

**PHS4700**

**Physique pour les applications multimédia**

Automne 2017

PAGE COUVERTURE **OBLIGATOIRE** POUR TOUS LES DEVOIRS

Numéro de devoir : 03

Numéro de l’équipe : 07

Numéro du groupe : 01

|  |  |  |
| --- | --- | --- |
| Nom: Bourgault | Prénom : Gabriel | matricule: 1794069 |
| Signature : |  |  |
| Nom: Chan | Prénom : Kevin Ka Hin | matricule: 1802812 |
| Signature : |  |  |
| Nom: Nguyen | Prénom : Kenny | matricule: 1794914 |
| Signature : |  |  |
| Nom: Silva-Pinto | Prénom : Nuno | matricule: 1799144 |
| Signature : |  |  |

Table des matières

[Introduction 2](#_Toc497076244)

[Théorie et équations 3](#_Toc497076245)

[Équations du mouvement à résoudre 3](#_Toc497076246)

[Équations du movement à résoudre 4](#_Toc497076247)

[Équations pour identifier une collision ou déterminer l’arrêt de la simulation 4](#_Toc497076248)

[Équations pour déterminer les vitesses linéaires et angulaires des deux autos tout juste après la collision 4](#_Toc497076249)

[Méthode de résolution des équations du mouvement et justification 4](#_Toc497076250)

[Intervalles de temps t choisis pour la résolution et justification 5](#_Toc497076251)

[Présentation et analyse des résultats 7](#_Toc497076252)

[Représentation visuelle des simulations 8](#_Toc497076253)

[Les six cas à évaluer 12](#_Toc497076254)

[Vérifications effectuées pour assurer la précision de nos simulations 13](#_Toc497076255)

[Conclusion 14](#_Toc497076256)

# Introduction

Ce troisième devoir a pour but d’étudier la collision entre deux autos glissant sur une surface. Ces derniers sont modélisés comme étant un bloc rectangulaire chacun. Ce problème est simulé en 2D.

Nous nous intéressons alors à une situation précise :

Les autos « a » et « b » se situent à un point de départ différent. Les deux autos ont une vitesse initiale alors qu’il roule sur la glace. L’auto « a » décide de freiner brusquement. Cela fait en sorte que l’auto « a » commence à glisser sur la glace tout en tournant autour de son centre de masse avec une vitesse angulaire constante. De son côté, l’auto « b » voit que l’auto « a » perd le contrôle et décide de freiner à un temps tb. L’auto « b » commence aussi à tourner autour de son centre de masse avec une vitesse angulaire constante ainsi qu’à glisser sur la glace. Les vitesses angulaires après freinage des deux autos sont constantes tant et aussi longtemps qu’il n’y a pas collision entre les deux.

Cette situation est simulée à l’aide de six ensembles de conditions initiales différents. Ainsi, l’objectif du devoir est de pouvoir simuler les trajectoires des deux autos dans le plan xy en fonction du temps, sous forme de graphique. De plus, il faudra aussi détecter s’il y a des collisions entre les deux autos. Si oui, notre simulation devra pouvoir déterminer les vitesses à partir du centre de masse des autos après la collision. Cela inclut aussi la vitesse angulaire. Finalement, une simulation est considérée terminée lorsqu’une collision survient ou lorsque la vitesse des autos est inférieure à 1 cm/s.

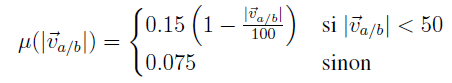
Ainsi, le rapport suivant contient la théorie ainsi que les équations auxquelles nous nous sommes tournés pour nous aider à programmer la fonction Devoir3.m. Cette section sera par la suite appuyée par la présentation ainsi que l’analyse des résultats que nous avons obtenus. Finalement, une brève discussion des problèmes rencontrés servira de conclusion à ce devoir.

# Théorie et équations

## Équations du mouvement à résoudre

Une seule force s’applique sur les voitures : la force de frottement. Durant la simulation, elle est constamment appliqué au véhicule A et seulement au véhicule après que le temps tb.

La force de frottement est donnée par l’équation 1. La balle étant un solide se déplaçant dans un fluide, soit l’air, elle subit des collisions élastiques et inélastiques avec les molécules de l’air. Cela résulte par une force de frottement. Elle ne s’applique que sur la partie immergée de la balle, ce qui donne alors lieu à un mouvement de rotation. Certaines valeurs ont été fournies dans l’énoncé du devoir : la masse volumique de l’air (air = 1.2 kg/m3), l’aire efficace de la balle de la balle ( A = ( R2b) ) ainsi que Cv = 0.5. b correspond à la vitesse de la balle. Cette équation est utilisée dans la situation 2 et la situation 3.



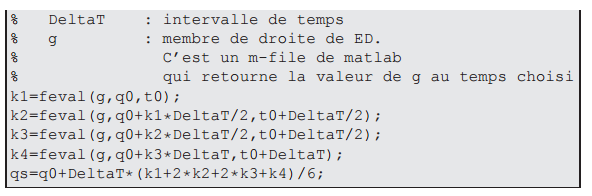
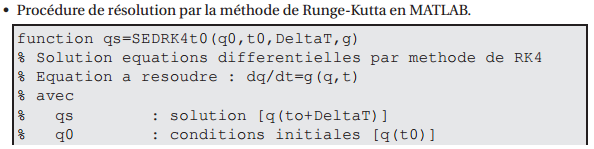
Équation 1

Pour isoler l’accélération de l’auto, on applique la relation fondamentale de la dynamique. Puisque la balle a une masse m et qu’un vecteur de force F agit sur celle-ci, l’auto aura un vecteur d’accélération a dans la même direction.

Équation 2

## 

## Méthode de résolution des équations du mouvement et justification

Pour résoudre nos équations différentielles de mouvement, nous avons fait appel à la méthode de Runge-Kutta classique d’ordre 4. Pour faire simple, l’algorithme se base sur une méthode itérative qui prédit puis corrige les estimations d’une solution. Cela tend à la rendre de plus en plus précise. Une procédure de résolution avec l’algorithme existe sur MATLAB.

Notre fonction « SEDRK4 » appelle ainsi Runge-Kutta d’ordre 4. Elle prend en paramètre une fonction (« rouler » ou « frottement »). Celle-ci donne les vecteurs d’accélération et de vitesse en fonction du temps. « rouler » n’applique aucune force sur l’auto. « frottement » prend en considération le frottement en implémentant l’équation 1. Pour trouver l’accélération de l’auto, on divise la force appliquée sur celle-ci par sa masse. Le paramètre q0 de Runge-Kutta est un tableau contenant la vitesse et les positions initiales x, y et z de l’auto. Le paramètre g quant à lui est un tableau qui contient sa vitesse et son accélération. Pour le DeltaT, on lui donne la valeur de notre pas. On définit enfin le temps t0 soit le moment initial de la simulation. La solution qs, sera un tableau avec la vitesse et les positions finales x, y et z de l’auto à la fin de l’intervalle de temps.

Le principe de notre simulation est d’appliquer l’algorithme de façon récursive après chaque intervalle de temps DeltaT. Le qs retourné à l’itération *n* devient alors le q0 de l’itération *n + 1*. Ce modèle continue jusqu’à la fin de la simulation soit lorsque les deux véhicules entrent en collision ou bien qu’ils s’immobilisent.

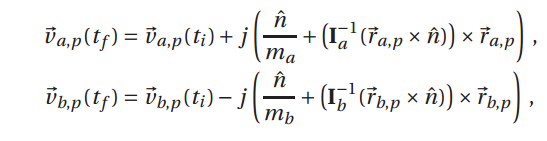
## Équations pour identifier une collision ou déterminer l’arrêt de la simulation

Afin de déterminer l’arrêt de la simulation, nous avons utilisé les algorithmes de détection des collisions vus en cours, tout en jumelant cela à une détection plus rapide :

1. Vérifier si les 2 autos, si représentées par des disques dont le rayon serait la distance entre le centre de masse et l’un des coins, entrent en collision.
   1. Si non, on quitte car il n’y a pas de collision.
2. Les sphères englobantes sont en collision, il y a des chances que nos voitures soient en collision. On applique alors la méthode des plans de division. Ce que nous avons fait est de vérifier si au moins un des coins de chaque auto était inclus dans l’autre auto. L’utilisation de cette méthode nous permet en même temps de déterminer le point exact de la collision.
   1. Si aucun coin n’est inclus dans l’autre véhicule, alors il n’y a pas de collision car il existe un plan de division.

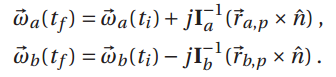
## Équations pour déterminer les vitesses linéaires et angulaires des deux autos tout juste après la collision

La vitesse linéaire des véhicules après la collision est donnée par la **loi des collisions sans frottement.** Les voitures effectuant des mouvements de rotations lors de leur glissement, la vitesse finale considère l’effet du moment de force en plus de l’effet force. Un effet angulaire est donc appliqué.

 La vitesse finale de la voiture A () et de la voiture B () est donné par ces équations :

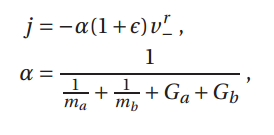
Équations 3 et 4

Elles impliquent les vitesses initiales, la normale, les masse et les vecteurs positions des points où la force est appliquée. On retrouve aussi le moment d’inertie et *j* qui seront détaillé plus bas.

Les équations des vitesses finales des autos A et B sont :

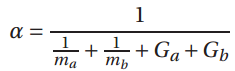
Équations 5 et 6

Pour résoudre celle-ci on a besoin des vitesses angulaires initiales en plus des vecteurs positions des points où la force est appliquée, de la normale, des moments d’inertie et de la variable *j.*

La variable j correspond à la composante normale à l’impulsion dont le sens de dernier est ce qui affecte la vitesse des véhicules. Le coefficient de restitution ε vaut 0.8 dans notre simulation. :

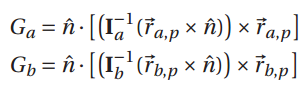
Équation 7

Pour calculé *j,* il faut calculer le facteur α donné ici. Il implique la masse des deux véhicules et des facteurs Ga et Gb :



Équation 8

Les facteurs Ga et Gb s’obtienne en mettant en relation les moments d’inerties, la normale et les vecteurs positions des points où la force est appliquée :



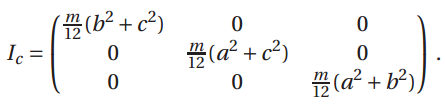
Équations 9 et 10

Nous utilisons enfin pour trouver *j* la vitesse relative du système avant la collision :



Équation 11

Finalement, le dernier élément entrant dans nos calculs est le moment d’inertie des véhicules. Les autos étant des solides réguliers, leur moment est facile est calculer. On applique alors cette matrice prenant comme variables les dimensions des véhicules en plus de leur masse :



Équation 12

Après cette longue application de formules, on arrive à trouver somme toute la vitesse et la vitesse angulaire finale des véhicules tout juste après leur collision.

## Intervalles de temps t choisis pour la résolution et justification

Après avoir fait des tentatives avec un pas fixe, nous avons finalement décidé d’implémenter un pas variable selon la distance parcourue et la proximité d’une collision. En effet, le pas est initialement de t = 0.01s. Cependant, lorsque nous détectons que les deux autos sont proches, nous réduisons le pas jusqu’à ce que les véhicules se déplacent de moins de 0.5cm par coup. Afin de déterminer si les véhicules sont proches, nous avons arbitrairement utilisé la condition suivante :

En effet, on compare la distance entre les deux centres de masse avec la distance combinée des deux rayons, cette-dernière majorée de 10% pour être complètement sûr.

# Présentation et analyse des résultats

**Tableau 1 : Conditions initiales pour les six cas à simuler**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tir** | **rai (m)** | **vai** | **rbi** | **vbi** | **tb** |
| 1 | [0 0] | [20 0 2] | [100 100] | [0 -20 -1] | 0.0 |
| 2 | [0 0] | [30 0 2] | [100 100] | [0 -30 -1] | 0.0 |
| 3 | [0 0] | [20 0 2] | [100 50] | [0 -10 0] | 1.6 |
| 4 | [0 0] | [10 10 1] | [25 10] | [10 0 0] | 0.0 |
| 5 | [0 0] | [20 0 2] | [100 50] | [0 -10 0] | 0.0 |
| 6 | [0 0] | [20 0 2] | [100 10] | [10 0 5] | 1.0 |

**Tableau 2 : Résultats des simulations des voitures a et b**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tir** | **Coll** | **tf (s)** | **raf** | **vaf** | **rbf** | **vbf** |
| 1 | 2 | 0,00260 | (1,3572 0,5000 0,8073) | (4,00 0,00 -2,53) |  |  |
| 2 | 3 | 0,00316 | (5,0086 0,90009 0,0110) | (10,0 1,00 -4,71) |  |  |
| 3 | 2 | 0,00232 | (1,3886 0,5000 0,8361) | (-5,00 0,00 -2,45) |  |  |
| 4 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |

## Représentation visuelle des simulations

Cette section offre un visuel de nos simulations. La trajectoire de l’option 1 est en bleu, l’option 2 en rouge et l’option 3 en jaune (voir version numérique).

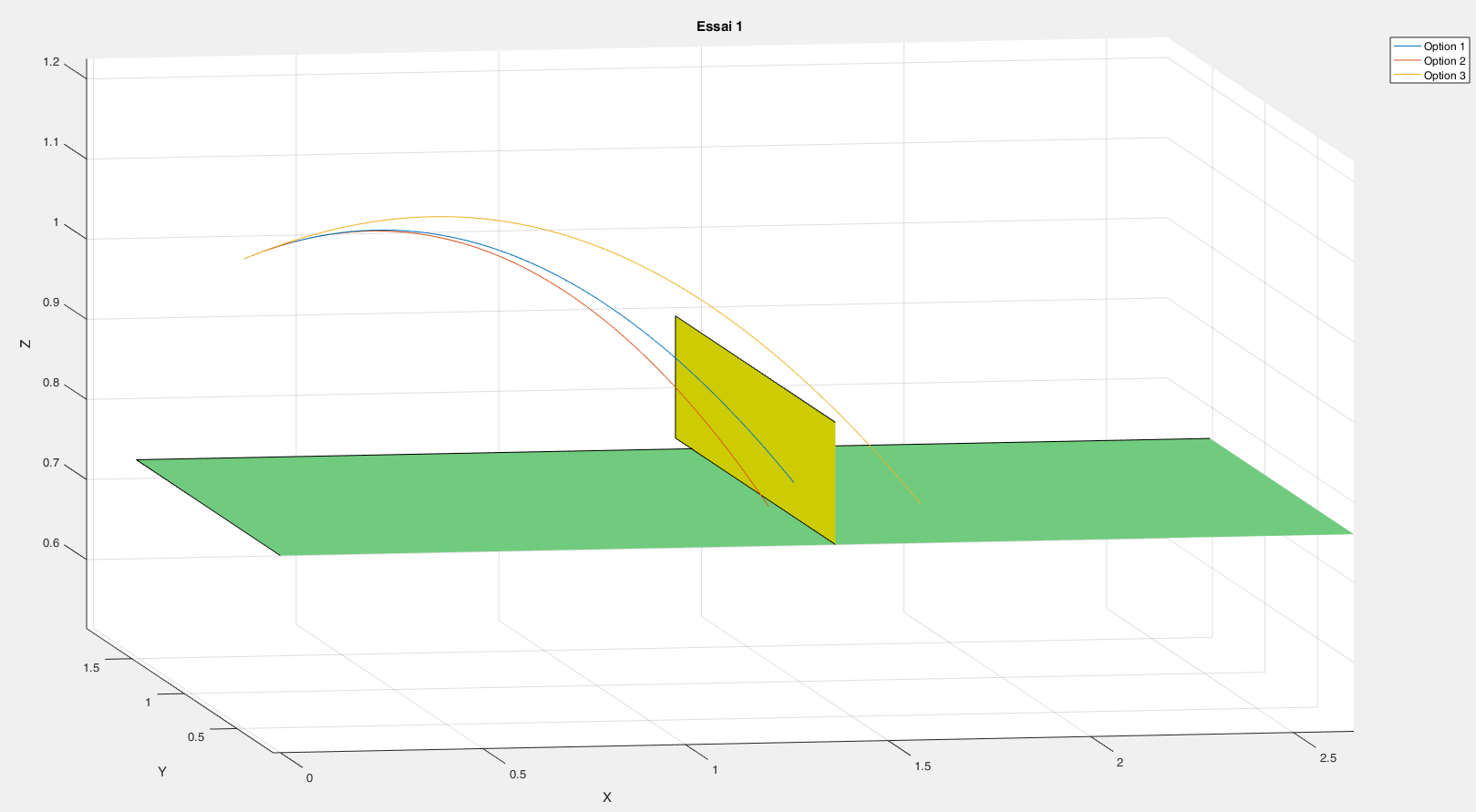


Figure 1 : Représentation graphique des coups de chacune des options lors de l’essai 1.

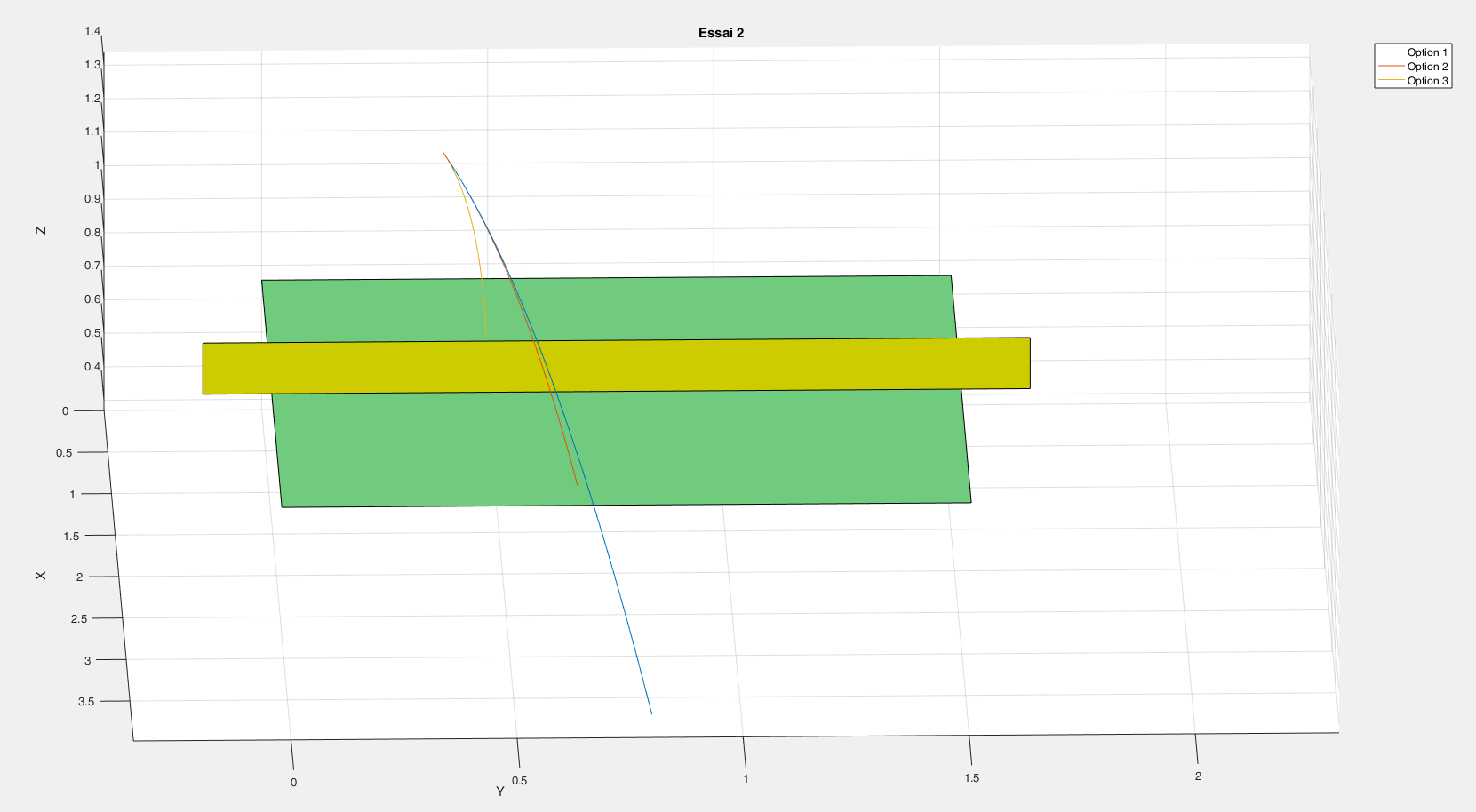


Figure 2 : Représentation graphique des coups de chacune des options lors de l’essai 2.

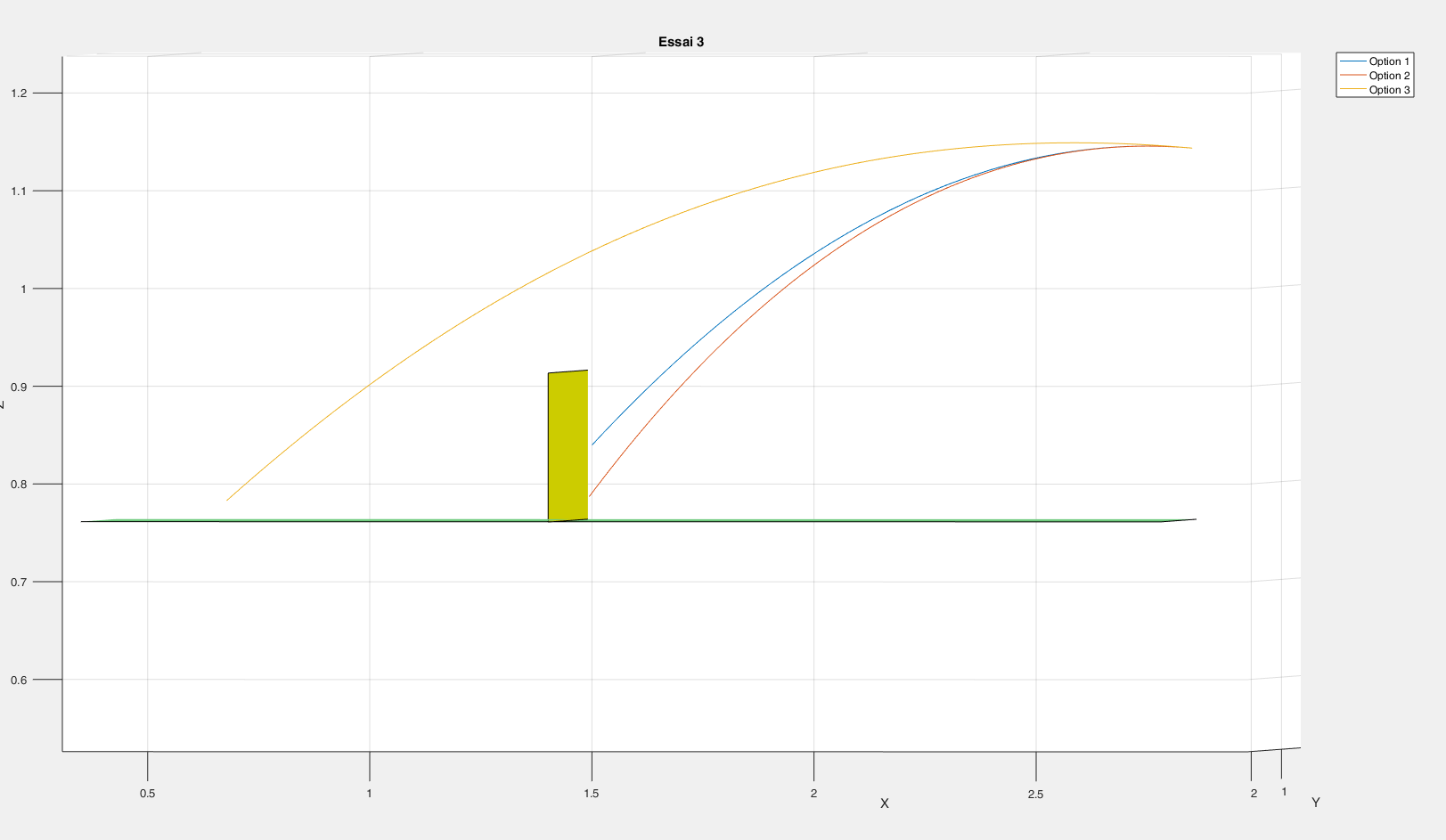


Figure 3 : Représentation graphique des coups de chacune des options lors de l’essai 3.

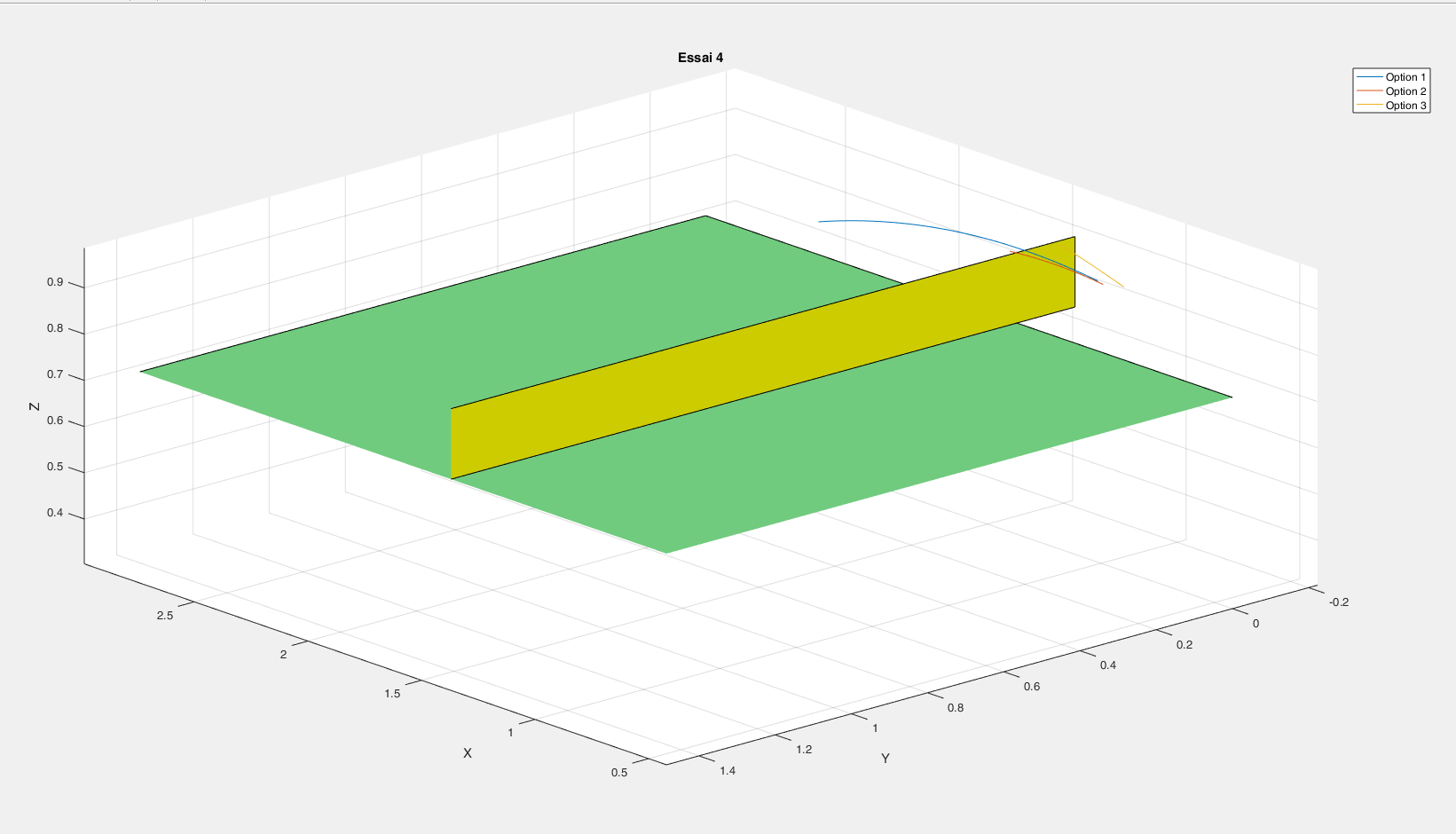


Figure 4 : Représentation graphique des coups de chacune des options lors de l’essai 4.

## Les six cas à évaluer

## Vérifications effectuées pour assurer la précision de nos simulations

Pour l’option 1, la plus simple, nous avons effectué les calculs à la main pour comparer avec le résultat obtenu par la simulation. Puisque les résultats étaient concluants, nous avons gardé les mêmes conditions (i.e. intervalle de temps) pour les autres options. Nous avons aussi vérifié que le point de départ de chacune des courbes correspondait aux conditions initiales de chacun des essais (**Tableau 1**). Nous avons aussi vérifié visuellement les collisions : cela nous a permis de valider la fiabilité de notre algorithme de détection de collision.

De plus, nous avons fait des hypothèses à l’aide des données brutes pour l’ajout de la force de Magnus : nous avons estimé qu’un *backspin* allait permettre à l’option 3 (Magnus) d’avoir une position finale plus loin que les autres. C’est ce qui arrive dans l’essai #3.

Aussi, nous avons validé que la vitesse finale diminue, comme attendu, lorsqu’on ajoute la force visqueuse.

Néanmoins, nous avons remarqué un manque de précision sur les collisions. Ainsi, nous aurions dû réduire le t d’enregistrement des collisions. Cela aurait offert une meilleure estimation de la position.

# Conclusion

Nous n’avons pas rencontré de difficultés