

1 Etape 1 : récupération des données en input

1. Champ de vitesse 3D, de dimension $(dim_x \times dim_y \times dim_z)$
2. SpaceFactor en mm/voxel (espace entre 2 voxels de la matrice 3D)
3. Positions des centres des billes et leur rayon respectifs (dimension : $nbredebille \times 4$)

2 Etape 2 : estimation de spaceMultiplier1

Paramètre calculé automatiquement par PA pour que, peu importe la taille du champ de vitesse, il ne soit pas trop grand ni trop petit : On sait que pour la matrice de 357 voxels sur x, avec son spaceFactor de 0.050439 mm/voxel, les dimensions sont agréables vis-à-vis du collider (pas trop petit, ni trop grand). La taille du milieu exploré dans notre cas est donc :

$$357 \times 0.050439 = 1.8mm \sim 2mm \quad (1)$$

On calcule alors le spaceMultiplier1 :

$$spaceMultiplier1 = \frac{2mm}{\max(dim_x, dim_y, dim_z) \times SpaceFactor} \quad (2)$$

3 Etape 3 : Paramètre ajustable d'espace et de temps

Choix par l'utilisateur (idéalement dans la fenêtre de choix des réglages d'affichage) d'un paramètre d'ajustement spatial (s'il veut modifier le ratio entre la taille du collider et le milieu exploré) et d'un paramètre d'ajustement temporel (s'il veut explorer un champ de vitesse où les vitesses sont trop lentes à son gout et qu'il veut les accélérer par exemple)

$$spaceMultiplier2 = 1 \text{ (valeur par defaut)} \quad (3)$$

$$TimeMultiplier = 1 \text{ (valeur par defaut)} \quad (4)$$

4 Etape 4 : Multiplication des échelles spatio-temporelles par ces valeurs

$$Vx = Vx \times spaceMultiplier1 \times spaceMultiplier2 \times TimeMultiplier \quad (5)$$

$$Vy = Vy \times spaceMultiplier1 \times spaceMultiplier2 \times TimeMultiplier \quad (6)$$

$$Vz = Vz \times spaceMultiplier1 \times spaceMultiplier2 \times TimeMultiplier \quad (7)$$

$$Position\ billes = Position\ billes \times spaceMultiplier1 \times spaceMultiplier2 \quad (8)$$

$$Rayon\ billes = Rayon\ billes \times spaceMultiplier1 \times spaceMultiplier2 \quad (9)$$