

CC2 Méthodes numériques : étude numérique de l'attracteur de Lorenz

Introduction :

Le but de cette étude est d'essayer de mettre en évidence un comportement chaotique numériquement. Les systèmes chaotiques obéissent généralement à des équations différentielles non linéaires très complexes pour lesquelles aucune solution analytique - même approchée - n'est envisageable. Le recours aux méthodes numériques est alors indispensable et se révèle particulièrement fructueux, si bien qu'il a permis la naissance de toute une nouvelle branche de la recherche en physique et en mathématiques encore très active aujourd'hui. La résolution numérique et le tracé de ses équations entraîne la rencontre d'objets fractals exotiques : les attracteurs étranges.

Ce bref projet se restreint à l'étude du plus célèbre de ceux-ci : l'attracteur de Lorenz en forme « d'ailes de papillons ».

I-Mise en contexte théorique et historique:

En 1963, le météorologue Edward Lorenz (1917-2008) étudia un modèle thermodynamique de l'atmosphère et de l'océan obtenu par simplification des équations de Navier-Stokes dans le but d'étudier des phénomènes de convection. Il aboutit ainsi à un système d'équations différentielles décrivant la dépendance et l'évolution temporelle des différentes variables thermodynamiques dans l'espace de phase. Ce système s'écrit comme suit :

$$\frac{dx}{dt} = s(y - x); \quad \frac{dy}{dt} = rx - y - xz; \quad \frac{dz}{dt} = xy - bz$$

Nous admettrons que ces équations peuvent s'appliquer à notre modèle d'atmosphère, leur obtention et démonstration dépassant largement le cadre de ce projet. x , y et z sont nos variables thermodynamiques telles que :

- x est proportionnel à l'intensité du mouvement de convection
- y est proportionnel au gradient de température horizontal
- z est proportionnel au gradient de température vertical

s et b sont des constantes thermodynamiques et mécaniques traditionnellement notées σ et β que nous fixerons – comme l'a fait Lorenz – à $s=10$ et $b=8/3$.

r , généralement noté ρ , est le nombre de Rayleigh. C'est le paramètre de contrôle de notre système : il déterminera - comme nous le verrons- l'apparition ou non du régime chaotique.

Nous ne savons encore pas bien à quoi nous attendre avec ce curieux système d'équations. Un peu de magie nous attend lors de sa résolution numérique et c'est d'ailleurs la première chose que fit Lorenz avec l'un des premiers calculateurs : le « Royal McBee LGP-300 ».

II-Étude numérique et interprétation des résultats :

a) le choix de la méthode :

Comparé à nos ordinateurs portables, le Royal McBee a l'air bien ridicule aujourd'hui et pourtant, le plus important reste de déterminer la méthode à utiliser pour résoudre notre système d'équations. La première idée a été de les résoudre simplement avec Euler. Cette méthode étant réputée source d'erreurs, il nous faut un moyen de juger si sa précision sera suffisante pour notre étude. Nous l'avons donc comparée à la méthode de Runge Kutta 4 –bien plus précise- en traçant les deux solutions obtenues avec les mêmes conditions initiales pour un nombre de points relativement petit ($n=10\,000$).

Rappelons pour cela la logique deux méthodes à travers l'exemple de la résolution de la première équation de l'attracteur de Lorenz (la résolution des deux autres équations suit exactement la même logique en parallèle {voir les programmes en Annexe}):

Euler : Prenons notre première équation en utilisant la définition de la dérivée :

$$\frac{dx}{dt} = \frac{x(t+dt) - x(t)}{dt} = s(y(t) - x(t)) \quad \Leftrightarrow \quad x(t + dt) = x(t) + s(y(t) - x(t))dt$$

Dont la résolution se traduit en algorithme comme : (en notant implicitement que les $y[i]$ sont calculés simultanément avec la même méthode)

- dt = pas de temps (définissant la précision)
- n = nombre de points de calcul
- x = tableau de $n+1$ cases (pour les conditions initiales)
- $x[0] = c$ (condition initiale à choisir)
- pour i allant de 1 à n :

$$x[i + 1] = x[i] + (s*(y[i] - x[i]) * dt) \quad (\text{avec } s \text{ fixé})$$

On « remplit » ainsi une à une toutes les cases de $x[i]$

- tracer x en tout point

RK4 : La méthode de Runge Kutta 4 équivaut à celle d'Euler mais en ajoutant 4 ordres de correction en calculant la fonction en quatre points intermédiaires et en faisant la moyenne des pentes. Cette méthode est ainsi beaucoup plus précise, (et demande ainsi beaucoup plus de temps à l'ordinateur) l'erreur totale accumulée étant seulement de l'ordre de dt^4 . On associe ainsi à chaque variable l'équation :

$$x(t + dt) = x(t) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) + O(dt^5)$$

Pour alléger l'écriture on notera ici $s(y-x) = f(x,y)$, on a alors :

- $k_1 = f(x,y)dt$
- $k_2 = f(x+k_1/2, y+K_1/2)dt$
- $k_3 = f(x+k_2/2, y+K_2/2)dt$
- $k_4 = f(x+k_3, y+K_3)dt$

Où les K_i sont déterminés parallèlement de la même manière avec l'équation de dy/dt .

Ce traduisant en algorithme par (similaire à Euler avant la boucle) : (des lettres différentes ont été utilisées dans le programme source pour des raisons de commodité)

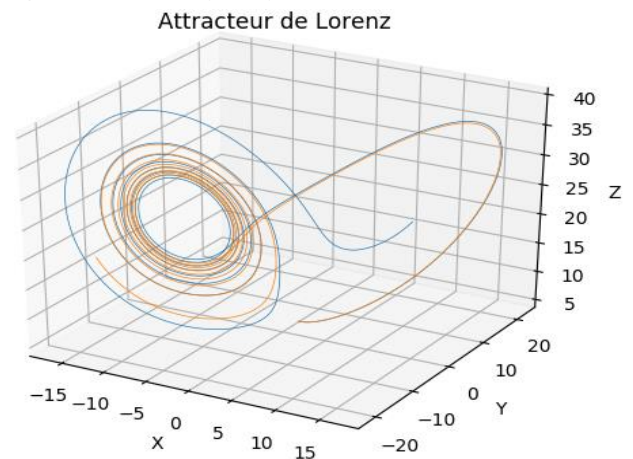
- $f(x,y) = s*(x-y)$
- $k_1 = dt*f(x[i], y[i],)$
- $k_2 = dt*f(x[i]+0.5*k_1, y[i]+0.5*K_1)$
- $k_3 = dt*f(x[i]+0.5*j_2, y[i]+0.5*K_2)$
- $k_4 = dt*f(x[i]+j_3, y[i]+K_3)$
- $x[i+1] = x[i] + (1/6) * (k_1 + (2*k_2) + (2*k_3) + k_4)$
- tracer x

Lors de l'étude de deux trajectoires nous demanderons également à notre algorithme d'afficher la distance finale séparant les deux parcours dans l'espace des phases pour estimer leur déviation $d = \sqrt{\sum (x'_i{}^2 - x_i^2)}$:

- $d = \text{sqrt}((x2[n]-x[n])**2 + (y2[n]-y[n])**2 + (z2[n]-z[n])**2)$
- afficher d

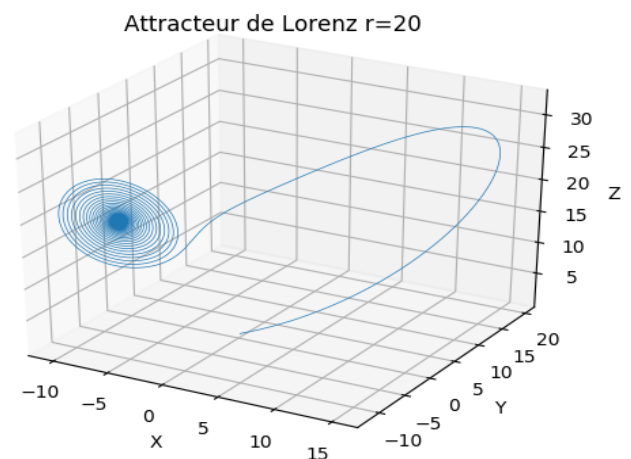
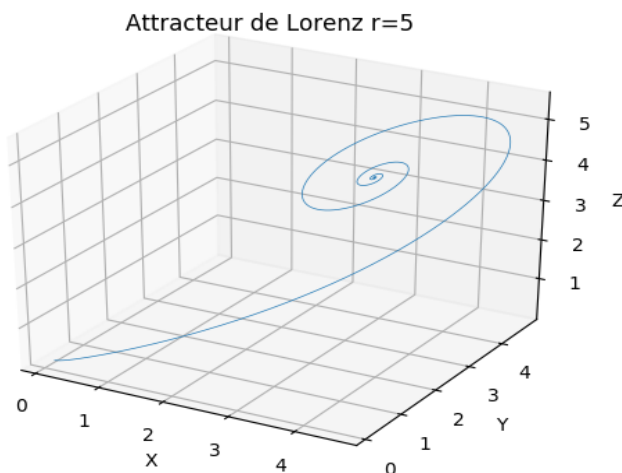
Non sans fierté, nous exécutons les deux algorithmes simultanément avec nos trois variables et les mêmes conditions initiales pour une étude comparative et obtenons la figure suivante : (r=28)

Essayons d'éviter de nous interroger sur la curieuse forme qui se dessine pour le moment et regardons d'abord les deux courbes. Euler est en bleu et RK4 en orange. Partant avec exactement les mêmes conditions initiales en bas à droite, les trajectoires dévient jusqu'à ne plus être comparables. On voit bien que la courbe bleue s'arrête en haut à droite alors que l'orange au milieu à gauche. L'algorithme affiche d=39.62 !! L'explication est que la méthode d'Euler accumule beaucoup trop d'erreurs pour être acceptable dans notre modélisation. La courbe ainsi obtenue n'est qu'à peine ressemblante à la solution exacte comme nous le verrons dans la suite de l'étude ...



b) r comme paramètre de contrôle et apparition d'un régime chaotique

Connaissant désormais la bonne méthode, nous allons maintenant tracer la solution obtenue grâce à notre algorithme RK4 en augmentant progressivement r avec $dt = 0.001$ et $n = 100\,000$ afin d'étudier son influence



Nous faisons partir notre système du point de coordonnées [0.1,0.1,0.1].

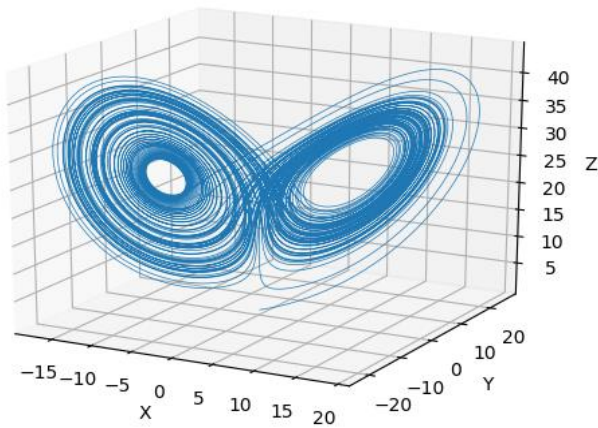
On constate que, pour r allant de 1 à 5, la trajectoire est une simple droite légèrement courbée dans l'espace des phases qui s'arrête en un point d'équilibre. Pour r allant de 5 à 24, elle se met désormais à tourner en spirale autour du point d'équilibre avant de l'atteindre. Plus r sera grand plus la spirale sera resserrée. Voilà deux comportements tout à fait classiques et attendus : notre système évolue, et finit par atteindre sagement un équilibre stable. Comparablement à une bille tournant dans un bol.

Quelque chose de curieux commence à se produire et s'amplifie en augmentant r ... mais c'est au point critique $r=25$ que notre trajectoire prend subitement une forme caractéristique d'ailerons de papillon. La figure ainsi formée n'est plus une figure géométrique habituelle, elle a une dimension fractionnaire entre deux et trois. Ce genre d'objets appelés fractales a une infinité de propriétés fascinantes qui justifient à elles seules la création de toute une branche très riche des mathématiques sur laquelle nous ne nous étendrons

(malheureusement) pas ici. Notons tout de même que les fractales décrivent de très nombreux objets physiques et biologiques dont l'étude serait tout simplement impossible sans méthode numérique.

Le parcours de notre système dans l'espace des phases est ainsi devenu complètement chaotique : Il tourne inlassablement autour de deux points d'équilibre en passant d'une aile à l'autre de manière tout à fait imprévisible et ne trouve pas de coordonnées finales précise et prévisible sur laquelle s'arrêter. Augmenter r ne change désormais plus le motif. Lorenz lui-même étudié le cas où $r=28$.

Attracteur de Lorenz $r=25$



Sur la figure jointe, le point de départ est bien visible en bas à droite. Plusieurs essais avec des conditions initiales très éloignées montrent que peu importe d'où partira le système il rejoindra toujours l'attracteur. Il tournoie donc chaotiquement autour de deux points d'équilibre mais respecte toujours une structure globale cohérente et identique : chaos n'est pas synonyme de n'importe quoi.

Ces systèmes ont pourtant quelque chose de vraiment particulier qui permet de les identifier hormis leurs trajectoires si exotiques, et c'est ce que nous allons tenter de mettre en évidence dans la prochaine section ...

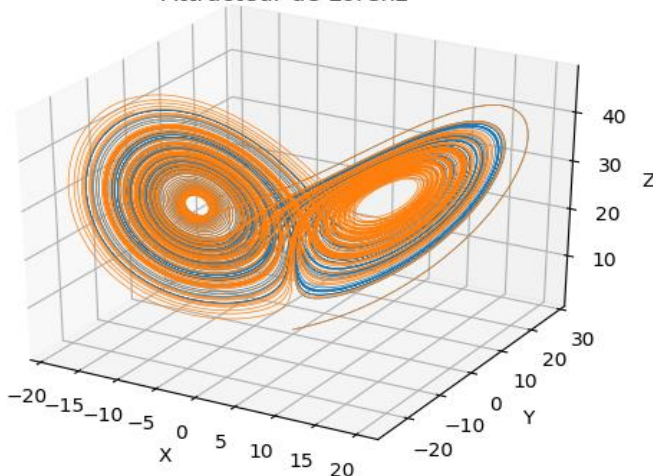
A noter que la même démarche avec Euler ne conduit pas aux mêmes paliers pour les valeurs de r . Il n'y a également plus de r critique clairement définis entraînant l'apparition du régime chaotique.

c) Le battement d'aile d'un papillon ...

Quelque chose de vraiment fascinant se produit si nous comparons deux trajectoires partant de conditions initiales très proches. Pour $r=28$ nous prenons pour la première trajectoire les coordonnées de départ $[0.01, 0.01, 0.01]$ (bleu) et $[0.001, 0.001, 0.001]$ (orange) pour la deuxième. Ces paramètres de départ étant très proches ($d_0=0.01$) nous nous attendons à ce que les deux trajectoires soient quasiment identiques et que finalement $d \approx 0.0$ comme pour tout système classique. Nous lançons donc le programme et obtenons la figure ci-dessous :

Il nous affiche également $d=25.42$. Cela paraît beaucoup. Est-ce une erreur ? Pour ce faire une idée nous relançons notre système avec les coordonnées initiales $[1, 1, 1]$ et $[2, 2, 2]$ en nous attendant à une plus grande divergence ... $d=12.32$!

Attracteur de Lorenz



Quid si nous rapprochons encore nos points de départ ? Nous prenons alors $[0.001, 0.001, 0.001]$ et $[0.002, 0.002, 0.002]$... $d=36.19$!!!!

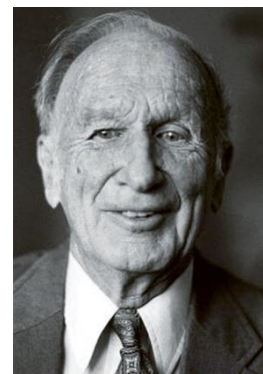
La même simulation avec $r=20$ conduit à $d=1.07e-6$. Résultat attendu pour un système classique. Ce n'est donc pas notre programme qui fonctionne mal.

Mais que se passe-t-il ? Il est impossible de déterminer avec certitude la trajectoire empruntée par notre système : il empruntera un chemin complètement différent pour un point de départ distant de $d_0=0.001$! Voilà la vraie caractéristique d'un système chaotique : l'imprédictibilité.

Admirez ici l'illustration parfaite de l'effet Papillon dont parlé Lorenz lui-même (photo plus bas) dans sa fameuse phrase : « Le battement des ailes d'un papillon au Brésil peut-il provoquer un ouragan au Texas ? ». Les systèmes chaotiques empêchent toutes précisions fiables à long terme concernant leur évolution, au grand dam des météorologues, directement concernés par des systèmes de ce type.

Pourtant, les lois qui les dirigent sont purement classiques et déterministes, contrairement aux systèmes quantiques fondamentalement probabilistes. Ainsi, le physicien voulant tout prévoir et maîtriser se voit remettre à sa place par ces deux grandes théories physiques du vingtième siècle. Le vertige ce fait sentir lorsque l'on ose imaginer la physique quantique chaotique ...

A titre de comparaison, la même analyse avec Euler conduit à des trajectoires stables qui divergent peu avec les conditions initiales enlevant ainsi à l'attracteur tout son intérêt. Le fait que les comportements chaotiques soient « lissés » par Euler, supposés pourtant diverger, est inattendu. Euler nous donne en fait une version « grossière » et linéaire de l'attracteur réel en ignorant toutes les petites variations d'ordre supérieures à un.



Conclusion :

Nous venons de découvrir que le concept de hasard – au sens premier du terme- pouvait avoir sa place dans des domaines aussi rigoureux que les sciences dures. Par là nous entendons que des systèmes se révèlent être tout simplement imprédictibles et c'est là le vrai secret de la physique du chaos. L'exemple très abstrait du modèle de convection de Lorenz n'est qu'un exemple parmi tant d'autre : le système solaire est lui-même un exemple de système extrêmement chaotique alors qu'il est considéré depuis des siècles comme une horloge parfaite. Le chaos est omniprésent dès que les approximations tombent pour se rapprocher du réel.

Ce projet a été chronophage de par les nombreuses difficultés rencontrées pour trouver la bonne méthode et réaliser que le problème de fond empêchant l'apparition du chaos venait de la méthode d'Euler. La complexité des résultats à interpréter cachant l'évidence en un premier temps. Il fut également difficile de se restreindre à 5 pages sur un sujet aussi prolifique et j'ai dû aller à l'essentiel.

Le résultat semble satisfaisant et méritait de loin toutes ces peines. Des améliorations pourraient tout de mêmes être envisagées comme l'utilisation d'une méthode à pas variable Rk4 pour pouvoir se permettre de réduire encore le pas et les écarts entre les conditions initiales. Une animation montrant l'évolution de la trajectoire dans l'espace des phases en temps réel aurait également pu être possible mais cela dépassait de loin le niveau de la présente étude.

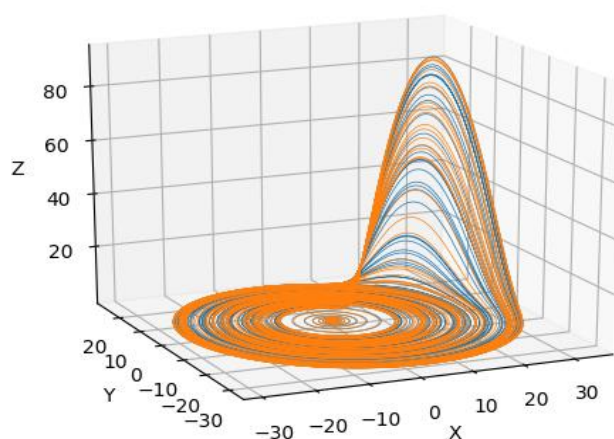
Ce travail a été extrêmement enrichissant tant sur le plan de la physique que des méthodes numériques voir même de la philosophie. La théorie du chaos et ses attracteurs méritent qu'on s'y attarde pour bien des aspects. L'aspect esthétique des attracteur et l'emblématique forme de celui de Lorenz entraîne la fascination de beaucoup, expliquant qu'il soit si souvent un sujet de la vulgarisation en physique et en mathématiques. Pourtant, une physique extrêmement complexe se cache à l'arrière-plan rendant l'accès aux documents spécialisés souvent très difficile. Nous n'avons fait qu'effleurer ici l'immensité ouverte par la percée du chaos dans les méthodes numériques et l'informatique, sans parler du passionnant domaine de la géométrie fractale lui aussi esthétique et marquant fortement l'esprit des non scientifiques : du chou Romanesco aux fractales de Mandelbrot.

Pour conclure, je ne peux me retenir de présenter brièvement « l'attracteur de Rössler » crée par Otto Rössler comme une version simplifiée de celui de Lorenz. En effet, celui-ci diverge grandement même sous Euler. Aucune interprétation physique n'en sera donnée. Son système d'équation s'écrit, avec x,y et z nos variables, a et b constants et fixés à 0.1 et c notre paramètre de contrôle :

$$\frac{dx}{dt} = -y - z; \quad \frac{dy}{dt} = x + ay; \quad \frac{dz}{dt} = b + z(x - c)$$

Voici la figure obtenue pour n=50000, dt=0.01 avec deux conditions initiales séparés de 0.01. On comprend alors immédiatement la fascination crée par ces « ovni mathématiques ».

Attracteur de Rossler



ANNEXE :

Voici le programme utilisé pour comparer la méthode d'Euler et RK4. Pour le tracé de différentes trajectoires c'est la partie RK4 qui a été dupliquée. Rössler a été résolu avec Euler uniquement.

```
Import numpy as np #numpy pour les tableaux
import matplotlib.pyplot as plt #matplotlib pour les tracés
from mpl_toolkits.mplot3d import Axes3D #fonctions pour la 3D
```

```
#variables du problème r, dt et n et les deux constantes b et s
```

```
s=10
r=28
b=8/3
```

```
dt = 0.001
n = 7000 #100000 pour le tracé des trajectoires complètes
```

```
# x,y,z et x2 y2 z2 sont des tableaux vide de n+1 éléments
```

```
x = np.empty((n + 1))
y = np.empty((n + 1))
z = np.empty((n + 1))
```

```
x2 = np.empty(n + 1)
y2 = np.empty(n + 1)
z2 = np.empty(n + 1)
```

```
#conditions initiales
```

```
x[0], y[0], z[0] = (1,2,5)
x2[0], y2[0], z2[0] = (1,2,5)
```

```
#Euler
```

```
for i in range(n):
    x[i + 1] = x[i] + (s*(y[i] - x[i]) * dt)
    y[i + 1] = y[i] + ( (r*x[i] - y[i] - x[i]*z[i])* dt)
    z[i + 1] = z[i] + ( (x[i]*y[i] - b*z[i]) * dt)
```

```
#RK4
```

```
#dx/dt
```

```
def U(x,y):
    return s*(y - x)
```

#dy/dt

```
def V(x,y,z) :  
    return r*x - y - x*z
```

#dz/dt

```
def W(x,y,z) :  
    return x*y - b*z
```

#les J sont les k associés à x, les K à y et les M à z. On a donc 12 k à définir

```
for i in range(n):  
    J1 = dt*U(x2[i], y2[i],)  
    K1 = dt*V(x2[i], y2[i], z2[i])  
    M1 = dt*W(x2[i], y2[i], z2[i])  
    J2 = dt*U(x2[i]+0.5*J1, y2[i]+0.5*K1)  
    K2 = dt*V(x2[i]+0.5*J1, y2[i]+0.5*K1, z2[i]+0.5*M1)  
    M2 = dt*W(x2[i]+0.5*J1, y2[i]+0.5*K1, z2[i]+0.5*M1)  
    J3 = dt*U(x2[i]+0.5*J2, y2[i]+0.5*K2)  
    K3 = dt*V(x2[i]+0.5*J2, y2[i]+0.5*K2, z2[i]+0.5*M2)  
    M3 = dt*W(x2[i]+0.5*J2, y2[i]+0.5*K2, z2[i]+0.5*M2)  
    J4 = dt*U(x2[i]+J3, y2[i]+K3)  
    K4 = dt*V(x2[i]+J3, y2[i]+K3, z2[i]+M3)  
    M4 = dt*W(x2[i]+J3, y2[i]+K3, z2[i]+M3)  
    # résolution de x(t), y(t) et z(t) :  
    x2[i+1] = x2[i] + (1/6) * (J1 + (2*J2) + (2*J3) + J4)  
    y2[i+1] = y2[i] + (1/6) * (K1 + (2*K2) + (2*K3) + K4)  
    z2[i+1] = z2[i] + (1/6) * (M1 + (2*M2) + (2*M3) + M4)
```

#tracé en 3D

```
fig = plt.figure()  
ax = fig.gca(projection='3d')  
  
ax.plot(x, y, z, lw=0.5) #tracer trajectoire Euler  
ax.plot(x2, y2, z2, lw=0.5) #tracer trajectoire RK4  
ax.set_xlabel("X ") #nom des axes  
ax.set_ylabel("Y")  
ax.set_zlabel("Z ")  
ax.set_title("Attracteur de Lorenz ") #titre
```

```
plt.show() #afficher
```

#afficher les coordonnées finales :

```
print(x[n], y[n] , z[n])  
print(x2[n], y2[n] , z2[n])
```

#calculer la distance finale d :

```
d = np.sqrt((x2[n]-x[n])**2 +(y2[n]-y[n])**2 + (z2[n]-z[n])**2)  
print('d=',d)
```