SIMULATION DE LA TRAJECTOIRE DE JAMES WEBB

On se propose de localiser les points de Lagrange pour le système Soleil-Terre et de modéliser le mouvement d'une sonde spatiale comme le James Webb au voisinage du point de Lagrange L2. La sonde de masse m est soumise à l'attraction gravitationnelle combinée du Soleil M_S et de la Terre M_T. La masse m est très faible par rapport aux masses des deux astres et ne perturbe pas leur mouvement.

On suppose que la distance Soleil-Terre ($1.496\ 10^{11}$ m) est constante et que le couple Soleil-Terre effectue un mouvement de rotation uniforme.

Les principaux objectifs de ce projet sont :

- Identifier au moins les trois premiers points de Lagrange en traçant les isovaleurs du potentiel incluant la contribution gravitationnelle et celle de la force centrifuge. Voir [https://fr.wikipedia.org/wiki/Point_de_Lagrange].
- Ecrire et implémenter les équations différentielles associées au mouvement de la sonde de masse m dans le référentiel en rotation uniforme. Elles seront intégrées numériquement à l'aide de la librairie GNU-gsl.
- Pour vous faciliter la tâche, un programme de simulation du mouvement d'une masse soumise à une force de rappel du type ressort vous est fourni.
- La représentation graphique de la trajectoire vous est demandée, vous utiliserez la librairie gfx d'utilisation très simplifiée et très efficace. De même que précédemment, un exemple vous est fourni.

Suggestions :

- Il est préférable dans ce type de problème d'utiliser des unités réduites, comme par exemple la distance Terre-Soleil comme unité de distances. Faire de même avec le temps.
- Calculer d'abord des trajectoires simples pour valider votre simulation avant de passer à des études plus complexes.

<u>Installation des librairies mathématique GNU-gsl</u> :

A titre d'information et aussi pour vous permettre de développer un projet utilisant ces librairies sur des machines Unix autres que celles de l'école, on résume les opérations nécessaires pour installer la librairie GNU-gsl.

- 1er cas : vous n'êtes pas l'administrateur de votre machine, autrement dit, vous ne pouvez pas vous connecter en tant que root :

Télécharger la librairie GNU-gsl gsl-latest.tar.gz depuis https://www.gnu.org/software/gsl/

Décompresser l'archive localement dans votre Home Directory en tapant :
tar –xzf gsl-latest.tar.gz
Placer vous dans le répertoire gsl-latest puis taper :
./configure --prefix=$HOME/**extralib**
Lancer la compilation et l'installation dans le répertoire local de votre Home Directory :
make ; make install

Pour compiler un programme utilisant la GNU-gsl, il est nécessaire d'indiquer le répertoire des fichiers include $HOME/**extralib/include** et le répertoire où se trouve la librairie $HOME/**extralib/lib**

```
gcc -I $HOME/extralib/include pgm.c -o pgm   -L $HOME/extralib/lib -lgsl -
lgslcblas -lm
```

- 2ème cas : l'administrateur de votre machine procède à l'installation de la librairie.

En se plaçant comme précédemment dans le répertoire gsl-2.X, exécuter **./configure** puis **make.** L'installation s'effectue en mode administrateur. Connecter vous en tant que **root** et dans le répertoire gsl-2.X, taper **make install**

Etudes préliminaires :

Lire la documentation de la librairie GNU-gsl concernant le traitement des équations différentielles. Tester le comportement de la librairie en créant de petits programmes tests comme celui donné ci-après.

Le programme ode.c décrit le mouvement d'une masse soumise à une force de rappel du type ressort :

$m\frac{d^2\mathbf{OM}}{dt^2} = -k\mathbf{OM}$ que l'on transforme en un système de 4 équations différentielles du premier ordre $\begin{cases} \frac{dx}{dt} = v_x \\ \frac{dy}{dt} = v_y \\ \frac{dv_x}{dt} = -\omega_0^2 x \\ \frac{dv_y}{dt} = -\omega_0^2 y \end{cases}$ avec $\omega_0^2 = k/m$ que nous avons fixé à 1 pour simplifier. On note $\mathbf{y} = \begin{Bmatrix} x \\ y \\ v_x \\ v_y \end{Bmatrix}$ et $\mathbf{f} = \begin{Bmatrix} v_x \\ v_y \\ -\omega_0^2 x \\ -\omega_0^2 y \end{Bmatrix}$ tels que $\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y},t)$.

```
/* Pour compiler le programme ode.c
      gcc -I /usr/extralib/include/ ode.c -o ode -lgsl -lgslcblas -lm

   Pour l'exécuter
      Export LD_LIBRARY_PATH = $HOME/extralib:$LD_LIBRARY_PATH
      ./ode
*/

#include <stdio.h>
#include <gsl/gsl_errno.h>
#include <gsl/gsl_matrix.h>
#include <gsl/gsl_odeiv.h>
#include <gsl/gsl_ieee_utils.h>
```

```
/* DIM dimension du systeme d'equations */
#define DIM 4

/* limite supérieure du pas d'intégration */
const double hmax=0.01 ;
```

/* ecriture du systeme d'equations differentielles du 1er ordre $\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y},t)$ */

```
/* eq du ressort en 2D sous la forme dyvec/dt = fvec */

int fonction (double t, const double *yvec, double *fvec, void *params)
{
  double x =yvec[0];
  double y =yvec[1];
  double vx=yvec[2];
  double vy=yvec[3];

  fvec[0] = vx;
  fvec[1] = vy;

  fvec[2] = -x;
  fvec[3] = -y;
```

$$\mathbf{f}\begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{pmatrix}$$

```
  return GSL_SUCCESS;
}

main()
{
  const gsl_odeiv_step_type * T = gsl_odeiv_step_rk8pd;
  gsl_odeiv_step * s = gsl_odeiv_step_alloc (T, DIM);

/* controle de la precision de calcul */
  gsl_odeiv_control * c = gsl_odeiv_control_y_new (0., 1.e-6);
  gsl_odeiv_evolve * e  = gsl_odeiv_evolve_alloc (DIM);
  gsl_odeiv_system sys = {fonction, NULL, DIM, NULL};

/* temps initial et temps final */
  double t = 0.0, tfin = 10.0;
/* pas initial d'integration des equations diff */
  double h = hmax;

/* conditions initiales {x0 y0 x0' y0'} */
  double y[DIM] = { 1.0, 1.0, 0., 0. };

  gsl_ieee_env_setup();

  for (;;)
    {
    int status = gsl_odeiv_evolve_apply (e, c, s,
                                          &sys,
                                          &t, tfin,
                                          &h, y);
    if (status != GSL_SUCCESS) break;

/* ---- limite sup du pas d'integration fixee par l'utilisateur */
```

```
    if (h>hmax) h=hmax;
/* ------------------------------------------------------------ */
    printf("%lf \t %lf \t %lf \t %lf\n", t, h, y[0], y[1]);
    if (t>=tfin) break;
    }

  gsl_odeiv_evolve_free(e);
  gsl_odeiv_control_free(c);
  gsl_odeiv_step_free(s);
}
```

Travail préliminaire :

1. Donner l'expression générale de la vitesse angulaire de rotation ω d'une planète de masse M_T tournant sur une orbite circulaire de rayon R autour du Soleil M_T, sans faire l'hypothèse $M_T << M_S$?
2. Donner les équations différentielles associées au mouvement de la sonde m dans le référentiel tournant du couple Soleil-Terre.
3. A partir de ces équations, retrouver la définition des points de Lagrange [https://fr.wikipedia.org/wiki/Point_de_Lagrange].
4. Vérifier numériquement la relation obtenue pour ω dans le cadre d'un sous-projet en prenant deux astres de masse identique, puis en prenant deux masses de rapport 2.

Travail numérique :

Des problèmes de perte de précision apparaissent lorsqu'on travaille sur des nombres ayant des ordres de grandeur très différents. Pour les éviter, les grandeurs physiques seront renormalisées. Les distances seront exprimées en distance Terre-Soleil, les temps en fonction de la période terrestre, les masses en fonction de celle du Soleil. Normaliser de même la constante de gravitation $G=6.67\ 10^{-11}\ N.m^2/kg^2$ en une grandeur sans dimension.
La méthode d'intégration temporelle des équations différentielles est Runge-Kutta ordre 8 avec contrôle d'erreur dont l'implémentation est fournie par la librairie gsl.

1. Vérifier numériquement la relation obtenue pour ω dans le cadre d'un sous-projet en prenant deux astres de masse identique, puis en prenant deux masses de rapport 2.
2. Tracer les isovaleurs du potentiel incluant la contribution gravitationnelle et celle de la force centrifuge. Voir [https://fr.wikipedia.org/wiki/Point_de_Lagrange].
3. Montrer numériquement qu'il est possible de placer la sonde sur une orbite dans le plan perpendiculaire à l'axe Soleil-Terre au voisinage du point L2.

Données astronomiques :

	Soleil	Terre
Masse en kg	$1.9891\ 10^{30}$	$5.9737\ 10^{24}$
Rayon de l'orbite en m	0	$1.49\ 10^{11}$