

# INTÉGRATION PAR MÉTHODE D'EULER ET LEAP-FROG

## Partie I

### Introduction

En astro-informatique, la méthode d'Euler est souvent utilisée pour intégrer numériquement les équation du mouvement des particules en interactions gravitationnelles. On va étudier ici le cas particulier d'une unique masse  $m$  soumise à l'attraction gravitationnelle d'une masse  $M$  bien plus importante placée au centre du repère. Les équations du mouvement s'écrivent alors de manière vectorielle

$$m \frac{d^2 \vec{r}}{dt^2} = -\frac{GMm}{r^2} \vec{e}_r$$

On a alors le choix d'intégrer les équations en coordonnées cartésiennes ou en coordonnées polaires. Celle-ci s'écrivent

$$\begin{cases} \ddot{x} = -\frac{GM}{(x^2 + y^2)^{3/2}} x \\ \ddot{y} = -\frac{GM}{(x^2 + y^2)^{3/2}} y \end{cases} \quad \text{ou} \quad \begin{cases} \ddot{r} - r \dot{\theta}^2 = -\frac{GM}{r^2} \\ r \ddot{\theta} + 2\dot{r} \dot{\theta} = 0 \end{cases}$$

En bons astro-informaticiens, nous prendrons  $G = 1$  et aussi, pour faire bonne mesure,  $M = 1$  de sorte que les équations soient vraiment simples à écrire. Il ne reste plus qu'à se concentrer sur l'algorithme.

## Partie II

### Méthode d'Euler

On va intégrer les deux systèmes d'équation par la méthode d'Euler pour vérifier si le choix du repère influence la précision des mesures. On va aussi comparer l'influence du pas de temps ( $dt=1e-2$  ou  $1e-3$  ou  $1e-4$ ) sur les résultats d'intégration, notamment sur la conservation de l'énergie (calcul de  $E = v^2/2 - GM/r$  à chaque pas de sortie). L'algorithme à considérer est donc le suivant:

```

1  ## Initialisation des valeurs:
2  ## (x,y) = (1,0)
3  ## (vx,vy)= (0,v0)  ## (v0=1 correspond à une trajectoire circulaire si G=M=1)
4  ##
5  ## On itère le nombre voulu de fois pour aller de t=0 à t=10
6  ##      On commence par calculer les incréments de vitesse et de position:
7  ##          dvx= ax*dt
8  ##          dx = vx*dt ...
9  ##      On met à jour vitesses, positions, énergie et temps
10 ##          vx+= dvx
11 ##          x += dx ...
12 ##          t += dt
13 ##          E = 0.5*v**2 - 1/r

```

Il suffira de stocker ces valeurs tous les  $10^{-2}$  unité de temps pour limiter le nombre total de points à afficher à 1000 et pouvoir facilement comparer les différentes versions suivant le pas de temps et le type de repère choisi. Il va donc falloir écrire deux fonctions `euler_cartesien(x0,vy0,dt)` et `euler_polaire(x0,vy0,dt)` qui acceptent toutes deux en entrée les valeurs de position<sup>1</sup> et de vitesse initiale (en coordonnées cartésiennes

<sup>1</sup>On imposera de surcroît  $x_0 > 0$  pour éviter d'avoir à traiter trop de cas particuliers dans la procédure.

à chaque fois) ainsi que la valeur du pas de temps  $\mathbf{dt}$ . Ces deux fonctions doivent renvoyer 4 tableaux  $\mathbf{x}, \mathbf{y}, \mathbf{t}, \mathbf{E}$  comprenant respectivement les coordonnées cartésiennes  $x$  et  $y$ , ainsi que les temps  $t$  et énergie  $E$  pour chacun des 1000 points que l'on utilisera pour l'affichage final. On rappelle les formules utiles permettant le passage initial de cartésiennes à polaires dans ces conditions ( $y_0 = 0$  et  $v_{x0} = 0$ ) :

$$r_0 = x_0, \quad \theta_0 = 0, \quad \dot{r}_0 = 0 \quad \text{et} \quad \dot{\theta}_0 = \frac{v_{y0}}{r_0}$$

et réciproquement à tout instant pour renvoi en fin de procédure:

$$x = r \cos \theta \quad \text{et} \quad y = r \sin \theta$$

Partie III

## Version Leap-Frog

La version « saute-mouton » de l'algorithme d'Euler a plusieurs avantages sur sa version simple: meilleures conservation de l'énergie et invariance par renversement du temps en sont deux facettes. Pour limiter les complications, on se contentera de l'implémenter en coordonnées cartésiennes, mais avec des conditions initiales arbitraires en  $x$  et en  $y$  pour pouvoir vérifier le caractère invariant par retournement temporel.

La méthode d'Euler précédente (à une seule variable) aurait pu s'écrire comme une suite de valeurs telles que

$$x_{n+1} = x_n + v_n \, dt \quad \text{et} \quad v_{n+1} = v_n + a_n \, dt$$

où l'accélération à l'étape  $n$  ne dépend que de  $x_n$ . En d'autres termes, les nouvelles valeurs des fonctions  $x$  et  $v$  sont calculées entièrement à partir des anciennes valeurs. Avec la méthode « saute-mouton », vitesse et position ne sont plus évaluées aux mêmes instants, la vitesse étant calculée pour des pas « demi-entiers »:

$$x_{n+1} = x_n + v_{n+1/2} \, dt \quad \text{et} \quad v_{n+1/2} = v_{n-1/2} + a_n \, dt$$

L'écriture semble bizarrement équivalente à la précédente, mais la différence vient de l'initialisation. En donnant un couple  $(x_0, v_0)$  de conditions initiales, on initialise bien  $x$  à  $x_0$ , mais pour la vitesse, on prend

$$v_{1/2} = v_0 + a_0 \frac{dt}{2}$$

de sorte que position et vitesse puisse jouer à saute-mouton à tour de rôle l'un par-dessus l'autre. Vous pouvez à présent écrire la fonction `leapfrog(position,vitesse,dt)`, où `position` et `vitesse` sont maintenant des doublets pour la position initiale  $(x_0, y_0)$  et la vitesse initiale  $(v_{x0}, v_{y0})$ , et la comparer à `euler(position,vitesse,dt)`, notamment au niveau de la conservation de l'énergie et de la réversibilité par renversement du temps (en d'autres termes, si on donne le point d'arrivée en  $t = 10$  pour les deux procédures et que l'on inverse le signe des vitesses, la particule se retrouve-t-elle à  $t = 20$  à l'endroit où elle se trouvait à  $t = 0$  ?).