

PROGRAMMER EN PYTHON FICHE N°4 : DÉTERMINATION DE LA MASSE DU SOLEIL

Cette ressource anticipe pour certains aspects sur le programme de la spécialité physique-chimie de terminale.

L'étude porte sur le mouvement de Mercure, les positions étant obtenues sur le site des éphémérides pour 11 dates démarrant le 1er janvier 2017 et espacées de 8 jours. Les variations du « vecteur vitesse » au cours du temps permettent d'estimer une valeur de la masse solaire.

Capacité numérique

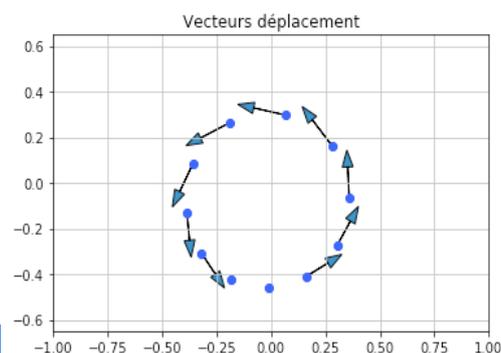
Utiliser un langage de programmation pour étudier la relation approchée entre la variation du vecteur vitesse d'un système modélisé par un point matériel entre deux instants voisins et la somme des forces appliquées sur celui-ci.

Représentation des vecteurs vitesse

Le script suivant fournit les directions et sens des vecteurs vitesse.

Un facteur d'échelle (5 ici) doit cependant être introduit pour que les vecteurs aient une longueur suffisante sur le graphe.

```
m=np.arange(len(t)-2)
for i in m :
    plt.arrow(x[i+1],y[i+1],5*(x[i+2]-x[i])/(t[i+2]-t[i]),5*(y[i+2]-y[i])/(t[i+2]-t[i]),head_width=0.05)
plt.plot(x,y,'bo')
plt.axis('equal')
plt.xlim(-1,1)
plt.ylim(-1,1)
plt.title("Vecteurs déplacement")
plt.grid()
plt.show()
```



Représentation des vecteurs « variation de vitesse »

Pour rendre l'étude quantitative, les valeurs de vitesses sont converties dans les unités correctes pour pouvoir définir les vecteurs variation de la vitesse. Pour cette manipulation des listes axl et ayl nécessite, on les transforme préalablement en tableau numpy : c'est le sens des lignes type :

```
ayln=1.5*10**11*np.array(ayl)/(24*3600)**2
```

Dans un deuxième temps, le script calcule, pour tous les points de mesure, la norme de la variation du vecteur vitesse. Ensuite, le produit de cette norme par le carré de la distance d^2 est calculé. D'après la deuxième loi de Newton (qui sera étudiée en enseignement de spécialité physique-chimie de la classe de terminale), en considérant Mercure uniquement soumis à l'action du Soleil, ce produit doit être égal à GMs, où G est la constante de gravitation universelle et Ms la masse du Soleil. Il faut ici prendre garde à redimensionner les listes de distances pour qu'elles contiennent autant de valeurs que de valeurs calculées d'accélération.

Il faut aussi prendre garde à bien faire correspondre les valeurs de distance et d'accélération calculées en décaler la liste des distances (c'est le sens de la commande `dr=d[2:]` qui enlève les 2 premiers termes de la liste d) et de `x[:-2]`, qui enlève les 2 derniers termes de la liste x).

On peut ensuite faire modéliser la fonction précédente par une fonction constante et faire afficher la valeur expérimentale de la masse du Soleil. Pour pouvoir faire varier l'échelle à loisir, il vaut mieux définir des grandeurs plus petites (ce qui explique la présence de 10^{20}).

Remarque : la commande `"%.1e"%M` sert à ne garder qu'un chiffre après la virgule pour M.

```
ayln=1.5*10**11*np.array(ayl)/(24*3600)**2
axln=1.5*10**11*np.array(axl)/(24*3600)**2
norma=(ayln**2+axln**2)**(1/2)
dx=x[:-2]*1.5*10**11
dy=y[:-2]*1.5*10**11
d=(dx**2+dy**2)**(1/2)

dr=d[2:]
GM=dr**2*norma
GMth=6.67*10**(-11)*2*10**30
n=np.arange(len(t)-4)
plt.plot(n,GM*10**(-20),'bo',label="points expérimentaux")
plt.legend()
plt.grid()

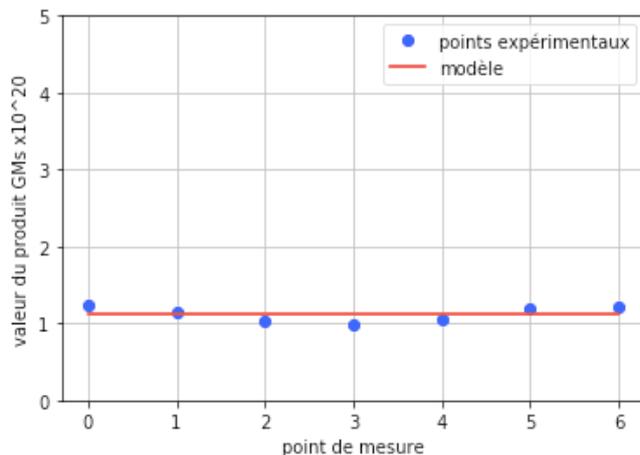
mod=np.polyfit(n,GM,0)
M=float(mod[0]/(6.67*10**(-11)))
print('la masse du soleil est de l ordre de', "%.1e"%M, 'kg')
GMmod=0*n+mod[0]
plt.plot(n,(GMmod)*10**(-20),'r-',label="modèle")
plt.legend()
plt.ylim(0,5)
plt.xlabel("point de mesure")
plt.ylabel("valeur du produit GMs x10^20")
```

Retrouvez éducol sur :



la masse du soleil est de l'ordre de 1.7×10^{30} kg

```
Text(0,0.5,'valeur du produit GMs x10^20')
```



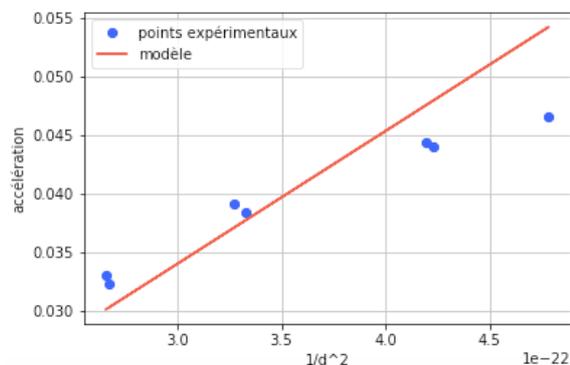
Conclusion

La masse obtenue est du même ordre de grandeur de la valeur tabulée.

Il est possible de faire tracer la norme de l'accélération en fonction de $1/\text{distance}^2$, pour vérifier que l'évolution est pratiquement linéaire, la pente de la fonction modèle étant alors le facteur GMs avec Ms obtenue précédemment.

```
drci=1/(dr)**2
plt.plot(drci,norma,'bo',label="points expérimentaux")
plt.legend()
plt.grid()
model=6.67*10**(-11)*1.7*10**30*drci
plt.plot(drci,model,'r-',label="modèle")
plt.legend()
plt.xlabel("1/d^2")
plt.ylabel("accélération")
```

```
Text(0,0.5,'accélération')
```



Retrouvez éduscol sur :

