

Luguy Gonthier / Stephan Kwan



Rapport final

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- Include

iv- 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.

$$\left\{ \begin{array}{lcl} \frac{dx}{dt} & = & \sigma(y - x) \\ \frac{dy}{dt} & = & x(\rho - z) - y \\ \frac{dz}{dt} & = & xy - \beta z \end{array} \right.$$

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 points 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 librairie et les fichiers à utiliser.

II- Corps du projet

i- Arborescence du projet

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

-bin : le dossier contenant le fichier exécutable (lorenz.exe)

-doc : le dossier contenant les fichiers pdf, à savoir le pré-rapport, le rapport final et les instructions d'utilisation.

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

-lib : le dossier contenant la librairie statique libmain.a.

-data : un dossier contenant le fichier gnuplot_param qui contient le paramétrage de gnuplot, le fichier position.dat qui sera le fichier où le programme écrira l'ensemble des coordonnées du point a chaque instant.

-Makefile : un fichier utilisé qui pour effectuer toutes les compilations afin de mettre à jour tout le projet.

-src : le dossier contenant la base même de notre programme, à savoir le code source (les fichier.c).

ii- Code Source

Le code source sera composé de 4 librairies contenant un ensemble de fonctions, utiles au programme principal (main.c).

1- initialisation.c

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

- les paramètres : σ , ρ , et β , fixés pour chaque trajectoire calculée
- l'incrément : dt
- le temps d'arrêt: T_{max}

Cette fonction aura en entrée une variable de type Parametre_s definie avec struct comme étant un ensemble de 5 floats.

2- fichier.c

Le fichier contenant 2 fonctions nécessaire à l'écriture des coordonnées dans un fichier.

- La première Creation_Fichier permet de créer un fichier.
- La deuxième Ecriture_Fichier permet d'écrire dans le fichier et prends en entrée un fichier et les trois coordonnées et le temps T du système.

3- calcul.c

Le fichier contenant les initialisation des positions initiales et le calcul des nouvelles positions.

- Creation_Coordo_Initial qui nous demande les positions initiales du système, et prend en entrée une variable de type coordo qui est une structure de trois floats.
- Position_Suivante qui calcul la nouvelle position du point et prend en entrée les anciennes coordonnées et le paramètre dt .

4- gnuplot.c

Le fichier qui contient la fonction de traçage sur gnuplot et prend en entrée un fichier.

iii- Include

Pour le bon fonctionnement de notre programme ainsi que les fichiers sources, nous utiliserons des includes.

Chacun des fichiers cités précédemment possèdera un include qui porte le même nom avec une terminaison différente, les includes se terminent par un.h.

Chaque includes contient les en-tête des fonctions contenant dans les fichiers sources.

Sauf les includes :

-calcul.h , qui contient la definition du type Coordonnees_s.

-initialisation.h, qui contient la définition du type Parametre_s.

iv- Fichier principal

Le programme principal peut se résumer en trois mots, initialisation, calculs et affichage. Ainsi, on le code dans le fichier principal nommé main.c appellera les fonctions qui initialiseront notre système et ainsi ces fonctions demanderont directement les paramètres à taper ainsi que les coordonnées initiales. On devra y inclure les includes définies précédemment.

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_suivante et Ecrire_Fichier, nous permettant ainsi d'obtenir la valeur de la nouvelle position pour chaque incrémentations et au fur et à mesure les stocker dans le fichier position.dat.

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

III- Difficulté rencontré

Les principales difficultés rencontrées sont surtout au niveau de l'utilisation des structures, pire, au niveau des pointeurs sur structure qui aboutissaient sur un segmentation fault. Mais également sur l'utilisation de gnuplot qui ne voulait pas afficher notre fonction, surtout nous qui avions voulu y mettre des couleurs.

Mais encore, sur l'écriture sur fichier, le fait de ne pas réussir à écrire les coordonnées dans le fichier.

IV- Conclusion

Nous allons créer deux bibliothèques 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.

