

Premier Rapport Du Projet Informatique

LIBRAIRIES

FRACTION

Librairie contenant tous les outils nécessaires pour travailler sous forme de fractions. Création d'un type et de fonctions de calcul associées.

(type)fraction :

Type reposant sur deux entiers p et q et représentant la fraction p/q .

(fct)nouvelle_fraction :

fraction nouvelle_fraction(int p, int q)

Crée un nouvel objet de type fraction à partir de deux entiers.

(fct)valeur_explicite :

double valeur_explicite(fraction a)

Retourne la valeur de la fraction sous forme décimale.

(fct)somme_fraction :

fraction somme_fraction(fraction a, fraction b)

Renvoie la fraction résultante de la somme de deux fractions.

Utilise :

nouvelle_fraction(int p, int q)

(fct) soustraire, multiplier, diviser construites de la même manière.

COORD

Librairie regroupant des outils pour travailler avec des coordonnées x,y,z .

(type)coord :

Type défini par trois fractions représentant un triplet de coordonnées (x,y,z) .

(fct)new_coord :

coord new_coord(fraction x, fraction y, fraction z)

Crée un nouvel objet de type coord à partir de trois fractions.

ENTREE

Librairie qui gère tous les échanges entre le programme et l'utilisateur. Vérifie que les entrées correspondent aux attentes du programme.

Pour le contenu, nous utiliserons la librairie du même nom faite en TP.

LORENTZ

Librairie qui gère l'évolution et le traitement d'un système de Lorentz

(fct)paramètre :

*void param(fraction * parametre, coord * position)*

Rentre, dans un tableau de fraction et un triplet de coordonnées, les paramètres initiaux du système de Lorentz qui va être simulé (à savoir sigma, rho, beta, Tmax, dt et xinit, yinit, zinit)

Utilise :

lire_entier()

(fct)calcul position :

*void calcul(fraction * parametre, coord * position)*

Calcule les nouvelles coordonnées du point au temps t+dt. Position directement modifiée dans "position" donc utilisation de variables temporaires (x,y,z) lors des calculs.

(fct)algo_lorentz :

void algo_lorentz()

Fonction « principale » pour un système de Lorentz qui, pour chaque incrémentation de dt (boucle), calcule la nouvelle position et l'écrit dans le fichier de data.

Utilise :

nouvelle_fraction(int p, int q)

new_coord(fraction x, fraction y, fraction z)

*param(fraction * parametre, coord * position)*

*calcul(fraction * parametre, coord * position)*

ecrire_dans_fichier()

AUTRE_SYS

Tout autre système qui pourrait être implémenté dans notre projet aura une librairie qui lui est propre organisée comme LORENTZ soit une fonction d'initiation des paramètres, une fonction de calculs et une fonction décrivant le déroulement du processus (dans la même optique que algo_lorentz).

FICHIER

Librairie liée à l'écriture dans un fichier et/ou à la lecture de celui-ci. Le contenu de cette librairie sera connu une fois que nous aurons appris à manier la lecture et l'écriture d'un fichier en C.

Cette librairie servira donc à sauvegarder par écriture dans un fichier les différentes positions/coordonnées au cours du temps.

TRACER

Librairie liée au traçage de l'évolution des positions/du système. Le contenu de cette librairie sera connu une fois que nous en saurons plus sur le fonctionnement de gnuplot.

Elle nous servira à tracer le résultat final.

DEPENDANCES ENTRE LIBRAIRIES

FRACTION : Pas de dépendance.

COORD : FRACTION.

ENTREE : Pas de dépendances.

LORENTZ : ENTREE, FICHIER, FRACTION, COORD.

AUTRE_SYS : ENTREE, FICHIER, FRACTION, COORD.

FICHIER : Dépendances inconnues

TRACER : Dépendances inconnues

ORGANISATION DU PROGRAMME

PRINCIPAL

Choix du système par l'utilisateur (Lorentz ou autre).

Exécution de la fonction principale liée au système choisi (algo_Lorentz ou autre).

Traçage du résultat final de l'évolution du système.