

Travail pratique

Simulation d'un système planétaire sur un plan à l'aide des lois de Newton

El Kharroubi Michaël
(et un tout petit peu Orestis Malaspinas)

But

- Simuler un phénomène physique vu en cours à l'aide du langage C
- Visualiser les interactions gravitationnelles au sein d'un système planétaire.

Théorie

Rappel théorique

Dans notre univers, tous les corps sont soumis à des forces. Une force est une grandeur permettant de quantifier pour un corps : la direction, le sens et l'intensité de son interaction avec les autres corps. Dans le cadre de ce travail pratique, nous nous intéresserons à l'une des quatre forces fondamentales, la force de gravitation. Pour rappel, les forces suivent les trois lois de Newton.

1. Si un corps est immobile (ou en mouvement rectiligne uniforme), alors la somme des forces qu'il subit, appelée force résultante, est nulle. ($\vec{F} = \vec{0}$).
2. La force résultante subit par un corps est égale à la masse de ce dernier multipliée par son accélération. ($\vec{F} = m\vec{a}_p$).
3. Si un corps A subit une force de la part d'un corps B , alors le corps B subit une force de réaction de sens opposé et de même intensité ($\vec{F}_{BA} = -\vec{F}_{AB}$).

La force de gravitation est une force qui apparaît entre tous les objets (ou corps) ayant une masse. Elle régit le mouvements des objets massifs (planètes, étoiles, trou noir, galaxies...). La force de gravité causée par un corps B et subie par un corps A s'obtient avec la formule suivante :

$$\vec{F}_{BA} = G \frac{m_A m_B}{\|\vec{r}_{AB}\|^3} \vec{r}_{AB},$$

où $G = 6.67 \cdot 10^{-11} \frac{\text{m}^3}{\text{kg} \cdot \text{s}^2}$, m_A , m_B sont les masses du corps A et B en [kg] et \vec{r}_{AB} le vecteur reliant A et B en [m].

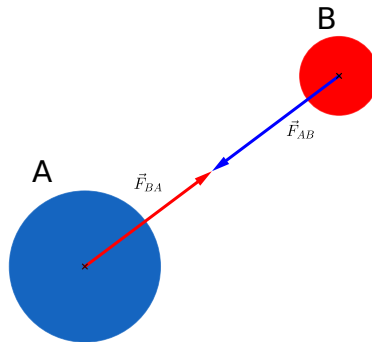


FIG. 1 : Exemple d'interaction gravitationnelle

Les orbites planétaires

Les planètes, dont la terre, tournent autour du soleil (ce n'est pas la terre le centre du système solaire comme on le croyait il y a longtemps).

L'orbite d'une planète n'est pas un cercle parfait, il s'agit en réalité d'une ellipse. Cette orbite ellipsoïdale est définie par trois paramètres :

1. Le demi-grand axe (a en mètres, **à ne pas confondre avec l'accélération \vec{a}**),
2. Le demi-petit axe (b en mètres),
3. L'excentricité (e sans unités).

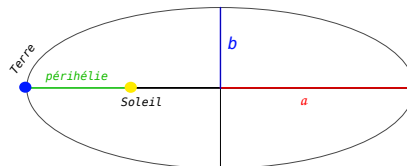


FIG. 2 : Exemple (volontairement exagéré) de l'orbite de la terre autour du soleil. Source : Alexis Durnat (Bureau A403).

Sur la figure @fig :e, vous observez différentes orbites pour différentes valeurs de e .

Simulation d'un système planétaire

Idée générale

Dans notre simulation, nous représenterons un système planétaire inspiré de notre système solaire où toutes les planètes sont sur le même plan (ce qui est une assez bonne approximation de ce qui se passe dans notre système solaire). Au centre nous aurons une étoile (fixe, encore une approximation) et un certain nombre de planètes qui orbitent autour de cette dernière.

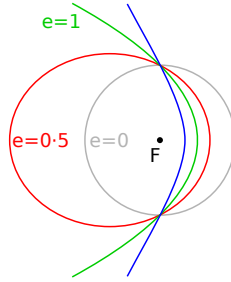


FIG. 3 : Différentes orbites en fonction de l'excentricité. Source : Wikipédia, <https://bit.ly/3x53F4A>.

Pour simuler un système planétaire, on peut effectuer les étapes suivantes :

1. Créer une étoile au centre de notre domaine.
2. On ajoute autant de planètes que l'on désire autour de l'étoile.
3. On définit nos conditions initiales (en particulier la vitesse de chaque planète).
4. On affiche puis on simule l'évolution :
 1. On affiche notre système ($\vec{x}_p(t)$).
 2. On calcule la force résultante sur chacune des planètes.
 3. On calcule la prochaine position de chacune des planètes ($\vec{x}_p(t + \Delta t)$).
 4. On met à jour la position des planètes de notre système.
 5. On revient à l'étape 4.1.

Évolution

Concentrons nous tout d'abord sur l'évolution de la simulation (nous verrons les conditions initiales dans un second temps). Pour déterminer les informations nécessaires, on commence par calculer la force résultante sur la planète p :

$$\vec{F}_p = \sum_{\{q \in \mathcal{C} \mid q \neq p\}} \vec{F}_{qp},$$

où \mathcal{C} est l'ensemble des corps célestes (les planètes et l'étoile) de votre simulation.

Ensuite, on calcule la nouvelle position à partir de sa position actuelle et de sa position précédente (on a également vu ça en cours). Pour ce faire, on reprend les équations du mouvement uniformément accéléré on a :

$$\begin{aligned} \vec{x}_p(t + \Delta t) &= \vec{x}_p(t) + \Delta t \vec{v}_p(t) + \frac{(\Delta t)^2}{2} \vec{a}_p(t) \\ \vec{x}_p(t - \Delta t) &= \vec{x}_p(t) - \Delta t \vec{v}_p(t) + \frac{(\Delta t)^2}{2} \vec{a}_p(t) \end{aligned}$$

{#eq :verlet}

Avec t et Δt en [s], $\vec{x}_p(t)$ la position de la planète p en [m], $\vec{v}_p(t)$ la vitesse de la planète p en [$\frac{m}{s}$] et $\vec{a}_p(t)$ l'accélération de la planète p en [$\frac{m}{s^2}$].

En additionnant les deux équations de @eq :verlet, on obtient :

$$\begin{aligned}\vec{x}_p(t + \Delta t) + \vec{x}_p(t - \Delta t) &= 2\vec{x}_p(t) + (\Delta t)^2 \vec{a}_p(t) \\ \vec{x}_p(t + \Delta t) &= 2\vec{x}_p(t) - \vec{x}_p(t - \Delta t) + (\Delta t)^2 \vec{a}_p(t)\end{aligned}$$

{#eq :mouvement}

On remarque donc qu'il n'est pas nécessaire de retenir l'évolution de \vec{v}_p pour calculer le prochain état de notre simulation. Nous n'avons besoin que des deux dernières positions ($\vec{x}_p(t), \vec{x}_p(t - \Delta t)$) et de l'accélération ($\vec{a}_p(t)$). Nous calculerons donc les \vec{x}_p itérativement et je vous laisse le soin de déduire comment nous pouvons obtenir $\vec{a}_p(t)$ à partir des formules données (oui cela est un exercice).

Conditions initiales

Si l'on regarde @eq :mouvement, on remarque que pour calculer $\vec{x}_p(t + \Delta t)$ en $t = 0$ ($\vec{x}_p(\Delta t)$), il nous faudrait la position $\vec{x}_p(t - \Delta t)$. Comme nous ne considérons pas de temps négatif dans notre simulation, nous allons fixer la valeur en $t = 0$ avec la formule @eq :mouvement. Ce qui nous donne :

$$\vec{x}_p(\Delta t) = \vec{x}_p(0) + \Delta t \vec{v}_p(0) + \frac{(\Delta t)^2}{2} \vec{a}_p(0)$$

Commençons par $\vec{v}_p(0)$. Pour rappel, l'orbite d'une planète est ellipsoïdale. Sans rentrer dans des détails qui dépassent le cadre de ce tp, nous pouvons connaître la vitesse à la périhélie (voir @fig :orbite) de l'orbite d'une planète autour de notre étoile. La périhélie est la distance la plus courte dans l'orbite d'une planète (autour du soleil, -hélie=soleil). Nous avons donc :

$$\vec{v}_p(0) = \sqrt{\frac{GM_\odot(1 + e_p)}{a_p(1 - e_p)}} \cdot \frac{\vec{r}_{p\perp}}{\|\vec{r}_{p\perp}\|},$$

où M_\odot est la masse de l'étoile en [kg], a_p est le demi-grand axe de l'orbite de la planète p , en [m], e l'excentricité de l'orbite de la planète p sans unités et \vec{r}_\perp est le vecteur perpendiculaire ($\vec{r}_{p\perp} = \begin{pmatrix} -r_{py} \\ r_{px} \end{pmatrix}$) au vecteur allant de la planète p à son étoile.

Par conséquent, puisque que nous connaissons la vitesse à la périhélie, nous placerons intelligemment la position initiale de la planète p , $\vec{x}_p(0)$, à la périhélie. Pour $\vec{a}_p(0)$, on peut le calculer de la même manière qu'avec n'importe quelle autre valeur de t .

Dans votre simulation vous utiliserez des données en mètres, vous aurez donc vraisemblablement (vu l'échelle cosmique) des valeurs en millions de kilomètres. Lors d'un précédent travail pratique, vous avez implementé une fonction permettant de convertir un vecteur en deux dimensions $\vec{r} \in \{[-1; 1]\}^2$ en coordonnées d'écran $\vec{c} \in \{[0; lignes[]] \times [0; colonnes[]]\}$. Pour obtenir un vecteur \vec{r} , vous devrez définir le rayon de votre écran en mètres R_S (p.ex : 110% du demi-grand axe de l'orbite de la planète la plus éloignée de l'étoile). Puis à partir de votre

vecteur position en mètres \vec{x}_p , vous obtiendrez la nouvelle position $\vec{r} = \frac{\vec{x}_p}{R_S}$. Vous pourrez ensuite convertir cette position en coordonnées d'écran \vec{c} grâce à votre fonction.

Énoncé

Dans le cadre de ce travail pratique, vous allez devoir implementer une simulation de système planétaire semblable à notre système solaire. Pour cela vous avez à votre disposition un squelette de code nommé `skeleton.tar`. Ce squelette est uniquement là dans le but de vous aider. Vous pouvez donc vous en passer (ce que je vous déconseille). En revanche, vous devez impérativement utiliser la méthode et les équations présentées dans la partie théorique de cet énoncé.

Si toutefois vous décidiez de prendre ce squelette (excellente décision, je vous en félicite), une fois téléchargé et décompressé, vous devrez effectuer les actions suivantes :

1. Remplacer le fichier `skeleton/vec2/vec2.c` par celui que vous avez réalisé durant un précédent travail pratique.
2. Prendre connaissance des fichiers `skeleton/planet/planet.h` et `skeleton/main.c`. Ces fichiers contiennent des commentaires pour vous guider.
3. Réaliser le travail pratique.

Ce travail est séparé en deux parties, qui valent respectivement 4.5 et 1.5 points (Pour un total de 6 (c'est dingue!!!)).

La première partie, consiste à simuler un système planétaire avec les quatre premières planètes (par ordre de distance) du système solaire (Mercure, Vénus, Terre (souriez, vous êtes simulé), Mars).

Dans un second temps vous devrez ajouter quelques planètes fictives (≥ 2), et essayez de faire varier les différents paramètres. C'est à dire :

- La masse
- La distance à l'étoile
- La forme de l'orbite en changeant la vitesse initiale via le demi-grand axe et l'excentricité.

Pour dessiner vos planètes et votre étoile, vous pouvez utiliser la fonction `draw_full_circle` qui a été ajoutée à cette occasion dans la librairie `skeleton/gfx`. Elle prend en paramètre un contexte SDL, un centre, un rayon et une couleur.

Pour que votre simulation marche, vous serez amené, à un moment ou à un autre, à devoir mettre des données (position de vos planètes, choix du Δt , masse, etc...). Vous devez dans le cadre de ce travail aller chercher ces données, et ce dans les bonnes unités (p.ex : distance en mètre et non en années lumière). Vous devrez indiquer dans votre rapport les données choisies (dans les bonnes unités) et leur source.

Validation

Pensez à valider que votre simulation fonctionne. Il y a différents degrés de validation.

Le premier consiste à observer si les planètes orbitent bien autour du soleil, que leurs orbites ne se croisent pas, qu'elles ne disparaissent pas dans l'infini intersidéral, etc. C'est une validation *à l'œil*.

Le second est de vérifier que les orbites des planètes se referment bien et après quelle période de temps et comparer avec les valeurs disponibles en ligne.

Question de la mort

Il s'agit ici de déterminer quelle est la planète qui en moyenne est la plus proche de la terre. A vous de déterminer une méthode pour faire cette mesure.

Travail à rendre par groupe de deux (strictement égal à deux)

- Un rapport succinct (environ 6 pages) présentant le travail réalisé, avec des images de ce dernier.
- Le repos git contenant le code réalisé.
- (Bonus) Une vidéo du résultat, si vous réussissez à créer un système cool avec une musique épique!

Ce travail est à rendre pour le 17.01.2024 à 23h59 au plus tard!

Quelques remarques sur le rapport

Le rapport doit être séparé en 3-4 parties principales : l'introduction, une partie description du problème et théorie, une partie résultat, et une partie conclusion.

Chacune de ces parties a ses spécificités, que nous allons rapidement résumer ici.

L'introduction

Cette partie sert à **poser le contexte** de votre travail. Elle introduit le sujet en donnant un contexte général. Attention, il ne s'agit pas ici d'écrire une banalité du type "on va simuler le système solaire parce que le prof de physique nous l'a demandé". Il faut plutôt expliquer à quoi peuvent servir les simulations numériques, et en particulier celles du système solaire. Puis, dire en quoi consiste votre travail tel que vous pourriez l'expliquer à une personne qui ne connaît pas grand chose à la physique et aux simulations numériques. Finalement, cette partie sert à poser "le plan" de votre travail et répondre à la question "que contient chaque chapitre?". N'hésitez pas à être très factuels : "Le chapitre 2 contient..., le chapitre 3 contient ...".

Partie théorique

Cette partie sert à décrire avec plus de précision le contenu de votre travail d'un point de vue théorique. Quelles sont les équations utilisées ? Comment est structuré votre code (pas besoin de mettre du code ici, plutôt décrire les fonctionnalités générales) ? Quelles sont les choses importantes à savoir pour reproduire vos simulations ? Si cela aide n'hésitez pas à utiliser des images que vous **commenterez** (une image sans commentaire c'est totalement inutile, car on ne sait pas ce que vous voulez dire avec). Il est très important de décrire toutes les notions dont vous avez besoin pour que la partie résultats soit compréhensible pour la personne qui lit votre travail.

Résultats

Dans cette partie vous décrivez les résultats obtenus. Quelles sont les simulations que vous avez effectuées ? Quel est l'état initial de votre simulation ? Quels sont les paramètres que vous avez utilisés ? Qu'y a-t-il de remarquable dans vos résultats ? Sur quelles parties faut-il attirer l'attention du lecteur ? Si vous en avez n'hésitez à utiliser des images ou des tableaux. Il faut bien entendu **commenter** les tableaux et images sinon ils sont inutiles.

Conclusion

La première fonction de la conclusion est de rappeler en quoi a consisté votre travail. Ensuite, il est nécessaire de faire ressortir les points importants de vos résultats (quitte à les répéter par rapport à la partie résultats). Il est fortement déconseillé de dire des choses comme “on a beaucoup appris dans ce travail, il était super”. Cela n'apporte pas grand chose à votre rapport (mémoire de bachelor dans le futur). Finalement, la conclusion doit ouvrir sur le “futur” :

- Comment est-ce que vous pourriez améliorer votre travail ?
- Que manque-t-il pour le rendre plus complet ?