

PM 2023

Système solaire réduit

1 Sujet

Dans le cadre du projet de mathématique, vous devrez concevoir (en C) le programme de simulation numérique du système solaire. Comme tout projet réaliste, c'est un projet transversal nécessitant des compétences de différents domaines :

- physique, algèbre et analyse : partie modélisation,
- mathématiques numériques : méthode d'intégration,
- algorithmique et production de code : informatique.

1.1 Système réel

Le système réel est composé d'une étoile (le Soleil) autour de laquelle gravitent des planètes. Ces planètes sont réparties en deux groupes, les planètes telluriques, au centre :

- Mercure,
- Vénus,
- Terre avec son satellite (la Lune),
- Mars,

les géantes gazeuses en périphérie :

- Jupiter,
- Saturne,
- Uranus,
- Neptune.

Les deux groupes précédents sont séparés par une ceinture d'astéroïdes (trace probable de l'échec de formation d'une planète supplémentaire).

1.2 Simulation demandée

Le système à simuler sera une simplification du système réel réduit aux trois éléments suivants :

- le Soleil,
- la Terre,
- la Lune

où seules agissent les forces de la gravitation universelle. À partir des conditions initiales fournies (positions et vitesses), il faudra simuler l'évolution temporelle par calcul des positions et vitesses de chacun des trois corps du système. On admettra que tous les corps du système (Lune incluse) gravitent dans un même plan (plan de l'écliptique) permettant ainsi de réduire le système à deux dimensions.

Le système à simuler reposera sur un système d'équations différentielles non linéaires du deuxième ordre. La méthode de résolution numérique imposée pour l'intégration est la méthode de Runge-Kutta d'ordre 4 explicite (classique). Référence la plus concise donnant un aperçu des coefficients utilisés : https://fr.wikipedia.org/wiki/Methodes_de_Runge-Kutta.

2 Spécifications

2.1 Repères et conditions initiales

Tous les objets appartiendront au même plan (écliptique) :

- le plan de l'écliptique sera muni d'un repère cartésien orthonormé centré en (0, 0), position initiale probable du Soleil,
- chaque corps sera assimilé à une masse ponctuelle,
- la position de chaque corps sera définie par un couple de coordonnées (cartésiennes) dans ce repère, exprimées en km,
- la vitesse de chaque corps sera définie par un couple de composantes (cartésiennes) dans ce repère, exprimées en km/s,
- la rotation gyroscopique individuelle de chaque corps est ignorée, en conséquence, l'inclinaison de la Terre est donc elle aussi ignorée et l'axe reliant les deux pôles célestes sera perpendiculaire au plan de l'écliptique,
- le plan sera vu du pôle nord céleste c'est-à-dire que la Terre devra tourner dans le sens trigonométrique (antihoraire).

2.2 Valeurs numériques

Les valeurs numériques relatives aux trois corps célestes, nécessaires à votre simulation, à savoir :

- les masses, ,
- les positions initiales (en x et y),
- les vitesses initiales (en x et y),

vous seront transmises lors de la première séance. Ces valeurs numériques seront contenues dans un fichier (texte) qui devra être traité par votre programme (cf. 3.3). Seule la constante universelle de la gravitation G est donnée ici :

$$G = 6,6742 \times 10^{-11} \text{ m}^3.\text{kg}^{-1}.\text{s}^{-2}$$

2.3 Livrable

Les sources de votre programme de simulation seront contenus dans un répertoire unique nommé `sysolxx` où `xx` sera remplacé par le numéro à deux chiffres qui vous sera fourni (le premier chiffre sera obligatoirement zéro, pour les nombres inférieurs à 10). Le livrable lui-même sera une archive du répertoire précédent (cf. paragraphes 3.2 et 4.3), selon le même principe de numérotation, cette archive sera nommée `sysolxx.tgz` (de préférence) voire `sysolxx.zip`.

ATTENTION : pour des raisons de portabilité, vous êtes limités à l'utilisation des entêtes définis par la norme SUS4. Pour faire simple, limitez-vous aux fichiers d'entêtes (bien suffisants) suivants :

- `float.h`,
- `locale.h`,
- `math.h`,
- `stdbool.h`,
- `stdlib.h`,

- `stdio.h`,
- `string.h`,
- `unistd.h`

Sur un système Unix (certifié SUS4), votre source doit impérativement pouvoir être compilé sans erreur (ni *warning*) par la commande suivante :

```
$ gcc -o sysolxx main.c -lm
```

soit, pour respecter l'état initial du squelette du projet :

```
$ gcc -o sysolxx main.c debug.c -lm
```

ou, encore plus simplement :

```
$ make
```

2.4 Exécution

Au moment de l'exécution, le programme devra admettre trois arguments, dans l'ordre :

- un entier : la durée totale de simulation exprimée en jours,
- un entier : le pas de calcul (période d'échantillonnage) en nombre de points par jour,
- une chaîne de caractères : le chemin d'accès au fichier des données numériques (2.2)

Par exemple (sur un système Unix ou compatible),

```
$ ./sysol03 365 100 STL.dat
```

lancera une simulation du système pour une durée totale de 365 jours, 100 points de calcul par jour, soit un total de 36500 points de calcul, à partir des données numériques fournies par le fichier `STL.dat`.

2.5 Présentation des résultats

Le programme devra afficher sur la sortie standard (`stdout`), les résultats bruts de simulation sous la forme d'une matrice à 22 colonnes, où

- la première ligne servira de ligne de titre (un label par colonne),
- les lignes suivantes représenteront les résultats numériques (notation scientifique) de simulation à raison d'un pas de calcul (à un instant donné) par ligne, dans l'ordre :
 - le temps (valeur de l'instant du pas de calcul en jours),
 - chaque groupe de six colonnes suivant présentera les résultats numériques d'un corps dans l'ordre d'apparition dans le fichier des données numériques (dernier argument de la ligne de commande). Ces six colonnes seront, dans l'ordre :
 - la position horizontale du corps dans le plan (en km),
 - la position verticale du corps dans le plan (en km),
 - la composante horizontale de la vitesse du corps dans le plan (en km/s),
 - la composante verticale de la vitesse du corps dans le plan (en km/s),
 - la composante horizontale de l'accélération du corps dans le plan (en km/s/s),
 - la composante verticale de l'accélération du corps dans le plan (en km/s/s),
 - les deux colonnes suivantes (20 et 21) présenteront les composantes horizontale et verticale de la position de la Lune par rapport à la Terre en km,
 - enfin, la 22^e et dernière colonne sera la distance entre Terre et Lune (longueur unique du segment liant le point Terre au point Lune, en km).

- le séparateur de colonnes sera le caractère de tabulation (`\t`), aussi bien pour la ligne de titre que pour les lignes suivantes.

Voici un exemple de résultat conforme au format attendu :

Temps ...	Soleil ...	Soleil ...	Soleil ...	Soleil ...	
0,0100e+00	#,####e+##	#,####e+##	#,####e+##	#,####e+##	...
0,0200e+00	#,####e+##	#,####e+##	#,####e+##	#,####e+##	...
0,0300e+00	#,####e+##	#,####e+##	#,####e+##	#,####e+##	...
0,0400e+00	#,####e+##	#,####e+##	#,####e+##	#,####e+##	...
0,0500e+00	#,####e+##	#,####e+##	#,####e+##	#,####e+##	...
0,0600e+00	#,####e+##	#,####e+##	#,####e+##	#,####e+##	...
0,0700e+00	#,####e+##	#,####e+##	#,####e+##	#,####e+##	...
0,0800e+00	#,####e+##	#,####e+##	#,####e+##	#,####e+##	...
0,0900e+00	#,####e+##	#,####e+##	#,####e+##	#,####e+##	...

Ce format permet une représentation graphique facile par simple redirection et importation dans un tableur (openoffice, libreoffice...).

Par exemple

```
$ ./sysol199 365 100 STL.dat >sysol199.csv
```

lancera la simulation sur 365 jours à raison de 100 points de calcul par jour, les données numériques initiales seront fournies par le fichier `STL.dat` et les résultats seront à récupérer dans le fichier `sysol199.csv`. L'extension `.csv` autorisera l'ouverture immédiate du fichier par un tableur.

ATTENTION, il ne sera toléré aucun autre message sur la sortie standard. Si vous désirez afficher d'autres informations, utilisez obligatoirement la sortie d'erreur (`stderr`). Les fonctions (ou plutôt macros) `debug` et `trace`, présentes dans le squelette initial (cf. 3.2), sont spécifiquement dédiées à cet usage.

2.6 Représentation de l'échelle temporelle

L'échelle temporelle de la simulation pouvant être longue (ordre de l'année terrestre), le temps sera exprimé par un nombre réel dont l'unité sera le jour terrestre à partir de zéro.

2.7 Documents à fournir

En plus de l'archive du code source de votre programme de simulation, vous devrez fournir un document (format pdf obligatoire), où vous

- développerez le modèle mathématique de votre système;
- si nécessaire, un dictionnaire des symboles (où les constantes de votre modèle mathématique feront référence aux noms de variables utilisés dans votre programme);
- développerez toutes les informations nécessaires à la compréhension et l'utilisation de votre programme;
- citerez toutes les sources utilisées.

Ce document

- est limité à deux pages (format A4),
- devra être nommé `sysolxx.pdf` selon le même principe qu'au paragraphe 2.3.

3 Divers

3.1 Carnet de route

La réalisation de ce projet est étalée sur trois séances de quatre heures.

La première séance sera consacrée à l'enregistrement définitif de la composition des groupes (binômes), en veillant à ce que chaque binôme possède les outils nécessaires à la résolution du projet (machine, compilateur, outils...). La compréhension complète du sujet et des spécifications par tous sera vérifiée. Enfin, la représentation des nombres en machine sera abordée sous formes de petits exercices.

La deuxième séance sera l'occasion d'un premier retour, c'est-à-dire que je répondrai aux questions que vous auriez à poser suite aux premiers développements de votre travail sur le projet. Comme la première séance, cette séance sera agrémentée de petits exercices destinés à illustrer les méthodes d'intégrations numériques (Euler et RK4).

Enfin, la troisième séance sera entièrement consacrée à la remise des projets. Chaque groupe qui me remettra son projet recevra en retour un rapport d'exécution qui sera un comparatif avec le résultat attendu. Vous aurez bien sûr la possibilité de modifier votre projet pour en soumettre différentes versions.

Quoi qu'il en soit, la date limite de remise du projet devrait à priori être fixée par la fin de la dernière séance, c'est-à-dire que le temps de développement du projet devrait s'étaler entre le début de la première séance et la fin de cette dernière séance.

Pour raison d'équité, afin que la durée de développement soit la même pour l'ensemble des groupes, le délai de remise du projet sera définitivement annoncé après l'inscription de l'ensemble des groupes.

3.2 Fourniture - archive

Une archive (tgz) vous sera fournie. Cette archive contiendra un répertoire nommé conformément à votre numéro de groupe. Ce répertoire contiendra les quatre fichiers suivants :

- `main.c`, véritable squelette de départ de votre programme
- `debug.h` et `debug.c`, module de fonctions dédiées aux messages de contrôle (dirigés vers `stderr`), vous êtes libres de ne pas les utiliser ;
- un `Makefile` générique.

Vous êtes libres de développer votre projet directement au sein du seul fichier `main.c` ou dans un ou plusieurs fichiers sources supplémentaires. Remarques complémentaires à propos du `Makefile` dans le paragraphe 4.3 suivant.

3.3 Fourniture - valeurs numériques

Les données numériques nécessaires à la simulation seront fournies par un fichier annexe. Ce fichier vous étant fourni, inutile d'envisager la vérification syntaxique de son contenu. Chaque corps céleste y est décrit par quatre lignes utiles qui, pour une lecture séquentielle, sont dans l'ordre :

- le nom : une ligne de texte qui n'est pas un commentaire,
- la masse : une ligne avec un nombre unique,
- les deux composantes de la position initiale : deux nombres séparés par un nombre quelconque d'espaces sur la même ligne,
- les deux composantes de la vitesse initiale : deux nombres séparés par un nombre quelconque d'espaces sur la même ligne,

Les lignes dont le premier caractère est un dièse (#) sont des lignes de commentaire qui doivent être ignorées, tout comme les lignes vides.

Les corps célestes peuvent y être décrits dans un ordre quelconque pour la simulation, attention toutefois à respecter l'ordre dans lequel sont attendus les résultats (cf. 2.5). Cela vous permet d'éditer votre propre fichier de simulation pendant la phase de développement et de vérification, réduit par exemple au simple système Terre-Lune.

3.4 Sources externes

Vous êtes libres d'utiliser les ressources documentaires que vous voulez, soyez cependant prudent et n'oubliez jamais de citer vos sources.

4 Notifications importantes

4.1 Respect des spécifications

ATTENTION, le strict respect des spécifications fonctionnelles du type :

- nom du programme source et de l'exécutable,
- ordre et format des arguments,
- ordre et format des valeurs résultantes,

sont obligatoires. Étant donné le grand nombre de groupes de projet, votre travail sera évalué par un script automatique qui ne fera pas de sentiment, l'indicateur calculé sera la somme cumulée des erreurs quadratiques de vos résultats de calculs par rapport aux résultats de calculs du programme de référence pour un même jeu d'arguments (le programme de référence utilise les coefficients de la référence du paragraphe 1.2 pour l'intégration numérique par RK4 et la valeur de G du paragraphe 2.2).

4.2 Localisation

La localisation française est exigée, c'est-à-dire que tous les nombres s'écrivent avec une virgule et non pas un point, c'est la raison pour laquelle, l'instruction

```
setlocale(LC_NUMERIC, "fr_FR");
```

doit absolument être la première instruction de votre programme (conformément au squelette fourni). En conséquence, tous les nombres (aussi bien en entrée qu'en sortie) seront au format naturel où la virgule sera le séparateur entre partie entière et partie décimale.

En cas d'erreur, sur un système du type Linux, vérifiez que les paquets `locales` et `locales-all` sont bien installés.

4.3 Compilation

En principe, sur un système du type Unix, que vous développiez votre code uniquement dans le fichier `main.c` ou dans d'autres fichiers, le fichier `Makefile` fourni conviendra à produire l'exécutable voulu.

Quel que soit le système sur lequel vous développerez votre programme, vous me livrez en retour, une archive (zip ou de préférence `tgz`) identique à la fourniture (3.2) uniquement enrichie du code source que vous aurez développé.

Sachez enfin que le `Makefile` fourni sera utilisé tel quel pour vérifier votre travail sur la plateforme d'essai. À vous de vous assurer que votre code est conforme.

4.4 Travail individuel par binôme

Plutôt que de récupérer la solution (complète ou partielle) d'un autre groupe (groupe ou binôme) préférez toujours la solution autonome. Vos sources seront soumis à un contrôle de similitudes sur l'ensemble des projets. En cas d'infraction, les groupes concernés (le producteur aussi bien que le consommateur seront sanctionnés).