

TP : simulation de planètes

Informatique pour tous

1. Écrire `import matplotlib.pyplot as plt` et `import numpy as np`
2. Écrire une fonction `cercle(c, R)` affichant un cercle de centre `c` (qui est un couple) et de rayon `R`. Cette fonction servira à afficher les planètes.
Pour cela on rappelle que `plt.plot(x, y)` dessine une ligne brisée passant par les points dont la liste des abscisses est `X` et la liste des ordonnées est `Y`.
On n'utilisera pas `plt.show()` dans `courbe`, mais seulement pour la tester.
3. Écrire une fonction `somme` renvoyant la somme des éléments d'une liste.
4. Écrire une fonction `ps(u, v)` renvoyant le produit scalaire de deux tableaux `u` et `v`.
5. Écrire une fonction `norme` renvoyant la norme d'un tableau.

Dans la suite, on représente une planète par une liste `[x, y, vx, vy, m]` à 5 éléments contenant les coordonnées de la planète, de son vecteur vitesse et sa masse.

6. Écrire une fonction `gravite(p1, p2)` renvoyant un tableau correspondant au vecteur de la force de gravité que la planète `p2` applique sur `p1`. On prendra $G = 1$.
7. Écrire une fonction `etape(L, dt)` ayant une liste `L` de planètes en argument et renvoyant une nouvelle liste contenant les planètes après un temps `dt`.
Il faut donc utiliser la méthode d'Euler, ce qui revient à dire que les nouvelles positions d'une planète sont $x + v_x \Delta t$, $y + v_y \Delta t$ et que sa nouvelle vitesse est $v_x + a_x \Delta t$, $v_y + a_y \Delta t$ (où l'accélération a_x et a_y sont obtenues par le PFD).
8. Écrire une fonction `simule(L, dt, n)` effectuant `n` étapes avec la liste `L` de planètes. À chaque étape il faut afficher les planètes (en utilisant `cercle`, avec par exemple un rayon de 1), et écrire `plt.show()`, `plt.pause(0.1)` (pour avoir le temps de voir l'animation) puis `plt.clf()` (pour mettre à jour le dessin).
Pour plus de clarté, on pourra écrire `plt.axis([0, 100, 0, 100])` (pour que les axes soient de 0 à 100) `plt.axis("equal")`.
9. Tester avec une étoile et une planète (par exemple avec la liste `L = [[50, 50, 0, 0, 70], [80, 50, 0, 20, 1]]`, où l'étoile est 70 fois plus massif), avec deux planètes de même poids (par exemple `L = [[25, 50, 0, -10, 30], [75, 50, 0, 10, 30]]`), en changeant les vitesses initiales...
On pourra prendre par exemple `dt = 0.1`