

Microméca :

Calcul de la trajectoire d'un point matériel

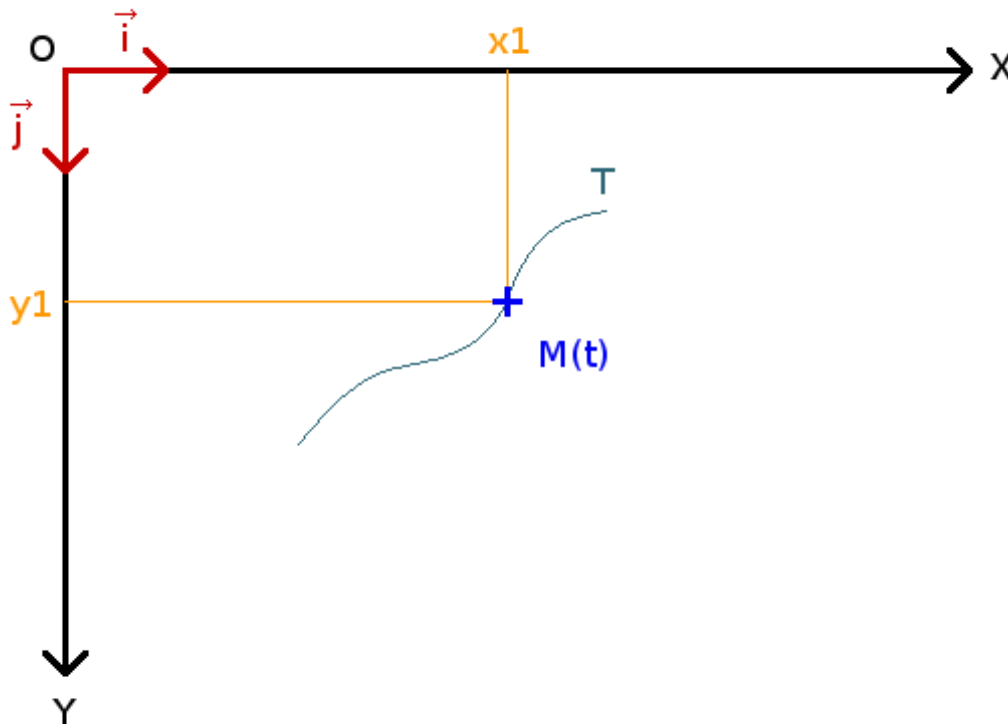
2ème loi de Newton

Méthode d'Euler

1 - Expression de la 2ème loi de Newton

La 2ème loi de Newton relie l'aspect cinématique - description de la trajectoire - et l'aspect dynamique - les forces. Un référentiel est choisi et le système étudié est le point matériel M soumis à des forces de résultante \vec{F} .

Les axes du repère sont choisis suivant la convention des écrans informatiques : origine O en haut à gauche, axe des abscisses x vers la droite et axe des ordonnées y vers le bas :



Dans l'enseignement de spécialité de première (2019), la 2ème loi de Newton est exprimée comme une variation du vecteur vitesse $\Delta\vec{V} = \vec{V}_2 - \vec{V}_1$ observée au bout d'une durée $\Delta t = t_2 - t_1$, liée à la résultante des forces \vec{F} , c'est à dire la somme vectorielle des forces appliquées à un instant t au point matériel de masse m :

$$\vec{F} = m \frac{\Delta\vec{V}}{\Delta t} \quad \text{soit :} \quad \Delta t \cdot \vec{F} = m \cdot \Delta\vec{V} \quad \text{ou encore :} \quad \Delta\vec{V} = \vec{V}_2 - \vec{V}_1 = \frac{\Delta t}{m} \vec{F}$$

Si la résultante des forces \vec{F} est connue, ainsi que la vitesse \vec{V}_1 à l'instant initial t_1 , la masse m du point matériel et l'intervalle de temps $\Delta t = t_2 - t_1$, alors la vitesse \vec{V}_2 à l'instant t_2 peut être calculée :

$$\vec{V}_2 = \vec{V}_1 + \frac{\Delta t}{m} \vec{F}$$

Ceci se traduit par une égalité entre les composantes suivant \vec{i} et \vec{j} de ces vecteurs :

$$V_{2x} = V_{1x} + \frac{\Delta t}{m} F_x \quad \text{et} \quad V_{2y} = V_{1y} + \frac{\Delta t}{m} F_y$$

Les membres de droite sont connus et permettent de trouver les composantes V_{2x} et V_{2y} de la vitesse à l'instant t_2 .

2 - Position et vitesse du point matériel M

La position du point matériel M est caractérisée par le vecteur \overrightarrow{OM} de coordonnées x et y .

Lorsque le point matériel M se déplace de la position $M_1(x_1; y_1)$ à la position $M_2(x_2; y_2)$, ses coordonnées varient de $\Delta x = x_2 - x_1$ et $\Delta y = y_2 - y_1$ entre les instants t_1 et t_2 .

La vitesse du point matériel est alors définie par :

$$\vec{V} = \frac{1}{t_2 - t_1} \overrightarrow{M_1 M_2}$$

soit une relation entre les coordonnées de ces vecteurs :

$$V_x = \frac{x_2 - x_1}{t_2 - t_1} = \frac{\Delta x}{\Delta t} \quad \text{et} \quad V_y = \frac{y_2 - y_1}{t_2 - t_1} = \frac{\Delta y}{\Delta t}$$

Connaissant la vitesse \vec{V} du point matériel et sa position initiale M_1 , on peut trouver les coordonnées x_2 et y_2 de sa position finale M_2 :

$$x_2 = x_1 + V_x(t_2 - t_1) = x_1 + V_x \cdot \Delta t \quad \text{et} \quad y_2 = y_1 + V_y(t_2 - t_1) = y_1 + V_y \cdot \Delta t$$

3 - Mise en oeuvre de ces relations : Microméca

Trois scripts en langage Python illustrent l'utilisation possible des relations qui précèdent. Le premier ([micromeca_01_02.py](#)) permet de calculer et de tracer la trajectoire d'un point matériel soumis à une force constante : son poids $\vec{P} = m \cdot \vec{g}$.

Le calcul de la trajectoire (méthode d'Euler) consiste à effectuer, pour un grand nombre de pas de temps très petits Δt , le calcul de l'évolution de la vitesse de \vec{V}_1 à \vec{V}_2 (lignes 63 et 64 du script), puis l'évolution de la position M_1 à la position M_2 (lignes 66 et 67 du script).

Le pas de temps Δt ne doit pas être trop grand : la vitesse doit pouvoir être considérée comme constante pour le calcul de x_2 à partir de x_1 et de y_2 à partir de y_1 .

Le pas de temps Δt ne doit pas être trop petit pour ne pas obtenir un nombre trop important d'itérations, ce qui entraînerait des erreurs d'arrondi qui s'accroissent d'itération en itération.

Le script [micromeca_02_02.py](#) fait exactement le même calcul, mais la force est calculée dans une fonction à chaque itération, ce qui ne change rien ici mais permet d'envisager de traiter le cas d'une force qui varie avec la position et la vitesse du point matériel.

Le script [micromeca_03_02.py](#) traite le cas d'une force de rappel vers un point central par un ressort. Un ressort est caractérisé par sa raideur en N.m^{-1} : plus le ressort est étiré, plus il exerce une force de rappel importante (lignes 59 et 64). Dans l'exemple de ce script, des valeurs différentes de la raideur ont été choisies pour l'étirement en x et en y (lignes 21 et 22).

4 - Conditions initiales

Le calcul précédent nécessite que des conditions initiales soient précisées : position initiale $(x_0; y_0)$ et vitesse initiale $(V_{0x}; V_{0y})$. Ceci est fait lignes 13 à 16 dans le script [micromeca_01_02.py](#). Sans ces conditions initiales, le calcul de proche en proche ne peut pas démarrer.

5 - Travail

- Comprendre le fonctionnement de ces scripts ([micromeca_01_02.py](#) est le plus commenté).
- Noter que l'expression de la 2ème loi de Newton (§1) et la définition de la vitesse (§2) sont utilisables telles quelles pour calculer numériquement la vitesse et la position du point matériel à tout instant, ainsi que sa trajectoire.
- Etudier l'impact du choix du pas de temps, ligne 24 du script : 0.01 ; 0.1 ; 0.001.
- Etudier l'impact de la position initiale puis de la vitesse initiale.
- Etudier l'impact de l'intensité de la pesanteur g dans le cas du point matériel soumis à son poids (ligne 18), ou de la raideur du ressort pour le 3ème script (lignes 21 et 22).
- Etudier l'impact de la masse du point matériel (ligne 17)

6 - Prolongement

Il est possible d'introduire toute autre expression de la force à laquelle le point matériel est soumis (fonctions `force_x()` et `force_y()` des scripts Microméca).

Le calcul de la trajectoire par l'algorithme d'Euler utilisé ici est simple à [comprendre](#) et intéressant dans son principe. Il existe cependant des algorithmes plus performants, à la fois plus rapides et plus précis. Les scripts [Mécablab](#) par exemple utilisent un algorithme de Runge-Kutta. L'étude d'algorithmes performants pour le calcul sur ordinateur relève de l'Analyse Numérique.