

Luguy GONTHIER / Stephan KWAN



# Rapport Projet S3

ILLUSTRATION TRAJECTOIRE D'UN POINT

INFORMATIQUE | 2018-2019

# Table des matière

## **I- Introduction**

## **II- Corps du projet**

i- Arborescence du projet

ii- Code Source

iii- Fichier principal

## **III- Difficultés**

## **IV- Conclusion**

## **I- Introduction**

L'objectif de notre projet est de donner une représentation graphique de la trajectoire d'un point définie par 3 équations dont chacune caractérise la vitesse du point dans l'espace en fonction de sa position, du temps  $t$  et durant une période  $T$ . Dans notre cas, les équations de Lorenz sont adaptées à notre étude.

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = x(\rho - z) - y \\ \frac{dz}{dt} = xy - \beta z \end{cases}$$

Cependant, les paramètres initiaux seront à définir par l'utilisateur du programme, c'est à dire la position définie par  $x, y, z$ , le point étant supposé initialement stationnaire, à l'instant  $t$  initial, pour un temps d'arrêt  $T_{\max}$ . De même, il faudrait créer un paramètre  $dt$  qui sera l'intervalle de temps entre chaque points, plus  $dt$  sera petit plus on pourra avoir une représentation se rapprochant davantage de la trajectoire réelle du point étudié (le nombre de point sur le graphique sera  $T_{\max}/dt$ ).

Dans la suite, nous allons entrer dans les détails, à savoir le corps de notre projet, les différentes fonctions dans chaque librairies et les fichiers à utiliser.

## **II- Corps du projet**

### **i- Arborescence du projet**

L'organisation étant primordiale, il est important de répartir notre projet dans différents dossiers/sous-répertoires, à savoir :

-bin : le dossier contenant les fichiers exécutables (fichier.exe, fichier.o)

-doc : le dossier contenant les fichiers utiles pour le code source et le présent rapport

-include : le dossier contenant les fichiers d'entêtes

-lib : le dossier contenant toutes les bibliothèques que l'on va répartir selon leurs utilités (fichier.h)

-lorentz.dat : un fichier qui stocke les données sous formes de temps, de  $x$ ,  $y$ ,  $z$ .

-Makefile : un fichier utilisé par le programme make pour pouvoir effectuer un ensemble d'actions tel que la compilation, l'archivage, la mise à jour.

-src : le dossier contenant la base même de notre programme, à savoir le code source (les fichier.c), un fichier principale (Lorentz.c), exécutable, qui va utiliser les fonctions définies au préalable et stocké en tant que librairie dans le dossier lib.

## ii- Code source

Le code source sera composé de deux principales bibliothèques contenant un ensemble de fonctions, utiles au programme principal.

### 1) Initialisation

Comme dit précédemment, les variables devront être entrées par l'utilisateur du programme, on aura donc une fonction qui récupérera :

- la position initiale :  $x, y, z$
- les paramètres :  $\sigma, \rho$ , et  $\beta$ , fixés pour chaque trajectoire calculée
- l'incrément :  $dt$
- le temps d'arrêt:  $T_{max}$

Cette fonction aura en entrée des floats, obtenus avec la fonction `scanf` pour chaque paramètres cités.

### 2) Position et vitesse

Cette bibliothèque contiendra les fonctions `calcul.h` de positions et vitesse, relatifs au système de Lorenz à un instant  $t+dt$ .

Ces fonctions prendront en entrées les positions et les paramètres et calculeront les différentes position en fonction du temps grâce aux équations différentielles de Lorenz données dans l'énoncé. Cette fonction renvoie alors 3 valeurs pour chaque coordonnée et les stocke dans un fichier Base qui servira de base de donnée.

### 3) Bibliothèques

Seront utilisées plusieurs bibliothèques dont :

- `stdio.h`, qui contient ce qui est nécessaire aux entrées-sorties.
- `calcul.h`, qui contient toutes les fonctions nécessaires aux calculs des coordonnées.

Les fonction.o présents dans les bibliothèques seront compilées en fonction.o contenus dans la librairie.a.

### iii- Fichier principal

Le programme principal peut se résumer en trois mots, initialisation, calculs et affichage. Ainsi, dans le sous-répertoire src, nous allons créer un fichier principal Lorentz.c dans lequel sera écrit le programme. On devra y inclure les librairies définies précédemment pour un soucis de clarté. Ce programme se composera d'une boucle allant de  $t=0$  à  $t=T_{max}$  avec  $dt$  comme intervalle entre chaque boucle, il y aura donc  $T_{max}/dt$  valeurs. Au sein de cette même boucle, se trouvera la fonction position, nous permettant ainsi d'obtenir la valeur de la position pour chaque incrémentation et au fur et à mesure les stocker dans le fichier Base.

On utilisera ici la fonction gnuplot, qui nous permettra, à partir des points obtenus dans le fichier Base, de tracer la courbe attendue.

Le fichier MakeFile sera lui, crucial, il permettra de compiler automatiquement l'ensemble des fichiers et effectuer les tests du programme. Seront ainsi indiqués dans le MakeFile, les commandes de terminal, les fichiers.

## III- Difficultés

Les difficultés rencontrées sont nombreuses, mais les principales concernent l'utilisation du MakeFile et la fonction gnuplot.

## IV- Conclusion

Nous allons créer deux librairies dans lesquelles seront condensées plusieurs fonctions utiles au programme final, dans un soucis de clarté et lisibilité. La difficulté résidera dans la bonne déclaration des fonctions qu'on réutilisera tout au long du projet. Le programme principal étant celui qui nous donnera la courbe attendue, il est important de réussir toutes les démarches évoquées. Ce rapport sera amélioré au cours de l'évolution du projet, celui-ci contient pour l'instant, uniquement des notions maîtrisées.