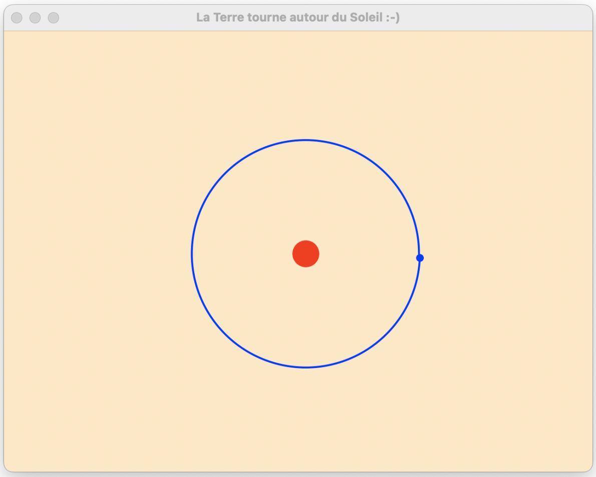
Python for physics dummies : problème 1
La Terre tourne autour du Soleil :-)

Dans ce devoir, nous allons intégrer les équations de Newton en utilisant l'expression du champ gravitationnel du soleil pour prédire la rotation de la Terre autour du soleil.



Dans le plan de l'orbite terrestre, les deux composantes du champ gravitationnel du soleil de masse M sont une fonction de la position dans le plan :

$$\vec{\mathbf{g}}(x,y) = -\frac{GM}{x^2+y^2}\begin{bmatrix} \dfrac{x}{\sqrt{x^2+y^2}} \\ \dfrac{y}{\sqrt{x^2+y^2}} \end{bmatrix}$$

avec l'origine placée au centre du Soleil.

Plus exactement, une spécifierons une position initiale $\vec{\mathbf{x}}_0 = (x_0, 0)$ et une vitesse initiale $\vec{\mathbf{v}}_0 = (0, v_0)$ de la Terre dans le plan de l'orbite et nous calculerons sa trajectoire orbitale dans le plan avec un programme `python`. Une méthode d'Euler explicite y sera utilisée pour obtenir la position et la vitesse de la Terre à des instants successifs $t_i = i\Delta t$ où l'intervalle de temps Δt est obtenu en divisant la période de rotation de la Terre par le nombre de pas n requis. En sélectionnant x_0 égal à la distance entre la Terre et le Soleil et en prenant v_0 égal à la vitesse de rotation de la Terre autour du Soleil, on obtiendra une orbite circulaire. La méthode d'Euler explicite pour obtenir la position de la Terre est définie par les deux relations de récurence ci-dessous :

$$\boxed{\begin{aligned} \vec{\mathbf{x}}_{i+1} &= \vec{\mathbf{x}}_i + \Delta t\ \vec{\mathbf{v}}_i \\ \vec{\mathbf{v}}_{i+1} &= \vec{\mathbf{v}}_i + \Delta t\ \vec{\mathbf{g}}(\vec{\mathbf{x}}_i) \end{aligned}}$$

où $\vec{\mathbf{x}}_i = \vec{\mathbf{x}}(i\Delta t)$ et $\vec{\mathbf{v}}_i = \vec{\mathbf{v}}(i\Delta t)$.

Pour obtenir une précision acceptable, nous utiliserons $n = 10000$ pour effectuer la rotation de la Terre. C'est une bonne idée de modifier cette valeur et d'observer que la précision augmente lorsque le nombre de pas augmente...

Des unités un peu particulières pour l'astrophysique...

Finalement, nous exprimerons toutes les grandeurs physiques en utilisant comme unité de masse, le kilogramme usuel, comme unité de temps, le jour et comme unité de distance, l'unité astronomique. L'unité astronomique correspond approximativement à la distance entre la Terre et le Soleil, soit environ 150 millions de kilomètres et est notée UA. Plus précisément, l'unité astronomique est définie comme valant exactement 149 597 870 700 mètres. Il est essentiel pour les calculs numériques d'exprimer les valeurs dans ces unités, car les très grands nombres obtenus dans les unités usuelles ne permettent pas d'effectuer le calcul sur une calculatrice ou un ordinateur qui ont une précision limitée et sont très peu doués pour manipuler des très grands ou de très petits nombres....

Et concrètement...

Votre mission consiste à :

1. Ecrire une fonction `G,M,m,v,T = earthData()` qui fournit la constante universelle de gravitation, la masse du Soleil, la masse de la Terre, la vitesse orbitale de la Terre et la période de rotation de la Terre en utilisant comme unités, le kilo, le jour et l'unité astronomique.

   Il n'y a rien à programmer : il suffit de calculer les bonnes valeurs en utilisant les ressources du livre de référence.

2. Ecrire ensuite une fonction `g = earthGravity(x,earthData)` qui fournit les deux composantes du champs gravitationnel pour un point `x` et en utilisant les paramètres matériels calculés par la fonction précédente founie en argument également. Il est essentiel de bien utiliser la fonction fournie pour calculer les paramètres matériels et de ne pas les recalculer dans la fonction car on souhaite que celle puisse fonctionner pour n'importe quelle étoile de l'Univers en modifiant ce dernier argument. Les arguments d'entrée `x` et de sortie `g` seront des vecteurs `numpy` de taille deux.

3. Finalement écrire une fonction `X = earthEuler(n,earthGravity,earthData)` qui calcule la trajectoire d'un planète soumise à un champs gravitationnel décrit par la fonction `earthGravity` et pour les paramètres matériels calculés par la fonction `earthData`. On utilisera `n` pas (ou itérations) pour calculer la trajectoire d'une orbite, en obtenant l'incrément temporel comme la période divisée par le nombre de pas :

   $$\Delta t = \frac{T}{n}$$

   Les positions aux temps $t_i = i\Delta t$ seront renvoyés dans un tableau `numpy` de taille $(n+1) \times 2$. Attention, les tableaux `numpy` sont numérotés à partir de zéro et la première ligne correspondra donc à la position initiale de la Terre. On prendre comme position initiale $\vec{\mathbf{x}}_0 = (1.0, 0.0)$ et $\vec{\mathbf{v}}_0 = (0, v)$. On sélectionnera comme vitesse initiale v la vitesse orbitale de la Terre qui aura été fournie par la fonction `earthData()`.

4. Ensuite pour tester, votre programme, on vous a fourni un tout petit programme tout simple `earthTest.py` qui devrait vous permettre de voir l'orbite circulaire de la Terre :-)

```
G,M,m,v,T = earthData()
n = 10000;
print(" === Gravitational constant = %9.2e [UA^3/kg day^2]" % G)
print(" === Mass of the Sun         = %9.2e [kg]" % M)
print(" === Mass of the Earth       = %9.2e [kg]" % m)
print(" === Speed of the Earth      = %9.2e [UA/day]" % v)
print(" === Period of the Earth     = %4.1f [day]" % T)
print(" === Number of time steps    = %d   " % n)
X = earthEuler(n,earthGravity,earthData)
print(" === Final position of the Earth after one revolution = [%4.1f,%4.1f] [UA]" % (X[-1,0],X[-1,1]))

import matplotlib
import matplotlib.pyplot as plt
matplotlib.rcParams['toolbar'] = 'None'
plt.figure("La Terre tourne autour du Soleil :-)")
plt.plot([0],[0],'ro',markerSize=20)
plt.plot(X[:,0],X[:,1],'-b')
plt.plot(X[-1,0],X[-1,1],'bo',markerSize=5)
plt.xlim(-1.5, 1.5)
plt.ylim(-1.5, 1.5)
plt.gca().set_aspect('equal',adjustable='box')
plt.axis("off")
plt.show()
```

5. Votre fonction (avec les éventuelles sous-fonctions que vous auriez créées) sera soumise sur `zouLab` sans y adjoindre le programme de test fourni ! Cette fonction devra être soumise via le web : ce travail est individuel et sera évalué. Cette évaluation permettra de gagner un demi-point bonus pour l'évaluation finale. Attention : toutes vos soumissions seront systématiquement analysées par un logiciel anti-plagiat. Faites vraiment votre programme seul... :-)

Si vous avez bien réalisé votre devoir, vous devriez obtenir la période de 365 jours pour la rotation de la Terre autour du Soleil et voir apparaître l'orbite circulaire sur votre écran, comme illustré sur la Figure de cet énoncé.

La méthode d'Euler explicite appliquée à la dynamique de la Terre :-)

En partant des équations de la dynamique de la Terre soumise à la force d'attraction du Soleil, nous pouvons écrire :

$$
\begin{aligned}
\frac{d\vec{\mathbf{x}}}{dt}(t) &= \vec{\mathbf{v}}(t)\\
\frac{d\vec{\mathbf{v}}}{dt}(t) &= \vec{\mathbf{g}}(\vec{\mathbf{x}}(t))
\end{aligned}
$$

$$\Big\downarrow \quad \text{En prenant un } \Delta t \text{ suffisamment petit...}$$

$$
\begin{aligned}
\frac{\vec{\mathbf{x}}(t+\Delta t)-\vec{\mathbf{x}}(t)}{\Delta t} &\approx \vec{\mathbf{v}}(t)\\
\frac{\vec{\mathbf{v}}(t+\Delta t)-\vec{\mathbf{v}}(t)}{\Delta t} &\approx \vec{\mathbf{g}}(\vec{\mathbf{x}}(t))
\end{aligned}
$$

$$\Big\downarrow \quad \text{En définissant } \vec{\mathbf{x}}_i = \vec{\mathbf{x}}(i\Delta t) \text{ et } \vec{\mathbf{v}}_i = \vec{\mathbf{v}}(i\Delta t)$$

$$
\begin{aligned}
\vec{\mathbf{x}}_{i+1} &= \vec{\mathbf{x}}_i + \Delta t\ \vec{\mathbf{v}}_i\\
\vec{\mathbf{v}}_{i+1} &= \vec{\mathbf{v}}_i + \Delta t\ \vec{\mathbf{g}}(\vec{\mathbf{x}}_i)
\end{aligned}
$$

Nous avons ainsi obtenu une manière numérique simple de calculer la trajectoire d'une orbite d'un champ gravitationnel connu. Toutefois pour que cette méthode (assez basique :-) fournisse des résultats acceptables en termes de précision, il est nécessaire d'avoir des pas de temps très petits et un grand nombre de pas pour calculer la trajectoire d'une révolution de la Terre. Typiquement, il faudra $n = 10000$ pour obtenir une orbite circulaire.... Le même calcul effectué avec un nombre beaucoup plus modeste de pas fournira une trajectoire totalement imprécise.

Il est aussi important de noter qu'utiliser des unités adéquates est primordial pour éviter d'entacher les résultats d'erreurs provenant du calcul en virgule flottante sur l'ordinateur.