

Corrigé devoir maison n°10

EXERCICE 1 : Charge d'un condensateur

Code associé (sur Capytale) : d016-5172079

1. Il suffit de prendre l'équation différentielle :

$$\frac{du_c}{dt} = \frac{E - u_c}{\tau} = f(u_c, t) \quad \text{où} \quad f(u_c, t) = \frac{E - u_c}{\tau}$$

Cela montre que, connaissant la valeur de u_c à un instant donné, il est possible de connaître la valeur de la dérivée première.

À l'aide de la méthode d'Euler, on peut obtenir :

$$u_c(t_{i+1}) = u_c(t_i) + f(u_c(t_i), t_i)h$$

2. à 7. Exemple de code :

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # Définition de la méthode d'Euler
5
6 def euler(f, y0, t0, tf, n):
7     # f : fonction de l'eq. diff.
8     # y0 : condition initiale
9     # t0 : instant initial
10    # tf : instant final
11    # n : nombre de pas de calculs
12
13
14    #Définition de la liste des temps (question 2)
15    liste_t = np.linspace(t0, tf, n)
16
17    #Définition du pas de calcul (question 3)
18    h = (tf-t0)/n
19
20    #Définition d'une liste de zéros
21    liste_y = np.zeros(n)
22    liste_y[0] = y0
23
24    #Remplissage de liste_y, calculé par récurrence (question 4)
25    for i in range(n-1):
26        liste_y[i+1]=liste_y[i]+f(liste_y[i],liste_t[i])*h
27
28    return liste_t, liste_y
```

```
29
30
31 # Définition de la fonction f (question 5)
32 def f_charge(uc,t):
33     # uc : valeur de uc à un instant t
34     # t : instant
35     return (5-uc)/10 # en V/ms
36
37 #Définition des paramètres de calcul (question 6)
38 y0= 0
39 t0 = 0
40 tf = 70
41 n = 1000
42
43 liste_t, liste_uc = euler(f_charge, y0, t0, tf, n)
44
45 plt.clf()
46 plt.plot(liste_t, liste_uc, '+k') # '+k' = courbe en croix noires
47
48 #Tracé de la solution analytique
49 liste_uc_th = 5*(1-np.exp(-liste_t/10)) # calcul de la solution
50 plt.plot(liste_t, liste_uc_th, '-r') # '-r' = courbe reliée en rouge
51
52 plt.xlabel('Temps (en ms)')
53 plt.ylabel('Tension u_c (en V)')
54
55 plt.show()
```

8. Dans la figure 1, on identifie bien la réponse exponentielle caractéristique de l'évolution du circuit RC. On trouve que la solution calculée par la méthode d'Euler (croix noires) reproduit bien l'évolution exacte obtenue par la calcul (courbe rouge). En diminuant le nombre de pas n de la méthode d'Euler, la solution calculée devient moins précise.

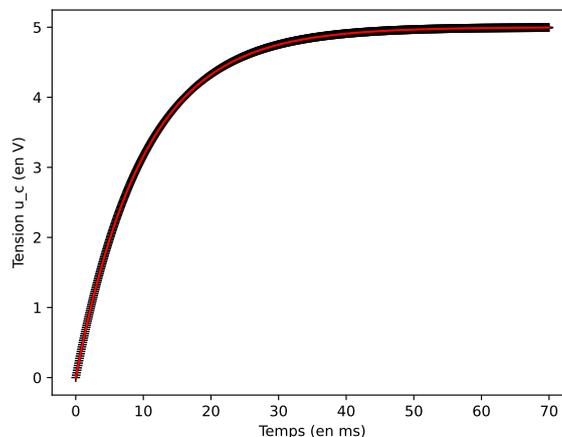


FIGURE 1 – Solution calculée avec la méthode d'Euler (croix noires) comparée à la solution analytique (courbe rouge).

9. En introduisant $\tilde{u}_c = \frac{u_c}{E}$ et $\tilde{t} = \frac{t}{\tau}$, on trouve :

$$\frac{d\tilde{u}_c}{d\tilde{t}} = \frac{du_c}{dt} \times \frac{\tau}{E}$$

Remarquons que $\frac{d\tilde{u}_c}{d\tilde{t}}$ est bien sans dimension.

10. L'équation différentielle de la charge du condensateur devient :

$$\frac{du_c}{dt} + \frac{u_c}{\tau} = \frac{d\tilde{u}_c}{d\tilde{t}} \times \frac{E}{\tau} + \tilde{u}_c \frac{E}{\tau} = \frac{E}{\tau}$$

En divisant tous les termes par E/τ , on obtient :

$$\frac{d\tilde{u}_c}{d\tilde{t}} + \tilde{u}_c = 1$$

L'équation obtenue est bien une équation adimensionnée.

EXERCICE 2 : Étude d'un pendule pesant

Code associé (sur Capytale) : 8159-6098345

- Il faut ici reprendre son cours de mécanique du solide. Il faut appliquer un théorème du moment cinétique par rapport à l'axe (Oz). On obtient alors l'équation demandée, avec $\omega_0^2 = \frac{mg\ell}{J}$.
- On cherche à exprimer $\frac{dX}{dt}$. On obtient à l'aide de l'équation différentielle :

$$\frac{dX}{dt} = \begin{pmatrix} \dot{\theta} \\ \ddot{\theta} \end{pmatrix} = \begin{pmatrix} \dot{\theta} \\ -\omega_0^2 \sin(\theta) \end{pmatrix}$$

On voit que les éléments de $\frac{dX}{dt}$ se calculent à l'aide des éléments de X . On peut donc écrire que : $\frac{dX}{dt} = F(X, t)$, où :

$$F\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, t\right) = \begin{pmatrix} x_2 \\ -\omega_0^2 \sin(x_1) \end{pmatrix}$$

3. à 7. Exemple de code :

[Cellule 1]

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.integrate import odeint
4
5 J = 0.07 #kg.m^2
6 m = 1.00 #kg
7 l = 0.15 #m
8 g = 9.81 #m.s^-2
9
10 w0= m*g*l/J

```

[Cellule 2]

```

1 def F(X,t):
2     # X : vecteur (liste ou np.array) comportant (x et dx/dt)
3     # t : instant (flottant) où est calculé la dérivée
4
5     return [X[1] , -(w0**2)*np.sin(X[0])]
6
7 #Paramètres du calcul
8 t0 = 0     # instant initial, en s
9 tf = 10    # instant final, en s
10 N = 1000  # nombre de pas
11
12 #Création de la liste de instants où seront calculées les solutions
13 liste_t = np.linspace(t0, tf, N)
14
15 #Conditions initiales
16 theta0 = 10*np.pi/180 # en rad, position initiale
17 thetap0 = 0            # en rad/s , vitesse angulaire initiale
18 X0 = [theta0, thetap0] # liste des cconditions initiales
19
20 #Résolution de l'équation différentielle
21 X_sol = odeint(F, X0, liste_t)

```

[Cellule 3]

```

1 #Représentation de la solution
2 plt.clf()
3
4 plt.plot(liste_t, X_sol[:,0], linestyle='-',label='theta =f(t)')
5
6 #Ajouter titre et unité aux axes
7 plt.xlabel('Temps (s)')
8 plt.ylabel('Theta (rad)')
9
10 #Affichage
11 plt.legend(loc='upper right')
12 plt.show()

```

7. On obtient l'évolution représentée en figure 2(a). On remarque que le système oscille autour de la position $\theta = 0$, avec une évolution semblant correspondre à des oscillations harmoniques.
8. Dans le cas où on se limite à des mouvement de faibles amplitudes, $|\theta(t)| \ll 1$ (en rad), alors on peut développer l'équation du mouvement à l'ordre 1 : $\sin(\theta) \approx \theta$, ce qui donne :

$$\ddot{\theta} + \omega_0^2 \theta = 0$$

On obtient l'équation différentielle d'un oscillateur harmonique. La solution générale s'écrit $\theta(t) = A \cos(\omega_0 t) + B \sin(\omega_0 t)$. Or les conditions initiales sont $\theta(0) = \theta_0$ et $\dot{\theta}(0) = 0$, d'où la solution :

$$\theta(t) = \theta_0 \cos(\omega_0 t)$$

9. Exemple de code : [Cellule 4] Comparaison avec l'approximation des petits angles

```

1  #Représentation de la solution
2  plt.clf()
3
4  plt.plot(liste_t, X_sol[:,0], linestyle='-',label='Sans approximation')
5  plt.plot(liste_t, theta0*np.cos(w0*liste_t),label='Petits angles')
6
7  #Ajouter titre et unité aux axes
8  plt.xlabel('Temps (s)')
9  plt.ylabel('Theta (rad)')
10
11 #Affichage
12 plt.legend(loc='upper right')
13 plt.show()

```

10. Comme observé sur la figure 2(b), on observe une bonne correspondance entre le résultat calculé avec la fonction `odeint` et le résultat littéral dans le cas des petits angles. L'approximation faite est donc satisfaisante dans ce cas.

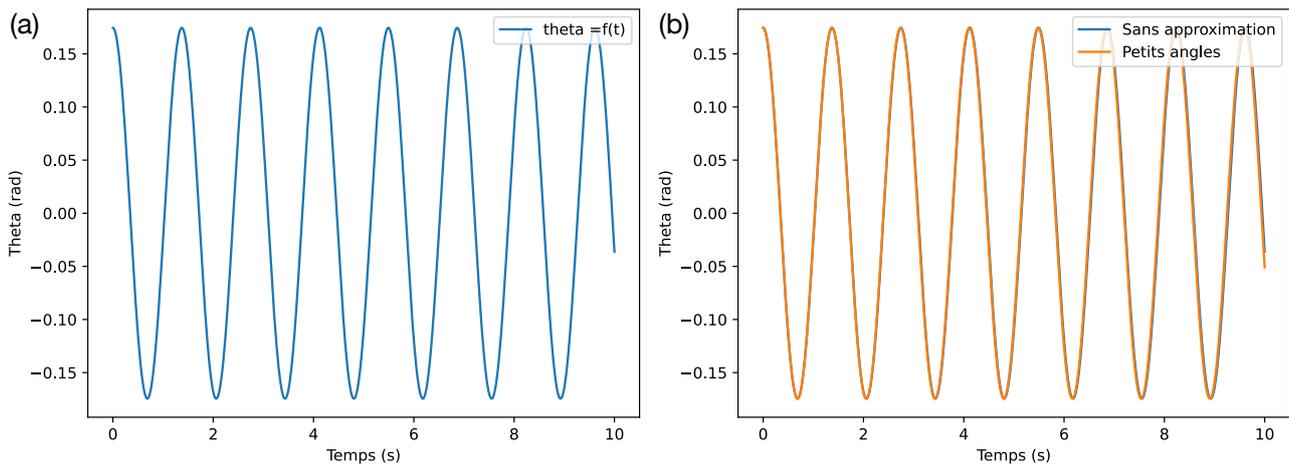


FIGURE 2 – (a) Évolution de $\theta = f(t)$ pour $\theta_0 = 10^\circ$. (b) Comparaison avec la solution de l'approximation des petits angles.

11. Exemple de code : [Cellule 5] Comparaison pour différentes positions initiales

```

1  plt.clf()
2
3  for theta0 in [10, 60, 90]: # pour chaque theta0, tracer l'évolution de theta(t)
4      X0 = [theta0*np.pi/180, thetap0]
5      X_sol = odeint(F, X0, liste_t)
6      plt.plot(liste_t, X_sol[:,0], linestyle='-',label='theta_0='+str(theta0)+'°')
7
8  plt.xlabel('Temps (s)')
9  plt.ylabel('Theta (rad)')
10

```

```

11 #Affichage
12 plt.legend(loc='upper right')
13 plt.show()

```

12. Le résultat obtenu est donné dans la figure 3(a). On remarque que l'amplitude des oscillations varie : plus θ_0 est important, plus l'amplitude des oscillations est grande. De plus, on remarque que les oscillations n'ont pas la même période : la période est plus longue lorsque θ_0 devient important. La période dépend donc des conditions initiales : **il n'y a pas isochronisme des oscillations** (un isochronisme approché est valable pour les petits angles).
13. Il suffit ici d'introduire un temps adimensionné. La grandeur caractéristique que l'on peut utiliser est ω_0 , en remarquant que $\tilde{t} = \omega_0 t$ est bien sans dimension. Ainsi :

$$\frac{d^2\theta}{dt^2} + \omega_0^2 \sin(\theta) = \omega_0^2 \frac{d^2\theta}{d\tilde{t}^2} + \omega_0^2 \sin(\theta) = 0 \implies \boxed{\frac{d^2\theta}{d\tilde{t}^2} + \sin(\theta) = 0}$$

On aurait pu également choisir d'introduire $\tilde{t} = t/T_0$, où $T_0 = \frac{2\pi}{\omega_0}$. L'équation adimensionnée obtenue ferait alors intervenir un facteur 2π .

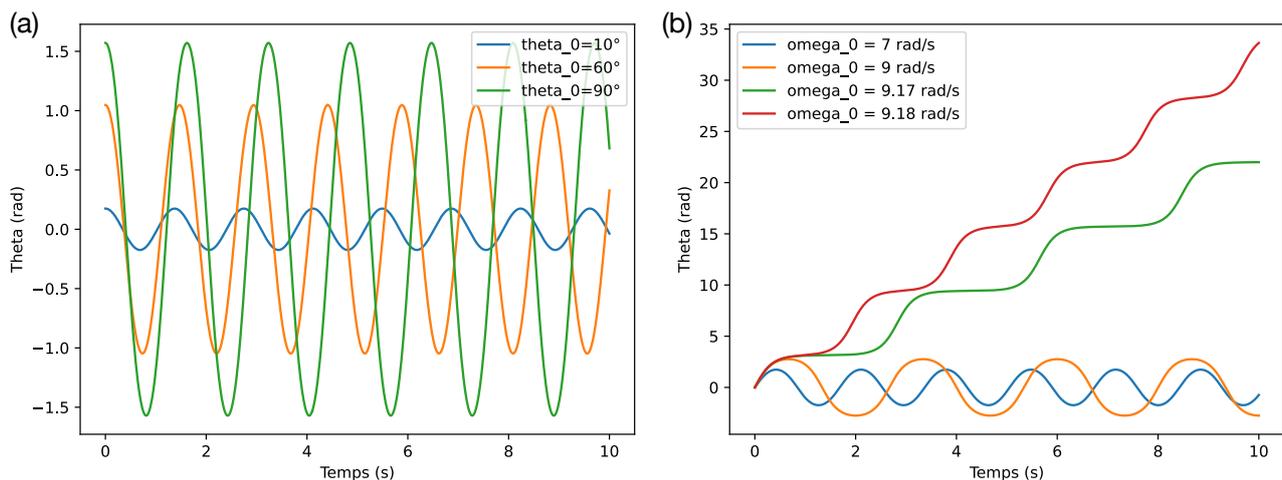


FIGURE 3 – (a) Influence de la position initiale sur la période des oscillations. (b) Influence de la vitesse initiale sur le mouvement du pendule.

Cas d'une vitesse initiale non nulle

14. Exemple de code : [Cellule 6]

```

1 N= 1000 #nombre de pas
2 t0= 0 #instant initial
3 tf= 15 #instant final, en s
4
5 plt.clf()
6
7 theta0=0
8 for thetap0 in [7, 9, 9.17, 9.18]: #en rad/s ; tracer l'évolution pour différents thetap0

```

```

9     X0 = [theta0, thetap0]
10    X_sol = odeint(F, X0, liste_t)
11    plt.plot(liste_t, X_sol[:,0], linestyle='-', label='omega_0 = '+str(thetap0)+' rad/s')
12
13    plt.xlabel('Temps (s)')
14    plt.ylabel('Theta (rad)')
15
16    plt.legend()
17    plt.show()

```

15. Le résultat est donné dans la figure 3(b). On remarque deux types de comportement : pour $\dot{\theta}_0 = 7$ ou 9 rad/s , le mouvement est oscillant (ou pendulaire), la position du pendule oscille entre deux angles extrêmes. Lorsqu'on augmente la vitesse de rotation initiale, soit pour $\dot{\theta}_0 = 9,17$ ou $9,18 \text{ rad/s}$, $\theta(t)$ ne fait que croître : on n'a donc plus un mouvement oscillant mais un mouvement révolutif. Le système fait donc des tours complets autour de l'axe de rotation (Oz).

Ce régime de mouvement révolutif n'est possible que si l'énergie mécanique est suffisamment importante pour que, lorsque le pendule atteint $\theta = 180^\circ$, la vitesse de rotation ne s'annule pas. Le système est conservatif, donc l'énergie mécanique est constante au cours du mouvement :

$$E_m = E_{m,0} \quad \text{avec} \quad E_{m,0} = \frac{1}{2} J \dot{\theta}_0^2 + E_p(\theta_0)$$

où $E_{m,0}$ est l'énergie mécanique initiale. Or ici, $E_p(\theta) = mg\ell(1 - \cos(\theta))$ (on a choisi l'origine des énergies potentielles pour $\theta = 0$). Ainsi $E_{m,0} = \frac{1}{2} J \dot{\theta}_0^2$.

À la limite, l'énergie mécanique initiale permet au pendule d'atteindre $\theta = 180^\circ$ avec $\dot{\theta} = 0$. Soit :

$$E_{m,0c} = E_p(\theta = 180^\circ) \implies \frac{1}{2} J \dot{\theta}_{0,c}^2 = 2mg\ell \implies \dot{\theta}_{0,c} = \sqrt{\frac{4mg\ell}{J}}$$

16. Exemple de code : [Cellule 7] Cas où $\dot{\theta}_0 = 7 \text{ rad}\cdot\text{s}^{-1}$

```

1     thetap0 = 7 #rad/s
2     theta0=0
3
4     X0 = [theta0, thetap0]
5
6     X_sol = odeint(F, X0, liste_t)
7
8     liste_ec = 1/2*J*X_sol[:,1]**2 #liste de l'énergie cinétique
9     liste_ep = m*g*l*(1-np.cos(X_sol[:,0])) #liste de l'énergie potentielle
10    liste_em = liste_ec + liste_ep
11
12    plt.clf()
13    plt.plot(liste_t, liste_ec, label='Énergie cinétique')
14    plt.plot(liste_t, liste_ep, label='Énergie potentielle')
15    plt.plot(liste_t, liste_em, label='Énergie mécanique')
16
17    plt.xlabel('Temps (s)')
18    plt.ylabel('Energie (J)')

```

```

19 plt.legend()
20 plt.show()

```

17. Exemple de code : [Cellule 8] Cas où $\dot{\theta}_0 = 10 \text{ rad}\cdot\text{s}^{-1}$

```

1  thetap0 = 10 #rad/s
2  theta0=0
3
4  X0 = [theta0, thetap0]
5
6  X_sol = odeint(F, X0, liste_t)
7
8
9  liste_ec = 1/2*J*X_sol[:,1]**2 #liste de l'énergie cinétique
10 liste_ep = m*g*l*(1-np.cos(X_sol[:,0])) #liste de l'énergie potentielle
11 liste_em = liste_ec + liste_ep
12
13 plt.clf()
14 plt.plot(liste_t, liste_ec, label='Énergie cinétique')
15 plt.plot(liste_t, liste_ep, label='Énergie potentielle')
16 plt.plot(liste_t, liste_em, label='Énergie mécanique')
17
18 plt.xlabel('Temps (s)')
19 plt.ylabel('Energie (J)')
20 plt.legend()
21 plt.show()

```

18. Pour $\dot{\theta}_0 < \dot{\theta}_{0,c}$ (figure 4(a)), le mouvement est pendulaire, le pendule oscille entre deux positions maximales. Lorsque le pendule atteint une position extrême, la vitesse s'annule et le pendule "repart dans l'autre sens". Ainsi, lorsque l'énergie potentielle est maximale, l'énergie

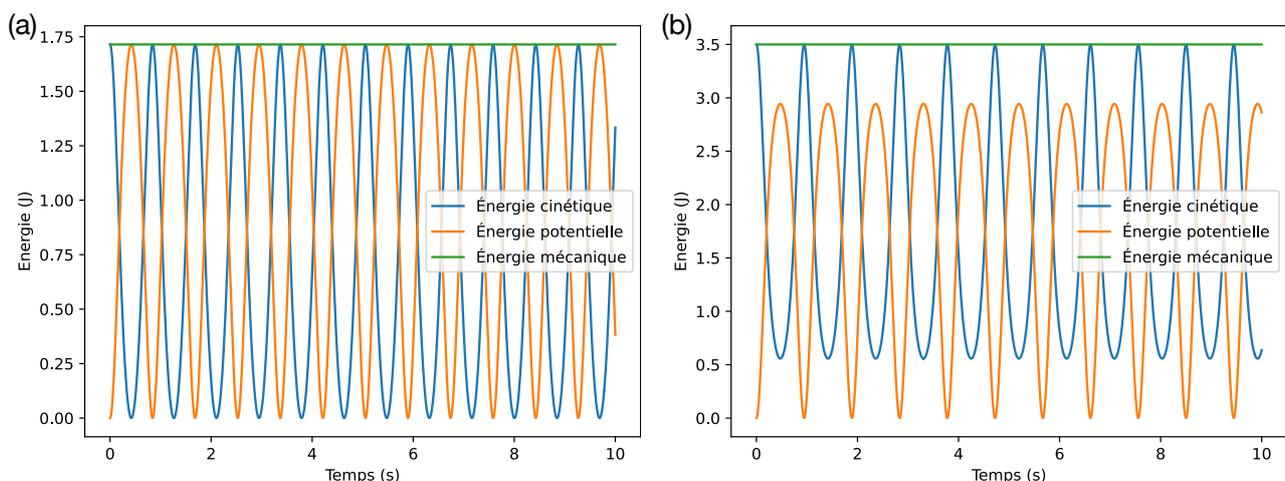


FIGURE 4 – Évolution des énergies cinétique, potentielle et mécanique pour (a) $\dot{\theta}_0 = 7 \text{ rad/s}$ et (b) $\dot{\theta}_0 = 10 \text{ rad/s}$.

cinétique s'annule. Au cours du mouvement, il y a une conversion réciproque d'énergie cinétique en énergie potentielle. Comme le mouvement est conservatif, l'énergie mécanique reste constante, comme observé sur le graphe.

Pour $\dot{\theta}_0 > \dot{\theta}_{0,c}$ (figure 4(b)), le mouvement est révolutif, le pendule tourne en continu autour de son axe de rotation. La vitesse de rotation ne s'annule donc jamais au cours du mouvement (elle atteint toutefois un minimum lorsque le pendule atteint $\theta = \pi [2\pi]$). La conversion réciproque d'énergie cinétique en énergie potentielle est donc partielle. L'énergie mécanique reste encore constante, le mouvement est toujours conservatif.

19. et 20. Exemple de code : **[Cellule 9]** Cas où $\dot{\theta}_0 = \dot{\theta}_{0,c}$

```

1  #Paramètres du calcul
2  t0 = 0      # instant initial, en s
3  tf = 50    # instant final, en s
4  N = 1000   # nombre de pas
5
6  #Création de la liste de instants où seront calculées les solutions
7  liste_t = np.linspace(t0, tf, N)
8
9  thetap0 = np.sqrt(4*m*g*l/J)
10 theta0=0
11
12 X0 = [theta0, thetap0]
13
14 X_sol = odeint(F, X0, liste_t)
15
16 plt.clf()
17 plt.plot(liste_t, X_sol[:,0], linestyle='-',label='omega_0 = '+f'{thetap0:.5}'+ ' rad/s')
18
19 plt.xlabel('Temps (s)')
20 plt.ylabel('Theta (rad)')
21 plt.legend()
22 plt.show()

```

21. Compte tenu des conditions initiales imposées, on s'attend à ce que le pendule tende vers la position $\theta = 180^\circ$ lorsque $t \rightarrow 0$ (car il atteint cette position avec une vitesse nulle). On s'attend alors à avoir une asymptote horizontale $\theta = \pi$ pour $t \rightarrow \infty$. Or quelle que soit la valeur de N choisie, on n'obtient pas le comportement attendu, comme présenté sur la figure 5.

De plus, en faisant varier N le nombre de pas de calculs, le comportement calculé varie, résultant en des mouvement par moments révolutifs et par moments pendulaires.

L'écart entre le calcul et la théorie provient de la méthode de résolution de l'équation différentielle : celle-ci introduit une erreur qui s'accumule au fil du calcul. Cette erreur empêche l'étude précise du cas limite. C'est pour la même raison que l'on observe une sensibilité au nombre de pas N : le système passe par moment d'un côté de la limite (mouvement pendulaire) à l'autre (mouvement révolutif) au cours de la résolution.

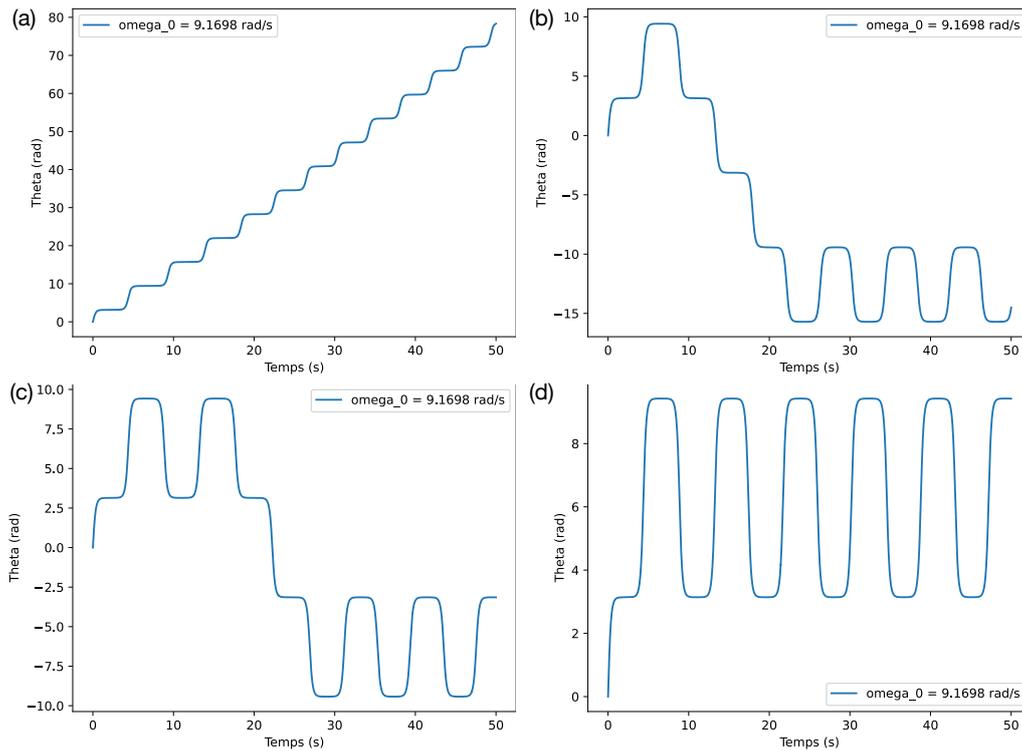


FIGURE 5 – Évolution de $\theta(t)$ pour (a) $N = 1000$, (b) $N = 1001$, (c) $N = 1003$ (d) $N = 1005$.

Prise en compte de frottements

22. On applique le théorème du moment cinétique par rapport à l'axe (Oz) comme dans le cas sans frottement, en ajoutant cette fois un couple de frottement fluide :

$$J\ddot{\theta} + \lambda\dot{\theta} + mgl \sin(\theta) = 0$$

On obtient l'équation différentielle demandée en introduisant $k = \frac{\lambda}{\omega_0 J}$.

23. Exemple de code :

[Cellule 10] Calcul de la solution avec des frottements

```

1 k = 0.1
2
3 def F(X,t):
4     # X : vecteur (liste ou np.array) comportant (x et dx/dt)
5     # t : instant (flottant) où est calculé la dérivée
6
7     return [X[1] , -k*w0*X[1]-(w0**2)*np.sin(X[0])]
8
9 #Paramètres du calcul
10 t0 = 0 # instant initial, en s
11 tf = 10 # instant final, en s
12 N = 1000 # nombre de pas
13
14 #Création de la liste de instants où seront calculées les solutions
15 liste_t = np.linspace(t0, tf, N)

```

```

16
17 #Conditions initiales
18 theta0 = 10*np.pi/180 # en rad, position initiale
19 thetap0 = 0 # en rad/s , vitesse angulaire initiale
20 X0 = [theta0, thetap0] # liste des cnoditions initiales
21
22 #Résolution de l'équation différentielle
23 X_sol = odeint(F, X0, liste_t)

```

24. Exemple de code : [Cellule 11] Affichage de $\theta(t)$

```

1 #Représentation de la solution
2 plt.clf()
3
4 plt.plot(liste_t, X_sol[:,0], linestyle='-',label='theta =f(t)')
5 plt.plot(liste_t, theta0*np.exp(-(k*w0/2)*liste_t),'--k') #enveloppe supérieure
6 plt.plot(liste_t, -theta0*np.exp(-(k*w0/2)*liste_t),'--k') #enveloppe inférieure
7
8 #Ajouter titre et unité aux axes
9 plt.xlabel('Temps (s)')
10 plt.ylabel('Theta (rad)')
11
12 #Affichage
13 plt.legend(loc='upper right')
14 plt.show()

```

25. La courbe obtenue est donnée en figure 6. On obtient bien le comportement d'un oscillateur amorti en régime pseudo-périodique : $\theta(t)$ oscille, avec une amplitude qui décroît exponentiellement au cours du temps. En résolvant l'équation différentielle, on peut prédire l'évolution de l'enveloppe exponentielle, dont le temps caractéristique de décroissance est $\tau = \frac{2}{k\omega_0}$.

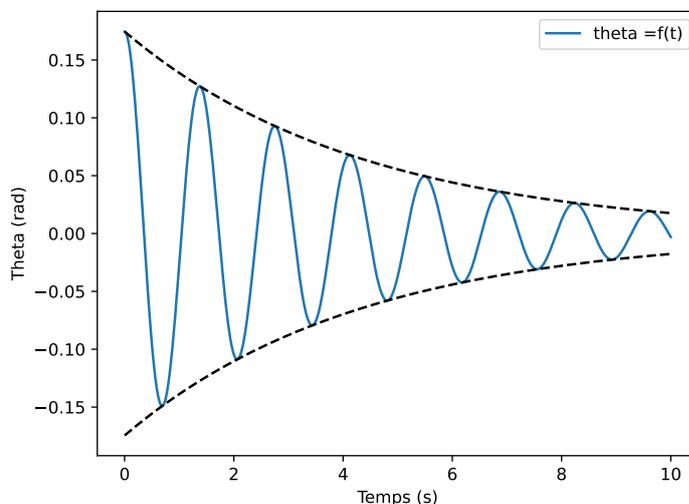


FIGURE 6 – Évolution de $\theta(t)$ pour $k = 0, 1$.

26. et 27. En faisant varier k , on fait varier le coefficient d'amortissement du système. Plus k est faible, plus les frottements sont faibles, et moins la décroissance exponentielle est rapide, comme le montre la figure 7(a). En augmentant k , on augmente les frottements, l'atténuation est plus rapide. À la limite où k devient grand, on n'observe plus d'oscillations : on se trouve alors dans un régime aperiodique, comme le montre la figure 7(d).

On aurait également pu introduire le facteur de qualité du système, afin de définir rigoureusement la frontière entre les différents régimes : $Q = \frac{1}{k}$. Ainsi pour $Q < 1/2$, soit $k > 2$, on se trouve avec un régime aperiodique.

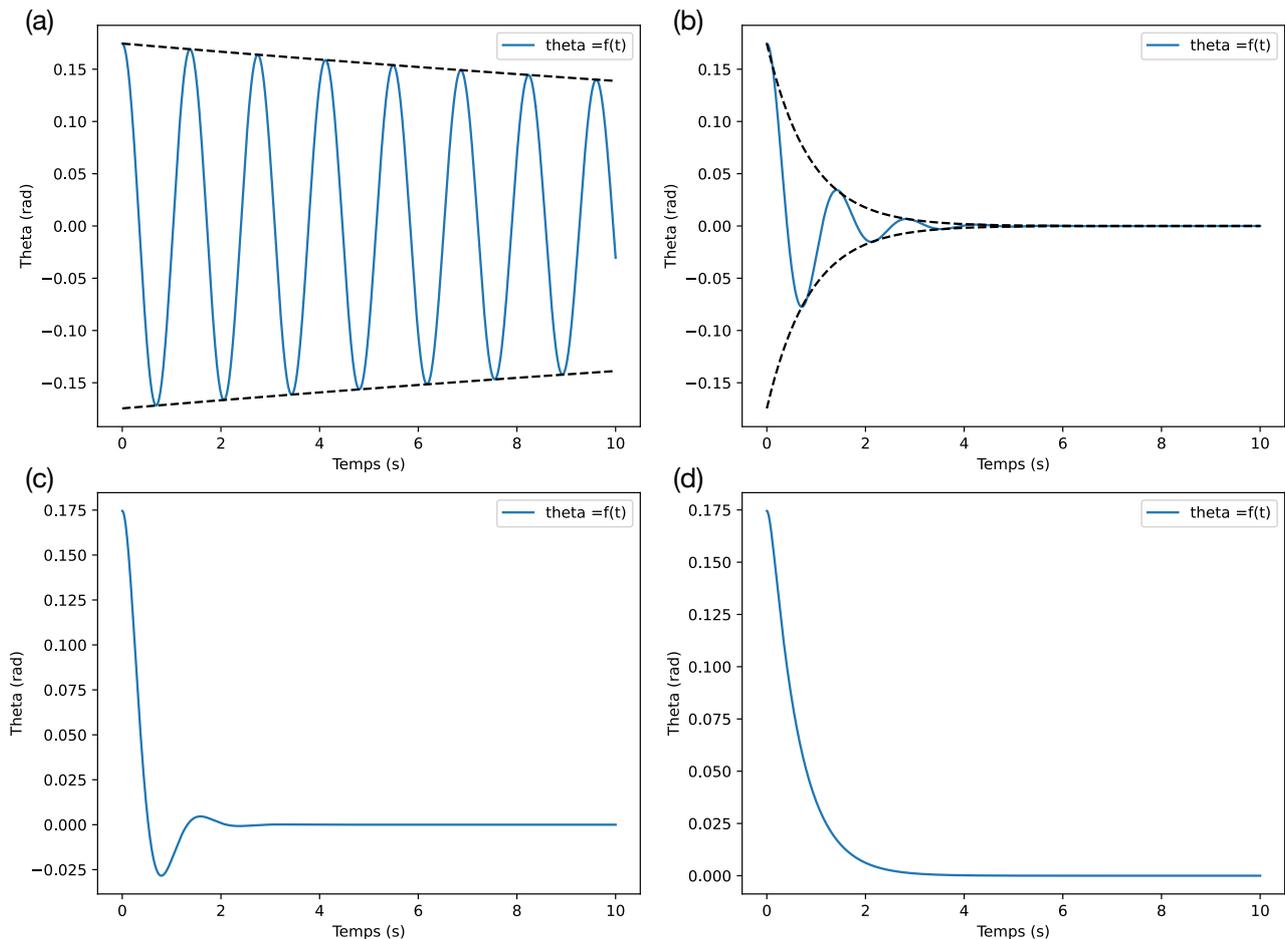


FIGURE 7 – Évolution de $\theta(t)$ pour (a) $k = 0,01$, (b) $k = 0,5$, (c) $k = 1$ et (d) $k = 3$.

28. Exemple de code : [Cellule 11] Évolution des énergies

```

1  liste_ec = 1/2*J*X_sol[:,1]**2 #liste de l'énergie cinétique
2  liste_ep = m*g*l*(1-np.cos(X_sol[:,0])) #liste de l'énergie potentielle
3  liste_em = liste_ec + liste_ep
4
5  plt.clf()
6  plt.plot(liste_t, liste_ec, label='Énergie cinétique')
7  plt.plot(liste_t, liste_ep, label='Énergie potentielle')
8  plt.plot(liste_t, liste_em, label='Énergie mécanique')
9
10 plt.xlabel('Temps (s)')
```

```

11 plt.ylabel('Energie (J)')
12 plt.legend()
13 plt.show()

```

29. Les courbes d'énergies cinétique et potentielle évoluent en opposition de phase entre des maxima et des annulations. On a donc toujours une conversion réciproque d'énergie cinétique en énergie potentielle au cours du mouvement pendulaire.

En revanche, la somme des deux courbes (l'énergie mécanique) décroît au cours du mouvement, car le système n'est plus conservatif : les frottements dissipent de l'énergie.

On remarque en particulier que la décroissance de l'énergie mécanique est maximale lorsque l'énergie cinétique est maximale : comme la force de frottement est proportionnelle à la vitesse, la puissance dissipée par les frottements est maximale lorsque la vitesse est maximale.

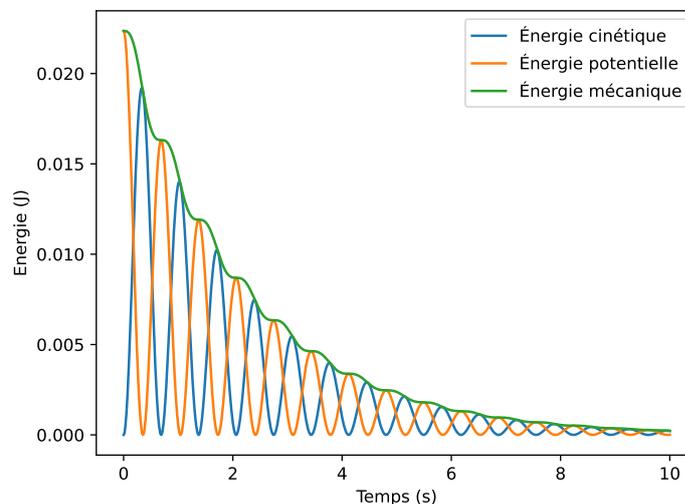


FIGURE 8 – Évolution des énergies cinétique, potentielle et mécanique dans le cas où $k = 0, 1$.

EXERCICE 3 : Mouvement d'un satellite autour de la Terre

Code associé (sur Capytale) : fa61-6205292

- On peut passer par un PFD appliqué dans le cas où le mouvement est circulaire. Sinon, on peut passer par une approche énergétique. Sachant que le rayon de la trajectoire est x_A , l'énergie mécanique du satellite s'exprime donc : $E_m = -GM_T m / (2x_A)$. Ainsi :

$$E_c + E_p = E_m \implies \frac{1}{2} m v_{0,c}^2 - \frac{GM_T m}{x_A} = -\frac{GM_T m}{2x_A} \implies v_{0,c} = \sqrt{\frac{GM_T}{x_A}}$$

L'application numérique donne $v_{0,c} \approx 6,89 \times 10^3$, m/s.

- On réalise le même raisonnement qu'à la question précédente, sachant que l'énergie mécanique dans le cas d'une trajectoire parabolique est nulle. Alors :

$$\frac{1}{2} m v_{0,\text{lib}}^2 - \frac{GM_T m}{x_A} = 0 \implies v_{0,\text{lib}} = \sqrt{\frac{2GM_T}{x_A}}$$

L'application numérique donne : $v_{0,\text{lib}} \simeq 9,74 \times 10^3 \text{ m/s}$.

3. En appliquant le PFD au satellite en se plaçant dans le référentiel géocentrique supposé galiléen, on obtient :

$$m\vec{a} = -\frac{GM_T m}{r^2} \vec{e}_r \implies \boxed{\vec{a} = -\frac{GM_T}{r^2} \vec{e}_r}$$

4. On projette le vecteur \vec{a} dans la base cartésienne :

$$\vec{a} = -\frac{GM_T}{r^3} (x\vec{e}_x + y\vec{e}_y) = -\frac{GM_T}{(x^2 + y^2)^{3/2}} (x\vec{e}_x + y\vec{e}_y)$$

En projetant selon les vecteurs de la base :

$$\begin{aligned} \ddot{x} &= -\frac{GM_T x}{(x^2 + y^2)^{3/2}} \\ \ddot{y} &= -\frac{GM_T y}{(x^2 + y^2)^{3/2}} \end{aligned}$$

5. En introduisant le vecteur X comme décrit dans l'énoncé, on peut alors exprimer :

$$\frac{dX}{dt} = \begin{pmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{d\dot{x}}{dt} \\ \frac{d\dot{y}}{dt} \end{pmatrix} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ -\frac{GM_T x}{(x^2 + y^2)^{3/2}} \\ -\frac{GM_T y}{(x^2 + y^2)^{3/2}} \end{pmatrix}$$

On remarque alors que les éléments de $\frac{dX}{dt}$ s'expriment intégralement à l'aide des éléments de

X (que sont x, y, \dot{x}, \dot{y}). Donc $\boxed{\frac{dX}{dt} = F(X, t)}$ où :

$$F \left(\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \right) = \begin{pmatrix} x_3 \\ x_4 \\ -\frac{GM_T x_1}{(x_1^2 + x_2^2)^{3/2}} \\ -\frac{GM_T x_2}{(x_1^2 + x_2^2)^{3/2}} \end{pmatrix}$$

6. à 9. Exemple de code :

```

1 import matplotlib.pyplot as plt
2 from scipy.integrate import odeint
3
4 G = 6.67e-11 #SI, constante de gravitation universelle
5 Mt = 5.97e24 #kg, masse de la Terre
6 Rt = 6370e3 #m, rayon de la Terre
7
8 def F(X,t):
9     # X : vecteur (liste ou np.array) comportant (x et dx/dt)
10    # t : instant (flottant) où est calculé la dérivée

```

```

11
12     return [X[2], X[3], -G*Mt*X[0]/(X[0]**2+X[1]**2)**(3/2),
13            -G*Mt*X[1]/(X[0]**2+X[1]**2)**(3/2)]
14
15 #Conditions initiales
16 xA= 8400e3
17 v0c = np.sqrt(G*Mt/xA)
18 v0lib = np.sqrt(2)*v0c
19
20 #Définition de X0
21 alpha=0 #en variant alpha entre 0 et 1, v0 varie entre v0c et v0lib
22 v0 = (v0lib-v0c)*alpha+v0c
23 X0 = [xA, 0, 0, v0]
24
25 #Paramètres du calcul
26 t0 = 0 # instant initial, en s
27 tf = 1e5 # instant final, en s
28 N = 1000 # nombre de pas
29
30 #Création de la liste de instants où seront calculées les solutions
31 liste_t = np.linspace(t0, tf, N)
32
33 #Résolution de l'équation différentielle
34 X_sol=odeint(F, X0, liste_t)
35
36 #Représentation de la solution
37 plt.clf()
38
39 plt.plot(X_sol[:,0], X_sol[:,1])
40
41 # DEBUT d'interdiction de modifier
42 # Code pour représenter la Terre.
43 plt.scatter(0,0,color='red',marker='+')
44 theta = np.linspace( 0 , 2 * np.pi , 150 )
45 radius = Rt
46 a = radius * np.cos( theta )
47 b = radius * np.sin( theta )
48 plt.plot( a, b, color='red' )
49 plt.axis('equal') #permet d'avoir des axes orthonormés
50 # FIN d'interdiction de modifier
51
52 plt.show()

```

10. On vérifie tout d'abord que pour $v_0 = v_{0,c}$ et $v_0 = v_{0,lib}$ les trajectoires prévues sont correctement calculées par le programme, comme représentées en figure 9.

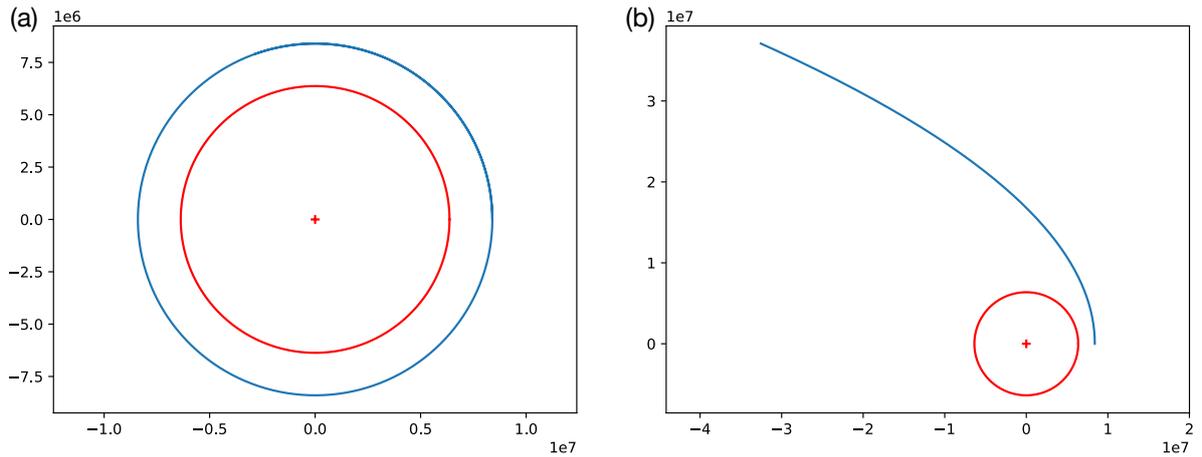


FIGURE 9 – Trajectoire du satellite pour (a) $v_0 = v_{0,c}$ et pour (b) $v_0 = v_{0,lib}$.

Ensuite, afin de faire varier v_0 sur l'intervalle $[v_{0,c}, v_{0,lib}]$, on pose

$$v_0 = v_{0,c} + \alpha(v_{0,lib} - v_{0,c}) \quad \text{avec} \quad \alpha \in [0, 1]$$

Ainsi, si $\alpha = 0$ on obtient $v_0 = v_{0,c}$ et si $\alpha = 1$, on obtient $v_0 = v_{0,lib}$, et en faisant varier α entre 0 et 1, on fait varier v_0 sur l'intervalle désiré.

On remarque que pour toute vitesse $v_0 \in]v_{0,c}, v_{0,lib}[$, on a toujours un état lié, la trajectoire est une ellipse. Plus la vitesse est importante, plus le demi-grand axe de l'ellipse devient grand, et jusqu'à tendre vers ∞ lorsque $\alpha \rightarrow 1$ (voir figure 10).

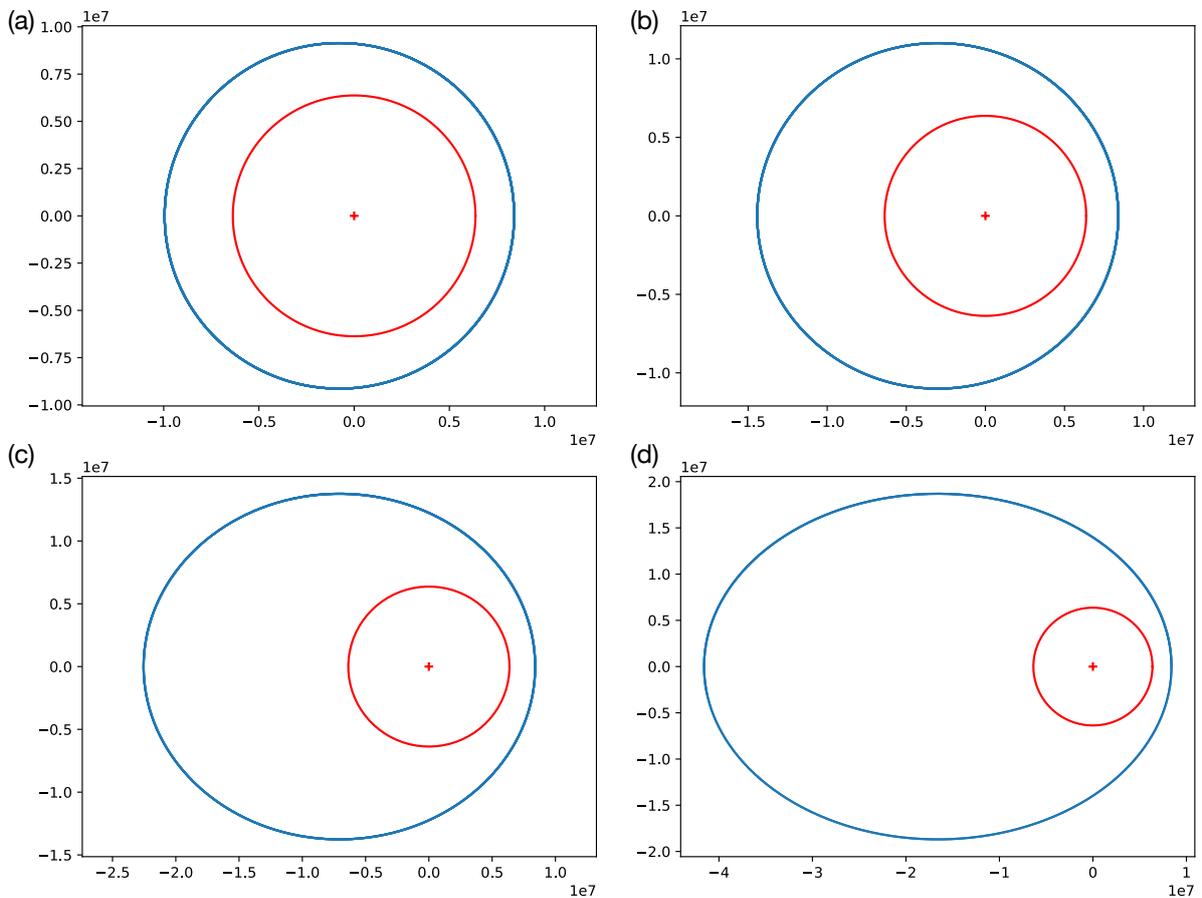


FIGURE 10 – Trajectoire du satellite pour (a) $\alpha = 0, 1$; (b) $\alpha = 0, 3$; (c) $\alpha = 0, 5$ et (d) $\alpha = 0, 7$.

11. Le code précédent reste inchangé, à l'exception de la ligne 22 où l'on écrira :

```
v0=alpha*v0c #alpha=0.98, 0.9, 0.6
```

12. On obtient les trajectoires représentées dans la figure 11. Dans les trois cas, la trajectoire est elliptique, et plus la vitesse initiale est faible, plus le demi-grand axe de l'ellipse est faible. Ainsi, pour $v_0 = 0,9v_{0,c}$ et $v_0 = 0,6v_{0,c}$, la trajectoire du satellite tracée sur le graphe coupe le cercle rouge représentant surface de la Terre, ce qui indique que le satellite s'écrase sur la surface de la Terre dans ces deux cas.

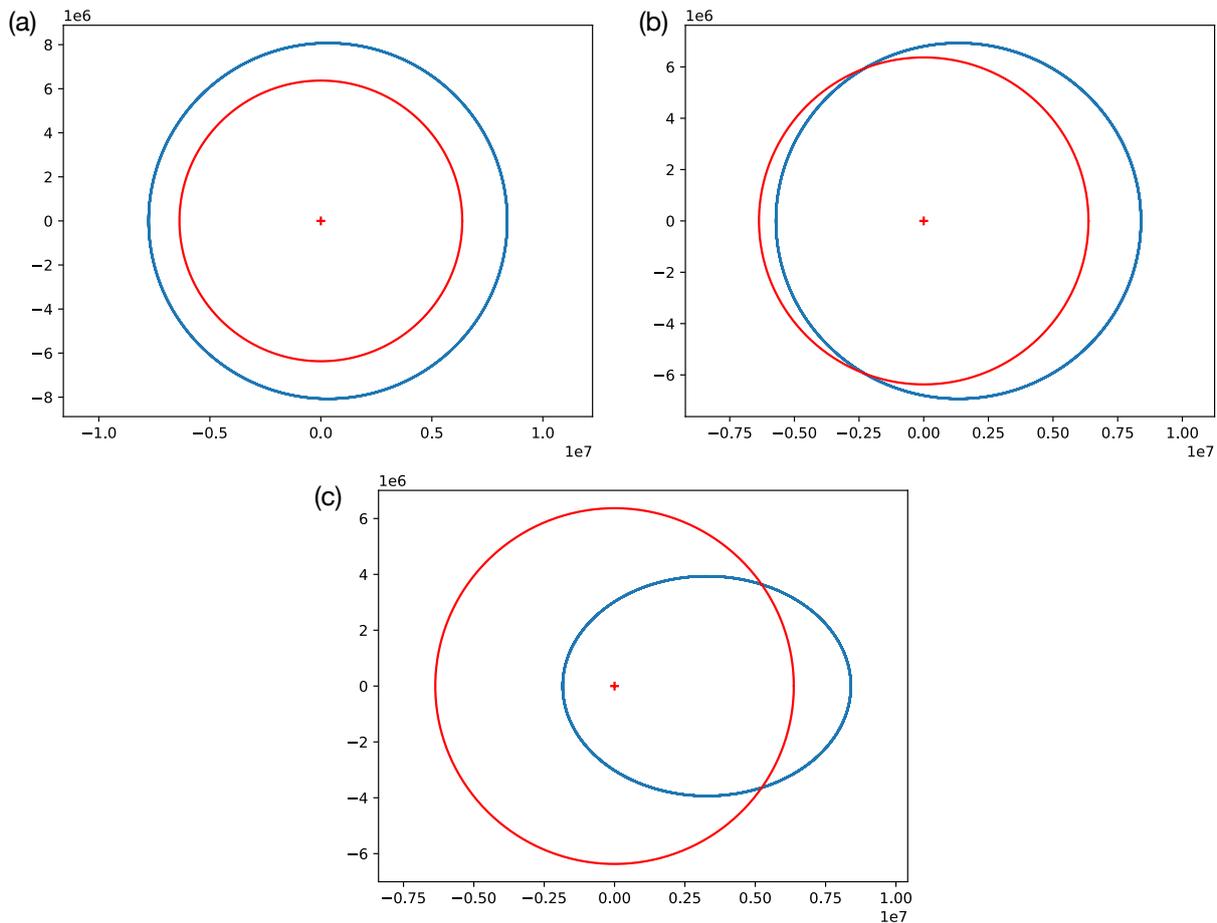


FIGURE 11 – Trajectoire du satellite pour (a) $v_0 = 0,98v_{0,c}$, (b) $v_0 = 0,9v_{0,c}$, (c) $v_0 = 0,6v_{0,c}$.

13. On adopte ici une approche énergétique. Dans le cas limite où le satellite rase la surface de la Terre sur sa trajectoire elliptique, le grand axe de la trajectoire est $R_T + x_A$. Ainsi l'énergie mécanique du satellite sur cette trajectoire s'écrit :

$$E_m = -\frac{GM_T m}{R_T + x_A} = -\frac{GM_T m}{2R_T + h} \quad \text{car } x_A = R_T + h$$

Dans ce cas, on peut utiliser l'expression de l'énergie mécanique à l'instant initial :

$$\frac{1}{2} m v_{0,\min}^2 - \frac{GM_T m}{R_T + h} = -\frac{GM_T m}{2R_T + h} \implies v_{0,\min} = \sqrt{\frac{GM_T}{R_T + h} \times \frac{2R_T}{2R_T + h}} = v_{0,c} \sqrt{\frac{2R_T}{2R_T + h}}$$

14. De nouveau, le code précédent reste inchangé, à l'exception de la ligne 22 où l'on écrira :

```
v0=v0c*np.sqrt(2Rt/(Rt+xA))
```

15. On obtient alors la trajectoire donnée dans la figure 12. La trajectoire est bien rasante lorsque le satellite passe par le péricée. L'application numérique donne $v_{0,\min} \simeq 6,39 \times 10^3$ m/s.

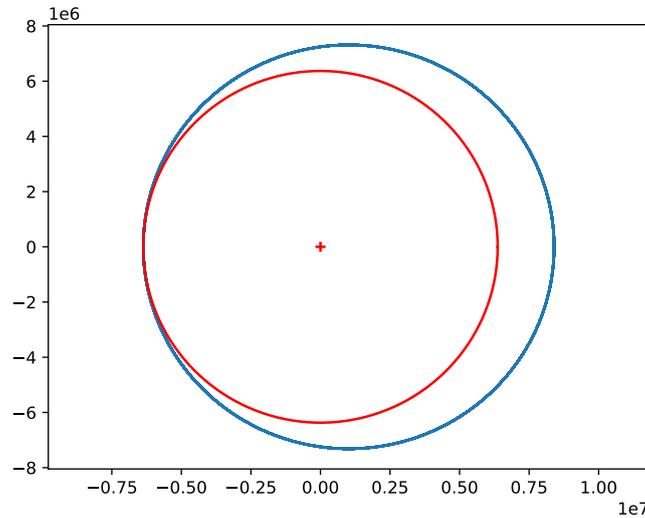


FIGURE 12 – Trajectoire du satellite pour $v_0 = v_{0,\min}$.

16. Dans le cas où l'interaction s'exprimerait en $\frac{1}{r^{3/2}}$, l'interaction ne serait alors plus newtonienne. La projection du vecteur accélération s'écrirait :

$$\ddot{x} = -\frac{GM_T x}{(x^2 + y^2)^{5/4}}$$

$$\ddot{y} = -\frac{GM_T y}{(x^2 + y^2)^{5/4}}$$

Évidemment, il faudrait rigoureusement introduire G^* à la place de G , dont la dimension serait adaptée à la nouvelle dépendance spatiale (pour que l'expression reste homogène).

En modifiant alors les lignes 12 et 13 par :

```
return [X[2], X[3], -G*Mt*X[0]/(X[0]**2+X[1]**2)**(5/4),
        -G*Mt*X[1]/(X[0]**2+X[1]**2)**(5/4)]
```

et également la ligne 22 par :

```
v0=50*v0c
```

on peut alors représenter la trajectoire, donnée dans la figure 13. On obtient alors une trajectoire non fermée : contrairement au cas d'une interaction newtonienne, le satellite ne retombe pas sur la même position après une révolution autour de la Terre.

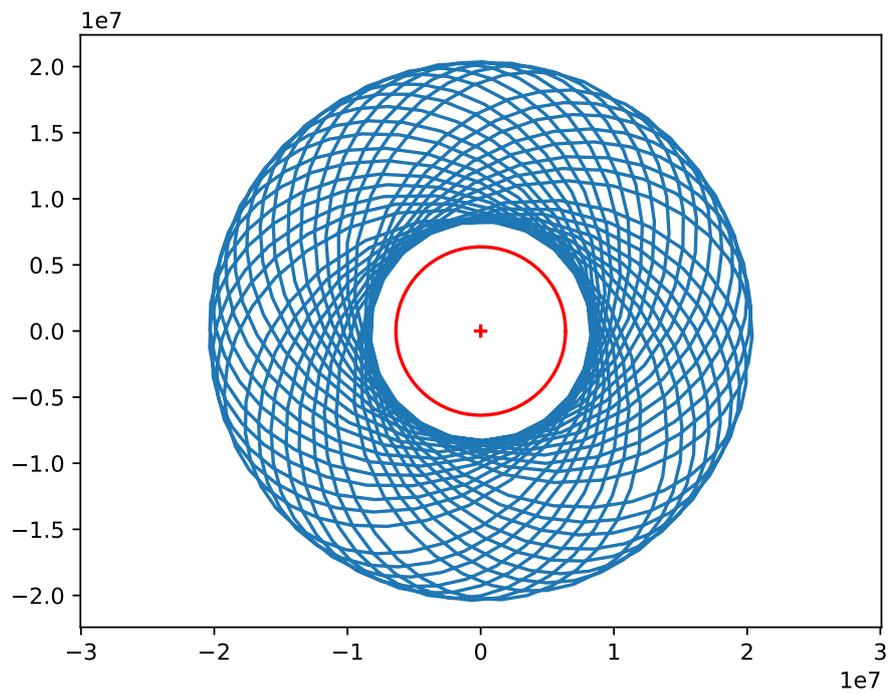


FIGURE 13 – Trajectoire du satellite dans le cas d'une interaction en $1/r^{3/2}$, avec $v_0 = 50v_{0,\text{lib}}$ et $t_f = 10^4$ s.