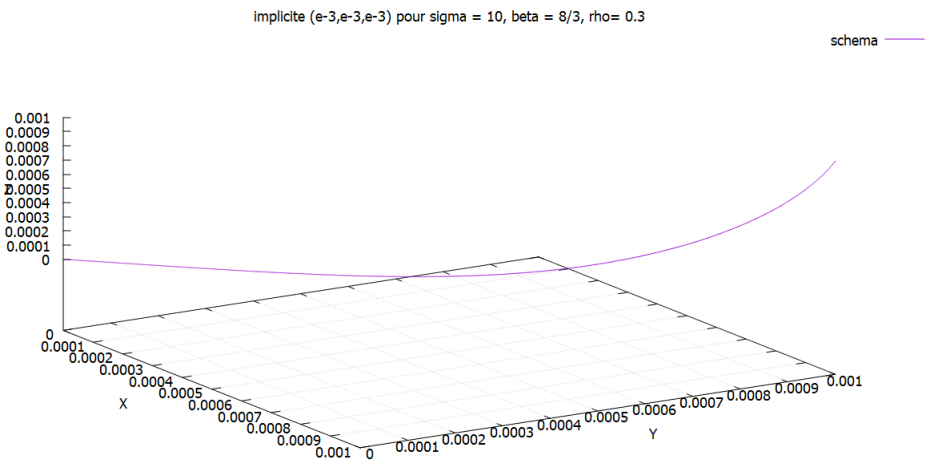


FIGURE 5 – Euler implicite à partir du point (0.001,0.001, 0.001) pour $\rho = 0.3$

Le point initialement placé en (0.001, 0.001, 0.001) converge vers (0,0,0) en accord avec notre analyse préalable. De plus, en n'importe quel point de l'espace il y aura toujours convergence vers cet unique point fixe.

Pour $\rho \in]1, 24]$:
Prenons maintenant $\rho = 13 > 1$. On obtient, avec une méthode identique :

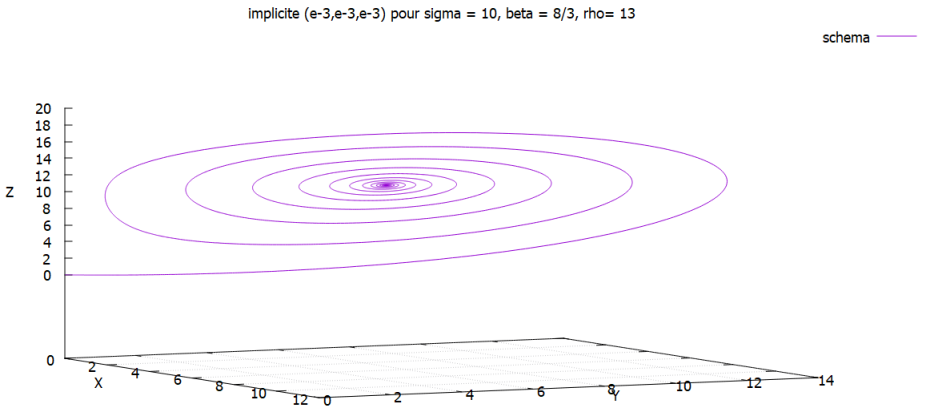


FIGURE 6 – Euler implicite à partir du point (0.001, 0.001, 0.001) pour $\rho = 13$

Cette fois ci, le point (0,0,0) est répulsif. La solution s'éloigne alors de ce point et se dirige vers le point attractif le plus proche (ici $(\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho-1)$). En prenant comme point initial le point (-0.001, -0.001, 0.001), on aurait eu convergence vers le point critique "conjugué" car plus proche.

NB : après plusieurs essais de valeurs de ρ différentes, il semble y avoir des conditions supplémentaires par rapport à notre hypothèse de distance minimale entre le point initial et les points critiques "conjugués".

Pour $\rho > 25$:
Enfin, en prenant un $\rho = 28 > 24.7$, on obtient :

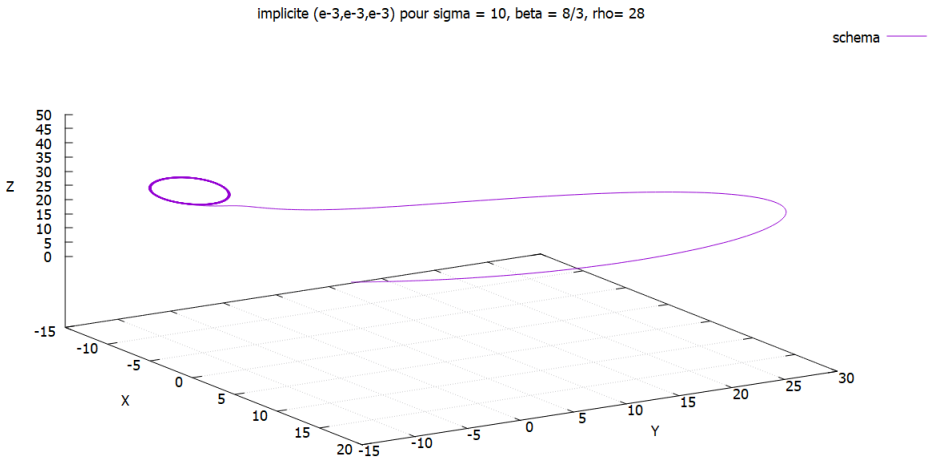


FIGURE 7 – Euler implicite à partir du point (0.001,0.001,0.001) pour $\rho = 28$) et $T = 10$

Néanmoins, il semble y avoir une convergence vers un des points critiques "conjugués". Ce n'est pas le cas dans la pratique. Il suffit d'augmenter le temps d'exécution du programme de $T = 10$ à $T = 30$ pour constater ce phénomène :

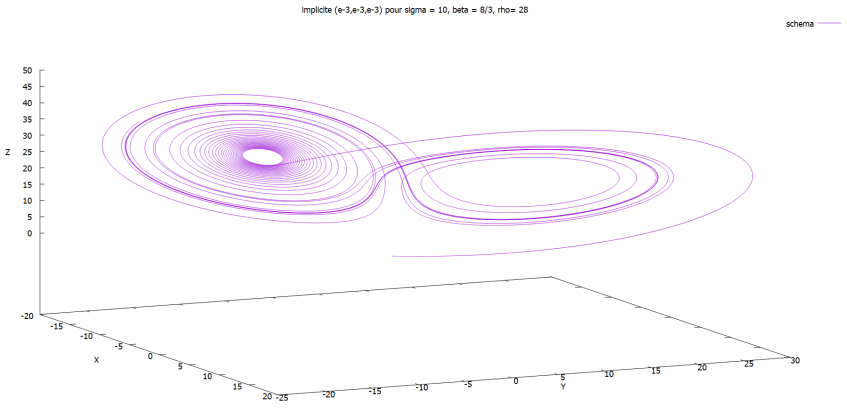


FIGURE 8 – Euler implicite à partir du point (0.001,0.001,0.001) pour $\rho = 28$) et $T = 30$

On retrouve le graphe "aile de papillon" représentant l'oscillation entre les deux points critiques "conjugués" présentés dans l'introduction de cette partie. Il n'y a pas de réelle convergence à long terme.

3.2 Au voisinage d'un autre point critique

On s'intéresse au comportement de notre programme en prenant un point initial proche d'un point critique conjugué. On choisit ici le point (10,10,20).

Rappelons que pour avoir existence d'un des deux autres points critiques, il faut nécessairement que $\rho > 1$.

De même que pour la partie précédente, il y a divergence de la solution pour l'origine et convergence vers un des deux points critiques "conjugués" pour des valeurs de $\rho < \rho_{critique}$. La figure 9 (voir ci-dessous) illustre alors un phénomène de convergence vers le point $(-\sqrt{\beta(\rho-1)}, -\sqrt{\beta(\rho-1}, \rho-1)$. Le résultat est toutefois un peu surprenant. En effet, la solution ne converge pas vers la valeur propre la plus proche (qui serait $(+\sqrt{\beta(\rho-1)}, +\sqrt{\beta(\rho-1}, \rho-1)$.

Le système est alors plus complexe qu'attendu mais converge néanmoins vers un de ses points critiques, comme prédit dans la partie analytique.

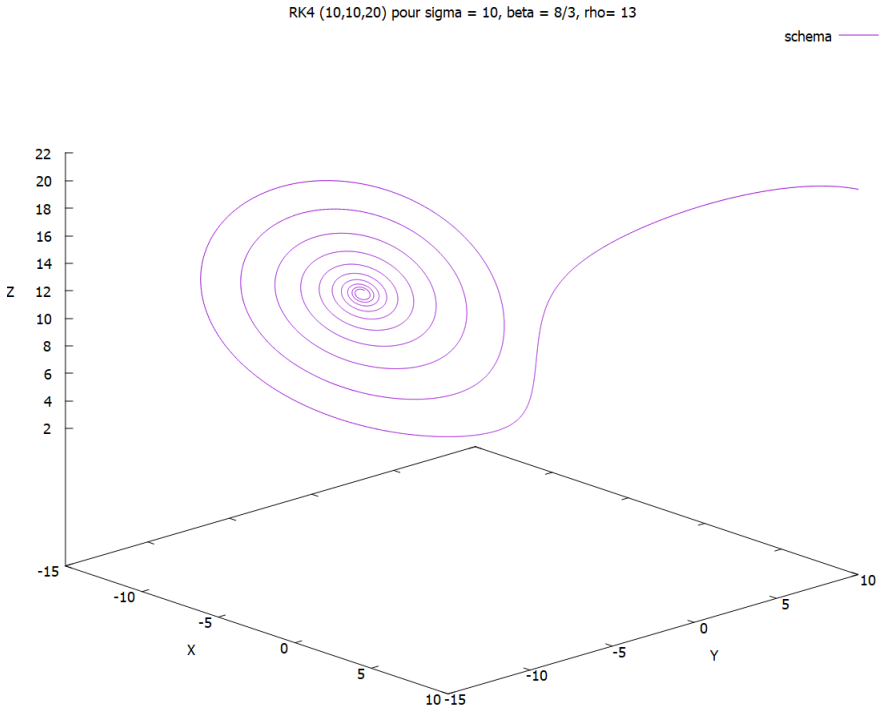


FIGURE 9 – Comportement de la solution avec la méthode RK4 au point (10,10,20) pour $\rho = 13$

La figure 10 ci-dessus illustre le phénomène chaotique déjà énoncé dans la partie précédente. La solution ne converge pas, puisqu'elle oscille entre les deux points critiques "conjugués". C'est ce phénomène particulier qui rend connu l'attracteur de Lorenz.

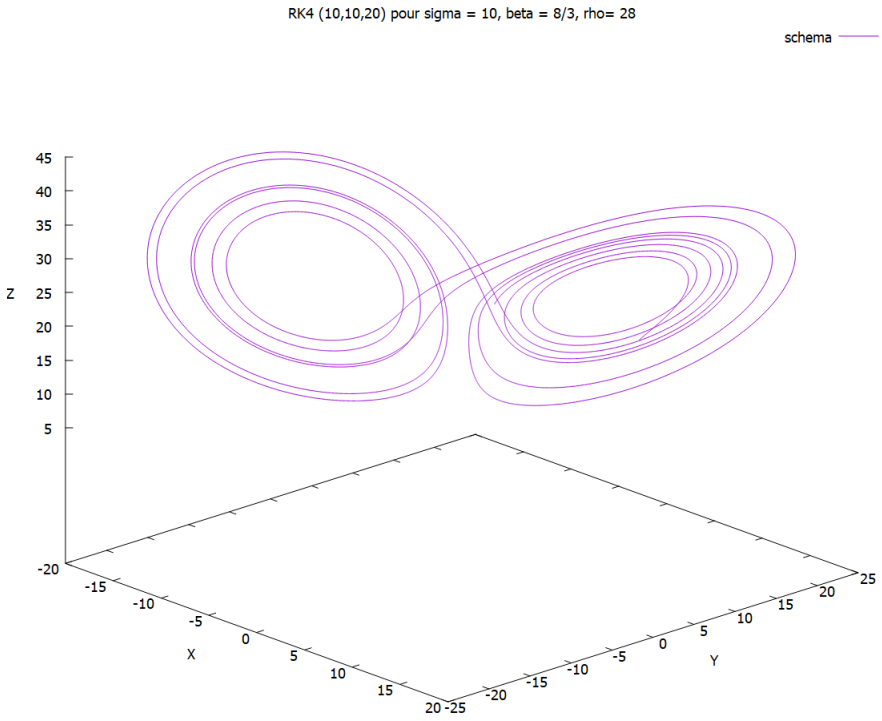


FIGURE 10 – Comportement de la solution avec la méthode RK4 au point (10,10,20) pour $\rho = 28$

4 Limite des schémas numériques

4.1 Euler implicite : dépendance du nombre de points N

Ci-dessous la méthode d'Euler implicite pour N=1000 et T=20 on a un dt=0.02.

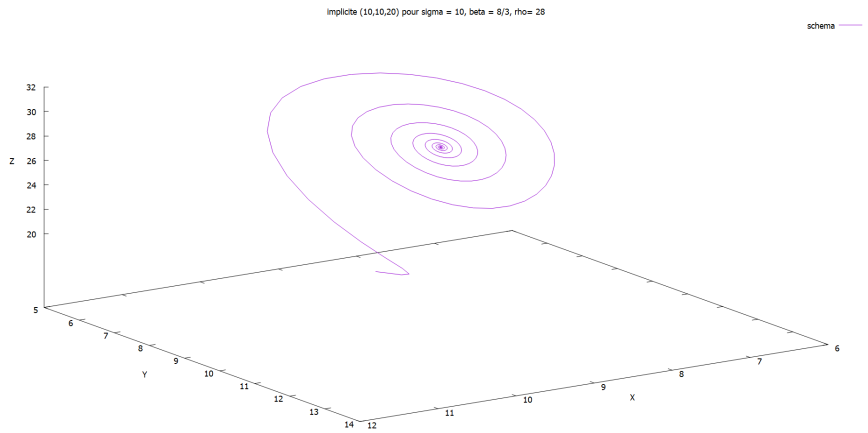


FIGURE 11 – Graphe d'Euler implicite pour N = 1000 au point (10,10,20) et $\rho = 28$

On remarque que le schéma numérique est attiré par seulement le point critique (8.485,8.485,27) ; ce comportement étrange semble s'expliquer part le fait que le dt ne soit pas suffisamment petit. Ci-dessous la méthode d'Euler implicite pour N=10000 avec un comportement comme on le prédisait.

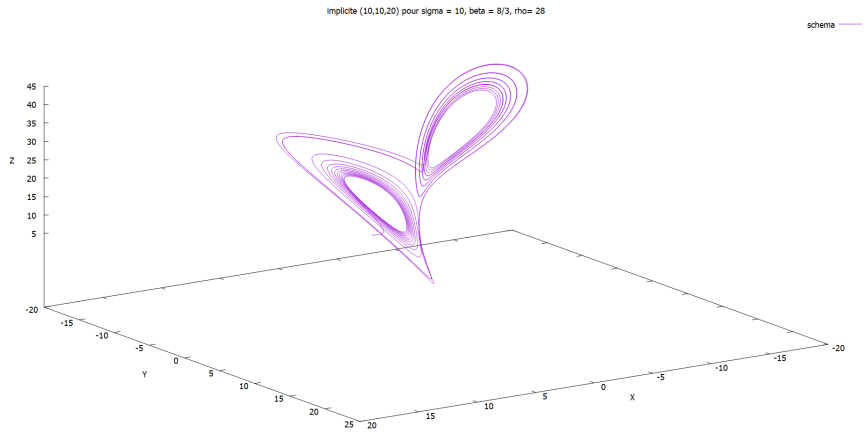
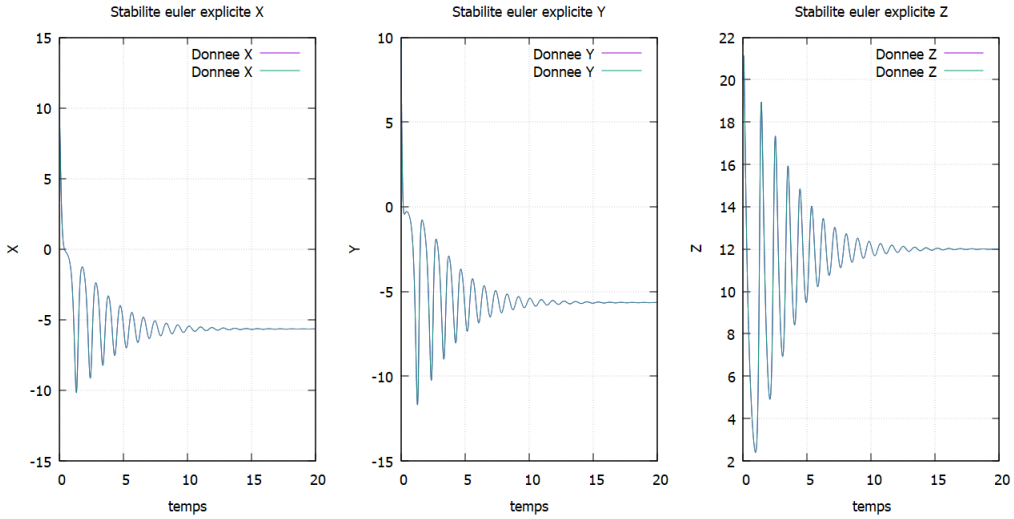


FIGURE 12 – Graphe d'Euler implicite pour N = 10000 au point (10,10,20) et $\rho = 28$

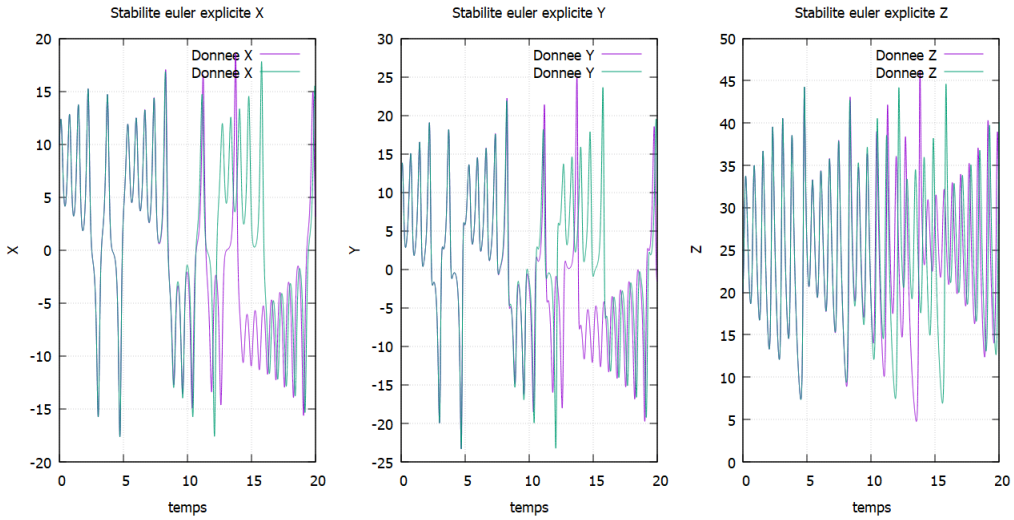
4.2 Stabilité

On s'intéresse dans cette partie à la stabilité des nos schémas numériques. Pour ce faire, on exécute le programme deux fois avec deux points très proches l'un de l'autre et on trace les deux évolutions des variables sur un même graphique. De plus, on tente d'étudier cette stabilité pour deux valeurs de ρ différentes, pour avoir un cas de point critique $(\pm\sqrt{\beta}(\rho-1), \pm\sqrt{\beta}(\rho-1), \rho-1)$ attractif ou répulsif.

4.2.1 Euler explicite



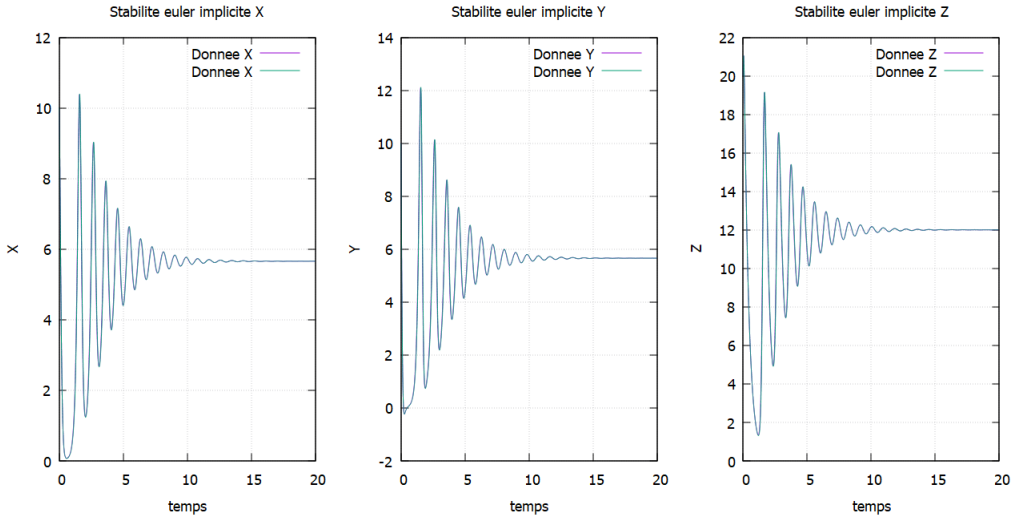
(a) Evolution de x, y, z en fonction du temps pour $\rho = 13$



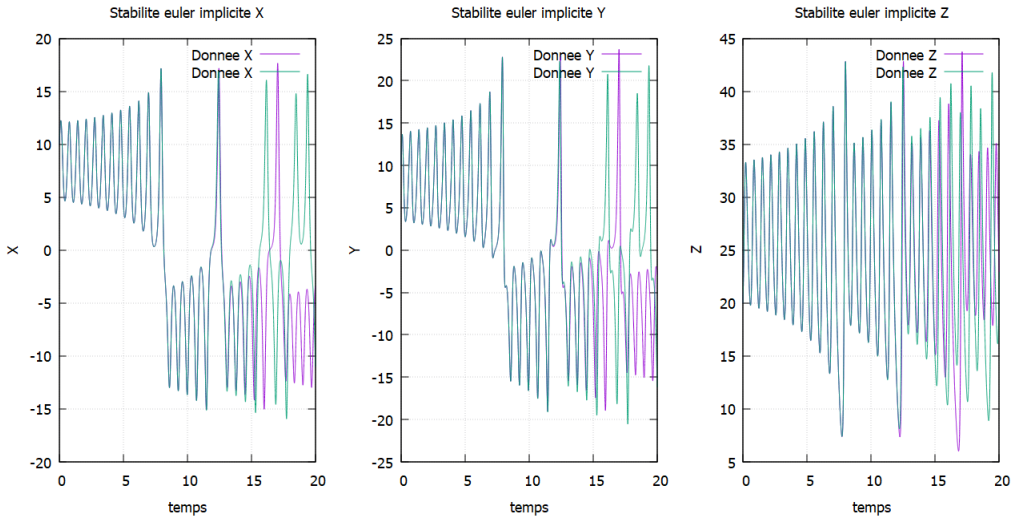
(b) Evolution de x, y, z en fonction du temps pour $\rho = 28$

FIGURE 13 – Stabilité numérique d'Euler explicite pour $\rho = 13$ et $\rho = 28$

4.2.2 Euler implicite



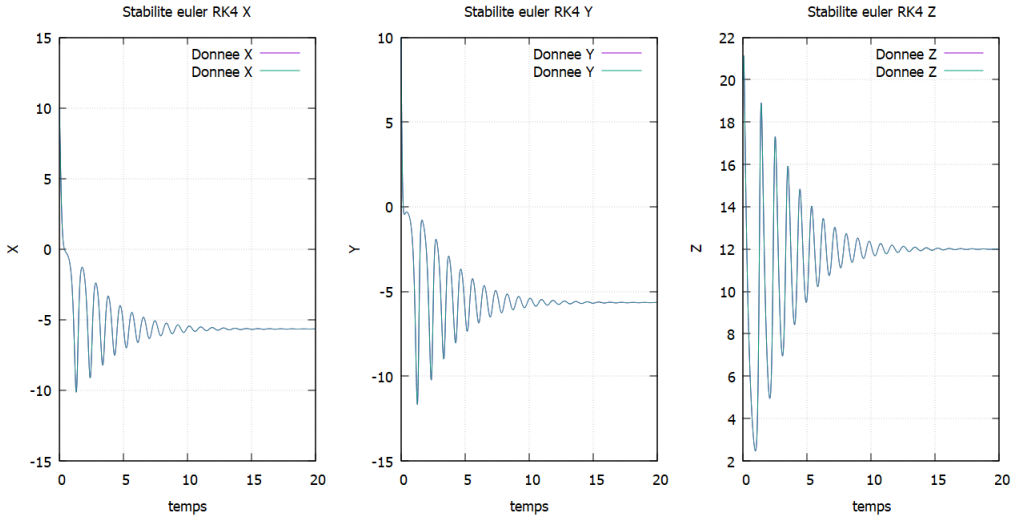
(a) Evolution de x, y, z en fonction du temps pour $\rho = 13$



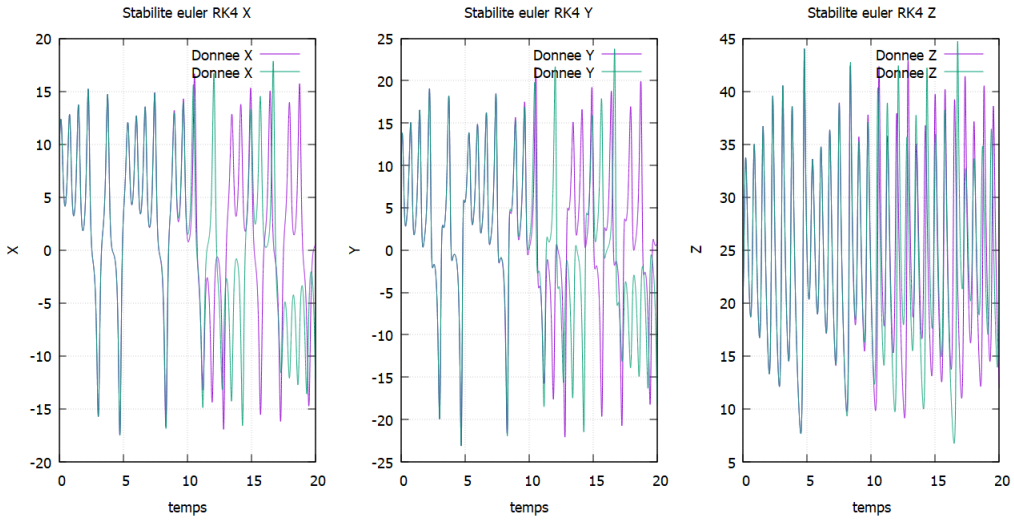
(b) Evolution de x, y, z en fonction du temps pour $\rho = 28$

FIGURE 14 – Stabilité numérique d'Euler implicite pour $\rho = 13$ et $\rho = 28$

4.2.3 Runge Kutta 4



(a) Evolution de x, y, z en fonction du temps pour $\rho = 13$



(b) Evoltion de x, y, z en fonction du temps pour $\rho = 28$

FIGURE 15 – Stabilité nuémrique de RK4 pour $\rho = 13$ et $\rho = 28$

4.2.4 Résultats de la stabilité numérique

Pour les trois méthodes numériques, nous remarquons que les schémas sont relativement stables dès lors que l'on a un $\rho < \rho_{critique}$. On rappelle qu'il n'y a pas de divergence vers l'infini pour des valeurs de $\rho > \rho_{critique}$ mais des sortes d'oscillations entre les deux points critiques.

4.2.5 Superposition des graphes implicites

Il y a un autre moyen de mettre en évidence cette différence de comportement en superposant les deux courbes aux points initiaux proches sur un même graphique. Ici, nous prenons les points $(10, 10, 20)$ et

$(10.01, 10, 20)$. Nous traçons alors trois graphiques pour des valeurs de ρ correspondants aux sous cas déjà présentés.
Il est toutefois important de préciser que l'on peut faire cette manipulation avec les trois schémas, mais ceux-ci n'apportent pas de matière supplémentaire à notre rapport.

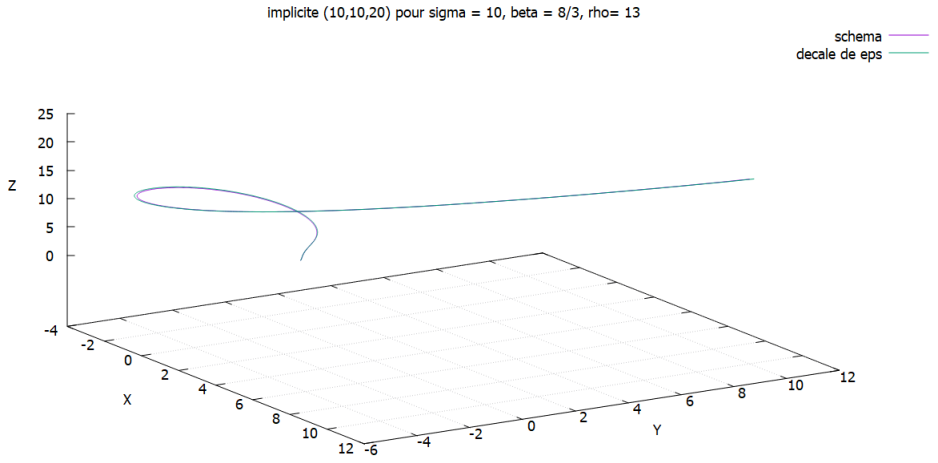


FIGURE 16 – Superposition de graphes avec points initiaux proches pour la méthode d'Euler implicite pour des valeurs de $\rho = 0.3 < 1$

Le graphe présenté ci-dessus semble avoir un unique graphique. En effet, le schéma a une très bonne stabilité numérique compte tenu de la "bonne" convergence vers 0.

Le graphe ci-dessous converge également. Néanmoins, on remarque un léger décalage qui s'atténue au fur et à mesure du temps, puisque les deux points convergent vers le même point critique. Il est intéressant de noter que l'on pourrait avoir deux points relativement proches qui convergeraient chacun vers une valeur propre différente.

Cependant, nous n'avons pas essayé suffisamment de points pour obtenir ce cas particulier.

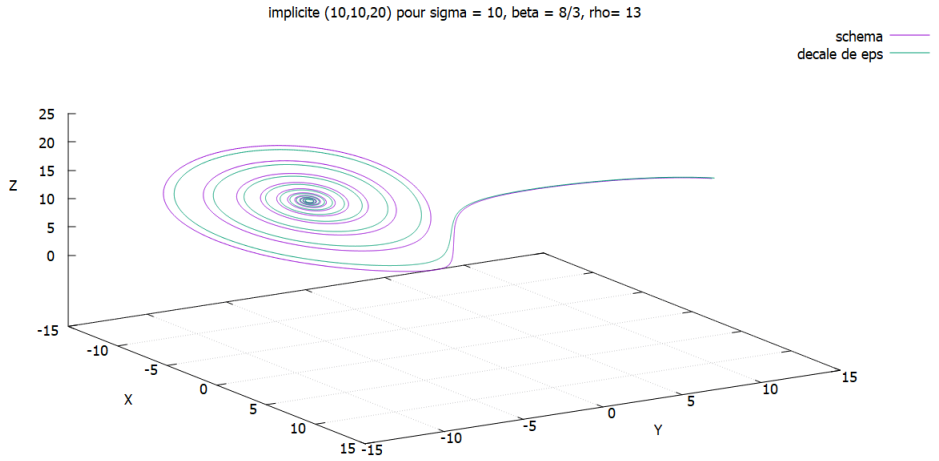


FIGURE 17 – Superposition de graphes avec points initiaux proches par la méthode d'Euler implicite pour des valeurs de $\rho = 13 < \rho_{critique}$

Le dernier graphe ci-dessous montre le cas d'oscillation entre les deux points critiques. Il est intéressant de remarquer que les comportements des solutions avec des points critiques différents ne sont pas identiques. En effet, on remarque un certain décalage, déjà exhibé dans la partie précédente en 1D.
Ce graphe met en évidence l'aspect chaotique du modèle, provenant d'un domaine difficilement prédictible (mécanique des fluides).

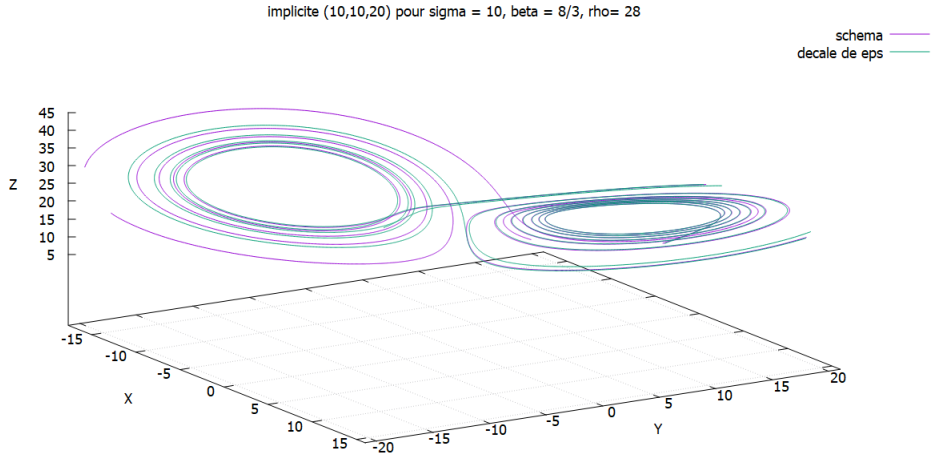
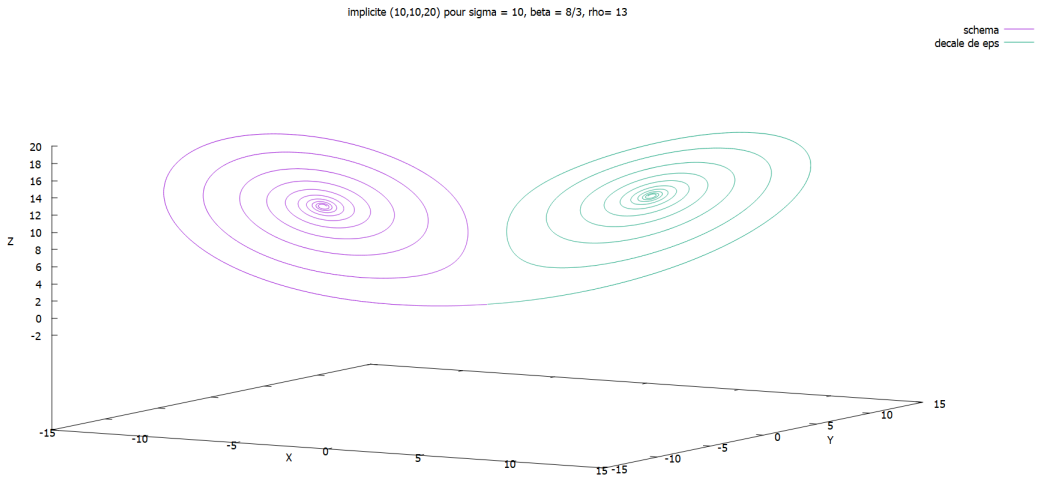
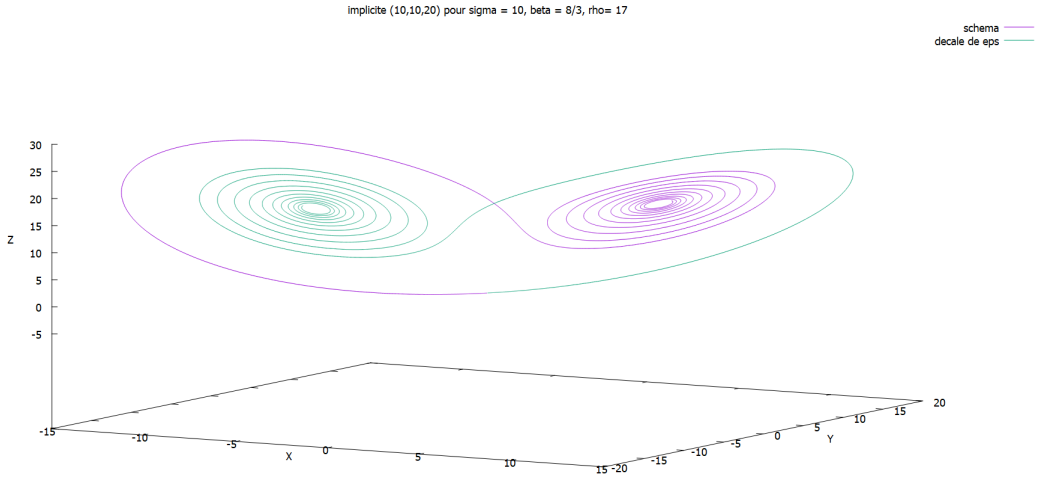


FIGURE 18 – Superposition de graphes avec points initiaux proches pour la méthode d'Euler implicite pour des valeurs de $\rho = 28 > \rho_{critique}$

De plus, comme énoncé dans la partie précédente, si on choisit astucieusement deux points initiaux proches, on peut trouver une convergence mettant en doute la stabilité.



(a) Pour $\rho = 13$



(b) Pour $\rho = 17$

FIGURE 19 – Convergence vers deux points critiques différents pour des points proches initialement

Les deux graphes ont été réalisés avec des valeurs de $\rho < \rho_{critique}$, ce qui assure la convergence vers un des points critiques (pour un nombre d'itération suffisant). Ayant choisi deux points initiaux à $(\pm 0.01, \pm 0.01, \pm 0.01)$, ceux-ci sont respectivement plus proches d'un des deux points critiques. On met en défaut la stabilité du schéma, pour les points situés sur le plan médian aux deux points critiques$(4x\sqrt{\beta(\rho-1)}+4y\sqrt{\beta(\rho-1)}=0)$.

De plus, remarquons qu'en modifiant la valeur de ρ à 17, nous obtenons une convergence, non pas vers le point le plus proche du point initial, mais vers son opposé. Cette particularité n'a pas été mise en évidence

dans la partie théorique, mais mériterait un approfondissement.

5 Conclusion rapide

Nous avons vu que le système de Lorenz se comportait différemment suivant les paramètres ρ σ et β. Les polynômes caractéristiques au point critique ont permis de déterminer des conditions sur les paramètres et un $\rho_{critique}$, une fois franchis les points critiques deviennent répulsifs. Ainsi une fois que les points critiques sont devenus répulsifs le phénomène de double attracteurs chaotiques apparaît. Nous avons mis en exergue l'aspect chaotique des doubles attracteurs, en effet ce phénomène entraîne un schéma instable.

6 Annexe

6.1 Euler explicite

```c
matrix_t* euler_explicite(matrix_t* x0, double T, double N, double sigma, double beta, double rho)
{
    int i;
    double dt = T/N;
    double x, y, z;
    double x1, y1, z1;
    double tn = 0;
    matrix_t * data = define_matrix();

    alloue_matrix(data, N+1, 4);
    x= get_elem(x0, 0, 0);
    y= get_elem(x0, 1, 0);
    z= get_elem(x0, 2, 0);

    set_elem(data, 0, 0, tn);
    set_elem(data, 0, 1, x);
    set_elem(data, 0, 2, y);
    set_elem(data, 0, 3, z);

    for(i = 1; i<=N; i++)
    {
        tn+=dt;
        x1 = x + dt*sigma*(y-x);
        y1 = y + dt*(x*(rho-z) - y);
        z1 = z + dt*(x*y - beta*z);

        x= x1;
        y= y1;
        z= z1;

        set_elem(data, i, 0,tn);
        set_elem(data, i, 1, x);
        set_elem(data, i, 2, y);
        set_elem(data, i, 3, z);

    }


    return data;
}
```

CODE_C/euler_explicite.c

6.2 Euler implicite

On commence ici par définir une fonction intermédiaire appelée "newton_lorenz" qui sera appelée dans "euler_implicite" par la suite :

```
double newton_lorenz(double x_ini, double y_ini, double z_ini,double T, double N, double sigma, double beta, double rho )
{
    int  kmax = 20;
    double y0 = y_ini+10;
    double eps = 1e-15;
    double y_resultat= 0;

    double h= T/N;
    double A= 1+h*sigma;
    double B = 1+h;
    double C= 1+beta*h;

    double f(double y)
    {
        return pow(A,2)*C*y_ini+ A*C*h*rho*x_ini - A*h*z_ini*x_ini + y*(A*C*pow(h,2)*rho*sigma - A*pow(h,2)*sigma*z_ini - pow(x_ini*h,2) -A*A*B*C)-y*y*2*x_ini * sigma*pow(h,3)-pow(y,3)*pow(h*h*sigma, 2);
    }

    double df(double y)
    {
        return (A*C*pow(h,2)*rho*sigma - A*pow(h,2)*sigma*z_ini - pow(x_ini*h,2) -A*A*B*C)-4*x_ini*sigma*pow(h,3)*y- 3 * pow(y*h*h,2)*sigma*sigma;
    }

    y_resultat = newton(kmax, y0, f, df, eps);
    return y_resultat;
}
```

CODE_C/euler_implicite.c

Celle-ci fait appel à l'algorithme de newton pour résoudre l'équation non linéaire en f que l'on a présenté dans la partie 2.

Ensuite, on définit notre méthode numérique implicite :

```c
matrix_t* euler_implicite(matrix_t* x0, double T, double N, double sigma, double beta, double rho)
{
    int i;
    double h = T/N;
    double A= 1+h*sigma;
    double C= 1+beta*h;
    double x, y, z;
    double tn = 0;
    matrix_t * data = define_matrix();

    alloue_matrix(data, N+1, 4);
    x= get_elem(x0, 0, 0);
    y= get_elem(x0, 1, 0);
    z= get_elem(x0, 2, 0);

    set_elem(data, 0, 0, tn);
    set_elem(data, 0, 1, x);
    set_elem(data, 0, 2, y);
    set_elem(data, 0, 3, z);

    for (i = 1; i<=N; i++)
    {
        tn+=h;
        y= newton_lorenz(x,y,z, T, N, sigma, beta, rho);

        x= (x + h*sigma*y)/A;
        z= (z + h*x*y)/C;

        set_elem(data, i, 0,tn);
        set_elem(data, i, 1, x);
        set_elem(data, i, 2, y);
        set_elem(data, i, 3, z);

    }

    return data;
}
```

CODE_C/euler_implicite.c

6.3 Runge Kutta 4

```c
matrix_t* RK4 (matrix_t* x0, double T, double N, double sigma, double beta, double rho)
{
    int i;
    double h = T/N;
    double k12, k22, k32, k42, k11, k21, k31, k41;

    double k13, k23, k33, k43;

    double yn1, yn;
    double xn1, xn;
    double zn1, zn;
    double tn = 0;

    matrix_t * data = define_matrix();
    alloue_matrix(data, N+1,4);

    xn= get_elem(x0, 0, 0);
    yn= get_elem(x0, 1, 0);
    zn= get_elem(x0, 2, 0);

    set_elem(data, 0, 0, tn);
    set_elem(data, 0, 1, xn);
    set_elem(data, 0, 2, yn);
    set_elem(data, 0, 3, zn);

    for(i=1;i<=N;i++)
    {
        tn+=h;
        k11=sigma*(yn-xn);
        k21=sigma*(yn-(xn+h*k11/2));
        k31=sigma*(yn-(xn+h*k21/2));
        k41=sigma*(yn-(xn+h*k31));
        xn1=xn+h*(k11+2*k21+2*k31+k41)/6;

        k12=rho*xn-yn-xn*zn;
        k22=rho*xn-(yn+h*k12/2)-xn*zn;
        k32=rho*xn-(yn+h*k22/2)-xn*zn;
        k42=rho*xn-(yn+h*k32)-xn*zn;
        yn1=yn+h*(k12+2*k22+2*k32+k42)/6;

        k13=xn*yn-beta*zn;
        k23=xn*yn-beta*(zn+h*k13/2);
        k33=xn*yn-beta*(zn+h*k23/2);
        k43=xn*yn-beta*(zn+h*k33);
        zn1=zn+h*(k13+2*k23+2*k33+k43)/6;

        set_elem(data, i, 0, tn);
        set_elem(data, i, 1, xn1);
        set_elem(data, i, 2, yn1);
        set_elem(data, i, 3, zn1);
        yn=yn1;
        xn=xn1;
        zn=zn1;

    }
return(data);

}
```

CODE_C/RK4.c

7 Bibliographie

Histoire de l'attracteur de Lorenz : https ://fr.wikipedia.org/wiki/Attracteur_de_Lorenz
Instabilites hydrodynamiques (bifurcation de Hopf) : https ://hmf.enseeiht.fr/travaux/CD9598/travaux/optmfn/IH/98PA
Détermination de $\rho_{critique}$: https ://cel.archives-ouvertes.fr/cel-00556972/document