

**PHS4700**


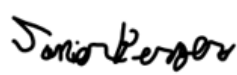


**Physique pour les applications multimédia**

**PAGE COUVERTURE OBLIGATOIRE POUR TOUS LES DEVOIRS**

**Numéro du groupe : 01**

**Numéro de l'équipe : 02**

**Numéro de devoir : 03**

Nom : Diop	Prénom : Abdul Hamid	Matricule : 2141605
Signature : 		
Nom : Berger	Prénom : Javier	Matricule : 2206989
Signature : 		
Nom : Ngandjui Tchuenta	Prénom : Ewald Jordan	Matricule : 2029689
Signature : 		
Nom : Jourba	Prénom : Alexandra	Matricule : 2413451
Signature : 		

## **Table des matières**

<b>1. Introduction.....</b>	<b>3</b>
<b>1. Résolution.....</b>	<b>4</b>
1.1. Simulation du mouvement.....	4
1.2. Détection des collisions.....	7
1.3. Calcul des vitesses après collision.....	8
<b>2. Présentation et analyse des résultats.....</b>	<b>9</b>
2.1. Résultats du tir 1.....	10
2.2. Résultats du tir 2.....	11
2.3. Résultats du tir 3.....	12
2.4. Résultats du tir 4.....	13
2.5. Résultats du tir 5.....	14
2.6. Résultats du tir 6.....	15
2.7. Analyse des résultats.....	16
<b>3. Conclusion.....</b>	<b>17</b>

# 1. Introduction

Le présent rapport a pour objectif d'étudier le déplacement et la collision de deux objets en mouvement : un bloc cubique et une balle.

À l'instant initial, un bloc cubique de masse et de volume connue est lancé depuis un canon, tandis qu'une balle, dirigée par un joueur concurrent, est lancée dans le but d'entrer en collision avec ce bloc et de le faire dévier.

La simulation se limite aux effets de la gravité, négligeant les frottements et autres résistances, pour simplifier les calculs et se concentrer sur la modélisation des trajectoires et de l'impact de la collision. De plus, la précision de cette simulation est cruciale, car elle doit permettre de prédire l'issue de chaque interaction entre les objets avec un degré de précision au millimètre près.

Ce devoir implique donc de programmer une fonction en Matlab capable de simuler et de suivre les trajectoires des objets, de détecter les éventuelles collisions, et de calculer les vitesses linéaires et angulaires des objets après l'impact. En plus de gérer les collisions, cette fonction devra également identifier le premier objet à toucher le sol si aucune collision n'a lieu. Une fois cette simulation achevée, les résultats obtenus fourniront des données cruciales, permettant de valider l'efficacité du moteur physique développé pour cette application.

Ainsi, ce rapport présentera le développement de la fonction, les principes théoriques sous-jacents, et les résultats obtenus, contribuant à une simulation de jeu réaliste et engageante pour les utilisateurs.

## 2. Résolution

Plusieurs notions vues en classe, notamment la détection des collisions ainsi que les résultats de celles-ci selon la méthode des conditions initiales, ont été implémentés dans le cadre de ce laboratoire. Notons que seule la force de la gravité affectera la sphère et le cube durant leurs trajectoires:

$$\vec{F}_g = m\vec{g}$$

où:

m est la masse de l'objet en question

$$\vec{g} = (0, 0, -9.8) \text{ m/s}^2$$

### 2.1. Simulation du mouvement

Pour simuler le déplacement des objets, nous utilisons l'algorithme de Runge-Kutta d'ordre 4.

#### Conditions Initiales :

- Bloc
  - masse: 1.2kg
  - longueur des arêtes: 0.08m
  - position initiale: (3, 3, 1)m
  - les vitesses linéaires et angulaires sont des paramètres
- Balle
  - masse: 0.05kg
  - rayon: 0.02m
  - Position initiale: (0, 0, 2)m
  - Vitesse angulaire initiale: (0, 0, 0) rad/s
  - La vitesse linéaire au moment du tir est un paramètre

#### Équations du mouvement :

- positions :

$$\frac{d\vec{r}_{\text{bloc}}}{dt} = \vec{v}_{\text{bloc}}$$

- vitesses linéaires :

$$\frac{d\vec{v}_{\text{bloc}}}{dt} = \vec{a}_{\text{bloc}} = \frac{\vec{F}_g}{m_{\text{bloc}}} = \vec{g}$$

$$\frac{d\vec{v}_{\text{balle}}}{dt} = \vec{a}_{\text{balle}} = \frac{\vec{F}_g}{m_{\text{balle}}} = \vec{g}$$

En l'absence d'autres forces que le poids, aucun moment de force ne pourra faire varier la vitesse angulaire des objets. Seule la collision pourra apporter des changements à ce niveau.

La résolution de l'équation différentielle vectorielle suivante permet d'obtenir la vitesse et la position des objets à un instant  $t$  :

$$\frac{d\vec{q}(t)}{dt} = \vec{g}(\vec{q}(t), t)$$

Avec les conditions initiales suivantes:

$$\vec{q}(0) = \vec{q}_0 = \begin{pmatrix} \vec{v}(0) \\ \vec{r}(0) \end{pmatrix}$$

Nous avons donc le vecteur d'état pour chaque objet de la forme suivant:

$$\vec{q}(t) = \begin{pmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \\ r_x(t) \\ r_y(t) \\ r_z(t) \end{pmatrix} = \begin{pmatrix} q_1(t) \\ q_2(t) \\ q_3(t) \\ q_4(t) \\ q_5(t) \\ q_6(t) \end{pmatrix}$$

Finalement, nous avons la fonction suivante :

$$\vec{g}(\vec{q}(t), t) = \begin{pmatrix} a_x(t) \\ a_y(t) \\ a_z(t) \\ q_1(t) \\ q_2(t) \\ q_3(t) \end{pmatrix}$$

Pour résoudre numériquement l'équation différentielle précédente, nous utilisons la méthode de Runge-Kutta d'ordre quatre afin de déterminer une approximation précise de la position et de la vitesse à un instant  $t+dt$ , où  $dt$  est un pas de temps déterminé.

Nous commençons par définir le pas temporel de résolution  $dt$ , établi en fonction des besoins de précision de la simulation. Dans notre cas, nous avons choisi :

$$dt=1\times 10^{-4}s$$

La solution est alors approximée itérativement à partir des conditions initiales, en utilisant la formule suivante :

$$\vec{q}(t_{i+1}) = \vec{q}(t_i) + \frac{\Delta t}{6} \left( \vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4 \right)$$

Avec :

$$\vec{k}_1 = g(\vec{q}(t_i), t_i)$$

$$\vec{k}_2 = g \left( \vec{q}(t_i) + \frac{\Delta t}{2} \vec{k}_1, t_i + \frac{\Delta t}{2} \right)$$

$$\vec{k}_3 = g \left( \vec{q}(t_i) + \frac{\Delta t}{2} \vec{k}_2, t_i + \frac{\Delta t}{2} \right)$$

$$\vec{k}_4 = g \left( \vec{q}(t_i) + \Delta t \vec{k}_3, t_i + \Delta t \right)$$

En pratique, cela se traduit par une boucle qui parcourt tous les instants  $t_i$  de 0 à  $t_{\max}$  avec un pas de  $dt$ . À chaque itération, nous mettons à jour la solution courante pour obtenir les nouvelles positions et vitesses :  $t_{i+1}=t_i+dt$

Nous appliquons la méthode RK4 pour mettre à jour le vecteur d'état du bloc qui contient les vitesses et positions du bloc.

Les vitesses angulaires sont mises à jour séparément si nécessaire, notamment après une collision.

L'orientation du bloc est représentée par un quaternion  $q_{\text{bloc}}$  à jour à chaque pas de temps en utilisant la démarche suivante :

1) On calcule l'angle de rotation pour le dt choisi :

$$\theta = \|\omega\| \cdot \Delta t$$

2) On crée un quaternion de rotation qui représente le changement en orientation :

$$q_{\text{rot}} = [\cos(2\theta); \sin(2\theta) \cdot \|\omega\| \omega]$$

3) On met à jour le quaternion avec la multiplication de quaternion (implémentation fourni):

$$q_{\text{bloc}}(t_i+1) = q_{\text{rot}} \otimes q_{\text{bloc}}(t_i)$$

4) Finalement, on normalise le quaternion en la divisant par son module

Notons que l'on peut se permettre d'utiliser cette méthode au lieu de runge kutta sans perdre de précision car la vitesse angulaire du bloc est constante durant sa trajectoire. Cela est dû à l'absence de forces tels que le frottement visqueux de l'air.

Ensuite, nous appliquerons la rotation au bloc à chaque itération en convertissant le quaternion en matrice de rotation:

$$\mathbf{R} = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix}$$

La même démarche est utilisée pour la balle.

## 2.2. Détection des collisions

Tout comme vu en classe, la détection de collision entre la balle de rayon  $r$  et le cube de dimensions  $a \times b \times c$  se fera selon l'une de ces trois types de collisions:

1. Surfaces : La sphère touchera ou pénétrera une face du parallélépipède si une droite partant de son centre, dans la direction normale à la surface, croise la surface plane à une distance  $d \leq r$  (intersection d'une droite et d'un plan).
2. Arêtes : Une des lignes de direction  $x$ ,  $y$  ou  $z$  et de longueur finie ( $-a/2 < x < a/2$ ,  $-b/2 < y < b/2$  et  $-c/2 < z < c/2$ ) qui correspond à une arête croise la sphère (intersection d'une droite et d'une sphère).
3. Coins : Un des coins se retrouve à l'intérieur de la sphère (intersection d'un point et d'une sphère).

## 2.3. Calcul des vitesses après collision

Lorsqu'une collision est détectée, nous calculons les nouvelles vitesses linéaires et angulaires des deux objets en appliquant les principes de conservation de la quantité de mouvement et en tenant compte du coefficient de restitution. Plus précisément, nous utiliserons la méthode des conditions initiales, qui s'appliquera bien ici car les collisions sont de courte durée.

Coefficient de Restitution ( $e$ ) : 0,8.

### Moments d'Inertie :

Avec  $I_3$  étant la matrice identité 3x3 :

$$I_{\text{bloc}} = \frac{1}{6} m_{\text{bloc}} a^2 I_3$$

$$I_{\text{balle}} = \frac{2}{5} m_{\text{balle}} R_{\text{balle}}^2 I_3$$

### Calcul de l'Impulsion :

L'équation de l'impulsion est la suivante:

$$\vec{J} = \frac{-(1+e)(\vec{v}_{\text{rel}} \cdot \vec{n})\vec{n}}{\frac{1}{m_{\text{balle}}} + \frac{1}{m_{\text{bloc}}} + \vec{n} \cdot \left( (I_{\text{balle}}^{-1}(\vec{r}_{\text{ballep}} \times \vec{n})) \times \vec{r}_{\text{ballep}} + (I_{\text{bloc}}^{-1}(\vec{r}_{\text{blocp}} \times \vec{n})) \times \vec{r}_{\text{blocp}} \right)}$$

où on a vitesse relative au point d'impact:

$$\vec{v}_{\text{rel}} = (\vec{v}_{\text{balle}} + \vec{\omega}_{\text{balle}} \times \vec{r}_{\text{ballep}}) - (\vec{v}_{\text{bloc}} + \vec{\omega}_{\text{bloc}} \times \vec{r}_{\text{blocp}})$$

Le vecteur  $\vec{n}$  étant le vecteur normal au point d'impact, dirigée de la balle vers le bloc, et les vecteurs  $\vec{r}_{\text{ballep}}$  et  $\vec{r}_{\text{blocp}}$  étant les positions des centres de masse des objets au point d'impact.

### Mise à jour des vitesses:

Finalement, les vitesses sont mises à jour:

$$\vec{v}'_{\text{bloc}} = \vec{v}_{\text{bloc}} - \frac{\vec{J}}{m_{\text{bloc}}}$$

$$\vec{v}'_{\text{balle}} = \vec{v}_{\text{balle}} + \frac{\vec{J}}{m_{\text{balle}}}$$

$$\vec{\omega}'_{\text{bloc}} = \vec{\omega}_{\text{bloc}} - \mathbf{I}_{\text{bloc}}^{-1}(\vec{r}_{\text{blocp}} \times \vec{J})$$

$$\vec{\omega}'_{\text{balle}} = \vec{\omega}_{\text{balle}} + \mathbf{I}_{\text{balle}}^{-1}(\vec{r}_{\text{ballep}} \times \vec{J})$$



### 3. Présentation et analyse des résultats

Dans cette deuxième partie, nous allons simuler 6 différents tirs, chacun avec des conditions initiales différentes. Ces conditions sont le temps à laquelle la balle est lancée, les vitesses linéaires et angulaires initiales du bloc ainsi que la vitesse linéaire initiale de la balle. Les sorties des simulations consisteront du résultat du tir, du temps final à la collision, des vitesses linéaires juste avant et juste après la collision et des vitesses angulaires juste avant et juste après la collision.

Tableau 2.1. - conditions initiales des 6 tirs

Tir	$t_l(s)$	$\vec{v}_{bloc}(0)(m/s)$	$\vec{\omega}_{bloc}(0)(rad/s)$	$\vec{v}_{balle}(0)(m/s)$
1	0.545454	(-2,-3,5)	(0,0,0)	(5,2,0.642424)
2	0.545454	(-2,-3,5)	(0,0,15)	(5,2,0.642424)
3	0.071429	(0,-6,3)	(0,0,0)	(7,0,0.40834)
4	0.071429	(0,-6,3)	(0,0,15)	(7,0,0.40834)
5	0.6	(-2,-3,5)	(-5,-5,0)	(5,2,0.642424)
6	0.1	(-2,-3,5)	(0,0,0)	(5,2,0.1)

### 3.1. Résultats du tir 1

Résultat : La balle a touché le bloc (0)

Temps final : 0.80960 s

Tableau 3.2. - résultats du tir 1

objet	$\vec{r}(t_f)(m/s)$	$\vec{v}(t_f^-)(m/s)$	$\vec{v}(t_f^+)(m/s)$	$\vec{\omega}(t_f^-)(rad/s)$	$\vec{\omega}(t_f^+)(rad/s)$
balle	$\begin{bmatrix} 1.3210 \\ 0.5284 \\ 1.8277 \end{bmatrix}$	$\begin{bmatrix} 5.0000 \\ 2.0000 \\ -1.9467 \end{bmatrix}$	$\begin{bmatrix} -7.3977 \\ 0.4187 \\ -1.9467 \end{bmatrix}$	$\vec{0}$	$\vec{0}$
bloc	$\begin{bmatrix} 1.3806 \\ 0.5709 \\ 1.8360 \end{bmatrix}$	$\begin{bmatrix} -2.0000 \\ -3.0000 \\ -2.9351 \end{bmatrix}$	$\begin{bmatrix} -2.0000 \\ -3.0000 \\ -2.9351 \end{bmatrix}$	$\vec{0}$	$\begin{bmatrix} 0.5121 \\ -4.0150 \\ 16.9005 \end{bmatrix}$

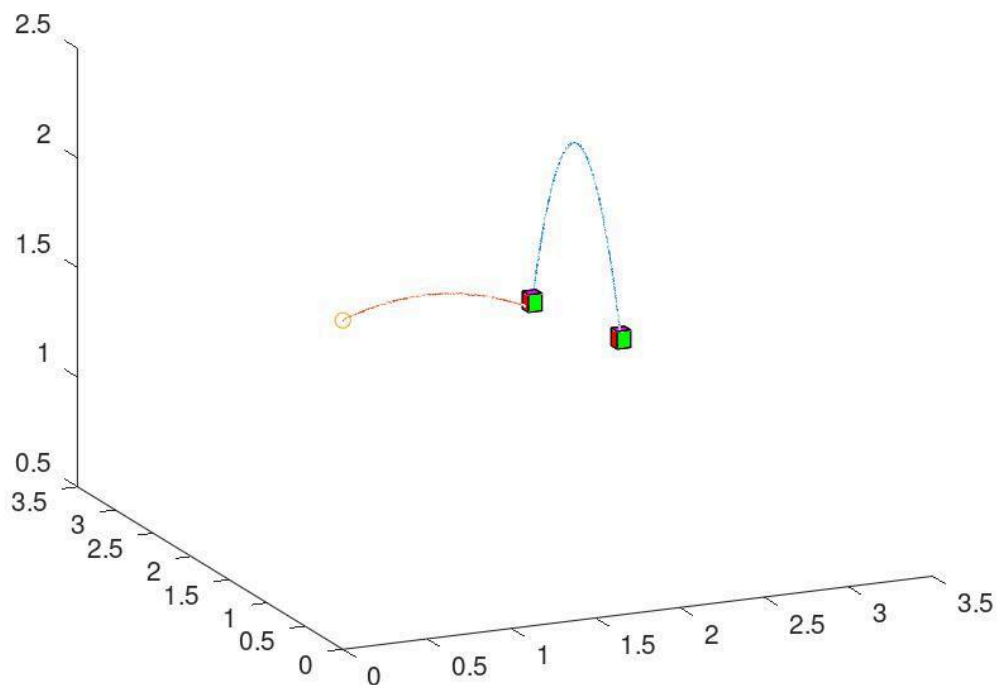


Figure 3.1. - trajectoires du tir 1

## 3.2. Résultats du tir 2

Résultat : La balle a touché le bloc (0)

Temps final : 0.81000 s

Tableau 3.3. - résultats du tir 2

objet	$\vec{r}(t_f)(m/s)$	$\vec{v}(t_f^-)(m/s)$	$\vec{v}(t_f^+)(m/s)$	$\vec{\omega}(t_f^-)(rad/s)$	$\vec{\omega}(t_f^+)(rad/s)$
balle	$\begin{bmatrix} 1.3230 \\ 0.5292 \\ 1.8269 \end{bmatrix}$	$\begin{bmatrix} 5.0000 \\ 2.0000 \\ -1.9507 \end{bmatrix}$	$\begin{bmatrix} -0.2617 \\ -9.9465 \\ -1.9507 \end{bmatrix}$	$\vec{0}$	$\vec{0}$
bloc	$\begin{bmatrix} 1.3798 \\ 0.5697 \\ 1.8348 \end{bmatrix}$	$\begin{bmatrix} -2.0000 \\ -3.0000 \\ -2.9390 \end{bmatrix}$	$\begin{bmatrix} -1.7808 \\ -2.5022 \\ -2.9390 \end{bmatrix}$	$\begin{bmatrix} 0.0000 \\ 0.0000 \\ 15.0000 \end{bmatrix}$	$\begin{bmatrix} 3.6844 \\ -1.6227 \\ -3.1823 \end{bmatrix}$

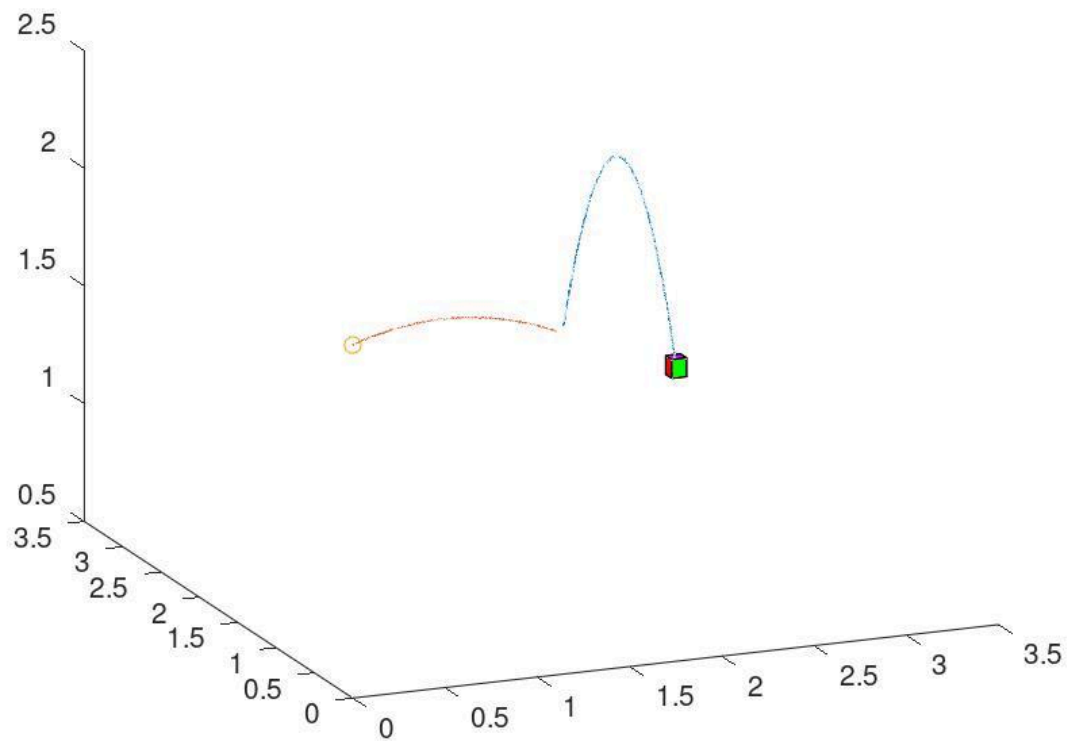


Figure 3.2. - trajectoires du tir 2

### 3.3. Résultats du tir 3

Résultat : La balle a touché le bloc (0)

Temps final : 0.49180 s

Tableau 3.4. - résultats du tir 3

objet	$\vec{r}(t_f)(m/s)$	$\vec{v}(t_f^-)(m/s)$	$\vec{v}(t_f^+)(m/s)$	$\vec{\omega}(t_f^-)(rad/s)$	$\vec{\omega}(t_f^+)(rad/s)$
balle	$\begin{bmatrix} 2.9428 \\ 0.0000 \\ 1.3057 \end{bmatrix}$	$\begin{bmatrix} 7.0000 \\ 0.0000 \\ -3.7116 \end{bmatrix}$	$\begin{bmatrix} -6.5381 \\ -6.7691 \\ -3.7116 \end{bmatrix}$	$\vec{0}$	$\vec{0}$
bloc	$\begin{bmatrix} 3.0000 \\ 0.0486 \\ 1.2901 \end{bmatrix}$	$\begin{bmatrix} 0.0000 \\ -6.0000 \\ -1.8206 \end{bmatrix}$	$\begin{bmatrix} 0.5641 \\ -5.7180 \\ -1.8206 \end{bmatrix}$	$\vec{0}$	$\begin{bmatrix} -4.1224 \\ 8.2448 \\ 10.5767 \end{bmatrix}$

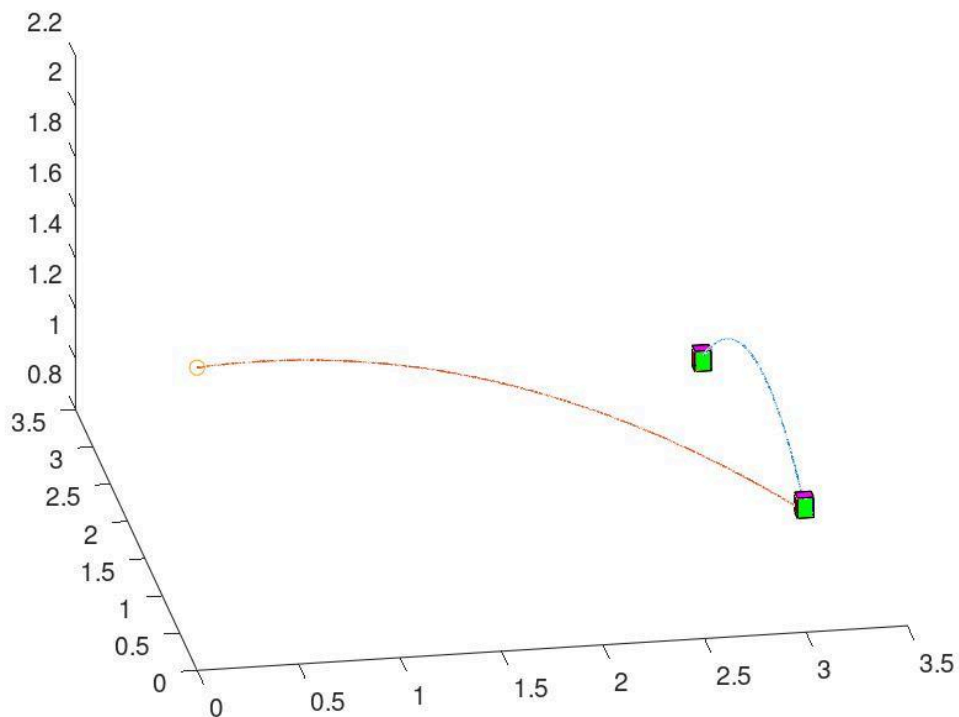


Figure 3.3. - trajectoires du tir 3

### 3.4. Résultats du tir 4

Résultat : La balle a touché le bloc (0)

Temps final 0.49290 s

Tableau 3.5. - résultats du tir 4

objet	$\vec{r}(t_f)(m/s)$	$\vec{v}(t_f^-)(m/s)$	$\vec{v}(t_f^+)(m/s)$	$\vec{\omega}(t_f^-)(rad/s)$	$\vec{\omega}(t_f^+)(rad/s)$
balle	$\begin{bmatrix} 2.9505 \\ 0.0000 \\ 1.3016 \end{bmatrix}$	$\begin{bmatrix} 7.0000 \\ 0.0000 \\ -3.7224 \end{bmatrix}$	$\begin{bmatrix} 0.4207 \\ -13.3134 \\ -3.7224 \end{bmatrix}$	$\vec{0}$	$\vec{0}$
bloc	$\begin{bmatrix} 3.0000 \\ 0.0420 \\ 1.2881 \end{bmatrix}$	$\begin{bmatrix} 0.0000 \\ -6.0000 \\ -1.8314 \end{bmatrix}$	$\begin{bmatrix} 0.2741 \\ -5.4453 \\ -1.8314 \end{bmatrix}$	$\begin{bmatrix} 0.0000 \\ 0.0000 \\ 15.0000 \end{bmatrix}$	$\begin{bmatrix} -7.0262 \\ 3.4722 \\ 0.0513 \end{bmatrix}$

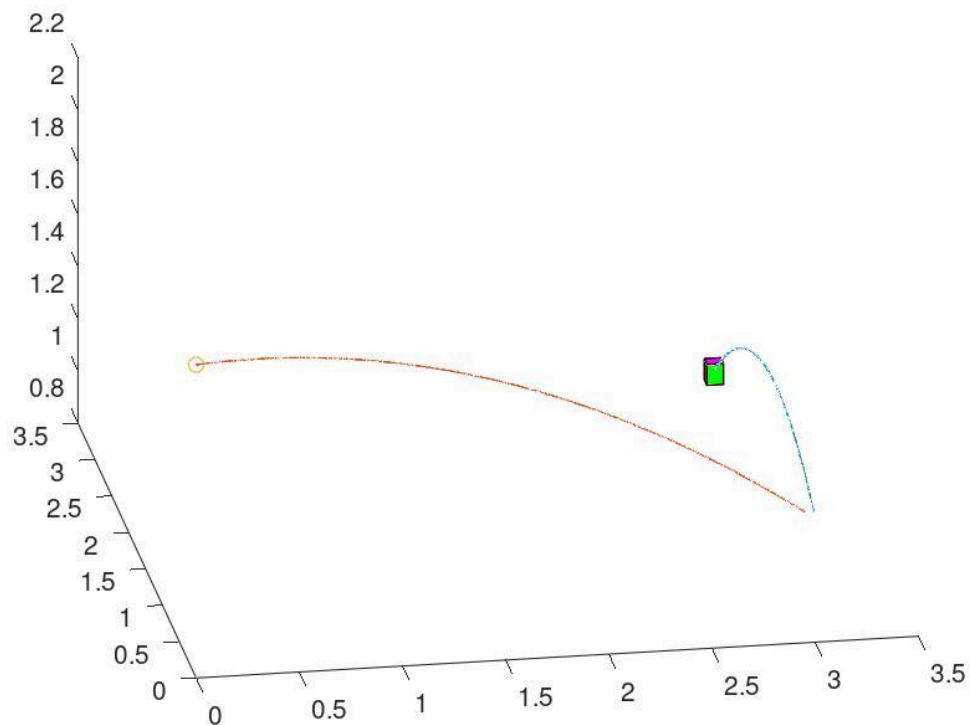


Figure 3.4. - trajectoires du tir 4

### 3.5. Résultats du tir 5

Résultat : Le bloc a touché le sol en premier (1)

Temps final : 1.19160 s

Tableau 3.6. - résultats du tir 5

objet	$\vec{r}(t_f)(m/s)$	$\vec{v}(t_f^-)(m/s)$	$\vec{v}(t_f^+)(m/s)$	$\vec{\omega}(t_f^-)(rad/s)$	$\vec{\omega}(t_f^+)(rad/s)$
balle	$\begin{bmatrix} 2.9580 \\ 1.1832 \\ 0.6651 \end{bmatrix}$	$\begin{bmatrix} 5.0000 \\ 2.0000 \\ -5.1553 \end{bmatrix}$	$\begin{bmatrix} 5.0000 \\ 2.0000 \\ -5.1553 \end{bmatrix}$	$\vec{0}$	$\vec{0}$
bloc	$\begin{bmatrix} 0.6166 \\ -0.5751 \\ -0.0002 \end{bmatrix}$	$\begin{bmatrix} -2.0000 \\ -3.0000 \\ -6.6787 \end{bmatrix}$	$\begin{bmatrix} -2.0000 \\ -3.0000 \\ -6.6787 \end{bmatrix}$	$\begin{bmatrix} -5.0000 \\ -5.0000 \\ 0.0000 \end{bmatrix}$	$\begin{bmatrix} -5.0000 \\ -5.0000 \\ 0.0000 \end{bmatrix}$

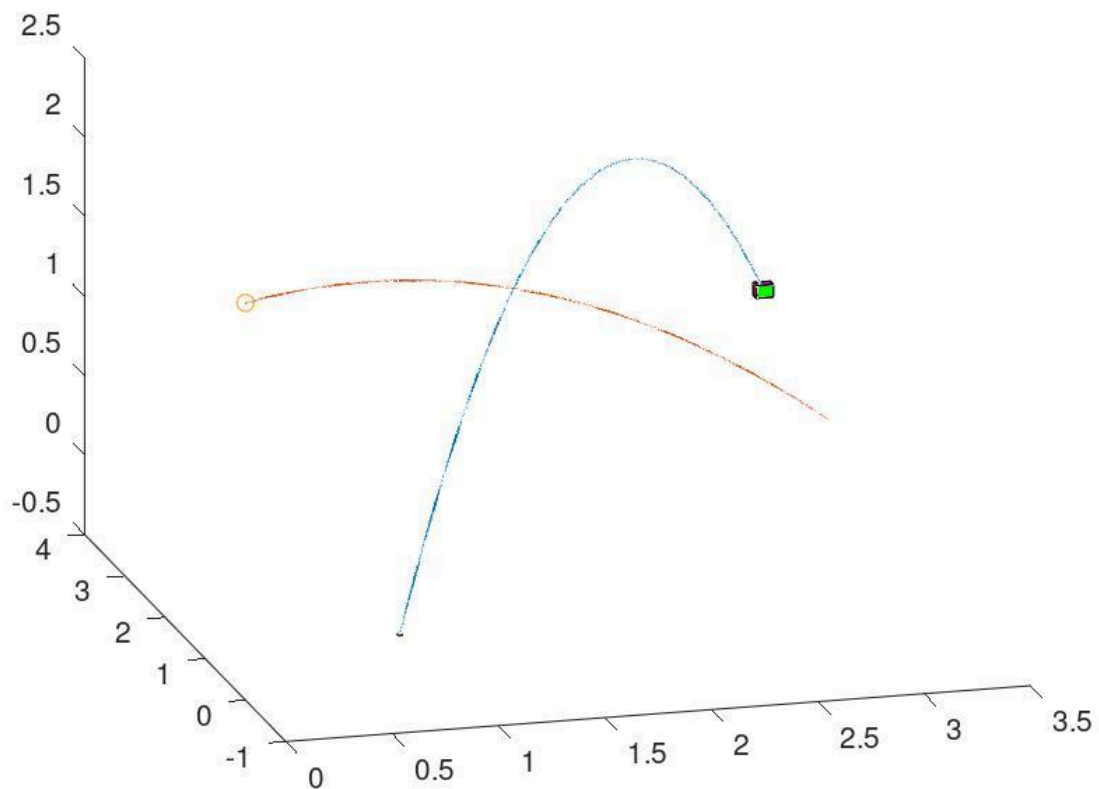


Figure 3.5. - trajectoires du tir 5

### 3.6. Résultats du tir 6

Résultat : La balle a touché le sol en premier (-1)

Temps final : 0.74910 s

Tableau 3.7. - résultats du tir 6

objet	$\vec{r}(t_f)(m/s)$	$\vec{v}(t_f^-)(m/s)$	$\vec{v}(t_f^+)(m/s)$	$\vec{\omega}(t_f^-)(rad/s)$	$\vec{\omega}(t_f^+)(rad/s)$
balle	$\begin{bmatrix} 3.2460 \\ 1.2984 \\ -0.0002 \end{bmatrix}$	$\begin{bmatrix} 5.0000 \\ 2.0000 \\ -6.2622 \end{bmatrix}$	$\begin{bmatrix} 5.0000 \\ 2.0000 \\ -6.2622 \end{bmatrix}$	$\vec{0}$	$\vec{0}$
bloc	$\begin{bmatrix} 1.5016 \\ 0.7524 \\ 1.9956 \end{bmatrix}$	$\begin{bmatrix} -2.0000 \\ -3.0000 \\ -2.3422 \end{bmatrix}$	$\begin{bmatrix} -2.0000 \\ -3.0000 \\ -2.3422 \end{bmatrix}$	$\vec{0}$	$\vec{0}$

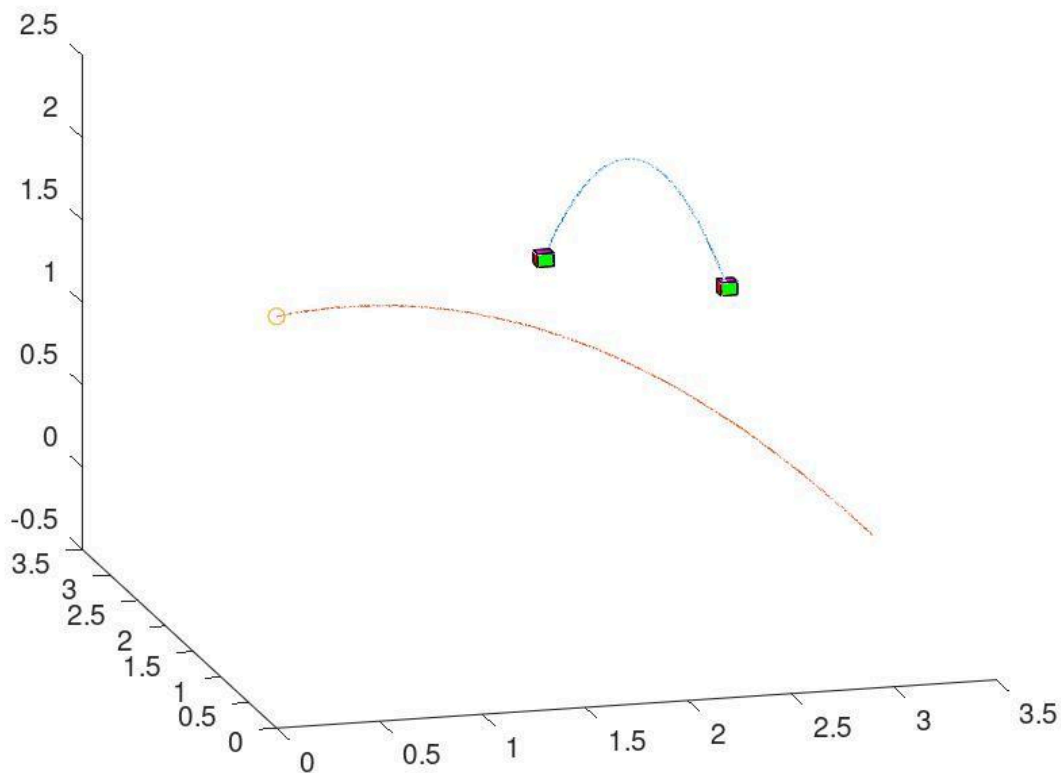


Figure 3.6. - trajectoires du tir 6

### 3.7. Analyse des résultats

Comme on peut observer aux pages précédentes, dans les 4 premiers tirs, la balle va toucher le bloc alors que dans les deux derniers, la balle touche le sol en premier sans percuter par le bloc.

Ce que l'on constate en analysant les simulations est que les deux premiers tirs ont des conditions initiales semblables à quelques différences près, le troisième et le quatrième ont eux aussi des conditions initiales similaires ainsi que le cinquième et le sixième.

Maintenant, si l'on étudie les deux premiers tirs comparativement, on constate que les différences entre le tir 1 et le tir 2 de la vitesse de la balle juste avant la collision sont mineurs tandis que la vitesse angulaire de la bloc avant la collision est nulle dans le premier tir et égale à 15 m/s dans une direction verticale vers le haut dans le deuxième. Suite à la collision, ce qui en résulte est une vitesse finale de la balle différente entre le tir 1 et le tir 2, la direction de la balle pour chacun des tirs est différente ainsi que leur vitesse scalaire. On constate par ailleurs que la vitesse linéaire  $\vec{v}(t_f^+)(m/s)$  pour le tir 2 est plus grande, ceci est dû au fait qu'au tir 2 le bloc a un mouvement de rotation ce qui a pour effet de percuter la balle avec une plus grande force.

Quant à la vitesse angulaire après la collision, alors que la vitesse angulaire de la balle était plus grande dans le cas du tir 2, on constate que la vitesse angulaire finale de la balle est considérablement plus élevée dans le cas du tir 1.



	Tir 1		Tir 2	
	Balle	Bloc	Balle	Bloc
$\vec{r}(t_f)(m/s)$	$\begin{bmatrix} 1.3210 \\ 0.5284 \\ 1.8277 \end{bmatrix}$	$\begin{bmatrix} 1.3806 \\ 0.5709 \\ 1.8360 \end{bmatrix}$	$\begin{bmatrix} 1.3230 \\ 0.5292 \\ 1.8269 \end{bmatrix}$	$\begin{bmatrix} 1.3798 \\ 0.5697 \\ 1.8348 \end{bmatrix}$
$\vec{v}(t_f^-)(m/s)$	$\begin{bmatrix} 5.0000 \\ 2.0000 \\ -1.9467 \end{bmatrix}$	$\begin{bmatrix} -2.0000 \\ -3.0000 \\ -2.9351 \end{bmatrix}$	$\begin{bmatrix} 5.0000 \\ 2.0000 \\ -1.9507 \end{bmatrix}$	$\begin{bmatrix} -2.0000 \\ -3.0000 \\ -2.9390 \end{bmatrix}$
$\vec{v}(t_f^+)(m/s)$	$\begin{bmatrix} -7.3977 \\ 0.4187 \\ -1.9467 \end{bmatrix}$	$\begin{bmatrix} -2.0000 \\ -3.0000 \\ -2.9351 \end{bmatrix}$	$\begin{bmatrix} -0.2617 \\ -9.9465 \\ -1.9507 \end{bmatrix}$	$\begin{bmatrix} -1.7808 \\ -2.5022 \\ -2.9390 \end{bmatrix}$
$\vec{\omega}(t_f^-)(rad/s)$	$\vec{0}$	$\vec{0}$	$\vec{0}$	$\begin{bmatrix} 0.0000 \\ 0.0000 \\ 15.0000 \end{bmatrix}$
$\vec{\omega}(t_f^+)(rad/s)$	$\vec{0}$	$\begin{bmatrix} 0.5121 \\ -4.0150 \\ 16.9005 \end{bmatrix}$	$\vec{0}$	$\begin{bmatrix} 3.6844 \\ -1.6227 \\ -3.1823 \end{bmatrix}$

Quant aux tirs 3 et 4, leur comparaison est semblable à celle du tir 1 et 2 étant donné qu'on retrouve les mêmes similitudes et les mêmes changements après la collision car les valeurs sont différentes mais les paramètres de différenciation restent les mêmes. Par exemple, la vitesse linéaire après la collision pour le tir 4 est considérablement plus grande que celle du tir 3, alors que les balles dans les deux cas on commence avec des vitesses initiales similaires au niveau de la norme et de l'orientation du vecteur. Ceci est dû au fait qu'au tir 2 le bloc a un mouvement de rotation ce qui a pour effet de percuter la balle avec une plus grande force.

Finalement, pour ce qui a trait au tir 5 et 6, les deux tirs de balle qui n'ont pas atteint le bloc. Dans le cas du tir 5, le bloc touche le sol en premier tandis que la balle atteint le sol en premier dans le tir 6. Dans les deux cas, le bloc a des conditions initiales similaires, seul la vitesse angulaire diffère. Par contre, la balle est lancée beaucoup plus tôt durant le tir 6 et avec une vitesse verticale (z) beaucoup moins élevée. Cela explique pourquoi la balle atteint le sol plus tôt que dans le tir 5. La rotation du bloc n'affecte pas de manière significative le résultat, c'est donc les conditions initiales liées à la balle qui causent la différence entre les deux tirs.

## Conclusion

Pour conclure, ce laboratoire a demandé la mise en application des notions de la dynamique des collisions, des méthodes de forces et la dynamique des solides. De plus, on a été porté à appliquer la détection des collisions sur des solides de formes régulières telle qu'une sphère et un cube.

Ces notions nous ont permis de simuler ce qui arriverait dans le cas d'une collision entre le ballon et le bloc. Après chaque collision, on doit recalculer la nouvelle vitesse linéaire et angulaire ainsi que son orientation pour chacun des objets (la balle et le bloc). Ceci a été possible grâce à la méthode des conditions initiales.

Nous avons rencontré quelques problèmes lors de la réalisation de ce laboratoire. En premier lieu, transformer les concepts physiques et mathématiques en Matlab a été une tâche complexe. Par exemple, l'algorithme de détection de collision a nécessité plusieurs essais pour trouver la bonne implémentation.

Deux principaux aspects de notre programme pourraient être améliorés. D'abord, notre méthode de collision avec le sol pourrait être améliorée. Pour la sphère, vérifier que son rayon soustrait à la composante  $z$  de son centre de masse ne soit pas inférieur ou égal à zéro est suffisante. Mais pour le cube, au lieu d'utiliser la même approche, il serait préférable de vérifier le contact avec le sol en tenant compte de son orientation. Bien que l'emphase soit mise sur les collisions sphères-blocs, ce détail nous donnerait une précision de quelques fractions de secondes.

Deuxièmement, notre programme vérifie les collisions sphères-blocs à chaque pas de temps en utilisant la démarche mentionnée plus haut. Cette démarche étant coûteuse en temps de calcul, on aurait pu traiter le bloc comme une sphère et vérifier s'il y a même une possibilité de collision à chaque pas de temps avant de commencer à utiliser la démarche et à calculer le point d'interception. Cela optimiserait notre temps de calcul de manière considérable.