T.P. numéro III

Discrétisation de la dynamique Newtonienne

L'objectif de ce TP est de créer un code C qui permette de construire les trajectoires de points matériels soumis à diverses forces (poids, frottement, gravité) de façon *numérique*, c'est-à-dire sans résoudre analytiquement l'équation différentielle de la dynamique :

$$\vec{f} = m\vec{\gamma} = m\frac{d\,\vec{v}}{dt} \tag{1}$$

Cette équation signifie qu'à un instant donné t on a

$$\frac{\vec{f}(t)}{m} = \lim_{dt\to 0}\left(\frac{\vec{v}(t+dt) - \vec{v}(t)}{dt}\right)$$

ou encore que

$$\vec{v}(t+dt) = \vec{v}(t) + \frac{\vec{f}(t)}{m}dt$$

quand dt tend vers 0. De même, par définition, la vitesse est la dérivée du vecteur position $\vec{OM}$ et on a

$$\vec{OM}(t+dt) = \vec{OM}(t) + \vec{v}(t)dt$$

quand dt tend vers 0.
Dans le cas de mouvements plans, on a $\vec{v}(v_X, v_y, 0)$, $\vec{OM}(x, y, 0)$ et $\vec{f}(f_x, f_y, 0)$ ce qui nous donne 4 équations :

$$v_x(t+dt) = v_x(t) + \frac{f_x(t)}{m}dt \tag{2a}$$

$$v_y(t+dt) = v_y(t) + \frac{f_y(t)}{m}dt \tag{2b}$$

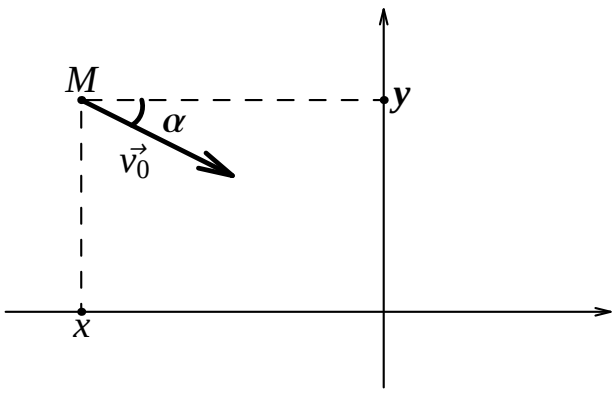
$$x(t+dt) = x(t) + v_x(t)dt \tag{2c}$$

$$y(t+dt) = y(t) + v_y(t)dt \tag{2d}$$

(2e)

On voit bien à partir de ces équations que si on connait $\vec{OM}$ et $\vec{v}$ à un instant t donné on est capable de déterminer $\vec{OM}$ et $\vec{v}$ à un instant ultérieur $t + dt$ et ainsi de construire pas à pas la trajectoire d'un point matériel au cours du temps.

1 Le principe d'inertie



Sauvegarder et éditer le code source du programme dynamique.c.
Dans ce programme, on fixe les conditions initiales du mouvement comme représentées sur le schéma. La norme v_0 de la vitesse initiale et l'abscisse initiale x sont fixées dans le code ; l'ordonnée intiale y et l'angle α que fait le vecteur vitesse avec l'horizontale sont demandés à l'utilisateur au moment de l'exécution.
Les résultats du programme sont écrits dans le fichier `trajectoire` à la fin de l'exécution.

Exercice III.1

1. Quel est le rôle de la fonction `position` dans ce programme ?
2. Y-a-t'il une force appliquée au point matériel ?
3. À l'aide d'une structure de boucle, modifier le programme principal (fonction main) pou calculer toute la trajectoire depuis l'instant initial jusqu'à ce qu'elle coupe l'axe Ox.
4. Compiler le code avec l'option `-lm` et exécuter le programme pour $\alpha = -30°$et $y = 10$. Tracer la trajectoire en exécutant la commande `xmgrace position`. Quelle est la nature de la trajectoire ?
5. Sauvegarder le fichier `trajectoire` sous un autre nom.

2 Trajectoires balistiques

On décide maintenant de soumettre la particuleà son poids $\vec{P} = -mg\vec{e_y}$. Cette force étant entièrement dirigée suivant l'axe Oy, la vitesse suivant l'axe Ox restera inchangée au cours du mouvement. La vitesse suivant l'axe Oy en revanche va varier selon l'équation (2b) qui se récrit ici :

$$v_y(t + dt) = v_y(t) - gdt$$

Exercice III.2

1. En s'inspirant de la fonction `position`, écrire la fonction `vitesse_y` qui calcule et renvoie la valeur de la nouvelle vitesse (suivant Oy) à l'instant $t + dt$ en fonction de la vitesse (suivant Oy) à l'instant t et du pas de temps dt (la constante `g` est déjà définie dans le code).
2. Ajouter les lignes de code dans la boucle du programme principal qui vont effectuer la calcul de la composante y de la vitesse (en appelant la fonction `vitesse_y`) à chaque nouveau temps.
3. Compiler et exécuter le programme pour $\alpha = 45°$et $y = 0$. Tracer la trajectoire. Quelle est la nature de cette trajectoire ?
4. À l'aide d'une instruction **if** plaçée à la fin de la boucle du programme principal, calculer l'altitude maximale y atteinte par le point matériel sur sa trajectoire. Calculer également la portée qui est la distance suivant l'axe Ox entre le point initial et le point final. Faire affichier ces deux valeurs par le programme.
5. Compiler et exécuter le programme pour les mêmes conditions initiales pour lire les valeurs de l'altitude maximale et de la portée affichées. Sauvegarder le fichier `trajectoire` sous un autre nom.

En plus de son poids, on soumet maintenant notre point matériel également à une force de frottement $\vec{f} = -\mu m\vec{v}$. Les équations (2a) et (2b) se récrivent alors

$$v_x(t + dt) = v_x(t) - \mu v_x(t)\, dt$$
$$v_y(t + dt) = v_y(t) - [g + \mu v_y(t)]\, dt$$

Exercice III.3

1. Modifier la fonction vitesse_y pour qu'elle prenne en compte la force de frottement et écrire la fonction vitesse_x, analogue de vitesse_y pour la composante x (la constante `mu` est déjà définie dans le programme).
2. Ajouter les lignes de code qui appellent vitesse_x dans la boucle du programme principal.
3. Compiler, exécuter et tracer la trajectoire avec les mêmes conditions initiales qu'à l'exercice III.2. Que constate-t-on ?
4. Sauvegarder le fichier `trajectoire` sous un autre nom.

3 Gravitation universelle

On soumet cette fois notre point matériel à la seule force de gravité $\vec{F_g} = -\frac{GMm}{r^2}\vec{e_r}$ qui se récrit en coordonnées cartésiennes

$$\vec{F_g} = -\frac{GMm}{r^3}\left(x\,\vec{e_x} + y\,\vec{e_y}\right)$$

avec $r = \sqrt{x^2+y^2}$. Les équations (2a) et (2b) se récrivent alors

$$v_x(t+dt) = v_x(t) - \frac{GMx(t)}{r^3(t)}\,dt$$
$$v_y(t+dt) = v_y(t) - \frac{GMy(t)}{r^3(t)}\,dt$$

Exercice III.4

1. Écrire une nouvelle fonction vitesse_grav qui permette de calculer les nouvelles valeurs des composantes x et y de la vitesse à l'instant $t+dt$.
   Dans le programme principal on fera appel à vitesse_grav (x,y,vx,dt) pour calculer la nouvelle valeur de v_x et à vitesse_grav (y,x,vy,dt) pour calculer la nouvelle valeur de v_y (la constante `GM` est déjà définie dans le code).
2. Modifier les lignes du code qui calculent la nouvelle vitesse pour qu'elles fassent appel à vitesse_grav .
3. On peut montrer analytiquement qu'en choisissant les conditions initiales $\alpha = 90$ et $y = 0$, avec les autres valeurs déjà fixées dans le code on doit obtenir une trajectoire circulaire.
   Compiler, exécuter, et tracer la trajectoire avec ces valeurs. Que constate-t-on ? Sauvegarder le fichier `trajectoire` sous un autre nom.
4. Diviser par 10 la valeur du paramètre `dt` dans le code et reprendre la question 3. Conclusion ?