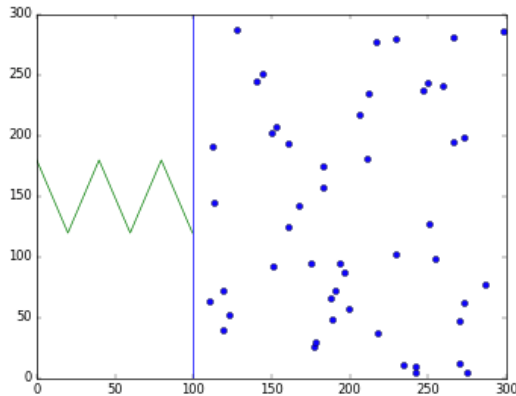


TP IPT : simulation de particules



Le but de ce TP est de réaliser une simulation de particules rebondissantes sur une paroi mobile sur un ressort. On supposera que cette paroi est verticale, comme sur l'image ci-dessus. Une fonction d'affichage est fournie, qui appelle la fonction `etape`, à remplir, devant mettre à jour les coordonnées des particules et du ressort après un temps `dt`.

Une particule est un tableau `numpy` contenant 5 flottants : son abscisse, son ordonnée, sa vitesse suivant les x et y et sa masse.

L'ensemble des particules est stocké dans une liste `particules`.

`ressort` est un tableau `numpy` contenant, dans l'ordre : l'abscisse de la paroi, sa vitesse suivant l'axe des x , sa raideur, sa longueur au repos et sa masse.

I Initialisation

1. Initialiser `ressort` avec des valeurs raisonnables (on prendra par exemple une raideur de 1.).
2. Écrire une fonction `init_particules` qui prend comme arguments le nombre de particules, la vitesse et la masse maximum d'une particule et qui initialise la liste `particules`, avec des valeurs aléatoires. Pour cela on utilisera `np.random.random()` qui renvoie un flottant uniformément au hasard entre 0. et 1.
Tester votre programme.

II Collision

Dans un premier temps, on considère que la paroi et le ressort sont immobiles.

1. Écrire une fonction `deplacer_particule` ayant une particule `p` en argument et faisant avancer `p` suivant son vecteur vitesse pendant un temps `dt`. On gèrera les collisions avec les parois : dans ce cas, on inverse son vecteur vitesse avant de déplacer la particule.
Tester votre programme en appelant `deplacer_particule` sur toutes les particules, dans `etape`.

On va maintenant autoriser le mouvement du ressort.

2. Écrire une fonction `force_ressort` renvoyant la force du rappel du ressort : $-k(x - x_0)$, où k est la constante de raideur, x sa position actuelle et x_0 sa position de repos.
3. Écrire une fonction `deplacer_ressort` appliquant la force de rappel au ressort et le déplaçant suivant son vecteur vitesse pendant un temps `dt`.
Tester votre programme en appelant aussi `deplacer_ressort` dans `etape`.

On souhaite prendre en compte les chocs des particules sur la paroi mobile. On suppose ces chocs élastiques : il y a conservation de la quantité de mouvement et de l'énergie cinétique.

1. Si une particule de masse m_1 , vitesse v_1 entre en collision avec la paroi de masse m_2 et vitesse v_2 , quelles sont les vitesses après collision ?
2. En déduire une fonction `choc` renvoyant ces nouvelles vitesses.
3. Modifier `deplacer_particule` de façon à ce quelle prennent en compte ces chocs.
Tester votre programme avec différentes vitesses initiales, raideur...

III Bonus

1. Appliquer une force de gravité aux particules.
2. Créer un deuxième ressort en face du premier.