TP 3: Équations différentielles

Philippe Després

Date de remise: 19 mars 2021

PHY-3500 – Physique numérique (H21)

Mécanique céleste - comètes

Plusieurs comètes orbitent autour du soleil, souvent selon des trajectoires très allongées. Elles passent la plupart de leur temps dans les régions lointaines du système solaire, où elles se déplacent lentement, mais s'approchent périodiquement près du soleil et peuvent être observées de la Terre (la comète C/2020 F3 (NEOWISE) a pu être observée à l'été 2020). Lorsqu'elles s'approchent du soleil (au périhélie), les comètes ont une vitesse supérieure à celle à l'aphélie ; ce système est un cas classique d'utilisation de méthodes de résolution d'équation différentielle à pas adaptatif.

L'équation différentielle (deuxième loi de Newton) qui régit le mouvement du système comète-soleil est

$$m\frac{d^2\mathbf{r}}{dt^2} = -\left(\frac{GMm}{r^2}\right)\frac{\mathbf{r}}{r} \tag{1}$$

où M est la masse du soleil, m la masse de la comète à la position $\mathbf{r}$, G la constante gravitationnelle et GMm/r^2 est la force entre les deux corps dans la direction $-\mathbf{r}/r$ (en direction du soleil).

Considérant que le mouvement est confiné dans un plan, deux dimensions sont nécessaires pour le décrire :

$$\frac{d^2x}{dt^2} = -GM\frac{x}{r^3}, \qquad \frac{d^2y}{dt^2} = -GM\frac{y}{r^3}, \tag{2}$$

où $r = \sqrt{x^2 + y^2}$.

a. Transformez ces deux équations du deuxième ordre en quatre équations du premier ordre.

b. Écrivez un programme pour résoudre ces équations à l'aide de la méthode Runge-Kutta à l'ordre 4, en utilisant un pas h constant. Utilisez le module `astropy` pour la valeur des constantes M et G, et les conditions initiales $x = 4 \times 10^9$ km et $y = 0$ (ce qui est environ la distance de l'orbite de Neptune) en plus de la vitesse à cette position : $v_x = 0$ et $v_y = 500$ m/s. Votre programme produira un graphique de la position de la comète (x vs y avec des points pour chaque pas).

Vous choisirez une valeur de pas h qui permettra de calculer avec précision au moins deux orbites complètes de la comète. Comme les orbites sont périodiques, elles devraient se superposer sur votre graphique. Si ce n'est pas le cas, c'est que le pas choisi est trop grand. Discutez de vos découvertes quant au choix du pas, et du temps de calcul associé. Vous pourriez développer une métrique quantitative, au-delà de la simple inspection visuelle, de ce que constitue un pas adéquat.

c. Modifiez votre programme RK4 pour obtenir une version à pas adaptatif. La justesse désirée δ sera de 1 km par année pour la position de la comète. Refaites le graphique de la trajectoire et commentez, en discutant de la vitesse d'exécution, de la densité de points sur la trajectoire, et de la justesse des calculs par rapport à la version de RK4 à pas constant.

Mécanique céleste - la Terre

La méthode de Verlet, pour rappel, est une variation de la méthode Leapfrog qui permet d'alléger les calculs pour certaines classes de problème, dont ceux liés au mouvement qui ont la forme

$$\frac{d^2\mathbf{r}}{dt^2} = \mathbf{f}(\mathbf{r}, t). \tag{3}$$

Avec des conditions initiales sur $\mathbf{r} = (x, y, ...)$ et $\mathbf{v} = d\mathbf{r}/dt$, la méthode consiste à faire un premier pas pour calculer

$$\mathbf{v}(t + \tfrac{1}{2}h) = \mathbf{v}(t) + \frac{1}{2}h\mathbf{f}(\mathbf{r}(t), t), \tag{4}$$

et de poursuivre en calculant successivement

$$\begin{aligned}
\mathbf{r}(t+h) &= \mathbf{r}(t) + h\mathbf{v}(t + \tfrac{1}{2}h), && (5)\\
\mathbf{k} &= h\mathbf{f}(\mathbf{r}(t+h), t+h), && (6)\\
\mathbf{v}(t+h) &= \mathbf{v}(t + \tfrac{1}{2}h) + \tfrac{1}{2}\mathbf{k}, && (7)\\
\mathbf{v}(t + \tfrac{3}{2}h) &= \mathbf{v}(t + \tfrac{1}{2}h) + \mathbf{k} && (8)
\end{aligned}$$

d. Utilisez la méthode de Verlet pour calculer l'orbite de la Terre autour du soleil, considérant que le mouvement est décrit par

$$\frac{d^2\mathbf{r}}{dt^2} = -GM\frac{\mathbf{r}}{r^3} \tag{9}$$

et sachant que la distance du périhélie est de 1.4710×10^{11} m et que la vitesse tangentielle à cette position est de 3.0287×10^4 m/s (utilisez `astropy` pour les autres constantes nécessaires). Utilisez un incrément temporel $h = 1$ heure. Représentez graphiquement plusieurs orbites, qui devraient appaître légèrement non-circulaires.

e. Modifier votre programme pour qu'il calcule aussi l'énergie potentielle $-GMm/r$ et cinétique $\frac{1}{2}mv^2$ à chaque pas, ainsi que la somme (énergie totale) de ces deux quantités. Rapportez ces valeurs en fonction du temps dans un graphique et commentez.

f. Utilisez maintenant RK2 pour calculer l'orbite de la Terre et l'énergie totale en fonction du temps, que vous rapportez sur le même graphique que l'énergie totale en fonction du temps calculée par la méthode de Verlet. Votre graphique devrait montrer le caractère symplectique de la méthode de Verlet vs RK2.

g. Utilisez maintenant la méthode de Bulirsch-Stoer pour calculer l'orbite de la Terre, avec une précision de 1 km par année (voir l'exemple 8.7 dans le Newman pour une implémentation, que vous pouvez réutiliser). Utilisez un intervalle H de une semaine. Comparez avec la précision obtenue avec la méthode de Verlet.

Mécanique céleste - Mars

Maintenant que vous êtes des professionnel.le.s, la NASA vous engage et vous demande de calculer la position de la planète Mars le 18 février 2021, jour d'atterissage de l'astromobile *Perseverance*, à partir des données connues le 30 juillet 2020, jour du lancement depuis la Terre.

Vous aurez besoin du module Python `jplephem`, soit les éphémérides utilisées par le JPL (`pip install jplephem`). Vous utiliserez les éphémérides `de421` (`pip install de421`), qui couvrent avec une bonne précision la période 1900–2050.

Le code suivant vous donnera la position et la vitesse initiales de Mars (attention aux unités utilisées, voir documentation à `https://pypi.org/project/jplephem/`) :

```
from astropy.time import Time

# éphémérides
import de421
from jplephem import Ephemeris

eph = Ephemeris(de421)

# dates
lancement=Time("2020-07-30")
atterissage=Time("2021-02-18")

# un nombre de jours juliens est attendu par la routine, d'où le .jd
# position en km, vitesse en km par jour
position, velocity = eph.position_and_velocity('mars',lancement.jd)
```

h. Utilisez Bulirsch-Stoer pour calculer la position de Mars 203 jours plus tard, soit le 18 février 2021 (jour de l'atterrissage). Ne lésinez pas sur la précision, ça coûte cher un astromobile. Notez que les calculs seront effectués en trois dimensions. Comparez votre position calculée à la valeur de l'éphéméride, soit `eph.position('mars',atterissage.jd)`. Commentez, notamment sur vos limites et celles de la méthode utilisée, peut-être en vous inspirant d'informations à cette adresse [1].

Instructions pour la remise

Le travail devra être complété en trinômes sous format de cahier de bord `jupyter` (.ipynb) et remis dans la boîte de dépôt créée à cette fin. Ce document contiendra **toutes informations pertinentes** permettant au lecteur d'apprécier vos résultats et conclusions, incluant le code

1. `https://en.wikipedia.org/wiki/Jet_Propulsion_Laboratory_Development_Ephemeris`

Python utilisé et d'éventuelles références bibliographiques. La qualité de la présentation est très importante (utilisation de sections, de graphiques appropriés, de mise en contexte, etc.).