PHS4700

Physique pour les applications multimédia

Automne 2015

PAGE COUVERTURE **OBLIGATOIRE** POUR TOUS LES DEVOIRS

Numéro de devoir : 03

Numéro de l'équipe : 14

Nom: Rose	Prénom : Alexandre	matricule: 1580973
Signature :		
Nom: Mainville	Prénom : David	matricule: 1636075
Signature :		
Nom: Gosselin	Prénom : Antoine	matricule: 1588443
Signature :		
Nom: Farvacque	Prénom : Dylan	matricule: 1684271
Signature :		

POLYTECHNIQUE MONTRÉAL
LE GÉNIE EN PREMIÈRE CLASSE

Table des matières

I – Description du problème

Pour ce présent devoir, notre tâche consiste à simuler le lancer d'une balle de tennis qui a pour cible une boîte de conserve remplie d'air en chute libre. Notre simulation devra être capable de fournir les composantes en X, Y et Z des vitesses et centres de masse selon le temps des différents objets. Enfin, en cas de collision entre la balle et la boîte, notre simulation devra indiquer la vitesse du centre de masse ainsi que la vitesse angulaire après la collision entre la boîte et la balle.

Nous nous intéresserons à quatre scenarios distincts :

- La boîte de conserve n'a pas de vitesse angulaire et le lancer de la balle ne possède que des vitesses selon l'axe des x et des z,
- Toujours avec une boîte conserve sans vitesse angulaire, le lancer de la balle aura aussi une vitesse en y
- Dans le troisième scenario, la balle n'aura qu'une vitesse en x et en y et la boîte de conserve aura une vitesse angulaire autour de l'axe des y
- Enfin, dans le dernier scenario, la boîte aura toujours une vitesse angulaire selon l'axe des y et le lancer de la balle aura une vitesse non nulle selon tous les axes.

Dans le présent rapport, un bref rappel des équations nécessaires à la simulation sera fait. Ensuite, notre simulation développée sur MATLAB sera présentée. Aussi, nous présenterons nos résultats et en ferons une analyse détaillée. Enfin, nous conclurons par une discussion sur les problèmes que nous avons dû surmonter au cours du devoir en ce qui a trait à la programmation et aux simulations.

II – Équations importantes

Équations du mouvement à résoudre

La force gravitationnelle sur la balle est donnée par une seule multiplication de constante sur l'axe de la hauteur (dans la situation présente l'axe des Z).

$$\vec{F}_g = m_b(0, 0, -9.8)^T$$

Équation 1

Nous utilisons comme équation pour le frottement visqueux la formule donnée dans l'énoncé.

$$\vec{F}_v = -kA\vec{v}$$

Équation 2

Les valeurs de k sont aussi données dans l'énoncé du travail

- pour la balle: $k = 0.1$ kg/(m^{2}s) et $A = \pi R^2_{\text{balle}}$;
- pour la boîte: $k = 0.1$ kg/(m^{2}s) et $A = R^2_{\text{boîte}} + h^2_{\text{boîte}}$;

Équations qui contrôlent la simulation et de détection de collision

Nous avons deux conditions d'arrêt :

- S'il y a une collision entre la balle et la boîte,
- Si la balle touche le sol.

Notre première équation de contrôle permet de déterminer si la balle a touché le sol, si c'est le cas, la simulation ce termine nous avons donc la condition suivante :

$$CDM(z)_{balle} - R_{balle} \leq 0$$

Pour déterminer s'il y a une collision entre la balle et la boîte, le processus de détection est un peu plus complexe. Comme décrit dans la section description du logiciel, une première vérification est faite en faisant une sphère englobante de rayon *r* tel que :

$$r = \sqrt{R_{boite}^2 + \ (h_{Boite}/2)^2}$$

Par la suite, si la distance entre les centres de masse de la boîte et de la balle est inférieure ou égale à la somme de r et de R_{balle}, nous nous trouvons dans une situation où il y a peut-être une collision. À ce moment, on vérifie plus précisément s'il y a une collision.

On commence par faire une rotation de notre système d axe afin que l'axe des Z passe par le vecteur centre du cylindre. Par la suite on fait une translation de l'origine de notre system d'axe au centre de masse de la boîte de conserve. Enfin, on recalcule les nouvelles coordonnées du centre de masse de la balle selon notre nouveau système d'axes.

Ensuite, nous avons plusieurs possibilités de collision :

- Collision sur le contour du cylindre
- Collision sur la surface inférieure ou supérieure
- Collision avec le côté du cylindre

Nous déterminons dans un premier temps s'il y a une collision avec le côté du cylindre. Pour ce faire, nous vérifions si la hauteur se situe entre le haut et le bas du cylindre. Si tel est le cas, nous avons une collision.

Sinon, on doit vérifier si le centre de la balle se trouve au-dessus ou sous le cylindre. Pour ce faire, nous calculons la distance *d* entre le centre de masse de la balle et l'axe des z :

$$d = \sqrt{CDM(x)^2_{balle} + CDM(y)^2_{balle}}$$

Si celle-ci est inférieure ou égale à la somme du rayon de la boîte et de la balle et que sa coordonnée en z se trouve entre le haut et le bas du cylindre, nous avons donc les conditions suivantes où *h* est la hauteur du cylindre :

$$d \leq R_{balle} + R_{cylindre}$$
$$CDM(z) \leq {}^{-h}\!/_{2} \text{ ou } {}^{h}\!/_{2} \leq CDM(z)$$

Si ces conditions sont satisfaites, nous avons probablement une collision entre la balle et le dessus ou le dessous de la boîte. Nous n'avons alors qu'à déterminer la distance entre le centre de la balle et les extrémités du cylindre. Si celle-ci est inférieure au rayon de la sphère, nous avons une collision. Nous déterminons alors les vitesses résultantes de la collision et notre simulation se termine. Toutefois, si ce n'est pas le cas, nous vérifions s'il y a une collision entre la balle et le rebord du cylindre. Nous commençons par trouver le point le plus proche de la sphère sur le contour du cylindre. De ce point nous calculons la distance avec le centre de la sphère et nous vérifions que la distance est inférieure au rayon de la sphère. Si tel est le cas, nous avons une collision. Finalement, nous calculons les vitesses résultantes des objets et la simulation s'arrête. Si aucune de ces conditions n'est respectée, alors la simulation continue.

Vitesses finales des objets

Les vitesses après la collision pour les deux objets impliqués sont données par les équations suivantes. Pour la calculer, on a besoin de la normale au point de contact, du moment d'inertie et d'informations données dans des équations subséquentes.

$$\vec{v}_{a,p}(t_f) = \vec{v}_{a,p}(t_i) + j\left(\frac{\vec{n}}{m_a} + \mathbf{I}_a^{-1}(\vec{r}_{a,p} \times \vec{n}) \times \vec{r}_{a,p}\right)$$
$$\vec{v}_{b,p}(t_f) = \vec{v}_{b,p}(t_i) - j\left(\frac{\vec{n}}{m_b} + \mathbf{I}_b^{-1}(\vec{r}_{b,p} \times \vec{n}) \times \vec{r}_{b,p}\right)$$

Équation 3

Les vitesses angulaires, quant à elles, sont données par les équations suivantes. Elles impliquent les vitesses angulaires avant la collision, du moment d'inertie et encore une fois de variables définies plus bas.

$$\vec{\omega}_a(t_f) = \vec{\omega}_a(t_i) + j\mathbf{I}_a^{-1}(\vec{r}_{a,p} \times \vec{n})$$
$$\vec{\omega}_b(t_f) = \vec{\omega}_b(t_i) - j\mathbf{I}_b^{-1}(\vec{r}_{b,p} \times \vec{n})$$

Équation 4

La variable j désigne la composante normale à l'impulsion. Ce terme est important puisque la vitesse ne sera qu'affecter seulement dans le sens de l'impulsion. Epsilon représente le coefficient de restitution.

$$j = -\alpha(1+\epsilon)v_{-}^{r}$$

Équation 5

Pour calculer *j*, on doit d'abord obtenir le facteur *α* donné par l'équation suivante. Elle implique les masses des objets ainsi que les facteurs G_a et G_b donnés plus bas.

$$\alpha = \frac{1}{\frac{1}{m_a} + \frac{1}{m_b} + G_a + G_b}$$

Équation 6

Les facteurs G_a et G_b entrant dans le calcul du *α* sont donnés par ces équations. Elles mettent en relation le moment d'inertie, la normale au plan de collision et le vecteur entre le centre de masse et le point de collision.

$$G_a = \vec{n} \cdot \left[\mathbf{I}_a^{-1} (\vec{r}_{a,p} \times \vec{n}) \times \vec{r}_{a,p} \right]$$
$$G_b = \vec{n} \cdot \left[\mathbf{I}_b^{-1} (\vec{r}_{b,p} \times \vec{n}) \times \vec{r}_{b,p} \right]$$

Équation 7

Enfin, le dernier élément entrant dans le calcul du *j* est la vitesse relative du système juste avant la collision.

$$v_r^- = \vec{n} \cdot (\vec{v}_{a,p}(t_i) - \vec{v}_{b,p}(t_i))$$

Équation 8

III – Méthode de résolution des équations du mouvement

Les équations de mouvement de la balle et du cylindre sont approximées numériquement à l'aide de Runge Kutta d'ordre 4. Nous fournissions à Runge Kutta les paramètres de pas et de vitesse. Avec cette méthode d'approximation numérique il est possible pour chaque itération de calculer la somme des forces qui est égale à la masse multiplié l'accélération, puis à l'aide des vitesses initiales de trouver les vitesses finales et les positions finales.

Pour calculer les forces de frottement visqueux, nous avons utilisé la formule donnée dans l'énoncé avec la constante *k* donnée dont seule la vitesse varie à chaque itération.

Pour la rotation de la boîte, nous incrémentons une variable contenant son vecteur de rotation d'une valeur égale à la vitesse angulaire divisé par le pas à chaque itération. Donc, lorsque nous avons besoin d'effectuer des calculs sur la boîte qui a subi une rotation, nous pouvoir calculer la matrice de rotation globale à l'aide des trois composantes de rotation en x, y et z. C'est la fonction *Rotation* dans notre programme qui est responsable d'obtenir les trois composantes de la matrice de rotation et de les multiplier.

Justification du pas

Le bon pas a été déterminé expérimentalement. L'idée étant de trouver un pas suffisamment petit pour que la précision minimale de 1mm soit respectée mais suffisamment grand pour que le temps d'exécution ne soit pas trop important. Avec un pas de 0,00005 nous obtenons une erreur maximale de 0,4734 mm dans toutes les simulations. On peut donc dire que notre précision est de 0,5 mm, ce qui est amplement suffisant pour respecter les contraintes du laboratoire.

Description des vérifications effectuées pour assurer la précision

Pour valider la précision de notre résultat lorsque nous obtenons une collision, nous calculons la profondeur d'enfoncement de la balle. Avec la méthode que nous utilisons pour calculer le point de collision, ses coordonnées se trouvent obligatoirement à la surface de la boîte. Ainsi, nous pouvons calculer la distance entre le centre de masse de la balle et le point de collision. La différence de cette distance avec le rayon de la balle correspond donc à l'enfoncement de la balle dans la boîte ou encore l'erreur maximale sur la position réelle du point de collision. Nous affichons le résultat en millimètres c'est pourquoi la formule ci-dessous multiplie le résultat par 1000.

```
(rayon_balle - norm(pointCollision - balle.CentreDeMasse(1:3,end)))*1000
```

Équation du calcul de précision
(en mm)

IV – Description du logiciel

Les simulations sont entièrement réalisées à l'aide du logiciel MATLAB. Tout d'abord on initialise nos objets, soit la balle et le cylindre. On définit chacune des propriétés disponibles dans l'énoncé pour ces deux objets et l'on calcule leur centre de masse pour les futures équations.

On définit ensuite un pas suffisamment petit afin d'obtenir la précision minimale requise. On définit également une rotation par pas en fonction de la vitesse angulaire initiale et du pas.
Une fois les diverses initialisations effectuées, on commence le début des simulations qui est une boucle while qui s'arrête lorsque la balle touche le sol (balle.CentreDeMasse(3,end) - balle.Rayon <= 0) ou qu'une collision avec le cylindre est confirmée. La boucle de simulation commence par incrémenter le temps de la simulation d'un pas. Ensuite on vérifie si le temps de lancer la balle est atteint ; Si c'est le cas on déplace la balle. On utilise RK4 afin de mettre à jour la position et la vitesse de la balle en fonction du temps courant. On effectue les mêmes opérations pour le déplacement du cylindre, sans toutefois avoir besoin de valider qu'un certain temps est atteint puisque le cylindre est en mouvement à partir du temps 0. Cependant en plus d'être déplacé, on effectue une rotation sur le cylindre à l'aide des matrices de rotations vu au TP1.

Suite au déplacement des deux objets, on effectue un test de collision. Afin de ne pas effectuer de validation complexe lorsque ce n'est pas nécessaire le test vérifie d'abord s'il y a collision entre la balle et une sphère englobant le cylindre. Si ce n'est pas le cas le test retourne 0, sinon il poursuit avec une validation plus sophistiqué. La validation supplémentaire effectue des projections dans le plan. Premièrement, on change le référentiel pour que la boîte soit droite et son centre soit à l'origine. Ensuite on projette dans le plan « xy ». Si la distance entre le centre de la balle et le centre du cylindre est inférieure à la somme des deux rayons on poursuit avec une validation en « xz ». Si le centre de la balle est à la même hauteur que le cylindre (c'est-à-dire entre les deux extrémités de celui-ci) alors il y a collision. Sinon on vérifie si les positions en x et y de la balle sont à l'intérieur du cylindre. Si c'est le cas on vérifie que le z de la balle est moindre que la hauteur du cylindre divisé par deux. Si le dernier cas n'est pas concluant il reste à valider si la collision a lieu avec le rebord du cylindre. En un premier temps il faut trouver le point le plus près de la balle sur le rebord du cylindre. Pour y arriver, on fait la projection du vecteur allant de l'origine au centre de la balle, puis on le normalise. Ceci nous donne un vecteur unitaire ayant la direction du point le plus proche de la sphère. Il suffit de multiplier par le rayon du cylindre pour obtenir les coordonnées. Ensuite, on calcule la distance entre le centre de la balle et le point le plus près du cylindre. Si cette distance est inférieure au rayon de la balle alors il y a collision.

Suite au test de collision, on réagit en mettant fin à la boucle si le résultat est positif. De plus, dans un tel cas, on effectue le calcul des vitesses et vitesses angulaires de la balle et de la boîte suite à cette collision. Pour se faire, on doit calculer le vecteur normal unitaire à la surface de collision au point de collision trouvé. On doit également calculer les moments d'inertie des objets à l'origine et effectuer une translation pour obtenir leur moment d'inertie par rapport à leur centre de masse. Ensuite, on doit calculer le vecteur entre les centres de masse des objets et le point de collision. Avec ces informations calculées, on peut trouver les vitesses et vitesses angulaires au moment suivant la collision avec les équations présentées à la section II.

Enfin, la dernière étape de la boucle de simulation est la validation que la balle n'a pas atteint le sol. Si c'est le cas on met fin à la simulation.

V – Résultats obtenus

Les résultats obtenus pour les quatre simulations sont donnés par les tableaux suivants. Ils indiquent le résultat de la simulation, à savoir si une collision est survenue ou non, le temps auquel s'est produit cette collision ou le contact avec le sol. De plus, la position, la vitesse et la vitesse angulaire avant la collision ainsi que la vitesse et la vitesse angulaire après la collision sont spécifiés pour la balle et la boîte.

La situation 1 avait comme conditions initiales, une vitesse de balle de (6.85, 0.0, 6.85) (m/s) lancée au temps 0,66 s. La boîte avant une vitesse angulaire initiale de (0, 0, 0) rad/s.

Situation 1					
Résultat	Collision bas de la boite				
Temps	1.0978				
	Balle				
	Position	**Vitesse initiale**	**w initial**	**Vitesse finale**	**w final**
x	3.0000	6.8095	0	6.8095	0
y	0	0	0	0	0
z	4.1596	2.5274	0	-9.0945	0
	Boite				
	Position	**Vitesse initiale**	**w initial**	**Vitesse finale**	**w final**
x	2.9901	0	0	0.6899	0
y	0	0	0	0	-0.0034
z	4.0516	-10.5748	0	-2.5434	0

La situation 2 avait comme conditions initiales, une vitesse de balle de (28, 0.5, 10) (m/s) lancée au temps 1,1 s. La boîte avant une vitesse angulaire initiale de (0, 0, 0) rad/s.

Situation 2					
Résultat	Collision côté de la boite				
Temps	1.2049				
	Balle				
	Position	**Vitesse initiale**	**w initial**	**Vitesse finale**	**w final**
x	3.0000	27.9603	0	14.9410	0
y	0	0.4993	0	11.0156	0
z	2.9734	8.9575	0	8.9575	0
	Boite				
	Position	**Vitesse initiale**	**w initial**	**Vitesse finale**	**w final**
x	2.9351	0	0	13.2522	-30.4162
y	0.0524	0	0	-8.1328	-0.0118
z	2.9943	-11.5854	0	-8.4290	-30.1380

La situation 3 avait comme conditions initiales, une vitesse de balle de (6.85, 0.0, 6.85) (m/s) lancée au temps 0,66 s. La boîte avant une vitesse angulaire initiale de (0, 2.3, 0) rad/s.

Situation 3					
Résultat	Collision contour				
Temps	1.0969				
	Balle				
	Position	**Vitesse initiale**	**w initial**	**Vitesse finale**	**w final**
x	3.0000	6.8096	0	2.7969	0
y	0	0	0	0	1.0e-18 * 0.6720
z	4.1696	2.5368	0	-6.4039	0
	Boite				
	Position	**Vitesse initiale**	**w initial**	**Vitesse finale**	**w final**
x	2.9836	0	0	5.9263	0
y	0	0	2.3000	0	2.3099
z	4.0492	-10.5658	0	0.2710	0

La situation 4 avait comme conditions initiales, une vitesse de balle de (28, 0.5, 10) (m/s) lancée au temps 1,1 s. La boîte avant une vitesse angulaire initiale de (0, 2.3, 0) rad/s.

Situation 4					
Résultat	Collision côté de la boite				
Temps	1.2051				
	Balle				
	Position	**Vitesse initiale**	**w initial**	**Vitesse finale**	**w final**
x	3.0000	27.9602	0	19.3689	0
y	0	0.4993	0	8.0231	0
z	2.9711	8.9555	0	12.2875	0
	Boite				
	Position	**Vitesse initiale**	**w initial**	**Vitesse finale**	**w final**
x	2.9407	0	0	6.8422	-1.1071
y	0.0525	0	2.3000	-5.8184	2.2993
z	2.9960	-11.5873	0	-13.9676	-1.0957

VI – Analyse des résultats obtenus

Dans un premier temps, nous pouvons observer que pour les quatre simulations nous avons eu une collision entre la balle et la boîte. Lors de la première simulation, la balle est entrée en collision avec le bas de la boîte, ce qui a eu pour effet de changer la grandeur et le sens de la vitesse en z de la balle. Après la collision, la boîte s'est retrouvée avec une rotation sur l'axe des y causé par l'impact de la balle. De plus, sa chute a été ralentie et une composante en x lui a été ajoutée.

Lors de la deuxième simulation une collision est survenue avec le côté de la boîte. Les résultats sont semblables à la première situation, mais avec des valeurs plus grandes puisque le lancer était beaucoup plus fort. Un changement à noter est que sous la force de l'impact, la boîte a commencé à monter (vitesse positive en z). De plus, la balle s'est retrouvée avec une vitesse angulaire suite à la collision.

Lors de la troisième simulation une collision est survenue avec le contour de la boîte. Lors de cette simulation la boîte avait une vitesse angulaire initiale non nulle. Suite à la collision, la balle a perdu de la vitesse et a changé de trajectoire en z pour se diriger vers le sol. La boîte quant à elle s'est vu remonter légèrement et a accéléré drastiquement en direction des x positifs. Selon nos résultats sa vitesse angulaire n'aurait pas changé, ce qui est très improbable vu le moment de force exercé par la force d'impulsion sur le rebord de la canne. Nous croyons donc qu'il s'agit d'une erreur.

Finalement pour la dernière simulation, la collision a lieu avec le côté de la boîte. Suite à la collision, la balle a changé de cap avec une vitesse en y grandissante. Toutefois le résultat montre une vitesse finale en z plus grande que la vitesse initiale, ce qui ne fait aucun sens. La boîte pour sa part a été accélérée vers les x positifs et les y négatifs, soit en direction opposée à celle de la balle, ce qui est attendu. Toutefois, la chute a été accélérée, ce qui est très improbable et probablement attribuable à la même erreur que celle de la vitesse en z de la balle. Finalement, on note une vitesse angulaire non nulle selon les trois axes pour la boîte après la collision.

Graphiques de la trajectoire

En rouge, la trajectoire de la boîte en chute libre. En bleu, la trajectoire de la balle lancée.

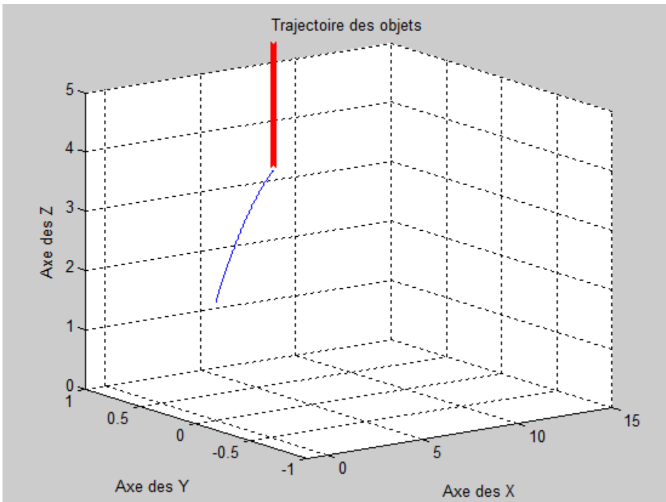

Figure 1: Simulation 1

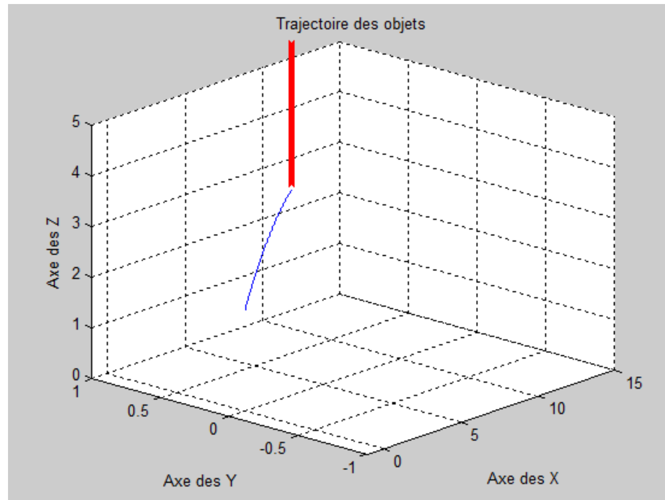

Figure 2: Simulation 2

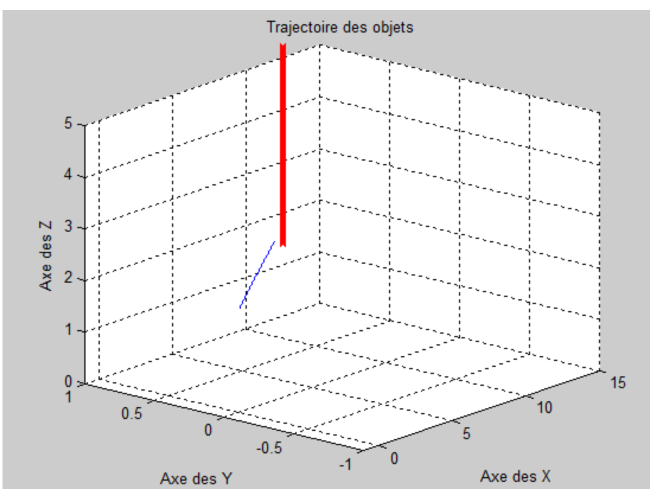

Figure 4: Simulation 3

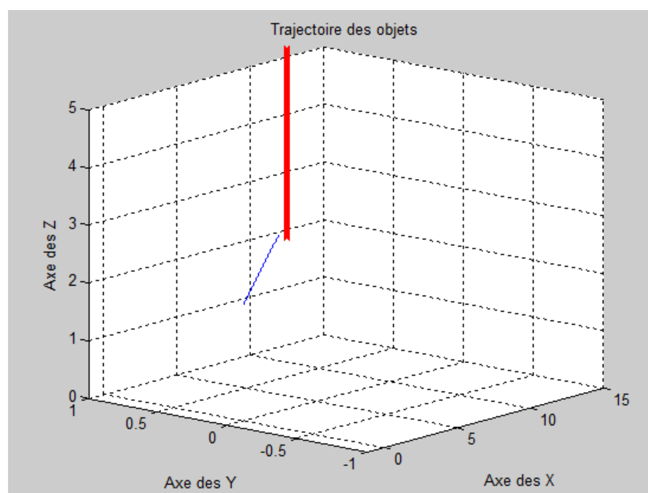

Figure 3: simulation 4

VII - Discussions sur le devoir

Comme lors du second laboratoire, nous avons tenté de mettre de l'ordre dans notre code en créant des objets contenant les propriétés physiques de la balle et de la boîte.

Nous avons rencontré des difficultés en ce qui a trait à la vérification de collision. Notre model initial détectait les collisions en vérifiant si plusieurs petites balles contenues dans le cylindre entraient en collision avec la balle lancée. Cette méthode était facile à implémenter puisque la détection de collision entre deux sphères est relativement simple. Cette modélisation du cylindre comme étant plusieurs sphères combinées n'était pas assez précise et nous avons dû laisser tomber cette méthode. Nous avons néanmoins conservé la pré-validation de collision avec une sphère englobant le cylindre, ce qui allège les calculs faits à chaque itération.