

# Projet en langage C : Modélisation de la trajectoire d'un point

Documentation :

## 1 Introduction

Dans ce projet, un simulateur de systèmes dynamiques a été implémenté, incluant des systèmes tels que Lorenz, un oscillateur simple et une spirale. L'utilisateur de ce système peut définir des conditions initiales, simuler des trajectoires et visualiser les résultats en 3D à l'aide de Gnuplot avec un affichage de la vitesse par couleur. De plus, le choix d'un système personnalisé est possible avec la mise à jour de la vitesse via la notation polonaise inversée (NPI).

## 2 Structure du Projet :

Le projet a été divisé en six fichiers principaux :

- **lorenz.c** : Définition des fonctions pour interagir avec les systèmes dynamiques.
- **lorenz.h** : Déclaration des structures et des fonctions.
- **main.c** : Gestion de l'exécution du programme en appelant les différentes fonctions et en gérant l'interaction avec l'utilisateur.
- **gnuplot.i.c** : Les définitions des fonctions qui initient **gnuplot**, génèrent des graphiques à partir des données, et ferment **gnuplot**.
- **gnuplot.i.h** : Définition des interfaces et structures pour interagir avec **gnuplot**
- **Makefile** : Automatiser le processus de compilation.

Les fichiers **gnuplot.i.c** et **gnuplot.i.h** font partie du module **gnuplot.i** disponible sur <https://github.com/mithodin/gnuplot.i>.

## 3 Structure de données :

**Coord** : Elle représente les coordonnées d'un point dans un espace à trois dimensions (x,y,z).

**Params** : Elle représente les paramètres utilisés par les systèmes dynamiques (sigma, beta, rho).

**SimSettings** : Elle représente les paramètres relatifs à la simulation (dt : incrément de temps et tmax : temps maximal de simulation)

**SysDynamique** : Elle représente la structure commune des systèmes dynamiques, et elle contient les fonctions d'initialisation, de mise à jour du système ainsi que les paramètres associés.

## 4 Fonctions principales :

**init(Coord \*point)** : L'utilisateur saisit la position initiale du point dans l'espace tridimensionnel.

**ask\_parametres\_lorentz()** : L'utilisateur saisit les paramètres (sigma, beta, rho) pour les systèmes dynamiques.

**ask\_simulation\_settings()** : L'utilisateur saisit les paramètres de simulations.

**actualiser\_nomsysteme(Coord \*point, Params \*para, double dt)** : Mise à jour des coordonnées d'un point dans un système dynamique en fonction de ses équations différentielles respectives.

**création\_sys(SysDynamique \*systeme, void (\*init\_sys)(Coord\*), void (\*actualiser\_sys)(Coord\*, Params\*, double), Params\* params)** : Initialisation d'un système dynamique, et affectation à une fonction d'initialisation et de mise à jour.

**choisir\_sys(SysDynamique \*systeme, Params \*params)** : Sélection par l'utilisateur du système dynamique à simuler.

**generer\_fichier(char \*nom\_fichier, void (\*fct\_actu)(Coord\*, Params\*, double), Coord \*pt, Params \*params, SimSettings \*sim)** : Il génère un fichier contenant les données de simulation pour les coordonnées x,y,z et la vitesse d'un point à chaque instant de la simulation.

**ask\_notation\_pol\_inv()** : L'utilisateur saisit une notation polonaise inversée (NPI).

**eval\_npi(const char\* np\_i, double x, double y, double z)** : Evaluation de l'expression en notation polonaise inversée en utilisant (x, y,z) et retourne le résultat de l'expression calculée.

**gnuplot(char\* nom\_fichier)** : Affiche les résultats graphiquement depuis le fichier de données par la fonction précédente, en représentant les points avec des couleurs correspondant à leur vitesse.

**gnuplot\_interface(char \*nom\_fichier)** : Elle fait la même chose que Gnuplot mais en utilisant le module `gnuplot.i`.

## 5 Exécution du programme

L'exécution du programme dans `main.c` se déroule comme suit :

1. L'utilisateur saisit les paramètres du système dynamique.
2. Configuration de la simulation
3. Choix du système dynamique à simuler.
4. Mise à jour des vitesses (Si l'utilisateur choisit le système personnalisée)  
si non il saisit les coordonnées (x,y,z)
6. Un fichier de données est généré, après la simulation.
7. Un graphique est généré à l'aide de Gnuplot.