

# Calcul de la Distance Entre la Comète de Halley et le Soleil à Partir de l'Algorithme de Verlet Utilisant le Langage MATLAB

Rasolomampiany G.<sup>1</sup>, Randriamaroson R.M.<sup>2</sup>, Randimbindrainibe F.<sup>3</sup>

<sup>1</sup>Laboratoire de recherche Sciences Cognitives et Applications (LR - SCA)

<sup>2</sup>Ecole Doctorale en Sciences et Techniques de l'Ingénierie et de l'Innovation (ED - STII)

<sup>3</sup>Ecole Supérieure Polytechnique Antananarivo (ESPA) - Université d'Antananarivo

BP 1500, Ankatso – Antananarivo 101 – Madagascar



**Résumé** – En exploitant l'algorithme de Verlet et le langage MATLAB. Ce travail consiste à calculer les distances, en fonction du temps, entre le soleil et la comète de Halley.

**Mots clés** – MATLAB- algorithme de Verlet- Soleil- Comète de Halley.

## I. INTRODUCTION

Il est fréquent de vouloir étudier un système dynamique conservatif sur une longue durée. Par exemple, la question de la stabilité du système solaire, dont il est nul besoin de rappeler l'importance. La solution analytique étant inaccessible, on met en place un traitement numérique du problème. Les algorithmes classiques d'ordre élevé (Runge-Kutta d'ordre 4) ont tendance à fournir une dynamique aux temps courts de bonne qualité mais ont le défaut de produire une dérive de l'énergie aux temps longs. Dans ce cas, on a recours à des méthodes numériques dites *symplectiques*, particulièrement adaptées aux systèmes conservatifs et supérieures aux méthodes classiques dans le sens où elles conduisent à une dérive de l'énergie faible aux temps longs.

Parmi ces algorithmes, on choisit de ce qu'on appelle algorithme de Verlet. Et pour illustrer son utilisation, un spécialiste [1] a calculé la variation de la distance entre le soleil et la comète de Halley avec cet algorithme, en le couplant avec le langage Java.

Dans cet article on va reprendre le même problème. Mais on utilisera le langage Matlab

Ce travail divisera en trois parties :

- La première partie sera consacrée à la présentation de l'algorithme de Verlet
- La partie suivante est pour la résolution du problème
- La dernière partie est la conclusion, dans lequel on mentionnera des perspectives de recherche dans l'avenir.

## II. L'ALGORITHME DE VERLET [2]

### 2.1 L'algorithme de Verlet à deux pas

L'algorithme de Verlet est parmi les plus simples à mettre en œuvre et largement utilisé, plus particulièrement en dynamique moléculaire. Il repose sur le développement de Taylor du vecteur position à l'ordre 3 aux instants  $t + h$  et  $t - h$  :

$$\begin{cases} \vec{r}(t+h) = \vec{r}(t) + h\vec{v}(t) + \frac{h^2}{2}\vec{a}(\vec{r}(t)) + \frac{h^3}{3!}\frac{d^3\vec{r}(t)}{dt^3} + O(h^4) \\ \vec{r}(t-h) = \vec{r}(t) - h\vec{v}(t) + \frac{h^2}{2}\vec{a}(\vec{r}(t)) - \frac{h^3}{3!}\frac{d^3\vec{r}(t)}{dt^3} + O(h^4) \end{cases} \quad (1)$$

En sommant ces deux équations, la vitesse disparaît et l'on obtient, aux erreurs d'ordre 4 près, le schéma itératif suivant :

$$\vec{r}_{n+1} = 2\vec{r}_n - \vec{r}_{n-1} + \vec{a}_n h^2 \quad (2)$$

Cet algorithme présente deux défauts :

- Il s'agit d'un schéma à deux pas : l'itération ne peut démarrer que si l'on connaît  $\vec{r}_0$  et  $\vec{r}_{-1}$ . Or, en général, les conditions initiales se résument par la donnée de  $\vec{r}_0$  et  $\vec{v}_0$ . Il est d'usage alors d'initier l'itération à l'aide d'un développement de Taylor :

$$\vec{r}_{-1} = \vec{r}_0 - \vec{v}_0 h + \frac{h^2}{2}\vec{a}_0 \quad (3)$$

Si l'on s'intéresse à l'évolution d'une grandeur faisant intervenir la vitesse (comme l'énergie cinétique), celle-ci se calcule à l'aide de la formule :

$$\vec{v}_n = \frac{\vec{r}_{n+1} - \vec{r}_n}{2h} \quad (4)$$

On voit donc que le calcul de la vitesse consiste à soustraire deux nombres voisins ce qui produit des erreurs d'arrondi importants. C'est pourquoi, on utilise en général une autre version algorithmique qui est mathématiquement équivalente à la version originale de Verlet mais qui présente l'intérêt d'être à un pas

## 2.2 Algorithme de Verlet à un pas

L'algorithme de Verlet à un pas repose sur le schéma numérique suivant :

$$\begin{cases} \vec{r}_{n+1} = \vec{r}_n + h\vec{v}_n + \frac{h^2}{2}\vec{a}_n \\ \vec{v}_{n+1} = \vec{v}_n + \frac{h}{2}(\vec{a}_n + \vec{a}_{n+1}) \end{cases} \quad (5)$$

On procédera donc ainsi :

1. Initialisation du pas  $h$ , de la durée  $T$ .
2. Initialisation des conditions initiales :  $t = 0$ ,  $\vec{r} = \vec{r}_0$  et  $\vec{v} = \vec{v}_0$ .
3. Définition de la fonction  $\vec{a}(\vec{r})$ .
4. Tant que  $t \leq T$  faire :
  - a) Calcul de  $\vec{a}_0 = \vec{a}(\vec{r})$ .

b) Nouvelle position :  $\vec{r} = \vec{r} + h\vec{v} + \frac{h^2}{2}\vec{a}_0$

c) Calcul de  $\vec{a}_1 = \vec{a}(r)$ .

d) Nouvelle vitesse :  $\vec{v} = \vec{v} + \frac{h}{2}(\vec{a}_0 + \vec{a}_1)$

e)  $t = t + h$ .

f) Enregistrement des données

Remarque

Cette méthode, de part sa qualité et sa facilité de mise en œuvre est l'une des plus utilisée en dynamique moléculaire.

## 2.3 Distance entre le soleil et de la comète de Halley en fonction du temps

### 2.3.1 Propriété particulière d'un système de deux particules en interaction

Pour un système de deux particules situés en  $M_1$  et  $M_2$  à l'instant  $t$ , de masse respectives  $m_1$  et  $m_2$ , en interaction dans un référentiel galiléen  $R(O, x, y, z)$ .  $O$  est un point fixe de  $R$ .  $R^*$  est le référentiel barycentrique de centre  $C$ , en translation par rapport au référentiel  $R$ . Si  $\vec{f}_{2 \rightarrow 1}$  la force que subit  $M_1$  et que  $\vec{f}_{1 \rightarrow 2}$  la force que subit  $M_2$ . Alors

$$\frac{m_1 m_2}{m_1 + m_2} \left( \frac{d^2 \overrightarrow{M_1 M_2}}{dt^2} \right)_{R^*} = \vec{f}_{1 \rightarrow 2} \quad (6)$$

### 2.3.1 Paramétrisation du problème

Notons

$M$  : la masse du soleil

$m$  : la masse du comète de Halley

$r$  : la distance entre le soleil et la comète de Halley

$G$  : la constante gravitationnelle

D'après (6) et [1] et que  $M \gg m$  alors

$$\frac{Mm}{M+m} \frac{d^2 \vec{r}}{dt^2} = \vec{f}_{1 \rightarrow 2} = -GMm \frac{\vec{r}}{r^3} \quad \text{d'où} \quad D_{\alpha}^2 \vec{r} = -GM \frac{\vec{r}}{r^3} \quad (7)$$

### 2.3.2 Utilisation de l'algorithme de Verlet [1]

On va décrire le mouvement de la comète dans plan  $(O, x, y)$  et on prend comme point de départ le point le plus éloigné (l'aphélie) comme point de départ. On a alors  $x_0 = r_{\max}$ ,  $v_{x_0} = 0$ ,  $y_0 = 0$ , et  $v_{y_0} = v_{\min}$ . A partir des résultats bien connus, nous avons  $r_{\max} = 5.28 \times 10^{12} m$  et  $v_{\min} = 9.13 \times 10^2 m.s^{-1}$ . L'application de l'algorithme de Verlet à ce problème nous donne

$$\begin{aligned}
 x^{(k+1)} &= x^{(k)} + \tau v_x^{(k)} + \frac{\tau^2}{2} g_x^{(k)} \\
 v_x^{(k+1)} &= v_x^{(k)} + \frac{\tau}{2} [g_x^{(k+1)} + g_x^{(k)}] \\
 y^{(k+1)} &= y^{(k)} + \tau v_y^{(k)} + \frac{\tau^2}{2} g_y^{(k)} \\
 v_y^{(k+1)} &= v_y^{(k)} + \frac{\tau}{2} [g_y^{(k+1)} + g_y^{(k)}]
 \end{aligned} \tag{8}$$

L'indice de pas du temps est donné entre parenthèses en exposant afin de le distinguer de l'indice de la composante x ou de l'indice de la composante y. Les composants de l'accélération sont donnés par

$$\begin{aligned}
 g_x &= -\kappa \frac{x}{r^3} \\
 g_y &= -\kappa \frac{y}{r^3}
 \end{aligned} \tag{9}$$

avec  $r = \sqrt{x^2 + y^2}$  et  $\kappa = GM$

Nous pouvons utiliser des unités plus spécifiques dans les calculs, par exemple, 76 ans comme unité de temps et le demi-grand axe de l'orbite  $a = 2,68 \times 10^{12}$  m comme unité de longueur. Alors on a  $r_{\max} = 1.97$ ,  $v_{\min} = 0.816$ ,  $\kappa = 39.5$

### 2.3 Programmation avec le langage MATLAB

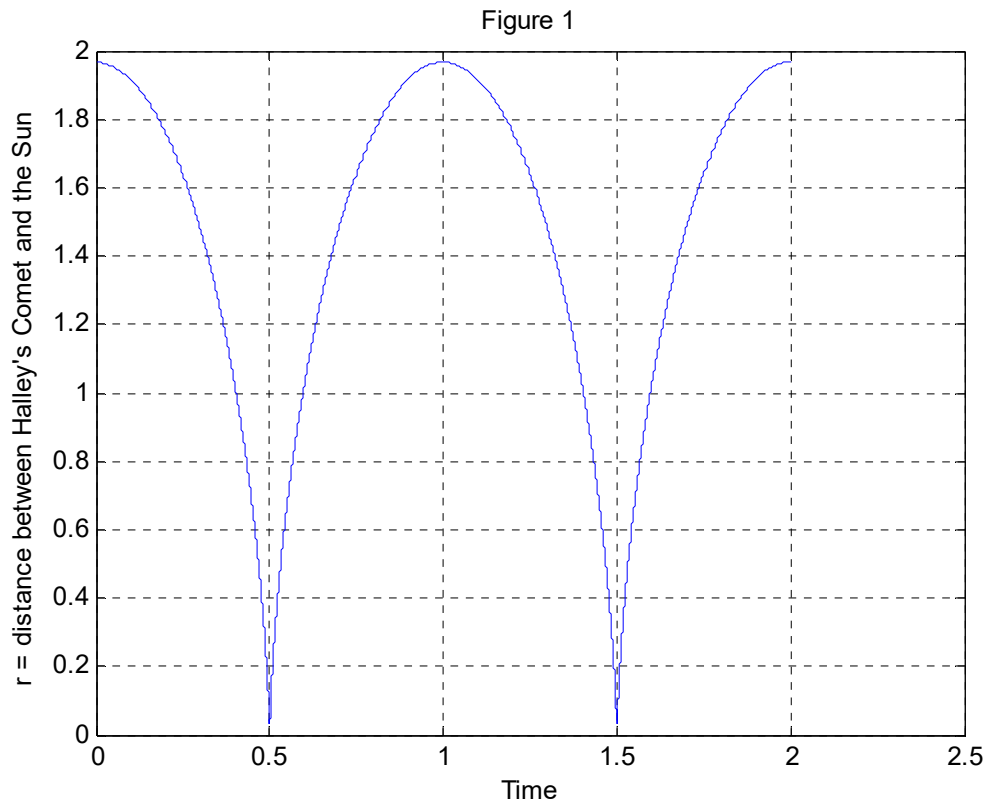
Le programme suivant est la mise en œuvre de l'algorithme décrit ci-dessus pour la comète de Halley.

```

close all
clc
clear
% Algorithme de Verlet
% Initialisation des variables
n= 20000 %input ('nombre de pas n=')
t = zeros(1,n);
x = t;
y = t;
r = t;
vx = t;
vy = t;
gx = t;
gy = t;
h = 2/n;

```

```
h2 = h^2/2;  
k = 39.478428;  
% Initialisation du problème  
x(1) = 1.966843;  
r(1) = x(1);  
vy(1) = 0.815795;  
gx(1) = -k / (r(1)^2);  
% Algorithme de Verlet pour la position et la vitesse  
for i = 1 : n  
    t(i+1) = h * (i+1);  
    x(i+1) = x(i) + h*vx(i) + h2*gx(i);  
    y(i+1) = y(i) + h*vy(i) + h2*gy(i);  
    r2 = x(i+1)^2 + y(i+1)^2;  
    r(i+1) = sqrt(r2);  
    r3 = r2 * r(i+1);  
    gx(i+1) = -k * x(i+1) / r3;  
    gy(i+1) = -k * y(i+1) / r3;  
    vx(i+1) = vx(i) + h * (gx(i+1)+gx(i))/2;  
    vy(i+1) = vy(i) + h * (gy(i+1)+gy(i))/2;  
end  
plot(t,r)  
grid on
```



La figure ci-dessus représente la distance entre la comète de Halley et le soleil, calculé avec le programme donné dans le texte. La période de la comète est utilisée comme unité de temps, et le demi-grand axe de l'orbite est utilisée comme unité de distance.

### III. CONCLUSION

On peut en conclure que le résultat obtenu à partir du langage Matlab est identique avec ce qui est obtenu dans [1]. La perspective de recherche prévue est de reprendre le même problème, en utilisant la version fractionnaire de l'équation de Newton.

### REFERENCES

- [1] Tao Pang . An Introduction to Computational Physics Second Edition. University of Nevada, Las Vegas. Cambridge University Press . © T. Pang 2006
- [2] Jimmy RousseL . Cours d'introduction à l'analyse numérique – femto-physique.fr. Copyright © 2021 Jimmy Roussel