PHS4700

Physique pour les applications multimédia

Automne 2015

Numéro de devoir : 3

Numéro de l’équipe : 18

|  |
| --- |
| Nom: Gagné Prénom : Alex matricule: 1689761  Signature : |
| Nom: La Rocque Carrier Prénom : Félix matricule:1621348  Signature : |
| Nom: Gamache Prénom : Mathieu matricule: 1626377  Signature : |
| Nom: Fedorov Prénom : Konstantin matricule: 1679095  Signature : |

# Description du problème à résoudre

Ce devoir est une continuité direct du devoir #2. Le devoir #2 consistait à simuler un lancer d’une balle, cette simulation devait prendre en compte plusieurs forces qui était appliqué sur la basse et nécessitait l’utilisation d’équation numérique. Le défi qu'amènes le devoir #3 est que non seulement il faut encore une fois effectué une simulation d’objet soumis à divers forces il faut maintenant gérer la collision. Gérer la collision consiste à détecter la collision et calculer comment réagit les objets impliqués dans la simulation.

Concrètement, le contexte de simulation est qu’une balle lancé peut ou pas frapper une boîte de conserve (un cylindre). Une partie du devoir va être de simuler la trajectoire de la basse et de la boîte avant la collision. Une autre partie va être capable de détecter une collision entre la balle et le sol ou la boîte et le sol, dans le cas où la balle ne frappe pas la boîte. Ces 2 premières parties étaient présente dans le devoir #2, donc on va pouvoir réutiliser. Les principales parties du devoir 3 vont consister à implémenter une fonction capable de détecter la collision et calculer la vitesse du centre de masse et angulaire des 2 objets après la collision. Les paramètres variable de la simulation sont la vitesse initial de la balle et de la boîte.

Les équations importantes pour les simulations (4 pts)

Dans le devoir #2, nous avons déjà discuter des différentes équations nécessaires au simulations. Plus spécifiquement comment utiliser les équations numériques pour calculer la trajectoire à des moments discrets ainsi que la détection d’une sphère avec le sol. Ce qui va être de nouveau va être la détection d’un cylindre qui tourne sur lui-même avec le sol, les équations pour détecter la collision et les équations pour calculer les forces appliqués durant la collision et en déduire la vitesse final.

2.1 Équations du mouvement à résoudre

### 2.1.1 Équations pour les accélérations non constante.

Notre fonction reçoit comme paramètre la vitesse initiale de la balle, la vitesse angulaire de la boîte ainsi que le temps t où la balle a été lancé. Nous connaissons les deux forces s’appliquant sur la balle et la boîte (force gravitationelle et force de frottement visqueuse). Nous cherchons à obtenir la position de la balle et de la boîte à plusieurs temps t. À partir de ces informations nous allons pouvoir calculer les collisions et vérifier les conditions d’arrêts. Nous avons la masse et la position initiale de la balle et de la boîte. Nous cherchons donc des équations qui mettent en relation r(t0), m et F et qui permettent de déterminer r(t).

|  |  |
| --- | --- |
| **Variable** | **Signifiation** |
| v(t) | Vitesse au temps t |
| v(t0) | Vitesse initiale |
| a(t) | Accélération au temps t |
| r(t) | Position au temps t |

|  |  |
| --- | --- |
| **Ce qu’on cherche** | **Équations** |
| La vitesse aux temps t |  |
| La position aux temps t |  |

Encore une fois on ne connaît pas directement l’accélération, par contre on connait la force exercé. Il nous faut donc des équations pour calculer l’accélération en fonction de la force

### 2.1.2 Équations pour trouver l’accélération en fonction de la force

En premier lieu, il va falloir être en mesure de transformer la force, chose qu’on connaît, en une accélération, quelque chose qu’on a besoins.

|  |  |
| --- | --- |
| **Variable** | **Signifiation** |
| a(t) | Accélération au temps t |
| Fnet(t) | Force net au temps t |
| m | Masse (kg) |

|  |  |
| --- | --- |
| **Ce qu’on cherche** | **Équations** |
| L’accélération au temps t |  |

On est donc en mesure de calculer l’accélération au temps t connaissant la force appliqué au temps t.

### 2.1.3 Équations de la force dans notre devoir

Pour ce devoir, 2 types de forces vont s’appliquer à notre balle et boîte.

**Force gravitationnelle**

La première force est la classique force gravitationnelle qui est constante et qui se calcul de la même façon pour la balle et la boîte. La force se fait vers le bas.

|  |  |
| --- | --- |
| **Variable** | **Signifiation** |
| Fg | Force gravitationnelle |
| mb | Masse de l’objet |

|  |  |
| --- | --- |
| **Ce qu’on cherche** | **Équations** |
| Force gravitationnelle |  |

**Force de frottement**

La balle et la boîte étant en mouvement subisse une force de frottement visqueuse avec l’air. Cette force s’applique vers le sens inverse de la vitesse de l’objet.

|  |  |
| --- | --- |
| **Variable** | **Signifiation** |
| Fv | Force de frottement |
| k | Coefficient |
| A | Air de l’objet |
| v | Vitesse de l’objet |

|  |  |
| --- | --- |
| **Ce qu’on cherche** | **Équations** |
| k pour la balle |  |
| k pour la boîte |  |
| A pour la balle |  |
| A pour la boîte |  |
| Force de frottement |  |

### 2.1.4 Équations pour calculer rotation d’un objet

Ce devoir introduit la nécessité de calculer la position angulaire de la boîte car celui ci a une vitesse angulaire. Par contre cette vitesse est constante, car il n’y pas de force à considérer qui agirait dessus.

Comme la vitesse angulaire est constante il est facile d’obtenir l’angle de la boîte à un temps t.

|  |  |
| --- | --- |
| **Variable** | **Signifiation** |
|  | Vitesse angulaire constante de la balle |
| R0 | Vecteur d’angle initial |
| R(t) | Vecteur d’angle au temps t |

|  |  |
| --- | --- |
| **Ce qu’on cherche** | **Équations** |
| L’angle de la boîte au temps t | R(t) = R0 + \* t |

Une fois qu’on connaît l’angle, on doit être en mesure de calculer la disposition de la boîte dans l’espace. Cela est possible en appliquant une rotation à chaque point de la boîte en fonction de son centre de masse. Cela est possible à l’aide des quaternion: on peut appliquer une rotation d’un angle x à un point par rapport à un vecteur unitaire u.

|  |  |
| --- | --- |
| **Variable** | **Signifiation** |
|  | Angle de la rotation |
|  | Axe arbitraire, auquel se fait la rotation |
|  | Quaternion unitaire utilisé durant l’équation |
|  | Vecteur qu’on veut appliquer une rotation |
|  | Quaternion du vecteur après rotation |

|  |  |
| --- | --- |
| **Ce qu’on cherche** | **Équations** |
| Quaternion du vecteur après rotation |  |
| Quaternion de rotation |  |
| Conjugué du quaternion de rotation |  |
| Quaterion de rotation |  |

2.2 Équations qui contrôlent la simulation

Il est important d’avoir des conditions d’arrêts pour être en mesure d’arrêter les calculs de la simulation. Sans ces conditions on pourrait gaspiller du temps de calcul et l’affichage graphique de la solution serait confondante. Dans ce devoir il existe 2 conditions d’arrêts, une s’il y a une collision entre la boîte et la balle, c’est à dire qu’un point de la balle se trouve à l’intérieur ou sur la surface de la boite, qui va être détaillé dans la section 2.3. L’autre condition d’arrêts est si la balle touche le sol.

Pour les formules suivantes, on assume que notre méthode va avoir un mécanisme de contrôle pour s’assurer que la balle ne passe pas tout droit.

|  |  |
| --- | --- |
| **Variable** | **Signifiation** |
| zi | Position de la balle au temps i, par rapport à l’axe des z |
| r | Rayon de la balle |
| zs | Position du sol, par rapport à l’axe des z |

|  |  |
| --- | --- |
| **Ce qu’on cherche** | **Équations** |
| Est-ce que la balle a traversé le sol | zi - r <= zs |

2.3 Équations qui contrôlent la détection de la collision balle-cylindre(1 pt)

Pour la détection de la collision, nous avons pris en compte deux cas. Le premier cas est celui ou la balle frappe le conteneur sur la partie courbe. Cet collision est la plus facile car nous avions a trouver lorsque la balle est exactement ou plus petite à une distance du rayon de la balle et rayon du cylindre selon l’axe de centre du cylindre.

La second cas de collision est lorsque la balle toucherais au bas ou au haut du conteneur cylindrique. Dans ce cas, nous regardons si le centre de la balle est à l’extérieur du plan que forment les deux cercle (couvercle et bas). Si c’est le cas, nous faisons une projection orthogonale du centre de la balle sur ces plans.

Si la balle se trouve a une distance plus grande que son rayon au bord du cylindre, il n’y aura pas de collision possible. Par contre, si celle-ci se trouve au moins a la distance de son rayon, on doit regarder le point de la projection pour déterminer si il y a contact.

Dans ce cas, nous déterminons le point de la balle qui est le plus proche du centre du cylindre et pouvons déterminer avec les deux distance entre les centre et ce point s’il y a collision.

|  |  |
| --- | --- |
| **Variable** | **Signifiation** |
| P | Plan coplannaire avec la face |
| C | Centre de masse de la balle |
| CP | Vecteur entre le centre de masse de la balle et le centre de masse projeté sur le plan |
| AC | Distance entre un point quelconque du plan et le centre de masse |
| n | Vecteur perpendiculaire au plan |

|  |  |
| --- | --- |
| **Ce qu’on cherche** | **Équations** |
| La projection orthogonale du centre de masse de la balle sur le plan dont fait partie la face circulaire de la boîte | projection.pngÀ modifier pour changer les noms de variable dans Word! |
| La projection orthogonale du centre de masse de la balle sur la droite perpendiculaire au plan | CP = A + (AC \* PA) / ||PA||² \* PA |
| La balle frappe la boîte sur son côté | distance = Norm(positionBalle - positionBoite)  si distance < rayonBalle + rayonboite, il y a collision |

2.4 Équations pour calculer vitesses après collision

Puisqu’il est demandé de seulement calculé la vitesse après la collision et non durant la collision on peut utiliser la méthode des conditions initiales. On suppose que le solide ne subit pas de déformation, ce qui va simplifier la simulation. Il faut d’abord introduire certaine notion de base avant de se lancer dans les détails des calculs. Ensuite on va regarder les équations pour voir l’impact de collision sur la vitesse relative, avec ces informations on va être en mesure de calculer l’impulsion, qui elle est nécessaire pour calculer la vitesse de déplacement et angulaire après la collision.

### 

### 

### Notion de base

|  |  |
| --- | --- |
| **Variable** | **Signifiation** |
|  | Quantité de mouvement du centre de masse des objets a et b |
|  | Moment cinétique des objets a et b autour de leur centre de masse respectif |
|  | Position du point par rapport au centre de masse des objets a et b ou la force est appliqué |
|  | Masse de l’objet a |
|  | Masse de l’objet b |

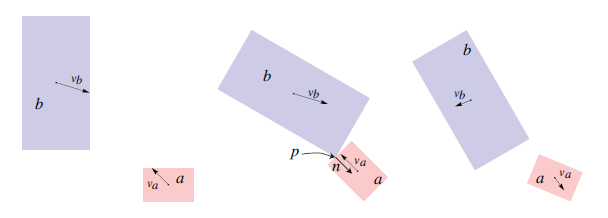
|  |  |
| --- | --- |
| **Ce qu’on cherche** | **Équations** |
| Quantité de mouvement |  |
| Moment cinétique |  |

### 

### 

### Calcul de la vitesse relative

Pour calculer la vitesse des solides après la collision, on doit d’abord calculer la vitesse relative du point p (point de collision) des 2 solides avant et après collision.



|  |  |
| --- | --- |
| **Variable** | **Signifiation** |
|  | Vitesse du solide a (son centre de masse) |
|  | Vitesse angulaire su solide a |
|  | Distance entre le centre de masse et du point p (point de collision) sur le solide a |

|  |  |
| --- | --- |
| **Ce qu’on cherche** | **Équations** |
| Vitesse du point p sur un solide |  |
| Vitesse relative du point p avant et après collision |  |
| Vitesse relative du point p avant et après la collision normales à la surface (ici solide a) |  |

### 

### 

### 

### Calcul de l’impulsion

Maintenant qu’on connaît les vitesses relatives la prochaine étape est de calculer la vitesse du solide de façon absolue. On y arrive en calculant l’impulsion.

|  |  |
| --- | --- |
| **Variable** | **Signifiation** |
|  | Normale qui part du centre jusqu’au point de collision |
|  | Matrice du moment cinétique inversé |
|  | Rayon entre centre de masse et point de collision |
|  | Coefficient de restitution |

|  |  |
| --- | --- |
| **Ce qu’on cherche** | **Équations** |
| Impulsion |  |
| Valeur de alpha |  |
| Valeur de G |  |

### Calcul de la vitesse de déplacement

Maintenant qu’on a l’impulsion on est en mesure de calculer la vitesse du centre de masse d’un solide après la collision.

|  |  |
| --- | --- |
| **Variable** | **Signifiation** |
|  | Impulsion |
|  | Normale qui part du centre jusqu’au point de collision |
|  | Vitesse avant collision |
| **m** | Masse du solide |

|  |  |
| --- | --- |
| **Ce qu’on cherche** | **Équations** |
| Vitesse du solide après la collision |  |

### Calcul de la vitesse angulaire

L’autre information demandé était de calculé la vitesse angulaire avant et après la collision.

|  |  |
| --- | --- |
| **Variable** | **Signifiation** |
|  | Vitesse angulaire initial du solide a avant la collision |
|  | Matrice inversé de l’intertie |
|  | Rayon entre le centre de masse et le point de collision |
|  | Normale entre le centre de masse et le point de collision |
|  | Impulsion |

|  |  |
| --- | --- |
| **Ce qu’on cherche** | **Équations** |
| Vitesse angulaire après la collision |  |

Méthodes de résolution des équations

Justification du pas de temps

Pour le pas de temps utilisé lors de ce devoir, nous avons utilisé une technique similaire au deuxième devoir puisque l’échelle de temps de ce devoir est pratiquement le même que lors de ce devoir. Le pas de temps est donc raffiné au fur et à mesure des itérations de simulations jusqu’à obtenir la précision désiré. Nous avons donc déterminé qu’un pas de temps initial de 0.01s est idéal pour ne pas faire trop de calcul en ayant la précision désiré. Le pas de temps de 0.02s que nous avons utilisé lors du devoir 2 n’était pas assez précis puisqu’il se pourrait alors que la collision ne soit pas détecté. Lors d’une détection de collision, nous retournons à l’itération avant la collision et nous raffinons le pas de temps pour obtenir la précision nécessaire.

À changer les mots contre plagiat

Description des mesures de vérifications de la précision

Nous avons 2 contraintes de précisions, la première étant que la réponse finale doit avoir une précision de ±1mm et la deuxième il faut que pour toute simulation soit capable de détecter les conditions d’arrêts: il ne faut pas que entre 2 temps consécutif on manque la cible ou le sol. Nous utilisons donc 2 mécanismes pour respecter ces contraintes.

## Équations numérique à résoudre

Puisque ça nous tente pas de résoudre analytiquement ces équations, on va employer des méthodes numériques. Qui consiste à trouver les valeurs de q(tn) basé sur la valeur de q(tn-1).

|  |  |
| --- | --- |
| **Variable** | **Signifiation** |
|  | Condition initiale (ici la position initiale de la balle) |
|  | Différentielle de q au temps t. (ici la vitesse de la balle au temps t) |
|  | Valeur de q au temps t. |
| ki | Approximation de pour la méthode Runge-Kutta |
| O(delta) | Erreur |

|  |  |
| --- | --- |
| **Ce qu’on cherche** | **Équations** |
| Méthode de Euler |  |
| Méthode de Runge-Kutta |  |

Dans notre cas les vecteurs g et q ont les significations suivantes:

|  |  |
| --- | --- |
| **Variable** | **Formule** |
| Vecteur ayant comme composant la vitesse et la position |  |
| Vecteur ayant comme composant l’accélération et la vitesse. |  |

Donc on part du principe que si on connaît la vitesse et la position au temps t, on est capable de trouver l’accélération à ce temps (dépendant de l’option choisi). Ensuite les méthodes numériques permettent d’estimer la dérivée de q, ce qui permet d’estimer la vitesse et la position au temps t+1.

Question d’avoir une meilleur précision nous allons utiliser la méthode de Runge-Kutta au lieu de la méthode d’Euler. Cette dernière à une erreur globale proportionnelle à delta-t et est donc moins précise du (delta-t)^4 de Runge-Kutta.

## Contrainte sur la réponse finale de ±1mm

Pour respecter la contrainte de précision de ±1mm sur l’endroit de la collision, nous avons utilisé une méthode similaire à celle utilisée pour le calcul du pas de temps. Lors de la détection de collision, le pas de temps que nous utilisons n’est pas assez précis pour respecter la précision, donc nous devons raffiner le pas de temps utilisé afin d’obtenir la précision nécessaire. Nous prenons donc la position de la balle une itération avant la détection de la collision et nous divisons le pas de temps en 2. Nous répétons le processus jusqu’à ce que la balle entre en collision avec la boîte et qu’elle ait bougé de moins de 1mm.

Description du logiciel matlab (3 pts)

Dans le logiciel, nous commençons par calculer les position de la balle, du cylindre et de ses deux extrémités dans le temps (avant et pendant le lancer de la balle).

À chaque itération de temps, nous calculons si la balle est entrée en collision avec le cylindre avec les deux cas expliquées plus haut (collision entre balle et surface courbe du cylindre, collision entre la balle et les deux extrémités du cylindre).

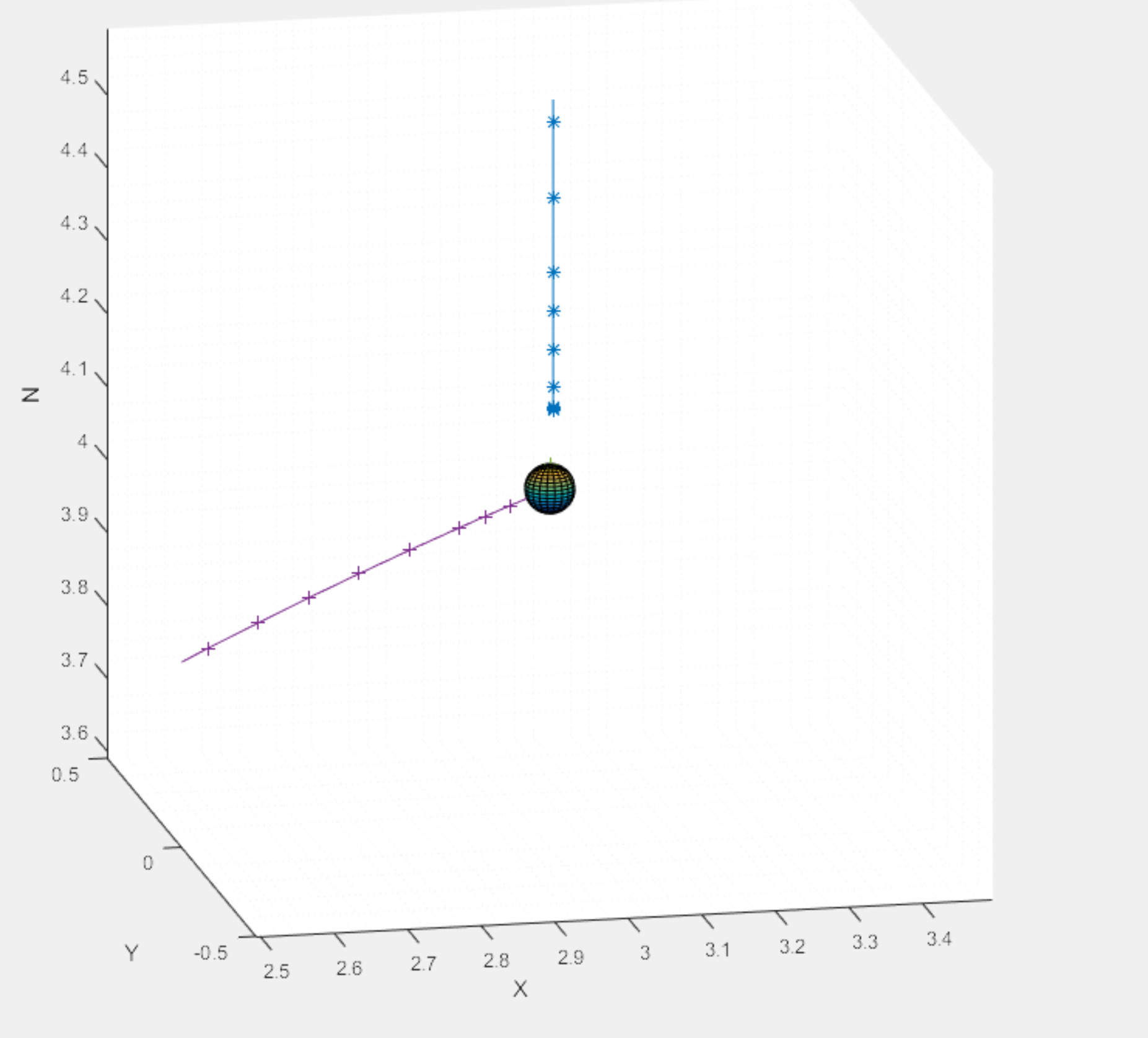
Si nous détectons une collision ou encore un enfoncement (dépassement de la collision), nous allons retourner dans le temps pour itérer avec un petit pas de temps pour déterminer à quel moment (avec une certaine précision) la collision a eu lieu et ci celle-ci est bien vérifié.

La collision est ensuite étudier avec le sol pour voir si la simulation doit s’arrêter.

Présentation et analyse des résultats (3 pts)

Tir #1:

Option 1 :

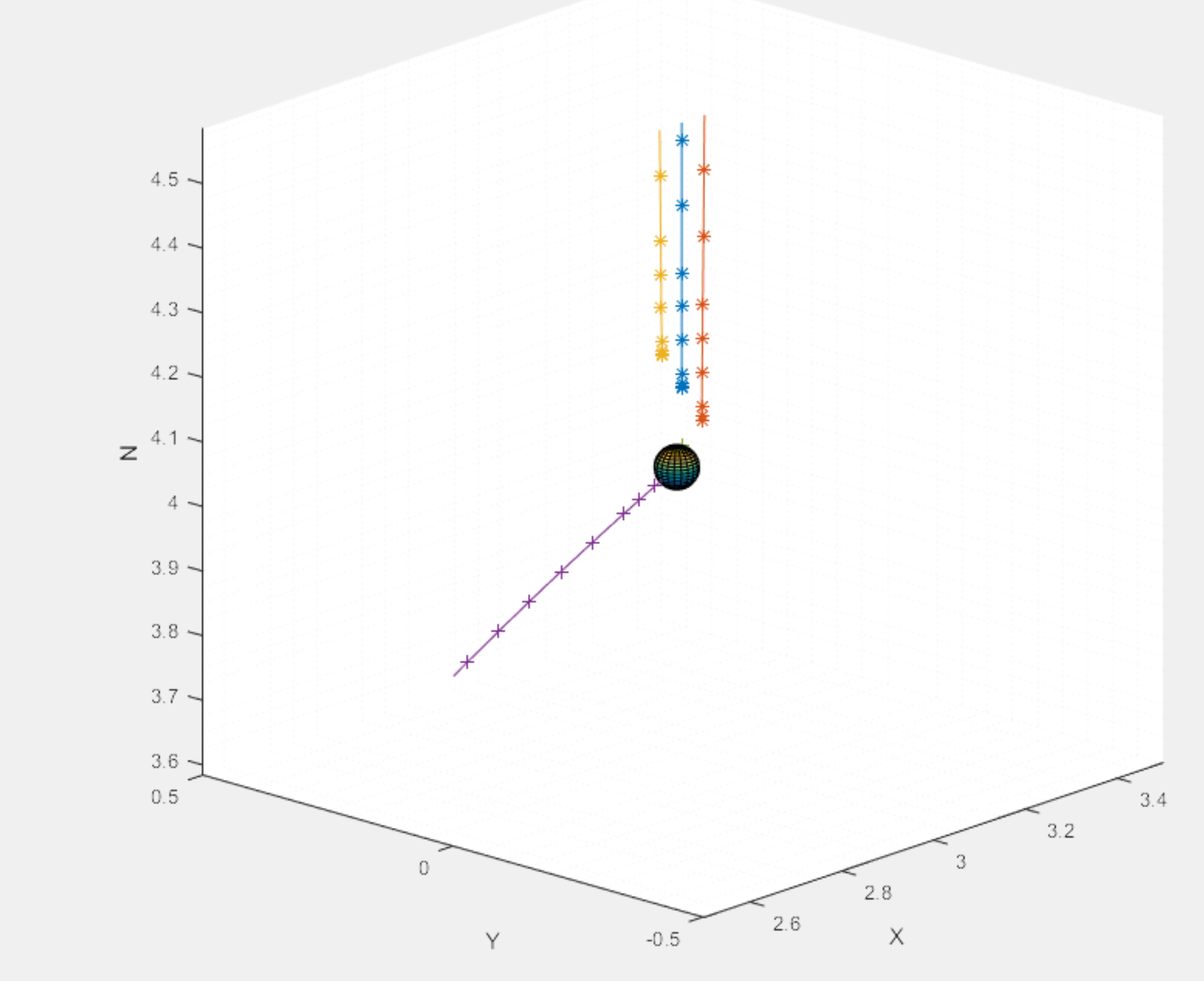


|  |  |  |  |
| --- | --- | --- | --- |
| Collision | Temps d’arrêt (s) | Position Balle Collision (m) | Position Boite collision (m) |
| Oui | 1.0979 | (2.9956, 0, 4.0568) | (3.0, 0, 4.1651) |

|  |  |  |  |
| --- | --- | --- | --- |
| Vitesse Boite avant collision (m/s) | Vitesse Boite après collision (m/s) | Vitesse Balle avant collision (m/s) | Vitesse Balle après collision (m/s) |
| (0,0,-10.5648) | (0,0,-2.0178) | (6.8318, 0, 2.5462) | (6.8318, 0, -8.5060) |

|  |  |  |  |
| --- | --- | --- | --- |
| Vitesse angulaire Boite avant collision (rad/s) | Vitesse angulaire Boite après collision (rad/s) | Vitesse angulaire Balle avant collision (rad/s) | Vitesse angulaire Balle après collision (rad/s) |
| (0,0,0) | (0, 15.1720, 0) | (0,0,0) | (0,0,0) |

Option 2:



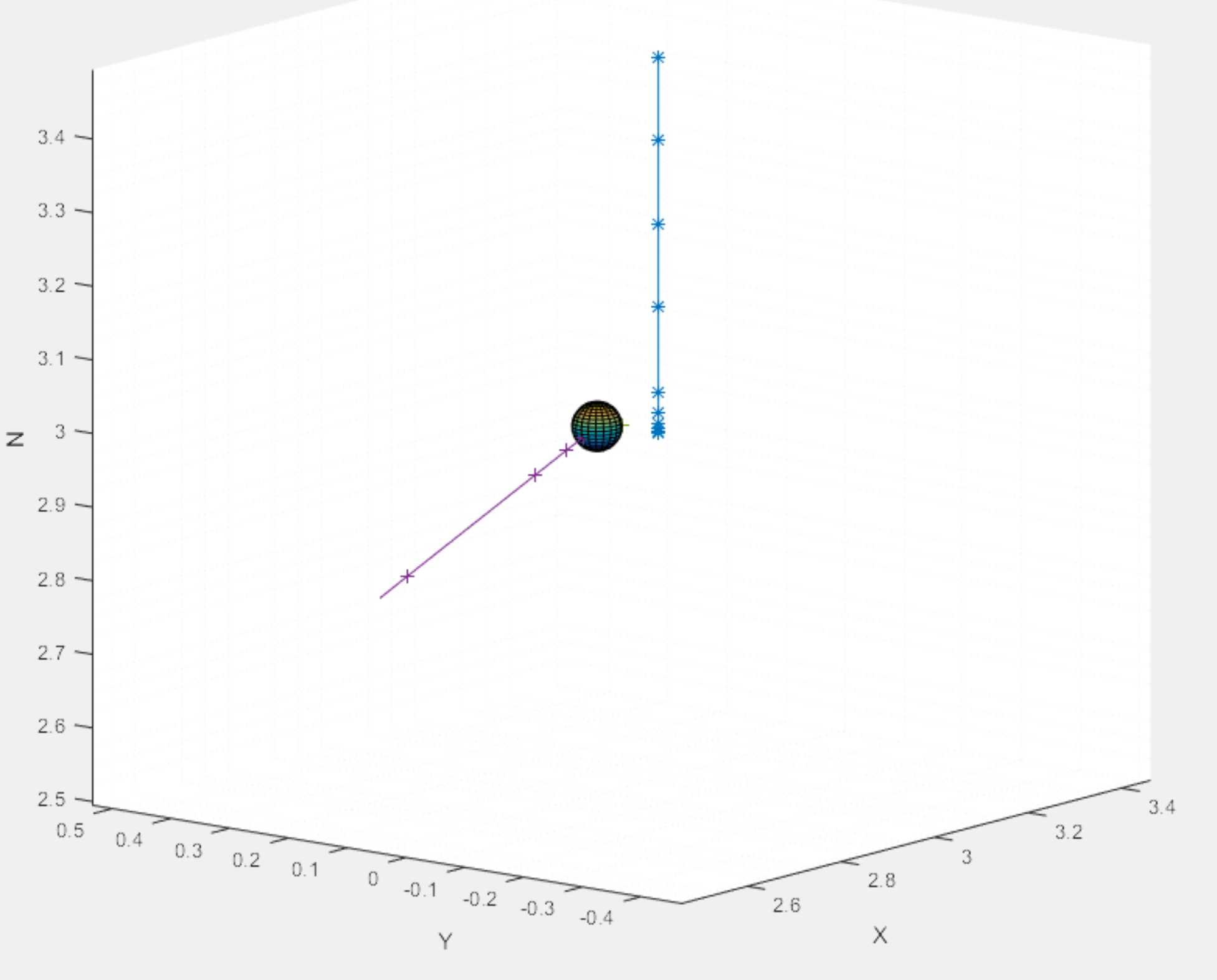
|  |  |  |  |
| --- | --- | --- | --- |
| Collision | Temps d’arrêt (s) | Position Balle Collision (m) | Position Boite collision (m) |
| Oui | 1.0970 | (2.9892, 0, 4.0544) | (3.0, 0, 4.1750) |

|  |  |  |  |
| --- | --- | --- | --- |
| Vitesse Boite avant collision (m/s) | Vitesse Boite après collision (m/s) | Vitesse Balle avant collision (m/s) | Vitesse Balle après collision (m/s) |
| (0,0,-10.5560) | (3.1274, 0, -3.5886) | (6.8318, 0, 2.5554) | (2.7878, 0, -6.4542) |

|  |  |  |  |
| --- | --- | --- | --- |
| Vitesse angulaire Boite avant collision (rad/s) | Vitesse angulaire Boite après collision (rad/s) | Vitesse angulaire Balle avant collision (rad/s) | Vitesse angulaire Balle après collision (rad/s) |
| (0, 2.3, 0) | (0, -118.3167, 0) | (0,0,0) | (0,0,0) |

Tir #2:

Option 1 :

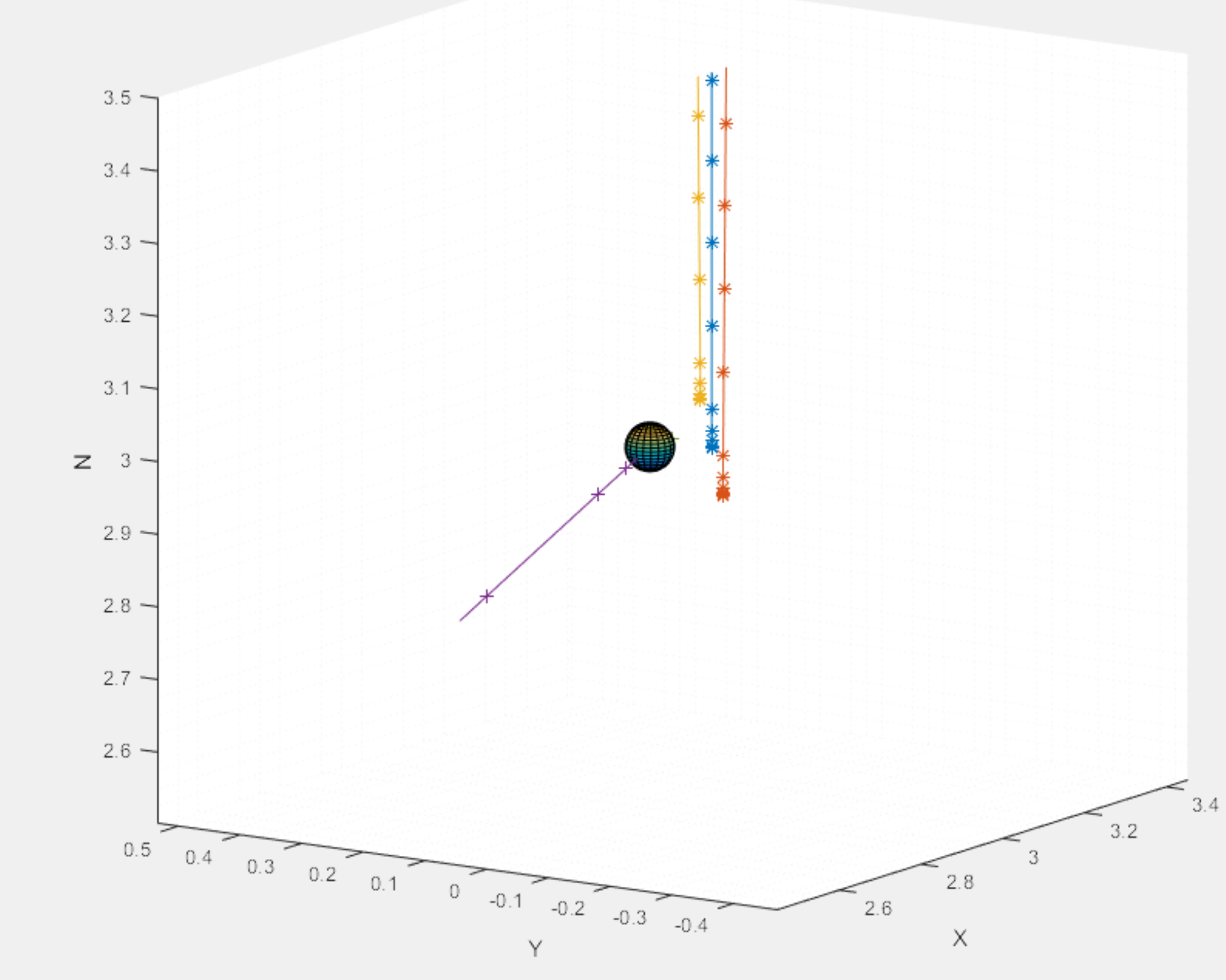


|  |  |  |  |
| --- | --- | --- | --- |
| Collision | Temps d’arrêt (s) | Position Balle Collision (m) | Position Boite collision (m) |
| Oui | 1.2048 | (2.9347, 0.0524, 2.9943) | (3.0, 0, 2.9812) |

|  |  |  |  |
| --- | --- | --- | --- |
| Vitesse Boite avant collision (m/s) | Vitesse Boite après collision (m/s) | Vitesse Balle avant collision (m/s) | Vitesse Balle après collision (m/s) |
| (0,0,-11.5735) | (10.6585, -8.5524, -11.5735) | (27.9822, 0.4997, 8.9665) | (14.1997, 11.5588, 8.9665) |

|  |  |  |  |
| --- | --- | --- | --- |
| Vitesse angulaire Boite avant collision (rad/s) | Vitesse angulaire Boite après collision (rad/s) | Vitesse angulaire Balle avant collision (rad/s) | Vitesse angulaire Balle après collision (rad/s) |
| (0,0,0) | (44.6611, 55.6592, 0) | (0,0,0) | (0,0,0) |

Option 2:



|  |  |  |  |
| --- | --- | --- | --- |
| Collision | Temps d’arrêt (s) | Position Balle Collision (m) | Position Boite collision (m) |
| Oui | 1.2046 | (2.9281, 0.0523, 2.9922) | (3.0, 0, 2.9839) |

|  |  |  |  |
| --- | --- | --- | --- |
| Vitesse Boite avant collision (m/s) | Vitesse Boite après collision (m/s) | Vitesse Balle avant collision (m/s) | Vitesse Balle après collision (m/s) |
| (0,0,-11.5713) | (9.9902, -8.7590, -7.6851) | (27.9822, 0.4997, 8.9688) | (15.0639, 11.8259, 3.9435) |

|  |  |  |  |
| --- | --- | --- | --- |
| Vitesse angulaire Boite avant collision (rad/s) | Vitesse angulaire Boite après collision (rad/s) | Vitesse angulaire Balle avant collision (rad/s) | Vitesse angulaire Balle après collision (rad/s) |
| (0, 2.3, 0) | (110.1545, 146.9508, 85.7010) | (0,0,0) | (0,0,0) |

Nous pouvons voir que dans notre dernier résultat, la vitesse angulaire de la boîte après collision est un peu trop

Discussion des défis rencontrés (1 pt)

### Lors de ce devoir, le langage MATLAB ne posait plus vraiment un problème puisque nous avons bien maîtrisé le langage lors des derniers devoirs. Notre plus grande difficulté lors de ce devoir a été de trouver un algorithme et les formules pour la détection de collision entre notre balle et notre boîte. Nous devions trouvé une manière de calculer si la balle frappe une des faces de la boîte, le côté de la boîte ainsi qu’une des arrêtes reliant le côté et une des faces de la boîte. Une fois cette difficulté surmonté, le reste du travail a été relativement facile puisque nous avons pu réutiliser une grande partie du code du Devoir 2 pour les calculs de force ainsi que pour le calcul de la position de la sphère en utilisant une méthode numérique.