

Introduction à la résolution des équations différentielles en Matlab

Iker Marginet Ballester

1 Van der Pol

1.1 Équations du mouvement

Nous considérons l'équation différentielle du pendule de Van der Pol

$$\frac{d^2\theta}{dt^2} - \varepsilon(1 - \theta^2)\frac{d\theta}{dt} + \theta = 0 \quad (1)$$

Nous transformons cette équation en un système du premier ordre en posant

$$\omega = \frac{d\theta}{dt} = \dot{\theta} \quad (2)$$

$$\frac{d\omega}{dt} = \varepsilon(1 - \theta^2)\omega - \theta \quad (3)$$

1.2 Définition des Variables

Nous définissons d'abord les paramètres de notre simulation, y compris les conditions initiales et les constantes.

Remarque : lorsque je définis un paramètre, je l'utilise pendant tout le code sans le redéfinir à chaque section.

```
clc; clear; close all;

% Paramètres (pas, temps et iterations) et conditions initiales
epsilon = 2;
T = input('Entrez la durée de simulation (s) : '); % 50 bon temps
h = input('Entrez le pas de temps (s) : '); % 0.01 pas ideal
N = T / h;

% Conditions initiales : [theta_0, omega_0]
CI = [0.5, 0.0; 1.0, -2.5];

% Fonction "anonyme" f(theta, omega)
f = @(theta, omega) epsilon * (1 - theta^2) * omega - theta;
```

1.3 Méthode d'Euler et Heun

Nous implémentons les méthodes d'Euler et de Heun pour résoudre le système d'équations différentielles.

```
% Boucle sur les conditions initiales
for i = 1:size(CI, 1)
    % Initialisation des variables pour Euler et Heun
    theta_e = zeros(1, N); omega_e = zeros(1, N);
    theta_h = zeros(1, N); omega_h = zeros(1, N);

    % Conditions initiales
```

```

theta_e(1) = CI(i, 1); omega_e(1) = CI(i, 2);
theta_h(1) = CI(i, 1); omega_h(1) = CI(i, 2);

% Boucle Euler
for n = 1:N-1
    theta_e(n+1) = theta_e(n) + h * omega_e(n);
    omega_e(n+1) = omega_e(n) + h * f(theta_e(n), omega_e(n));
end

% Boucle Heun
for n = 1:N-1
    i1_theta = h * omega_h(n);
    i1_omega = h * f(theta_h(n), omega_h(n));

    i2_theta = h * (omega_h(n) + i1_omega);
    i2_omega = h * f(theta_h(n) + i1_theta, omega_h(n) + i1_omega);

    theta_h(n+1) = theta_h(n) + 0.5 * (i1_theta + i2_theta);
    omega_h(n+1) = omega_h(n) + 0.5 * (i1_omega + i2_omega);
end

% Tracer les résultats pour les deux méthodes
figure;
plot(theta_e, omega_e, 'DisplayName', 'Euler');
hold on;
plot(theta_h, omega_h, 'DisplayName', 'Heun');
title(['Portrait de phase pour les CI']);
xlabel('\theta'); ylabel('\omega');
legend;
hold off;
end

```

1.4 Portraits de phase pour le pendule de Van der Pol

Les conditions initiales (θ_0, ω_0) jouent un rôle crucial dans l'évolution du système. Elles déterminent la trajectoire initiale et influencent la vitesse de convergence vers le cycle limite ou l'instabilité.

Condition initiale 1 : $(\theta_0 = 0.5), (\dot{\theta}_0 = 0.0)$

Condition initiale 2 : $(\theta_0 = 1.0), (\dot{\theta}_0 = -2.5)$

Observations

Peu importe les valeurs initiales (θ_0, ω_0) , toutes les trajectoires convergent rapidement vers le cycle limite.

- Si θ_0 est proche de zéro, la trajectoire commence avec de faibles oscillations instables, mais elles grandissent et se stabilisent autour du cycle limite.
- Si θ_0 est grand, la stabilisation est plus rapide, car le système est déjà proche de l'amortissement stabilisateur.

Interprétation physique

- Le cycle limite agit comme un attracteur global, forçant toutes les trajectoires à adopter une oscillation stable d'amplitude et de période fixes.
- Ce comportement est utilisé pour modéliser les circuits oscillants en électronique et les rythmes biologiques, où un système dynamique atteint naturellement une oscillation stable indépendamment des perturbations initiales.

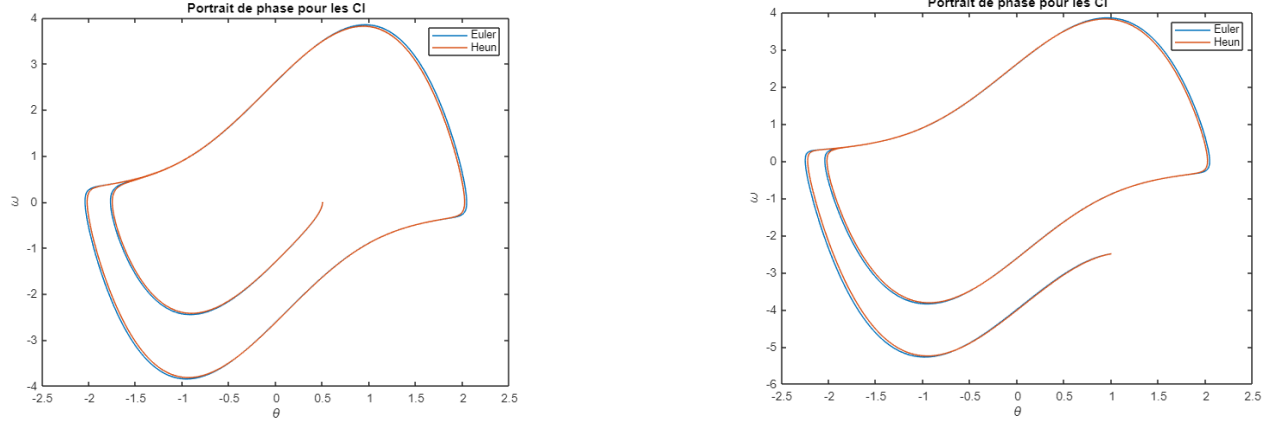


Figure 1: Portraits de phase pour les premières et deuxième conditions initiales, respectivement.

1.4.1 Cas $\varepsilon \rightarrow 0$: Oscillations parfaites, pas d'amortissement

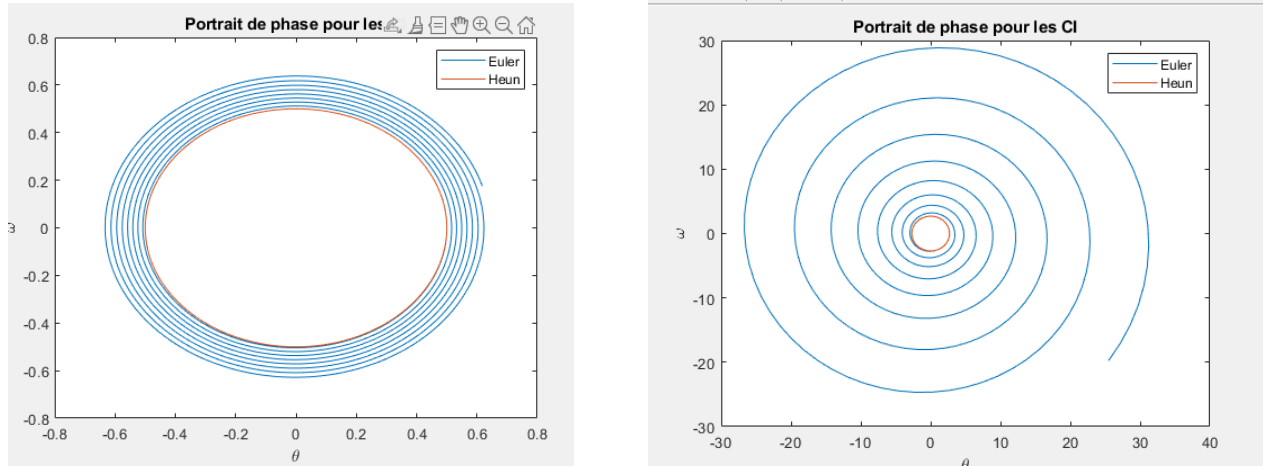


Figure 2: Comportement de l'oscillateur lorsque $\varepsilon \rightarrow 0$

Lorsque $\varepsilon = 0$, l'équation devient $\frac{d^2\theta}{dt^2} + \theta = 0$, qui est une équation différentielle linéaire homogène à coefficients constants.

Observations

Les conditions initiales définissent entièrement la trajectoire, car aucune dissipation ni stabilisation n'est présente.

- Les trajectoires sont des ellipses parfaites dans l'espace des phases (θ, ω) .
- Si θ_0 et ω_0 sont plus grands, l'ellipse est plus large, mais sa forme reste inchangée.

Interprétation physique

- Il s'agit d'un système conservatif, où l'énergie initiale est parfaitement préservée.
- Ce régime est observé dans les pendules idéaux, les circuits LC parfaits et les modèles quantiques de particules sans interaction.

1.4.2 Cas $\varepsilon < 0$: Instabilité exponentielle, divergence des trajectoires

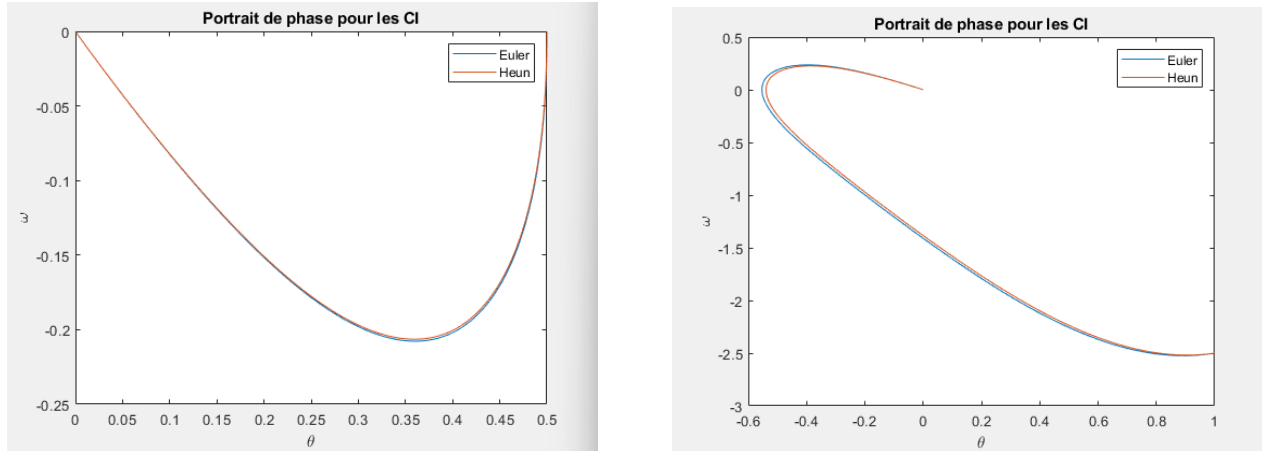


Figure 3: Comportement de l'oscillateur lorsque $\varepsilon < 0$

Observations

L'effet des conditions initiales est majeur, car le système est instable.

- Pour des petites valeurs de θ_0 , l'oscillation démarre lentement mais diverge ensuite très vite.
- Pour des grandes valeurs de θ_0 , la divergence est encore plus rapide, atteignant des valeurs extrêmes en peu de temps.

Interprétation physique

- Une instabilité auto-entretenu est présente, ce qui est typique des réactions explosives et des systèmes amplifiés de façon incontrôlée.
- En électronique, cela correspond à une rétroaction positive non contrôlée, où une petite fluctuation initiale est amplifiée sans limite.

1.4.3 Cas $0 < \varepsilon < 1$: Transition douce vers le cycle limite

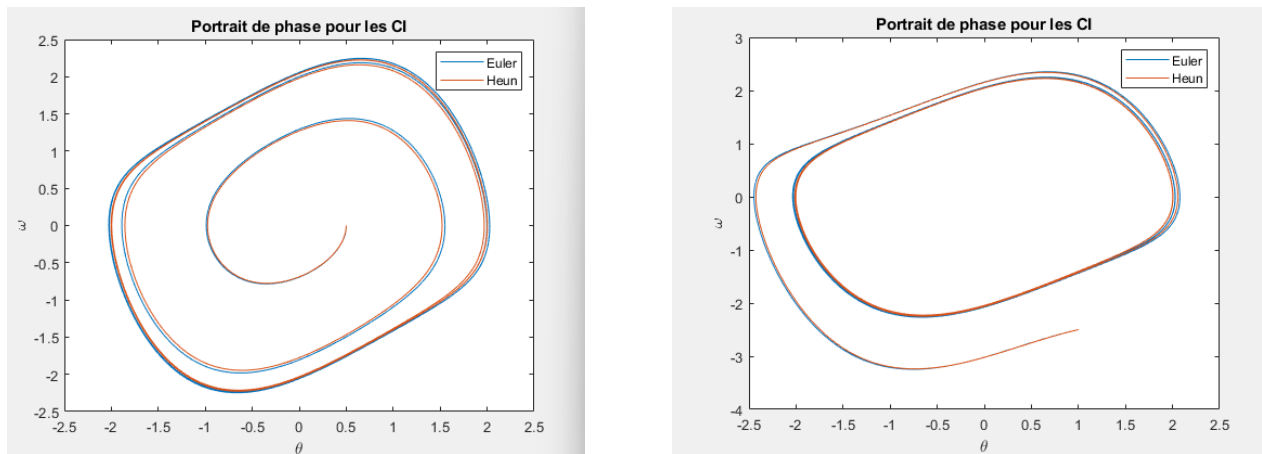


Figure 4: Comportement de l'oscillateur lorsque $0 < \varepsilon < 1$

Observations

Le cycle limite est plus faible, et la convergence dépend davantage des conditions initiales.

- Pour θ_0 petit, les trajectoires restent proches d'un oscillateur harmonique avant de converger lentement.
- Pour θ_0 grand, la convergence est plus rapide, mais les oscillations restent faiblement non linéaires.

Interprétation physique

- Ce régime est une zone intermédiaire où la non-linéarité est assez faible pour que les conditions initiales influencent l'évolution transitoire du système.
- Ce comportement est typique des circuits amplifiés faiblement ou de certains systèmes mécaniques où un amortissement variable est présent.

1.4.4 Conclusion générale sur l'influence des conditions initiales

Influence des conditions initiales

- $\varepsilon > 1$: Faible, convergence rapide vers un cycle limite attractif stable.
- $0 < \varepsilon < 1$: Moyenne, influence transitoire, convergence progressive.
- $\varepsilon = 0$: Forte, les conditions initiales définissent entièrement la trajectoire.
- $\varepsilon < 0$: Extrême, divergence exponentielle, amplification des perturbations initiales.

Résumé visuel

ε	Influence des conditions initiales
$\varepsilon > 1$	Peu importe θ_0 , l'oscillateur se stabilise rapidement.
$0 < \varepsilon < 1$	La trajectoire initiale influe légèrement avant d'atteindre le cycle limite.
$\varepsilon = 0$	L'oscillateur suit exactement la trajectoire définie par (θ_0, ω_0) .
$\varepsilon < 0$	Une petite différence dans les conditions initiales entraîne une explosion des oscillations.

1.5 Influence du Pas de Temps h

- Un pas trop grand entraîne des erreurs significatives (surtout avec Euler car plusieurs passages).
- Un pas trop petit augmente le temps de calcul.
- Heun permet des pas plus grands tout en conservant la précision.
- Pour un temps T trop grand, il faudrait encore autre méthode de résolution

1.6 Analyse d'erreurs

*% On ajoute ce but de code (dans le code précédent) pour comparer les méthodes numériques
↪ d'Euler et de Heun pour la résolution d'une équation différentielle, en calculant
↪ l'erreur relative entre les solutions théoriques et expérimentales pour différentes
↪ valeurs du pas de temps h.*

```
h_err = 10.^(-2:-1:-6);  
erreurs = [];
```

% Calcul de l'erreur relative

```

theta_th = max(theta_h);
theta_exp = max(theta_e);
err_rel = abs((theta_exp - theta_th) / theta_th);
erreurs = [erreurs, err_rel];
end

% Tracé des solutions en fonction du temps
figure;
plot(t, theta_e, 'r', 'DisplayName', 'Euler');
hold on;
plot(t, theta_h, 'b', 'DisplayName', 'Heun');
xlabel('Temps (s)');
ylabel('\theta');
title(['Évolution de \theta en fonction du temps pour h = ', num2str(h)]);
legend;
grid on;
hold off;
end

% Affichage du tableau des erreurs
fprintf('h Erreur Relative\n');
for i = 1:length(h_err)
fprintf('%e %e\n', h_err(i), erreurs(i));
end

% Tracé unique de log(erreur relative) vs log(h)
figure;
loglog(h_err, erreurs, '-o');
xlabel('log(h)');
ylabel('log(erreur relative)');
title('Erreur relative en fonction du pas de temps h');

```

1.7 Analyse graphique

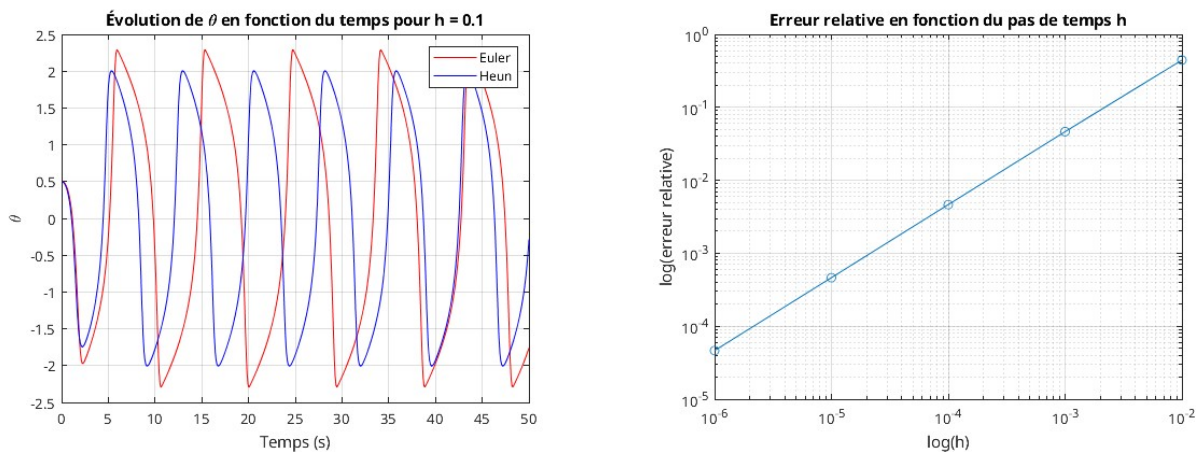


Figure 5: Erreur relatif

L'erreur relative est définie par :

$$e_{rel} = \left| \frac{\theta_{Euler}^{max} - \theta_{Heun}^{max}}{\theta_{Heun}^{max}} \right| \quad (4)$$

Le premier graphique représente deux ondes sinusoïdales déphasées. Elles correspondent aux solutions des méthodes d'Euler et de Heun. Ces solutions sont approchées et varient selon le pas de temps h .

Choix des valeurs de h

- h varie de 10^{-2} à 10^{-6} .
- Cela permet d'observer l'effet des petits pas de temps.
- Un grand h entraîne une erreur plus forte.
- Un petit h réduit l'erreur mais augmente le coût de calcul.

Résultat du graphique log-log

- Le graphique log-log montre une droite.
- Cela indique un comportement de convergence typique.
- L'erreur suit une loi de puissance en fonction de h .

Méthode avec plus d'erreur

- La méthode d'Euler produit plus d'erreurs.
- Euler est une méthode d'ordre 1.
- Heun, d'ordre 2, est plus précise et réduit l'erreur.

2 Dynamique d'un satellite en orbite terrestre

Nous considérons le mouvement d'un satellite autour de la Terre dans un plan Oxy , sous l'influence gravitationnelle de la Terre seule. L'équation du mouvement est donnée par :

$$\frac{d^2x}{dt^2} = -\frac{GM_t}{r^3}x \quad (5)$$

$$\frac{d^2y}{dt^2} = -\frac{GM_t}{r^3}y \quad (6)$$

$$v_x = \frac{dx}{dt}, \quad v_y = \frac{dy}{dt} \quad (7)$$

$$\frac{dv_x}{dt} = -\frac{GM_t}{r^3}x \quad (8)$$

$$\frac{dv_y}{dt} = -\frac{GM_t}{r^3}y \quad (9)$$

- avec :
- $G = 6.67 \cdot 10^{-11} \text{ Nm}^2\text{kg}^{-2}$ (constante gravitationnelle)
 - $M_t = 5.9736 \cdot 10^{24} \text{ kg}$ (masse de la Terre)
 - $r = \sqrt{x^2 + y^2}$

2.1 Simulation Numérique de la Trajectoire du Satellite

Nous simulons la trajectoire d'un satellite en utilisant la méthode de Heun vectorisée et la fonction `ode23` de MATLAB. La valeur correcte du pas de temps h doit être suffisamment petite pour assurer une bonne précision sans ralentir excessivement la simulation. Une valeur correcte est $h = 1 \text{ s}$, et $T_{max} = 250000 \text{ s}$ ce qui garantit une erreur numérique faible tout en étant efficace.

2.2 Fonction : m_heun_vecto.m

```
function Y = m_heun_vecto(CI, N, h, f)
    % Initialisation du tableau des solutions (4 variables : x, y, vx, vy)
    Y = zeros(4, N);
    Y(:, 1) = CI;

    for n = 1:N-1
        k1 = f(0, Y(:, n)); % Première estimation
        k2 = f(0, Y(:, n) + h * k1); % Estimation corrigée

        Y(:, n+1) = Y(:, n) + (h/2) * (k1 + k2);
    end
end
```

2.3 Fichier principal : simulation_satellite.m

```
% Paramètres et Conditions initiales
G = 6.67e-11; Mt = 5.9736e24; RO = 4.223e7; v0 = 3071; ms = 1000;
R1 = RO; v1 = 1.2 * v0;
CI1 = [RO; 0; 0; v0]; CI2 = [R1; 0; 0; v1];

% Choix de Tmax et du pas de temps h par l'utilisateur
Tmax = input('Entrez la durée de simulation (s) : ');
h = input('Entrez le pas de temps (s) : ');
N = Tmax / h;

% Définition de la fonction différentielle
f = @(t, Y) [Y(3); Y(4);
    -G * Mt * Y(1) / (Y(1)^2 + Y(2)^2)^(3/2);
    -G * Mt * Y(2) / (Y(1)^2 + Y(2)^2)^(3/2)];

% Appel de la méthode Heun vectorisée
Y1_h = m_heun_vecto(CI1, N, h, f);
Y2_h = m_heun_vecto(CI2, N, h, f);

% Résolution avec ode23
[t1, sol1] = ode23(f, [0 Tmax], CI1);
[t2, sol2] = ode23(f, [0 Tmax], CI2);

figure;
% Tracé des trajectoires pour CI1
subplot(1,2,1);
plot(Y1_h(1, :), Y1_h(2, :), 'r', 'DisplayName', 'Méthode Heun - CI1');
hold on;
plot(sol1(:,1), sol1(:,2), 'g--', 'DisplayName', 'ode23 - CI1');
hold off;
xlabel('x (m)'); ylabel('y (m)');
title('Trajectoire du satellite - CI1');
legend show;
axis equal;

% Tracé des trajectoires pour CI2
subplot(1,2,2);
plot(Y2_h(1, :), Y2_h(2, :), 'b', 'DisplayName', 'Méthode Heun - CI2');
```



```

hold on;
plot(sol2(:,1), sol2(:,2), 'k--', 'DisplayName', 'ode23 - CI2');
hold off;
xlabel('x (m)'); ylabel('y (m)');
title('Trajectoire du satellite - CI2');
legend show;
axis equal;

```

2.4 Résultats

2.4.1 Analyse pour $T = 10,000$ s, $h = 1$ s avec CI1 et CI2

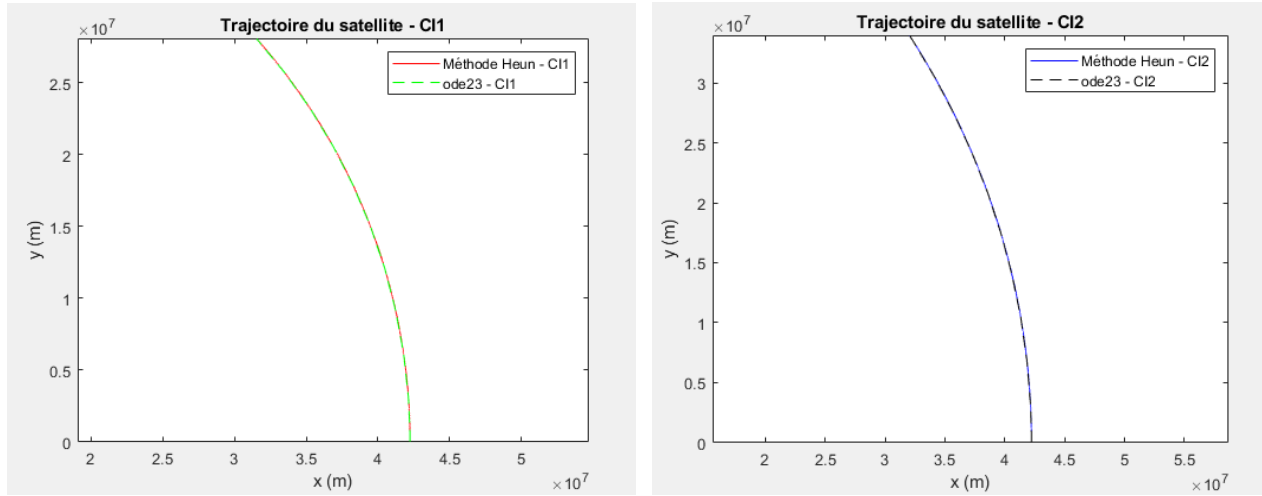


Figure 6: Comportement du satellite sous les conditions initiales pendant une période de 10 000 s

Conditions Initiales

- CI1 : $R_0, v_0 \rightarrow$ orbite circulaire ou légèrement elliptique.
- CI2 : $1.2 v_0 \rightarrow$ orbite plus large et plus excentrique.

Observations des graphes

- Pour CI1, la trajectoire obtenue avec Heun et ode23 est très proche d'un cercle.
- Pour CI2, on observe une orbite plus grande et plus elliptique.

Méthode de Heun

- L'algorithme conserve bien la forme de l'orbite sur cette courte durée.
- Il suit une précision stable, mais légèrement plus rigide que ode23.
- L'erreur d'approximation reste faible car on est sur une période courte.

Méthode ode23

- Ode23 ajuste dynamiquement le pas de temps en fonction de l'évolution de l'orbite.
- Sur 10,000 s, les différences avec Heun sont faibles, mais on voit une légère dissymétrie dans l'évolution de l'orbite.

Interprétation Physique

- CI1 correspond à une orbite stable, avec une faible excentricité.
- CI2 montre que l'augmentation de la vitesse initiale pousse l'orbite à devenir plus elliptique.
- Les différences entre Heun et ode23 sont très légères sur cette durée, car l'accumulation d'erreur n'est pas encore significative.

2.4.2 Analyse pour $T = 250,000$ s, $h = 1$ s (figure complète de l'orbite)

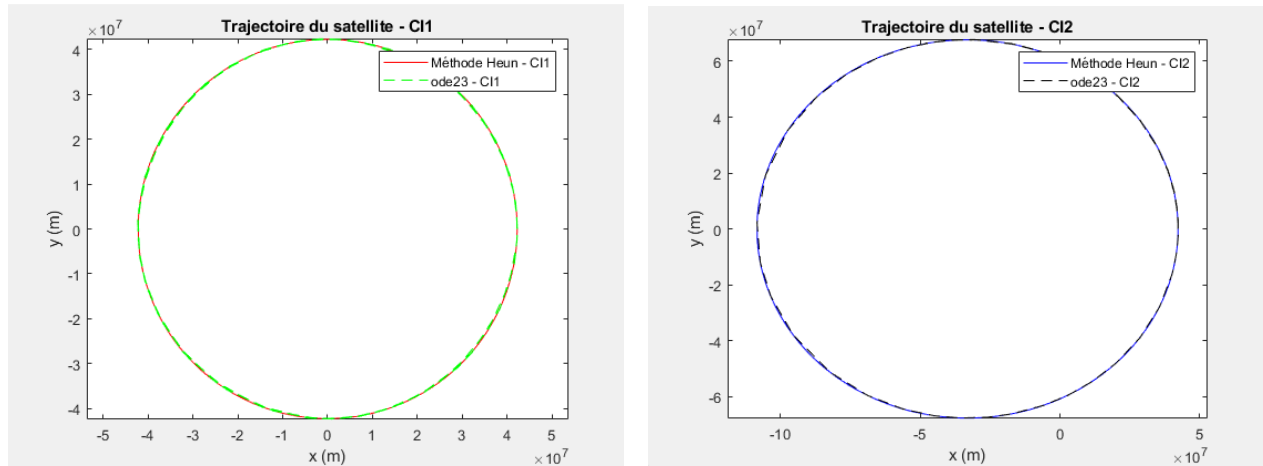


Figure 7: Comportement du satellite sous les conditions initiales pendant une période de 250 000 s

Observations des graphes

- Sur une durée plus longue, les erreurs numériques deviennent plus visibles.
- Heun maintient une orbite stable et répétitive.
- Ode23 commence à dériver légèrement à chaque passage.

Méthode de Heun

- Fonctionne bien pour des intégrations longues, mais introduit de petites erreurs cumulées.
- L'orbite reste globalement inchangée, sauf pour quelques petites déviations numériques.

Méthode ode23

- On observe une légère dérive orbitale après plusieurs passages.
- Comme ode23 ajuste dynamiquement le pas de temps, chaque orbite est légèrement différente, ce qui crée une variation progressive.
- Cette variation peut être due à l'accumulation d'erreurs d'approximation sur les longues durées.

Interprétation Physique

- Plus la durée d'intégration est longue, plus les méthodes numériques peuvent générer des erreurs accumulées.
- Heun est plus rigide, maintenant une orbite presque identique à chaque révolution.
- Ode23 est plus précis localement, mais introduit de petites erreurs globales, ce qui cause une variation orbitale progressive.

2.4.3 Analyse avec CI3 : $0.5 R_0, 0.5 v_0 \rightarrow$ Grande ellipse

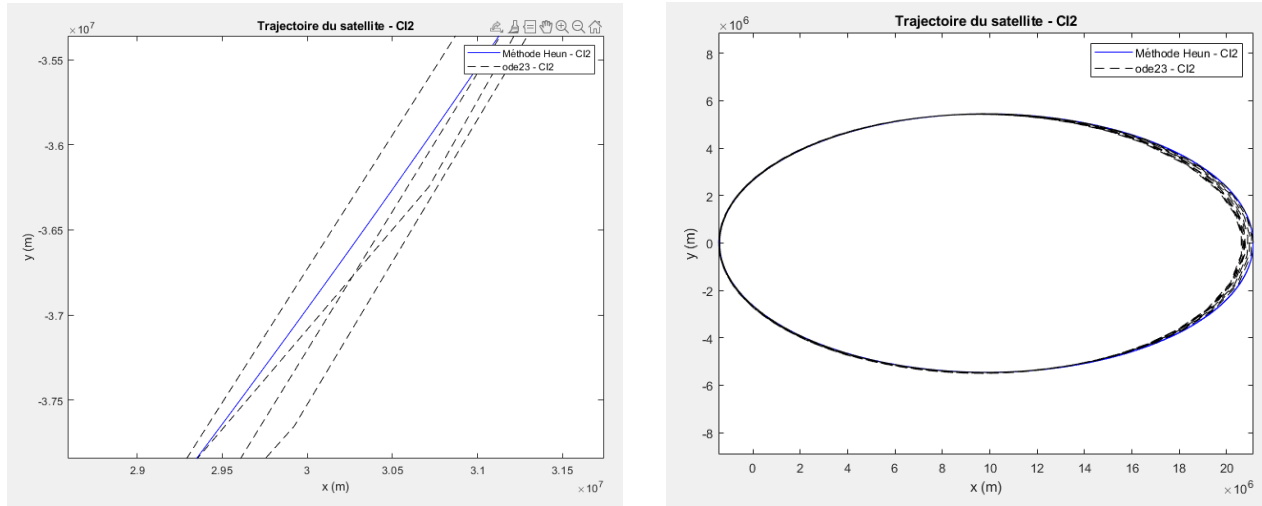


Figure 8: Conditions initiales elliptiques et passages multiples pour la fonction ode23

Conditions Initiales

- $CI3 = (0.5 R_0, 0.5 v_0)$
- La réduction de R_0 signifie que le satellite commence plus près de la Terre.
- La réduction de v_0 signifie que l'orbite sera plus elliptique car la vitesse ne compense pas totalement la gravité.

Observations des graphes

- L'orbite devient une ellipse très allongée.
- L'altitude minimale du satellite est beaucoup plus basse, ce qui crée une variation de vitesse très marquée selon la loi de Kepler.

Méthode de Heun

- Conserve la forme de l'ellipse avec des passages identiques à chaque révolution.
- Les erreurs restent faibles, ce qui signifie qu'il suit bien la trajectoire prévue.

Méthode ode23

- Génère des orbites légèrement différentes à chaque révolution.
- L'ellipse semble légèrement se déplacer à cause des ajustements dynamiques du pas de temps.

Interprétation Physique

- CI3 montre clairement l'effet des conditions initiales sur l'orbite : une vitesse plus faible entraîne une orbite plus aplatie.
- Cette configuration est similaire aux orbites des comètes ou des satellites en orbite très elliptique.
- Encore une fois, Heun conserve bien une orbite répétitive, alors que ode23 introduit de légères variations orbitales.

2.4.4 Pourquoi Heun donne des passages identiques et ode23 pas ?

Observations

- Heun donne exactement la même orbite à chaque passage.
- Ode23 montre une légère dérive orbitale après chaque révolution.

Explication Numérique

- Heun applique une correction constante à chaque étape, ce qui maintient une rigidité forte dans la trajectoire.
- Ode23 ajuste le pas de temps dynamiquement, ce qui fait que les points de calcul ne sont pas toujours exactement au même endroit à chaque passage.
- Ces petits ajustements s'accumulent sur plusieurs orbites, ce qui explique la légère variation orbitale à long terme.

Explication Physique

- En réalité, les orbites des satellites ne sont jamais parfaitement répétitives car il y a des perturbations dues aux effets de la Terre (aplatissement, atmosphère, Lune, Soleil).
- Ode23, en modélisant l'évolution plus dynamiquement, est plus proche d'une orbite réelle, où de petites corrections naturelles existent à chaque passage.

2.4.5 Conclusion Générale

Expérience	Heun	ode23
T = 10k, CI1 et CI2	Orbites précises et stables	Orbites précises et ajustées dynamiquement
T = 250k, CI1 et CI2	Orbites presque identiques	Orbites légèrement modifiées à chaque passage
CI3 (grande ellipse)	Orbite stable et elliptique	Orbite légèrement fluctuante
Passages multiples	Orbite exactement répétée	Légère dérive à chaque révolution

Résumé visuel

- Heun → Précis mais rigide, orbites toujours identiques.
- Ode23 → Plus flexible, orbites légèrement différentes à chaque révolution.
- CI3 montre l'impact des conditions initiales avec une orbite beaucoup plus allongée.

3 Étude de l'Énergie

3.1 Equations du mouvement

Nous calculons l'énergie mécanique totale à chaque instant avec la relation suivante :

$$E_t = \frac{1}{2}m_s(v_x^2 + v_y^2) - \frac{m_s M_t G}{\sqrt{x^2 + y^2}} \quad (10)$$

3.2 Fonction : calculer_energie.m

Nous définissons d'abord une fonction pour calculer l'énergie mécanique :

```
% Fonction pour calculer l'énergie
function E = calculer_energie(vx, vy, x, y, ms, G, Mt)
    E = 0.5 * ms * (vx.^2 + vy.^2) - ms * G * Mt ./ sqrt(x.^2 + y.^2);
end
```

3.3 Fichier principal : calcul_energie.m

```
% Calcul des énergies pour la méthode de Heun
E1_h = calculer_energie(Y1_h(3, :), Y1_h(4, :), Y1_h(1, :), Y1_h(2, :), ms, G, Mt);
E2_h = calculer_energie(Y2_h(3, :), Y2_h(4, :), Y2_h(1, :), Y2_h(2, :), ms, G, Mt);

% Calcul des énergies pour la méthode ode23
E1_ode = calculer_energie(sol1(:, 3), sol1(:, 4), sol1(:, 1), sol1(:, 2), ms, G, Mt);
E2_ode = calculer_energie(sol2(:, 3), sol2(:, 4), sol2(:, 1), sol2(:, 2), ms, G, Mt);

T = linspace(0, 10000, length(E1_h)); % Vecteur temps pour Heun
t1 = linspace(0, 10000, length(E1_ode)); % Vecteur temps pour ode23 (CI1)
t2 = linspace(0, 10000, length(E2_ode)); % Vecteur temps pour ode23 (CI2)

% Tracé des résultats
close all;

% Figure pour la méthode Heun
figure(1);
subplot(2, 1, 1);
plot(T, E1_h, 'r-', 'DisplayName', 'Heun - CI1');
hold on;
plot(T, E2_h, 'b--', 'DisplayName', 'Heun - CI2');
hold off;
xlabel('Temps (s)');
ylabel('Énergie (J)');
title('Comparaison Heun - CI1 et CI2');
legend show;

subplot(2, 1, 2);
plot(t1, E1_ode, 'g-', 'DisplayName', 'ode23 - CI1');
hold on;
plot(t2, E2_ode, 'm--', 'DisplayName', 'ode23 - CI2');
hold off;
xlabel('Temps (s)');
ylabel('Énergie (J)');
title('Comparaison ode23 - CI1 et CI2');
legend show;

figure(2);
subplot(2, 1, 1);
[ax, h1, h2] = plotyy(T, E1_h, t1, E1_ode);
set(h1, 'Color', 'r', 'DisplayName', 'Heun - CI1');
set(h2, 'Color', 'g', 'DisplayName', 'ode23 - CI1');
xlabel('Temps (s)');
```

```

ylabel(ax(1), 'Énergie (J)'); ylabel(ax(2), 'Énergie (J)');
title('Comparaison CI1');
legend([h1, h2]);

subplot(2, 1, 2);
[ax, h1, h2] = plotyy(T, E2_h, t2, E2_ode);
set(h1, 'Color', 'b', 'DisplayName', 'Heun - CI2');
set(h2, 'Color', 'm', 'DisplayName', 'ode23 - CI2');
xlabel('Temps (s)');
ylabel(ax(1), 'Énergie (J)'); ylabel(ax(2), 'Énergie (J)');
title('Comparaison CI2');
legend([h1, h2]);

```

3.4 Résultats

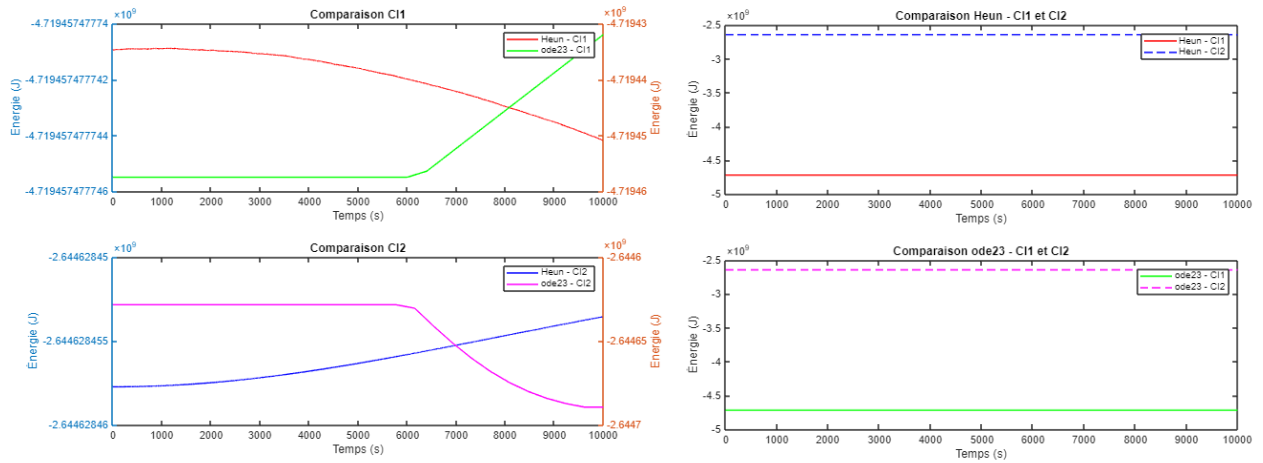


Figure 9: Démonstration de la conservation de l'énergie et défauts numériques

3.4.1 Analyse de la Conservation de l'Énergie avec CI1 et CI2

1. Conservation théorique de l'énergie

Dans un système conservatif comme une orbite idéale (sans frottement ni perturbations externes), cette énergie doit rester constante au cours du temps.

Dans les graphes obtenus :

- Globalement, l'énergie semble bien conservée.
- Si on zoome avec `plotyy`, on observe de petites oscillations et variations dans l'énergie totale, de l'ordre de 10^{-2} à 10^{-3} .

Ces imperfections sont visibles dans les deux méthodes (Heun et `ode23`), ce qui montre qu'elles ne sont pas spécifiques à une seule méthode d'intégration, mais plutôt liées aux erreurs numériques intrinsèques.

3.4.2 Explication des imperfections dans la conservation de l'énergie

Ces variations ne signifient pas que l'énergie physique du système est réellement modifiée, mais plutôt qu'il existe des erreurs d'approximation dans les méthodes numériques utilisées.

A. Erreurs d'arrondi (erreurs de machine)

- Les calculs numériques sont effectués avec une précision finie (généralement en double précision avec 10^{-15} comme précision machine).
- Les opérations répétées sur de longues simulations accumulent ces erreurs, ce qui crée de légères variations de l'énergie totale.

B. Erreurs d'intégration

- Méthode de Heun : c'est une méthode d'ordre 2, donc elle introduit des erreurs, c'est-à-dire que la vraie trajectoire est légèrement déformée à chaque pas de temps.
- Ode23 ajuste dynamiquement le pas, mais ce faisant, il introduit de petites fluctuations numériques qui affectent l'énergie.
- Ces erreurs sont minimales à chaque étape, mais elles s'accumulent progressivement au fil des itérations.

C. Approximation des dérivées

- Dans les équations différentielles, on approche les dérivées par des différences finies.
- Cette approximation introduit toujours une petite erreur, qui se traduit par de légères fluctuations dans l'énergie.

4 Introduction de la lune dans les équations

4.1 Équations du mouvement

Nous considérons un satellite soumis à l'influence gravitationnelle de la Terre et de la Lune. L'équation différentielle régissant son mouvement est donnée par :

$$\frac{d^2 \mathbf{r}}{dt^2} = -G \left(\frac{M_t \mathbf{r}}{\|\mathbf{r}\|^3} + \frac{M_l (\mathbf{r} - \mathbf{r}_l)}{\|\mathbf{r} - \mathbf{r}_l\|^3} \right) \quad (11)$$

- avec :
- $M_l = \frac{M_t}{81}$: masse de la Lune,
 - \mathbf{r} : position du satellite,
 - \mathbf{r}_l : position de la Lune.

Nous supposons que la Lune est située à une distance constante $R = 4 \cdot 10^8$ m et qu'elle suit une orbite circulaire autour de la Terre.

4.2 Code pour l'introduction de la lune

```
% Paramètres physiques
```

```
Ml = Mt / 81; R2 = 4e8;  
Tmax = input('Entrez la durée de simulation (s) : ');
```

```
% Conditions initiales de la Lune (orbite circulaire autour de la Terre)
```

```
v_lune = sqrt(G * Mt / R2);  
CIL = [R2; 0; 0; v_lune];
```

```
% Définition de la fonction différentielle avec la Terre, la Lune et le satellite
```

```

f_sat = @(t, Y) [Y(3); Y(4);
-G * Mt * Y(1) / (Y(1)^2 + Y(2)^2)^(3/2) - G * Ml * (Y(1) - CIL(1)) / ((Y(1) - CIL(1))^2
↳ + (Y(2) - CIL(2))^2)^(3/2);
-G * Mt * Y(2) / (Y(1)^2 + Y(2)^2)^(3/2) - G * Ml * (Y(2) - CIL(2)) / ((Y(1) - CIL(1))^2
↳ + (Y(2) - CIL(2))^2)^(3/2)];

f_lune = @(t, Y) [Y(3); Y(4);
-G * Mt * Y(1) / (Y(1)^2 + Y(2)^2)^(3/2);
-G * Mt * Y(2) / (Y(1)^2 + Y(2)^2)^(3/2)];

% Résolution par la méthode de Heun pour le satellite
s_h_s = m_heun_vecto(CI1, N, h, f_sat);

% Résolution par la méthode de Heun pour la Lune
s_h_s = m_heun_vecto(CIL, N, h, f_lune);

% Résolution avec ode23
[t, s_ode_sat] = ode23(f_sat, [0 Tmax], CI1);
[t, s_ode_lune] = ode23(f_lune, [0 Tmax], CIL);

% Tracé des trajectoires
figure; hold on; grid on; axis equal;
plot(s_h_s(1, :), s_h_s(2, :), 'y', 'DisplayName', 'Satellite - Heun');
plot(s_ode_sat(:,1), s_ode_sat(:,2), 'g--', 'DisplayName', 'Satellite - ode23');
plot(s_h_s(1, :), s_h_s(2, :), 'b', 'DisplayName', 'Lune - Heun');
plot(s_ode_lune(:,1), s_ode_lune(:,2), 'p--', 'DisplayName', 'Lune - ode23');
plot(0, 0, 'ro', 'MarkerSize', 10, 'MarkerFaceColor', 'r', 'DisplayName', 'Terre');

xlabel('x (m)'); ylabel('y (m)');
title('Trajectoires du satellite et de la Lune autour de la Terre');
legend show;
xlim([-1.5*R2, 1.5*R2]); ylim([-1.5*R2, 1.5*R2]);

```

4.3 Analyse des résultats

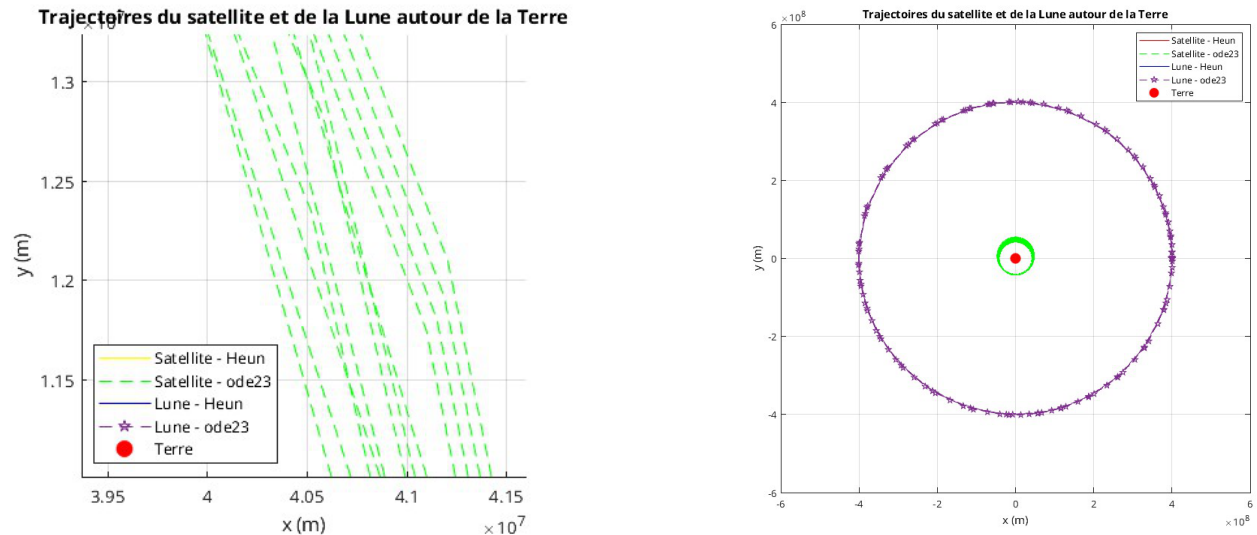


Figure 10: Tracé terre-satellite-lune et défauts (influence gravitationnelle)

Les trajectoires obtenues montrent que :

- Le satellite subit des perturbations gravitationnelles dues à la présence de la Lune, pouvant altérer son orbite et nécessiter des corrections.
- Une résonance gravitationnelle entre le satellite et la Lune peut induire des variations significatives de trajectoire.
- L'utilisation de `ode23` assure une bonne précision pour les simulations à court terme, mais des méthodes plus avancées comme `ode45` ou des intégrateurs symplectiques pourraient être préférables pour les simulations à long terme.
- Une étude complémentaire en utilisant des variations des conditions initiales permettrait d'évaluer la sensibilité de l'orbite du satellite à la position de la Lune.

4.4 Erreurs et limitations

- **Hypothèse d'orbite circulaire pour la Lune :** en réalité, la trajectoire de la Lune est légèrement elliptique avec une excentricité de 0.0549.
- **Négligence de l'influence d'autres corps célestes :** l'effet gravitationnel du Soleil et des autres planètes peut modifier la trajectoire du satellite.
- **Simplification des forces :** la pression de radiation solaire et la résistance atmosphérique à haute altitude (exosphère) ne sont pas prises en compte.
- **Approximations numériques :** l'utilisation de `ode23` peut introduire des erreurs d'intégration qui s'accumulent sur le long terme, et un pas de temps adaptatif pourrait améliorer la précision.
- **Modèle en 2D :** une extension en 3D permettrait d'obtenir une modélisation plus réaliste du mouvement orbital.

5 Conclusion Générale

Ce projet a permis l'exploration de la résolution numérique des équations différentielles appliquées à deux systèmes dynamiques : l'oscillateur de Van der Pol et la dynamique orbitale d'un satellite, en tenant compte de l'interaction avec la Lune. L'analyse des trajectoires a révélé l'influence significative des conditions initiales et des paramètres sur l'évolution du système, avec des comportements allant de la stabilité à l'instabilité.

La comparaison des méthodes d'intégration a montré que la méthode de Heun, malgré sa raideur, est très stable et prédictive en termes de trajectoires orbitales, surtout lorsqu'il s'agit du satellite et de la Lune, qui sont soumis à des forces dynamiques complexes. Cependant, la méthode `ode23` est très flexible, ayant adopté des modifications adaptatives dans les pas de temps ; cependant, elle est associée à de légères erreurs à long terme, surtout dans les systèmes auxquels une haute sensibilité est causée par de légères perturbations, par exemple, la Lune. Encore une fois, l'évaluation de l'énergie a confirmé l'accumulation conjointe des erreurs numériques, indiquant les limitations intrinsèques des méthodes utilisées pour simuler des systèmes dynamiques sur de longues périodes.

En conclusion, cette étude met en évidence le rôle crucial joué par le choix des algorithmes d'intégration et des paramètres de simulation pour garantir des résultats à la fois précis et fiables. L'exploration des trajectoires du satellite et de la Lune illustre également l'impact de la précision des méthodes d'intégration sur la représentation des interactions gravitationnelles dans des systèmes dynamiques complexes. Une compréhension plus approfondie de telles dynamiques est essentielle pour les applications futures, en particulier dans les domaines de l'exploration spatiale et de la simulation des systèmes planétaires élémentaires.