

Calculs d'éphémérides avec Python

Limoges : jeudi 6 mai 2021

par Rémi METZDORFF et Laurent ASTIER

Lycée Suzanne Valadon – 87000 Limoges

remi.metzdorff@orange.fr

laurent.astier@ac-limoges.fr

Cet article présente un module Python permettant de calculer la position de nombreux objets du système solaire et son utilisation dans le cadre des nouveaux programmes de lycée, où la programmation est devenue incontournable. La première partie présente le cadre théorique dans lequel sont réalisés les calculs de position. La deuxième décrit sommairement le fonctionnement général du module disponible en ligne. Enfin, plusieurs applications directement exploitables en classe par l'enseignant et les élèves sont présentées, notamment dans le cadre du thème mouvement et interaction des programmes de lycée.

INTRODUCTION

Lors de la réforme du lycée en 2019, la programmation est entrée dans les nouveaux programmes. C'est le langage Python qui est préconisé : il offre les avantages d'une syntaxe épurée, tout en conservant une efficacité remarquable. En témoignent ses nombreuses applications, depuis la création de jeux vidéo jusqu'à la détection d'ondes gravitationnelles, en passant par le *machine learning*, l'interfaçage, le traitement de données, la finance, etc. Cet article explore une utilisation de Python en lycée, avec la création et l'exploitation d'un module nommé `celestialbody` utilisé pour générer rapidement et simplement les éphémérides de nombreux astres du système solaire.

Le module `celestialbody` permet de calculer la position d'un corps du système solaire à partir des équations du mouvement dans le cadre du problème de Kepler. Retrouver les lois du mouvement à partir des données générées par ce programme ne présente donc pas d'intérêt : ce n'est pas l'objectif du module. Par ailleurs, les positions calculées ici sont bien des positions approximatives, notamment dans le cas des astéroïdes et comètes où l'influence des huit planètes du système solaire conduit à des perturbations importantes. Pour ces petits objets, la résolution numérique des équations du mouvement dans le cadre d'un système à N corps est envisageable [1]. Si des éphémérides précises sont requises, on se référera plutôt à [2, 3].

1. PRINCIPE DU CALCUL DE POSITION

1.1. Le problème de Kepler



On s'intéresse ici au système à deux corps, formé par le Soleil et l'objet choisi dont la masse est supposée négligeable devant celle du Soleil. La seule force considérée est la force d'attraction gravitationnelle de Newton et le mouvement du corps est étudié dans un référentiel considéré comme galiléen. Le problème revient donc à l'étude d'un mouvement à force centrale conservative dont on rappelle ici les résultats importants :

- ♦ la conservation du moment cinétique, qui permet d'établir la planéité du mouvement et la conservation de la vitesse aréolaire. On démontre ainsi la deuxième loi de Kepler (loi des aires) puis la troisième pour des trajectoires fermées.
- ♦ la conservation de l'énergie mécanique, qui permet de différencier deux catégories de solutions : les états

liés et les états de diffusion. On établit alors l'équation polaire de la trajectoire [4], une conique caractérisée par son excentricité e . On se restreindra au cas des orbites elliptiques conformément à la première loi de Kepler pour lesquelles $e \in [0; 1[$. On a alors :

$$\rho(v) = \frac{a(1 - e^2)}{1 + e \cos v}, \quad (1)$$

où ρ est la distance au foyer, v la position angulaire par rapport au périhélie et a le demi-grand axe de l'ellipse (le demi-petit axe sera noté b).

L'équation polaire de la trajectoire permet d'obtenir l'allure de l'orbite. Pour connaître la position d'un corps à un instant t donné, il faut déterminer l'évolution temporelle de la position angulaire $v(t)$ aussi appelée anomalie vraie.

1.2. Anomalies

En effet, sauf dans le cas d'une trajectoire circulaire, la position angulaire v n'évolue pas linéairement avec le temps. Il est ici nécessaire d'utiliser quelques astuces géométriques et de construire le cercle circonscrit à l'ellipse appelé cercle auxiliaire. On introduit alors deux nouveaux angles : l'anomalie excentrique E et l'anomalie moyenne M (Fig. 1).

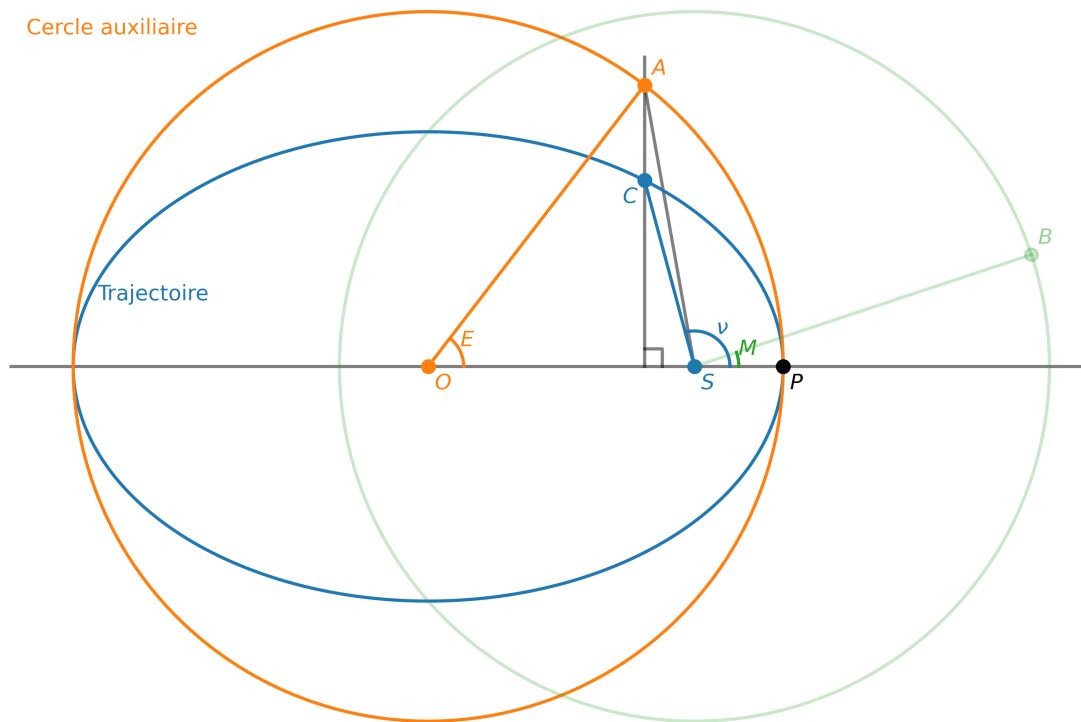


Figure 1 – Représentation des anomalies, angles utilisés pour le repérage de la position d'un astre sur son orbite.

On utilise la deuxième loi de Kepler : l'aire $\mathcal{A}(t)$ balayée par le segment SC est proportionnelle au temps écoulé depuis le passage au périhélie $t - t_p$. Elle s'exprime en fonction de l'aire totale de l'ellipse πab et de la période de révolution T calculée avec la troisième loi de Kepler :

$$\mathcal{A}(t) = \frac{\pi ab}{T} (t - t_p) = M(t) \times \frac{ab}{2}. \quad (2)$$

On a ici introduit l'anomalie moyenne $M(t)$ qui repère la position angulaire qu'aurait le corps si son orbite était circulaire et de même période :

$$M(t) = \frac{2\pi}{T} (t - t_p). \quad (3)$$

On peut ensuite remarquer que la trajectoire elliptique est obtenue après une compression du cercle auxiliaire d'un facteur b/a selon l'axe vertical. À un facteur b/a près, l'aire $\mathcal{A}(t)$ est donc égale à l'aire du secteur POA du cercle auxiliaire, moins celle du triangle SOA :

$$\mathcal{A}(t) = \frac{b}{a} \times \left[\frac{a^2 E(t)}{2} - \frac{ae \times a \sin E(t)}{2} \right]. \quad (4)$$

On obtient alors l'équation de Kepler qui lie l'anomalie excentrique et l'anomalie moyenne en combinant les équations (3) et (4) :

$$M(t) = E(t) - e \sin E(t). \quad (5)$$

Le lien entre l'anomalie excentrique et l'anomalie vraie peut finalement être obtenu après « quelques » efforts trigonométriques :

$$\tan \frac{v}{2} = \sqrt{\frac{1+e}{1-e}} \tan \frac{E}{2}. \quad (6)$$

1.3. Résolution numérique de l'équation de Kepler

L'équation de Kepler (Éq. (5)) ne possède pas de solution analytique. Pour trouver la valeur de l'anomalie excentrique $E(t)$, on doit résoudre cette équation numériquement à chaque instant t [5]. On procède par itération en commençant avec :

$$E_0 = M(t) + e \sin M(t).$$

Les valeurs suivantes sont calculées avec :

$$E_{n+1} = E_n + \Delta E_n,$$

où

$$\Delta E_n = \frac{\Delta M_n}{1 - e \cos E_n},$$

et

$$\Delta M_n = M(t) - (E_n - e \sin E_n).$$

Dans le cadre de ce travail, une précision satisfaisante est atteinte quand $|\Delta E_n| < 10^{-6}$. Pour des orbites d'excentricité faible, la convergence est très rapide puisque $M \approx E$ et seules quelques itérations sont nécessaires.

Il suffit maintenant de connaître l'orientation de la trajectoire par rapport à un référentiel bien choisi pour déterminer à tout instant la position du corps dans l'espace.

1.4. Un mot sur les repères

Tant que l'on ne s'intéresse qu'à un seul objet en orbite autour du Soleil, le repère le plus naturel est le repère héliocentrique orbital $Sxyz$ dont l'origine est le centre du Soleil. Le mouvement du corps est alors contenu dans le plan Sxy et se fait dans le sens direct, avec l'axe Sx orienté vers le périhélie et l'axe.

Pour la suite, il est nécessaire de choisir un repère commun à tous les corps du système solaire : le repère écliptique héliocentrique $SXYZ$. Son origine demeure le centre du Soleil S mais l'axe SX est cette fois orienté vers le point vernal (position de la Terre lors de l'équinoxe de printemps) et forme avec l'axe SY le plan de l'écliptique. Finalement, l'axe SZ est orienté pour former un repère direct dans lequel le mouvement de la Terre, comme celui de la plupart des éléments du système solaire, se fait dans le sens direct.

1.5. Les paramètres orbitaux

L'évolution du système à deux corps obéit à un système de trois équations différentielles du second ordre (une pour chaque coordonnée d'espace) qui nécessitent donc six conditions initiales données à une date connue pour aboutir à une solution unique. Donnés à une date de référence t_0 appelée époque, les paramètres orbitaux permettent de donner simplement ces conditions initiales sous la forme de six paramètres indépendants du temps :

- ◆ le demi-grand axe a ;
- ◆ l'excentricité e ;
- ◆ l'inclinaison I ;
- ◆ la longitude du nœud ascendant Ω ;
- ◆ l'argument du périhélie ω ;
- ◆ la date de passage au périhélie t_p .

Les deux premiers paramètres décrivent la taille et la forme de l'orbite tandis que les trois suivants correspondent aux angles qui donnent l'orientation de l'ellipse par rapport au repère écliptique héliocentrique. Finalement, le dernier permet d'obtenir la position du corps sur son orbite à une date donnée.

Les paramètres orbitaux indiqués précédemment sont donnés sous différentes formes selon qu'il s'agit de comètes, d'astéroïdes ou de planètes. Pour les comètes, la distance du périhélie q remplace le demi-grand axe et on a $q = a(1 - e)$. Pour les astéroïdes, la date de passage au périhélie est remplacée par la donnée de la valeur de l'anomalie moyenne M_0 à l'époque. Finalement, pour les planètes :

- ♦ la longitude du périhélie ϖ remplace l'argument du périhélie et on a : $\omega = \varpi - \Omega$;
- ♦ la longitude moyenne L_0 remplace l'anomalie moyenne et on a : $M_0 = L_0 - \varpi$.

Tous les paramètres orbitaux utilisés pour la suite sont accessibles librement sur la base de données très conséquente du Jet Propulsion Laboratory (NASA). À ce jour, elle comprend bien sûr les huit planètes du système solaire, plus d'un million d'astéroïdes et près de quatre mille comètes.

1.6. Position d'un astre à une date quelconque

Pour déterminer la position d'un corps à une date quelconque t , on procède donc comme suit¹ [5].

- ♦ On commence par calculer l'anomalie moyenne à la date t , soit en utilisant sa définition (Éq. (3)) soit d'après sa valeur à l'époque :

$$M(t) = M_0 + \frac{2\pi}{T}(t - t_0).$$

- ♦ Vient ensuite la résolution numérique de l'équation de Kepler (Éq. (5)) détaillée précédemment pour obtenir la valeur de l'anomalie excentrique $E(t)$.
- ♦ Le calcul des coordonnées du corps dans le repère héliocentrique orbital se fait directement en utilisant le cercle auxiliaire (Fig. 1) :

$$\vec{r}_o(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} a(\cos E(t) - e) \\ a\sqrt{1 - e^2} \sin E(t) \\ 0 \end{bmatrix}.$$

- ♦ On calcule les coordonnées dans le repère héliocentrique écliptique en effectuant successivement trois rotations d'axe SZ et d'angle ω , d'axe SX et d'angle I et d'axe SZ et d'angle Ω :

$$\vec{r}_e(t) = \mathcal{R}_Z(\Omega)\mathcal{R}_X(I)\mathcal{R}_Z(\omega)\vec{r}_o(t),$$

soit finalement :

$$\vec{r}_e(t) = \begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \end{bmatrix} = \begin{bmatrix} x(t) \times (\cos \omega \cos \Omega - \sin \omega \sin \Omega \cos I) - y(t) \times (\sin \omega \cos \Omega + \cos \omega \sin \Omega \cos I) \\ x(t) \times (\cos \omega \sin \Omega + \sin \omega \cos \Omega \cos I) - y(t) \times (\sin \omega \sin \Omega - \cos \omega \cos \Omega \cos I) \\ x(t) \times \sin \omega \sin I + y(t) \times \cos \omega \sin I \end{bmatrix}$$

2. LE MODULE CELESTIALBODY

Le module `celestialbody` présenté ici est construit autour de la *classe* `CelestialBody` [7]. Pour le langage Python, une *classe* permet de définir un nouveau type d'objet auquel sont associées des propriétés et des fonctionnalités. L'accès à ces différents paramètres se fait à l'aide de commandes très simples. La *classe* `CelestialBody` définit donc un nouvel objet qui modélise l'astre choisi, récupère ses paramètres orbitaux et calcule sa position à une date définie par l'utilisateur.

2.1. Démarrer avec CelestialBody

Le module contient les cinq fichiers essentiels au fonctionnement du programme : le programme `celestialbody.py` où est définie la *classe* `CelestialBody` et les fichiers de données regroupant les paramètres orbitaux des corps pris en charge par la *classe*. Pour accéder aux fichiers mis à jour, consulter [8] et [9] : `p_elem_t2.txt` rassemble les paramètres orbitaux de planètes, `ELEMENTS.COMET` ceux des comètes, `ELEMENTS.NUMBR` et `ELEMENTS.UNNUM` ceux des astéroïdes. Tout autre programme utilisant la *classe* `CelestialBody` devra se trouver dans le même dossier que ces cinq fichiers.²

¹ Pour tenir compte de termes correctifs, l'algorithme du module est légèrement différent de la procédure décrite ici mais le principe reste le même.

² Il faudra mettre à jour la variable `PYTHONPATH` du système d'exploitation sinon.

Pour une première utilisation, il est conseillé d'utiliser l'application Jupyter Notebook installée avec la distribution Anaconda ou un autre environnement du même type disponible grâce aux ENT par exemple, et de suivre le notebook `tutorial_fr.ipynb` joint au code source. Il permet de découvrir pas à pas le fonctionnement du module.

2.2. Fonctions de base

Après l'import des modules utiles, la première étape consiste à créer l'objet `CelestialBody` correspondant à l'astre choisi :

```
mars = CelestialBody('Mars')
```

Le choix du corps se fait à partir de son nom anglophone : 'Mars' pour la planète Mars, '1P/Halley' pour la comète de Halley, etc.

On choisit ensuite la date à laquelle doivent être réalisés les calculs grâce à la fonction `datetime` du module `datetime` de Python :

```
mars.date = datetime(2021, 2, 18)
```

La position de Mars à cette date est alors donnée en unités astronomiques dans le repère éclipique héliocentrique par la commande :

```
mars.position
```

qui renvoie :

```
(-0.0057727483433337445, 1.5698184461545464, 0.03297198596449348)
```

2.3. Éphémérides

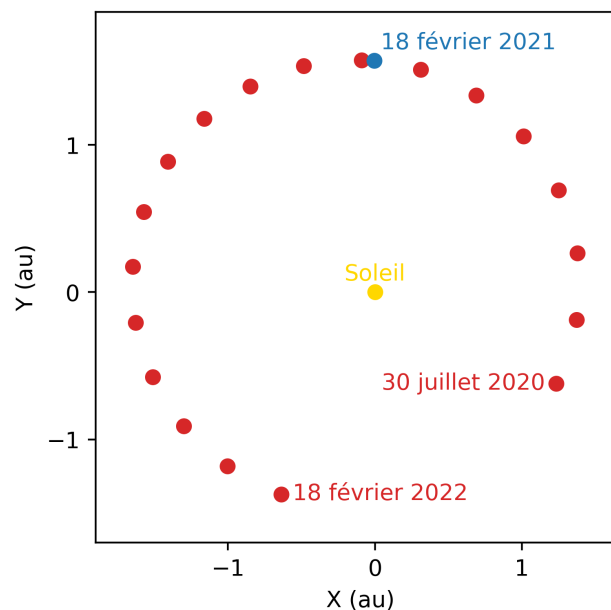


Figure 2 – Positions successives de la planète Mars dans le repère éclipique héliocentrique entre le lancement de Perseverance et son anniversaire de présence sur la planète rouge (en année terrestre).

L'intervalle entre deux positions consécutives est d'environ 30 jours.

L'objectif premier du module étant le calcul d'éphémérides entre des dates spécifiques et à intervalle de temps choisi, on y parvient avec la commande :

```
mars.data('position', start=debut, stop=fin, step=pas)
```

On obtient alors les coordonnées des positions successives de la planète Mars, dans le repère éclipique héliocentrique, entre les dates `debut` et `fin`, avec un intervalle de `pas` jours entre deux dates.³ Avec cette fonction, le calcul des éphémérides est refait systématiquement ce qui nécessite de travailler dans le même dossier que le module et qui peut être chronophage selon la machine utilisée et la quantité de données demandées.

³ Si ces paramètres ne sont pas spécifiés, cette fonction et la suivante ont des paramètres par défaut qui permettent de visualiser correctement l'orbite de la plupart des corps célestes.

Pour pallier ce problème, on peut créer très simplement un fichier de données contenant les coordonnées des positions successives de l'astre. La commande :

```
mars.data_position_txt(start=debut, stop=fin, step=pas)
```

crée le fichier `mars.txt` qui contient les mêmes données que celles calculées précédemment.

On peut finalement représenter graphiquement ces données à l'aide de la bibliothèque `matplotlib` de Python (Fig. 2).

3. EXPLOITATION EN CLASSE

Quelques utilisations possibles de ce module sont présentées ci-dessous. Il est aussi possible d'avoir un aperçu de quelques autres pistes envisagées en explorant les fichiers disponibles en ligne [7].

3.1. Représenter les positions successives d'un système

La première activité s'inscrit dans le cadre du programme de seconde. Elle a pour objectif de travailler sur la capacité numérique « représenter les positions successives d'un système [...] à l'aide d'un langage de programmation » du thème mouvement et interactions. En utilisant les coordonnées des positions de plusieurs astres, les élèves sont également amenés à déterminer les échelles spatiales adaptées à l'étude de ces mouvements (Fig. 4). Tous les fichiers « prof » et « élève » nécessaires à cette activité sont disponibles avec le module `celestialbody` dans le dossier « `activites/tp_tchouri` ».

3.1.1. Présentation de l'activité

On s'intéresse ici à trois objets : la Terre, Jupiter et la comète 67P/Churyumov–Gerasimenko, plus connue sous le nom de Tchouri. Dans un premier temps, les élèves doivent parcourir rapidement les fichiers de données afin de comprendre d'où viennent les coordonnées utilisées et ce à quoi elles correspondent. Ils sont ensuite amenés à s'approprier et modifier le programme `position_planetes.py` fourni pour afficher les positions de ces différents astres avec la commande `plt.plot`, puis à ajuster les limites du graphique pour observer toute leur trajectoire avec les commandes `plt.xlim` et `plt.ylim` (Fig. 3).

Tout au long de l'activité, les élèves sont amenés à commenter la nature du mouvement des différents objets observés. Enfin, pour travailler sur la capacité « identifier les échelles temporelles [...] pertinentes de description d'un mouvement », un supplément proposé aux élèves les invite à déterminer les périodes de révolution de Tchouri et Jupiter avec les informations contenues dans les différents fichiers à leur disposition.

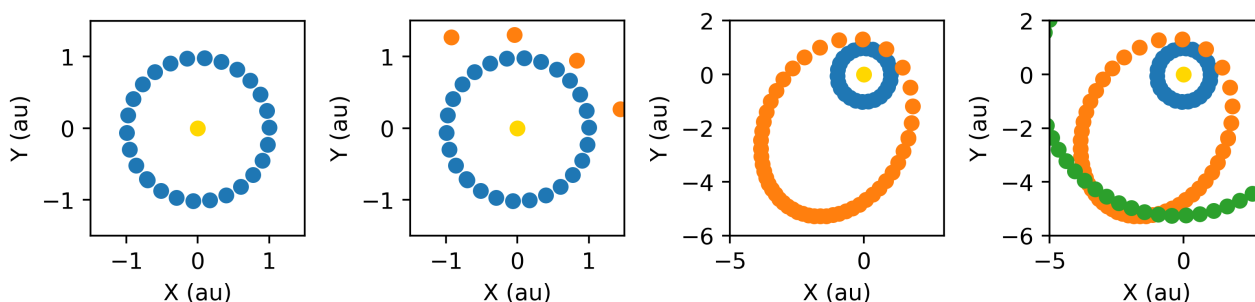


Figure 3 – Graphes successivement obtenus lors de l'activité. Le programme fourni n'affiche que la position du Soleil (en jaune) et celles de la Terre (en bleu). Les élèves le modifient pour représenter Tchouri (en orange), puis ajustent les limites des axes pour observer toute la trajectoire. Ils répètent ce travail avec Jupiter (en vert) pour finalement obtenir le graphe de la Fig. 4.

3.1.2. Déroulement de la séance

Cette activité a été proposée en mars, lors de plusieurs séances de TP d'une durée de 1h25 à quatre demi-classes d'environ 18 élèves, dans une salle équipée de neuf postes informatiques. EduPython est installé sur chaque machine et tous les fichiers nécessaires à l'activité étaient placés dans un espace de partage accessible sur le réseau du lycée. Deux sujets étaient fournis pour mettre en place une différenciation pédagogique : TP Tchouri 1 et TP Tchouri 2, le second étant adressé aux élèves ayant besoin d'un accompagnement plus important. Un supplément était aussi disponible pour les plus rapides. Les élèves étaient répartis dans des binômes imposés par l'enseignant de sorte que chaque élève soit associé à un autre de niveau comparable. Le choix de l'attribution de chaque sujet à un groupe donné s'est fait sur la base des résultats d'un autre TP de programmation sur la production d'un son avec Arduino, qui avait permis d'évaluer l'aisance des élèves sur

une activité mobilisant des compétences similaires. Il s'agit de la deuxième séance réalisée avec Python en physique-chimie, la première ayant été faite en début d'année comme une initiation. Les élèves doivent produire un compte-rendu ramassé en fin de TP et trois « appels prof » pendant l'activité permettent d'évaluer certaines compétences pendant la séance.

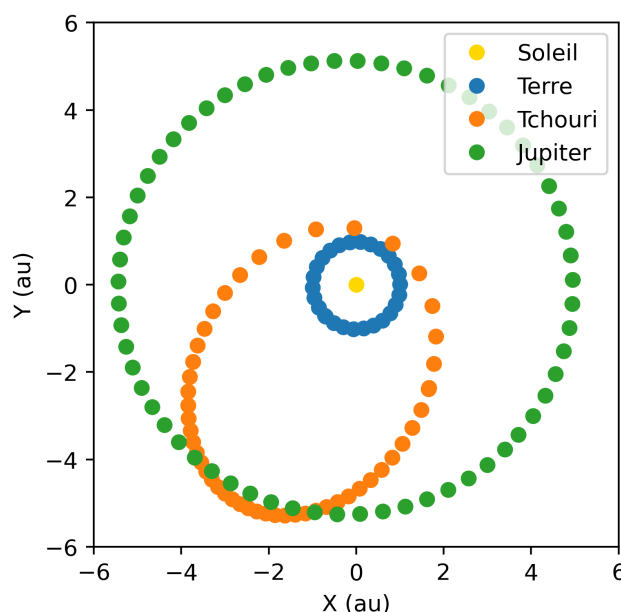


Figure 4 – Graphe obtenu par les élèves en fin de séance.

En début de séance, deux demi-classes d'une même classe ont manifesté une certaine réticence à l'idée d'une activité utilisant Python mais plusieurs élèves se sont étonnés en fin de séance : « Ah ! Mais c'est facile en fait ! ». Cette activité a été dans l'ensemble très bien réussie par les élèves puisque près de 85 % des binômes sont parvenus à obtenir le graphe de la Fig. 4 en fin de séance (Tab. 1). Le choix des groupes a semblé pertinent puisque la majorité des élèves ayant eu le sujet 1 ont demandé le supplément. Le supplément était par ailleurs nécessaire, puisque près de la moitié des groupes l'ont abordé même si tous ne l'ont pas terminé.

Sujet proposé	TP Tchouri 1	TP Tchouri 2
Nombre de groupes concernés	16	17
Nombre de groupes ayant convenablement terminé le sujet	15	14
Nombre de groupes ayant demandé le supplément	11	3

Tableau 1 – Bilan des résultats obtenus par les élèves pour l'activité.

3.1.3. Évolutions possibles de l'activité

Dans cette activité, les objets ont été choisis pour leur trajectoire quasiment contenue dans le plan de l'écliptique, dont la taille et la forme invite naturellement l'élève à changer d'échelle spatiale pour observer convenablement le mouvement de chaque corps, en plus de fournir un prétexte pour parler de la mission Rosetta avec le retour prochain de la comète Tchouri « près » de la Terre. Grâce à la base de données conséquente accessible avec le module `celestialbody`, il est possible de choisir d'autres objets pour adapter l'activité à l'actualité scientifique, aux affinités de l'enseignant et à celles des élèves.

Pour travailler davantage la capacité « identifier les échelles temporelles [...] pertinentes de description d'un mouvement », il est possible de laisser les élèves choisir eux-mêmes l'intervalle de temps qui permet de décrire correctement la trajectoire et l'évolution de la vitesse des différents objets. Dans l'activité présentée ici, le fichier ne contient que les données nécessaires à l'obtention des graphes (Fig. 3 et 4). On pourrait aussi fournir aux élèves des fichiers contenant les positions calculées tous les jours pendant 20 ans par exemple. En modifiant deux paramètres dans le programme (`pas` pour la période d'échantillonnage et `duree` pour la durée d'observation), les élèves pourraient choisir ceux qui permettent d'observer une révolution complète avec un nombre de points suffisant pour que la trajectoire soit clairement visible, sans en avoir trop pour distinguer l'évolution de la vitesse (Fig. 5).

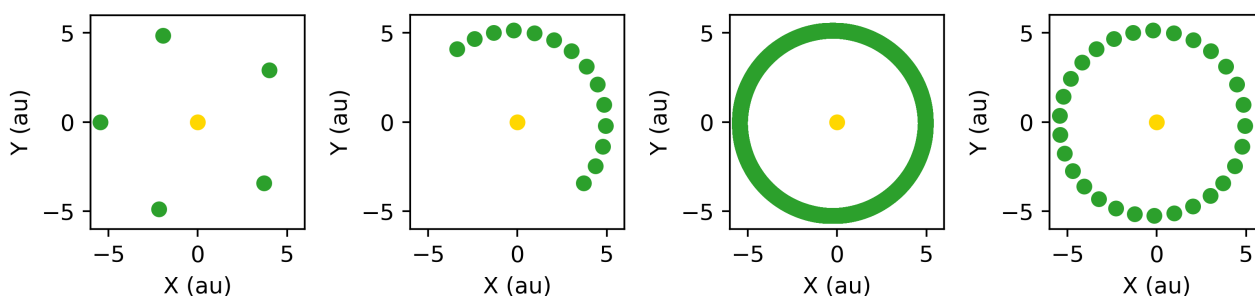


Figure 5 – Sur le dernier graphe, le choix d'échelles temporelles adaptées permet de décrire convenablement le mouvement de Jupiter autour du Soleil.

3.2. Troisième loi de Kepler

Cette deuxième proposition d'activité est à destination d'élèves de terminale en spécialité physique-chimie, pour travailler sur la capacité numérique « exploiter, à l'aide d'un langage de programmation, des données astronomiques ou satellitaires pour tester les deuxième et troisième lois de Kepler ». L'objectif est de mesurer la période de révolution d'un objet en orbite autour du Soleil et le demi-grand axe de sa trajectoire à partir des données calculées avec `celestialbody`, puis de comparer les valeurs obtenues à celles des huit planètes du système solaire, issues d'observations astronomiques [10]. Ici encore, les fichiers sont disponibles dans le dossier « `activites/tp_kepler` » [7].

3.2.1. Mesurer les paramètres orbitaux d'un corps

Dans un premier temps, les élèves doivent modifier le programme `position_comete.py` pour déterminer « expérimentalement » les paramètres orbitaux exploités dans la troisième loi de Kepler, pour une comète dont la trajectoire est peu inclinée par rapport au plan de l'écliptique. Les paramètres `t_ech` et `n_positions` correspondent respectivement à la période d'échantillonnage et au nombre d'observations, le produit $(n_{\text{positions}} - 1) \times T_{\text{ech}}$ donne la période d'observation. En choisissant bien les valeurs de ces paramètres, on obtient donc la période de révolution de l'astre. Le demi-grand axe se mesure directement dans la fenêtre graphique qui s'ouvre à l'exécution du programme. Là encore, une différenciation est possible en choisissant des objets dont la trajectoire est plus ou moins facilement exploitable (Fig. 6).

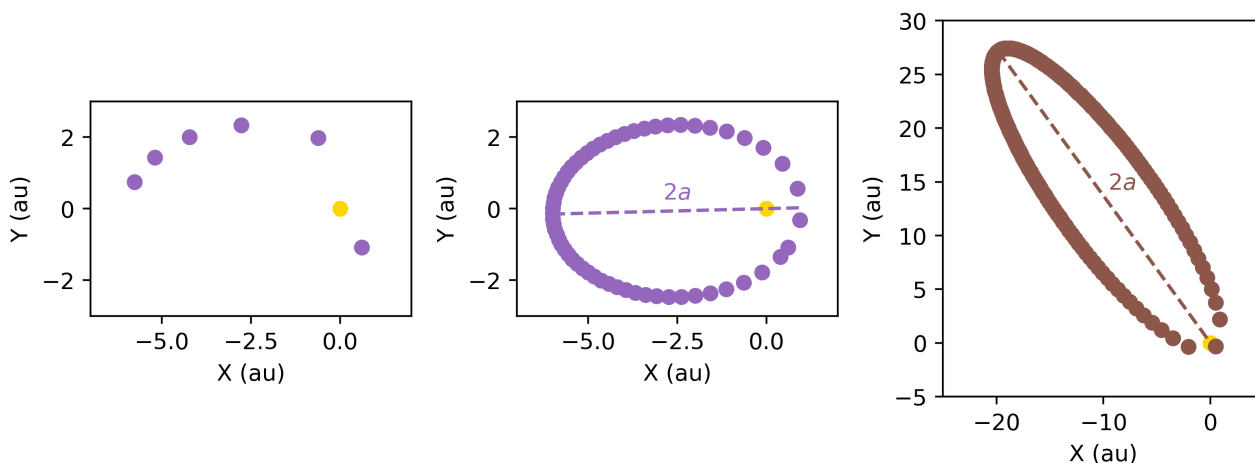


Figure 6 – Représentations graphiques des positions de la comète Finlay (à gauche) et de la comète de Halley (à droite) permettant la mesure de a et T . Avec son excentricité élevée, la détermination de la période de révolution de la comète de Halley requiert plus d'itérations. De plus, l'orientation de son orbite rend nécessaire l'utilisation du théorème de Pythagore pour la détermination du demi-grand axe.

3.2.2. Vérification de la troisième loi de Kepler

En s'appuyant sur le programme `kepler.py`, les élèves vérifient ensuite la troisième loi de Kepler avec des observations astronomiques auxquelles on ajoute les données mesurées précédemment. L'ajustement des données par une droite permet finalement, par exemple de déterminer la masse du Soleil (Fig. 7).

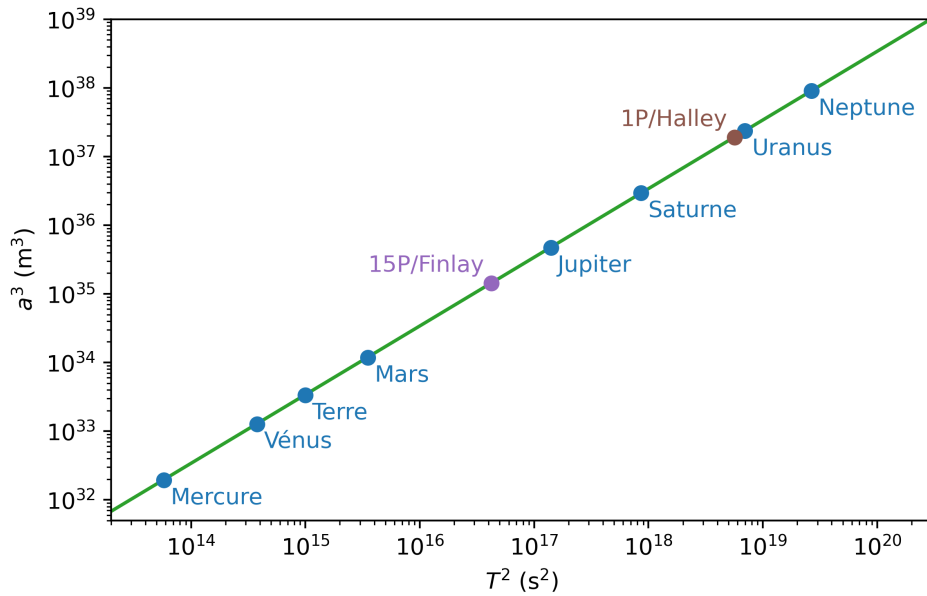


Figure 7 – Vérification de la troisième loi de Kepler avec quelques objets du système solaire. La droite correspond à l’ajustement des données issues des différentes mesures. Son coefficient directeur permet de calculer la masse du Soleil et on trouve $M_{\odot} = 2,0 \times 10^{30}$ kg.

3.3. Illustrations

Puisque le module permet d’accéder facilement aux coordonnées de positions des astres, il est possible de créer de nombreuses illustrations et animations pour appuyer certains points du programme. Un exemple est visible ici (Fig. 8), les fichiers `ex_illustrations.ipynb` et `ex_animations.ipynb` disponibles avec le module `celestialbody` en proposent d’autres [7].

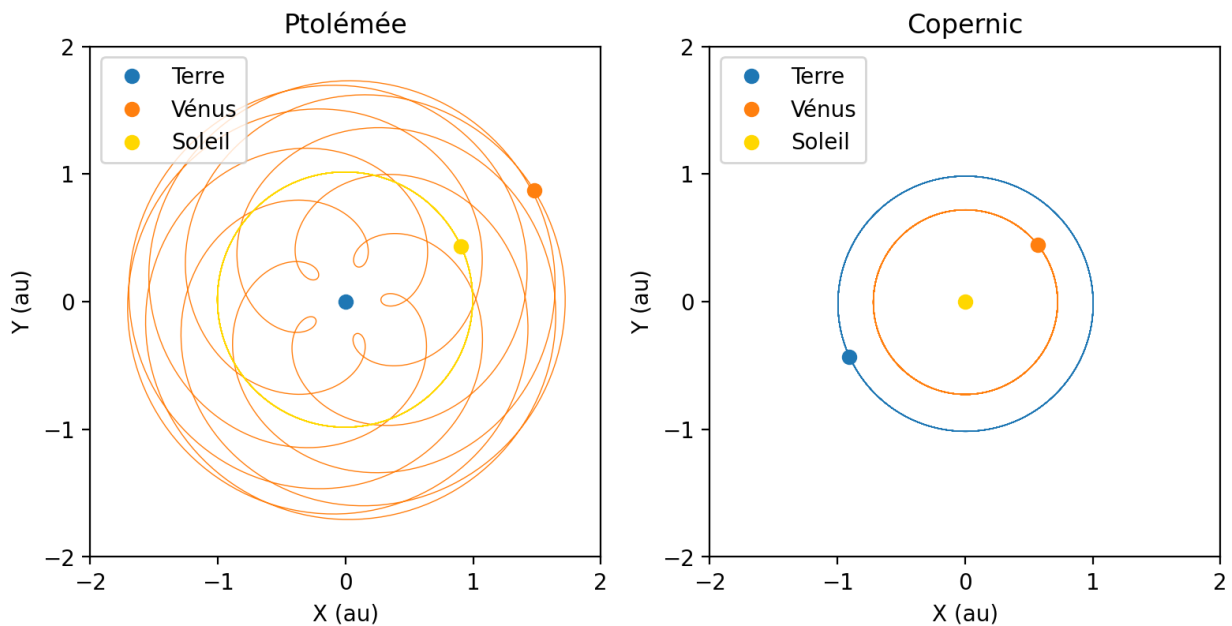


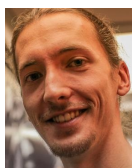
Figure 8 – Illustration de l’effet du changement de référentiel sur la nature du mouvement de quelques objets du système solaire. La figure de gauche se répète quasiment à l’identique en raison de la résonance orbitale qui existe entre la Terre et Venus.

CONCLUSION

Le module `celestialbody` facilite donc la création d’activités exploitant de diverses manières des données astronomiques. Le grand nombre d’objets disponibles permet d’adapter l’activité au contexte scientifique du moment, mais aussi et surtout à la diversité des élèves. Certaines applications ont été identifiées et développées dans cet article, d’autres sont accessibles en ligne [7], mais il y en reste sans doute beaucoup encore à découvrir.

BIBLIOGRAPHIE

- [1] F. Legrand, « Système solaire : orbites képlériennes », <https://www.f-legrand.fr/scidoc/docmml/sciphys/meca/kepler/kepler.html>, consulté le 20/04/21.
- [2] « Miriade Ephemeris Generator », <http://vo.imcce.fr/webservices/miriade/?forms>, consulté le 20/04/21.
- [3] « HORIZONS web interface », <https://ssd.jpl.nasa.gov/horizons.cgi>, consulté le 20/04/21.
- [4] B. Salamito *et al.*, *Physique tout-en-un MPSI-PTSI*, Paris : Dunod, 2^e édition, 2016.
- [5] E. M. Standish, *Keplerian Elements for Approximate Positions of the Major Planets*, https://ssd.jpl.nasa.gov/txt/aprx_pos_planets.pdf, consulté le 20/04/21.
- [6] J. Tatum, *Celestial Mechanics*, <https://phys.libretexts.org/@go/page/6783>, consulté le 20/04/21.
- [7] R. Metzdorff, « celestialbody », <https://github.com/remimetzdorff/celestialbody>, consulté le 20/04/21.
- [8] « Keplerian Elements for Approximate Positions of the Major Planets », https://ssd.jpl.nasa.gov/?planet_pos, consulté le 20/04/21.
- [9] « Small-Body Orbital Elements », https://ssd.jpl.nasa.gov/?sb_elem, consulté le 20/04/21.
- [10] « Planetary Fact Sheet », <https://nssdc.gsfc.nasa.gov/planetary/factsheet/>, consulté le 20/04/21.



Rémi METZDORFF

Professeur de physique-chimie

Lycée Suzanne Valadon

Limoges (Haute-Vienne)