



## PHS4700


### Physique pour les applications multimédia Automne 2017

PAGE COUVERTURE **OBLIGATOIRE** POUR TOUS LES DEVOIRS

**Numéro de devoir : 02**

**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  |
| Théorie et équations .....  | 3  |
| Équations du mouvement à résoudre .....   | 3  |
| Équations pour déterminer l'arrêt de la simulation (collision) .....              | 4  |
| Méthode de résolution des équations du mouvement et justification .....           | 5  |
| Intervalles de temps $\Delta t$ choisis pour la résolution et justification ..... | 6  |
| Présentation et analyse des résultats.....  | 6  |
| Représentation visuelle des simulations .....                                     | 7  |
| Option 1 : Force gravitationnelle seulement.....                                  | 11 |
| Option 2 : Force gravitationnelle et force visqueuse .....                        | 11 |
| Option 3 : Force gravitationnelle, force visqueuse et force de Magnus .....       | 12 |
| Vérifications effectuées pour assurer la précision de nos simulations.....        | 13 |
| Conclusion .....  | 14 |

## Introduction

Ce deuxième devoir a pour but d'étudier la trajectoire d'une balle au tennis sur table. Pour ce faire, nous avons programmé une application permettant la simulation de cette trajectoire. De plus, les principales caractéristiques de jeu seront simulées. Cela inclut un système de coordonnées de références où l'origine se trouve au sol aligné avec le coin droit de la table, une surface de jeu, un filet ainsi qu'une balle sphérique.

Nous nous intéressons alors à trois situations précises :

1. Seule la force gravitationnelle agit sur la balle.
2. La force gravitationnelle ainsi qu'une force de frottement visqueux agissent sur la balle.
3. La force gravitationnelle, la force de frottement visqueux ainsi qu'une force de Magnus agissent sur la balle.

Chacune de ces situations est testée à l'aide de quatre coups différents. Ainsi, au total, nous avons 12 simulations. Finalement, une simulation se termine lorsque la balle touche le filet, lorsque la balle touche la surface de la table ou lorsque la balle touche le sol.

Pour pouvoir simuler les trois situations précédentes, nous avons programmé une fonction Matlab nous permettant de tout d'abord déterminer si le coup a réussi ou non. Dans le cas échéant, on différencie un coup échoué en trois catégories : la balle frappe la table du côté du joueur, la balle frappe le filet ou la balle est frappée à l'extérieur de la table (après avoir touché le sol). Celle-ci détermine aussi les variables suivantes lorsque la simulation se termine : le temps (en secondes), les positions finales en  $x$ , en  $y$  et en  $z$  (en m) du centre de masse de la balle, le vecteur vitesse final du centre de masse de la balle (en m/s).

Ainsi, le rapport suivant contient la théorie ainsi que les équations auxquelles nous nous sommes tournés pour nous aider à programmer la fonction Devoir2.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

Trois forces sont appliquées à la balle dans nos simulations: la force gravitationnelle, la force de frottement visqueux et l'effet Magnus.

La force gravitationnelle sur la balle est donnée par l'équation 1. Elle correspond à l'interaction gravitationnel entre la balle qui est attiré par la Terre. Celle-ci indique que la force ne s'applique sur l'axe des z. Cette équation est utilisée dans les trois situations.

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

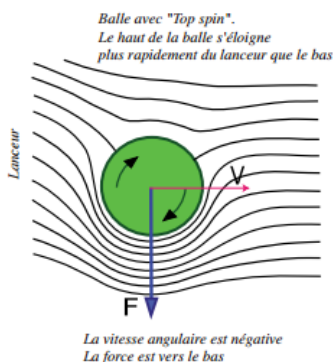
Équation 1

La force de frottement visqueux est donnée par l'équation 2. 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 ( $\rho_{\text{air}} = 1.2 \text{ kg/m}^3$ ), l'aire efficace de la balle de la balle ( $A = (\pi R_b^2)$ ) ainsi que  $C_v = 0.5$ .  $\vec{v}_b$  correspond à la vitesse de la balle. Cette équation est utilisée dans la situation 2 et la situation 3.

$$\vec{F}_v(\vec{v}_b(t)) = -\frac{\rho C_v A}{2} |\vec{v}_b(t)| \vec{v}_b(t)$$

Équation 2

L'effet de Magnus est donnée par l'équation 3. Elle est engendrée quand un solide en rotation (la



balle) se déplaçant dans un fluide (l'air). Avec le frottement visqueux, son effet va changer la vitesse du courant d'air entourant la balle. Cette perturbation sera proportionnelle à la vitesse de la balle par rapport à l'air. Comme la balle est en rotation, l'effet Magnus sera dissymétrique. Cet effet est causé par le ralentissement (accélération) du fluide en contact avec la balle dû au mouvement de rotation et qui en plus est en opposition ou en combinaisons avec le mouvement du fluide. La valeur de  $C_M$  est égale à 0.29 : elle nous a été fournie par l'énoncé du devoir.  $\rho$  correspond à la masse volumique de l'air : c'est la même qu'à

l'équation 2. Le rayon  $R_b$ , la vitesse angulaire  $\vec{\omega}_b$  ainsi que  $\vec{v}_b$  correspondent à ceux de la balle. Cette équation est utilisée dans la situation 3 uniquement.

$$\vec{F}_M(\vec{v}_b(t), \vec{\omega}_b(t)) = 4\pi C_M \rho R_b^3 (\vec{\omega}_b(t) \times \vec{v}_b(t))$$

*Équation 3*

Pour isoler l'accélération de la balle, 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, la balle aura un vecteur d'accélération  $a$  dans la même direction.

$$\vec{F} = m\vec{a}$$

*Équation 4*

### Équations pour déterminer l'arrêt de la simulation (collision)

Afin de déterminer l'arrêt de la simulation, nous avons opté pour une formule simple de détection de collision :

*Il y a collision lorsqu'au moins une des composantes de la balle est en intersection avec un des plans.*

Dans la situation, nous avons défini 3 plans : le filet, la table et le sol. Ainsi, lorsque nous détectons qu'une des extrémités de la balle était contenue dans les bornes du plan étudié, on savait que nous avions obtenu une collision, et donc une condition d'arrêt de la simulation.

Le pseudocode est le suivant :

Si bornes $X_{\min}$  > cm $X$  + rayon || bornes $X_{\max}$  < cm $X$  – rayon

Alors collision

Si bornes $Y_{\min}$  > cm $Y$  + rayon || bornes $Y_{\max}$  < cm $Y$  – rayon

Alors collision

Si bornes $Z_{\min}$  > cm $Z$  + rayon || bornes $Z_{\max}$  < cm $Z$  – rayon

Alors collision

Sinon, pas de collision

## 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.

- Procédure de résolution par la méthode de Runge-Kutta en MATLAB.

```
function qs=SEDRK4t0(q0,t0,DeltaT,g)
% Solution equations differentielles par methode de RK4
% Equation a resoudre : dq/dt=g(q,t)
% avec
%   qs      : solution [q(to+DeltaT)]
%   q0      : conditions initiales [q(t0)]
%   DeltaT  : intervalle de temps
%   g       : membre de droite de ED.
%           C'est un m-file de matlab
%           qui retourne la valeur de g au temps choisi
k1=feval(g,q0,t0);
k2=feval(g,q0+k1*DeltaT/2,t0+DeltaT/2);
k3=feval(g,q0+k2*DeltaT/2,t0+DeltaT/2);
k4=feval(g,q0+k3*DeltaT,t0+DeltaT);
qs=q0+DeltaT*(k1+2*k2+2*k3+k4)/6;
```

Notre fonction « SEDRK4 » appelle ainsi Runge-Kutta d'ordre 4. Elle prend en paramètre une fonction (g1, g2 ou g3). Celle-ci donne les vecteurs d'accélération et de vitesse en fonction du temps. « g1 » correspond l'équation ne considérant uniquement que la gravité pour le calcul de la vitesse et de l'accélération. « g2 » ajoute au calcul précédent le frottement visqueux. Enfin, « g3 » ajoute aussi le calcul de l'effet de Magnus aux forces prises en compte précédemment. Pour trouver l'accélération de la balle, 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 la balle. Pour le DeltaT, on prend le temps écoulé auquel on rajoute une variation de temps après une itération (le pas). La solution qs, sera un tableau avec la vitesse et les positions finales x, y et z de la balle.

## Intervalles de temps $\Delta t$ choisis pour la résolution et justification

Après quelques essais et quelques ajustements suite à ces derniers, nous avons finalement opté pour un  $\Delta t = 0.00001s$ . Cette valeur nous permettait d'avoir un moins haut risque de manquer un enregistrement de collisions tout en gardant une précision adéquate pour la position finale.

## Présentation et analyse des résultats

**Tableau 1 : Conditions initiales pour les quatre coups à simuler**

| Essai | rbi (m)<br>Position initiale du cm | vbi (m/s)<br>Vitesse initiale du cm | wbi (rad/s)<br>Vitesse angulaire |
|-------|------------------------------------|-------------------------------------|----------------------------------|
| 1     | (0,00 0,50 1,10)                   | (4,00 0,00 0,80)                    | (0,00 – 70,00 0,00)              |
| 2     | (0,00 0,40 1,14)                   | (10,00 1,00 0,20)                   | (0,00 100,00 – 50,00)            |
| 3     | (2,74 0,50 1,14)                   | (–5,00 0,00 0,20)                   | (0,00 100,00 0,00)               |
| 4     | (0,00 0,30 1,00)                   | (10,00 – 2,00 0,20)                 | (0,00 10,00 – 100,00)            |

**Tableau 2 : Résultats des simulations**

| Option   | Essai | Coup | tf (s)<br>Temps de la simulation | rbf (m)<br>Position finale du cm | vbf (m/s)<br>Vitesse finale du cm |
|--|-------|------|----------------------------------|----------------------------------|-----------------------------------|
| 1 : Force gravitationnelle seulement                           | 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     | 3    | 0,00306                          | (4,6971 -0,6394 0,0129)          | (10,0 -2,00 -4,40)                |
| 2 : Force gravitationnelle et force visqueuse                  | 1     | 1    | 0,00266                          | (1,2956 0,5000 0,7777)           | (3,32 0,00 -2,51)                 |
|  | 2     | 0    | 0,00248                          | (2,5725 0,6573 0,7751)           | (6,99 0,70 -2,43)                 |
|  | 3     | 2    | 0,00244                          | (1,3824 0,5000 0,7834)           | (-4,12 0,00 -2,50)                |
|  | 4     | 2    | 0,00172                          | (1,3523 0,0295 0,9251)           | (8,28 -1,66 -1,17)                |
| 3 : Force gravitationnelle, force visqueuse et force de Magnus | 1     | 0    | 0,00304                          | (1,6701 0,5000 0,7779)           | (3,43 0,00 -2,11)                 |
|  | 2     | 2    | 0,00175                          | (1,3889 0,4743 0,9281)           | (8,04 0,03 -2,80)                 |
|  | 3     | 0    | 0,00313                          | (0,5663 0,5000 0,7792)           | (-4,11 0,00 -1,67)                |
|  | 4     | 2    | 0,00175                          | (1,3605 -0,1078 0,9060)          | (7,75 -3,20 -1,37)                |

## 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).

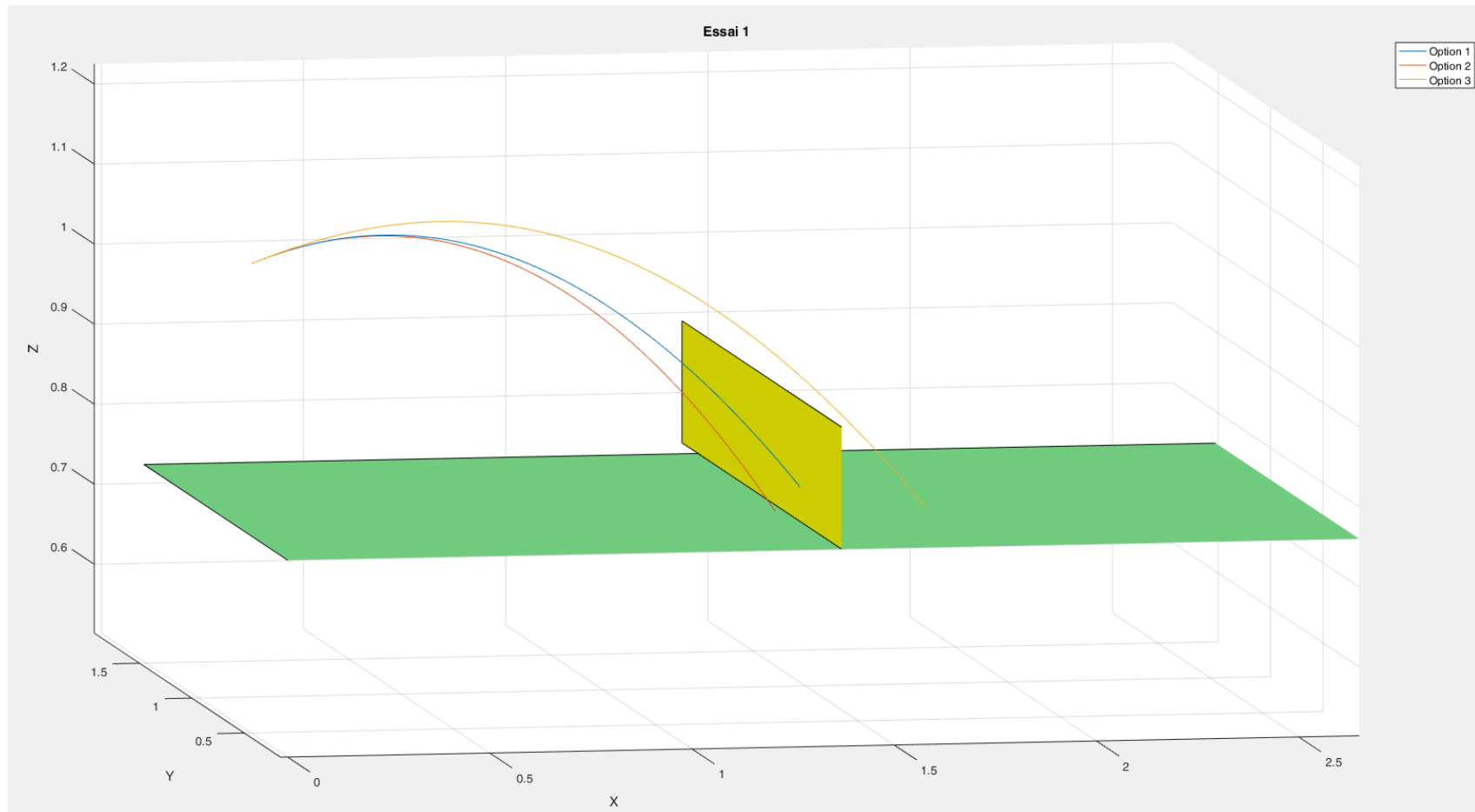


Figure 1 : Représentation graphique des coups de chacune des options lors de l'essai 1.



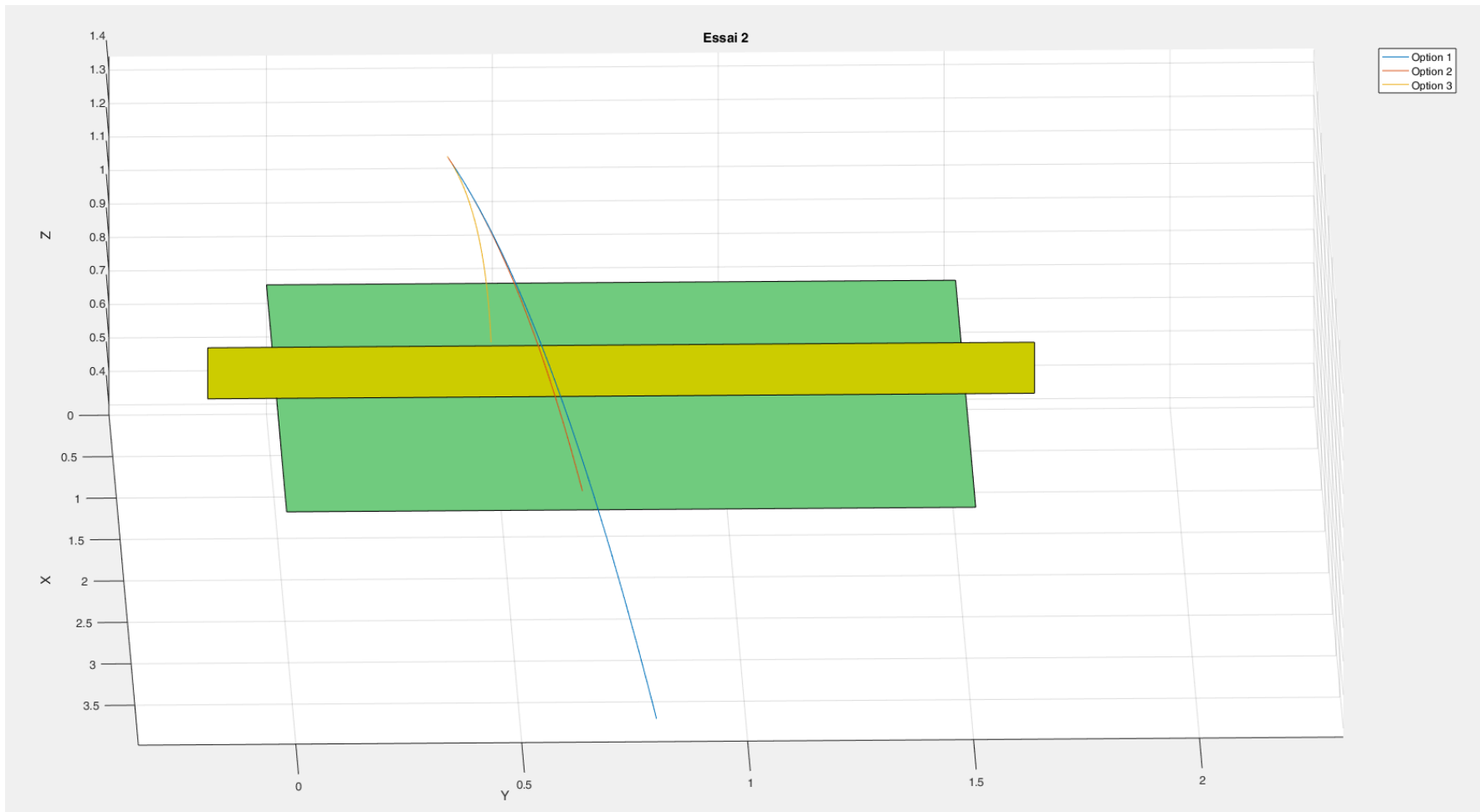


Figure 2 : Représentation graphique des coups de chacune des options lors de l'essai 2.

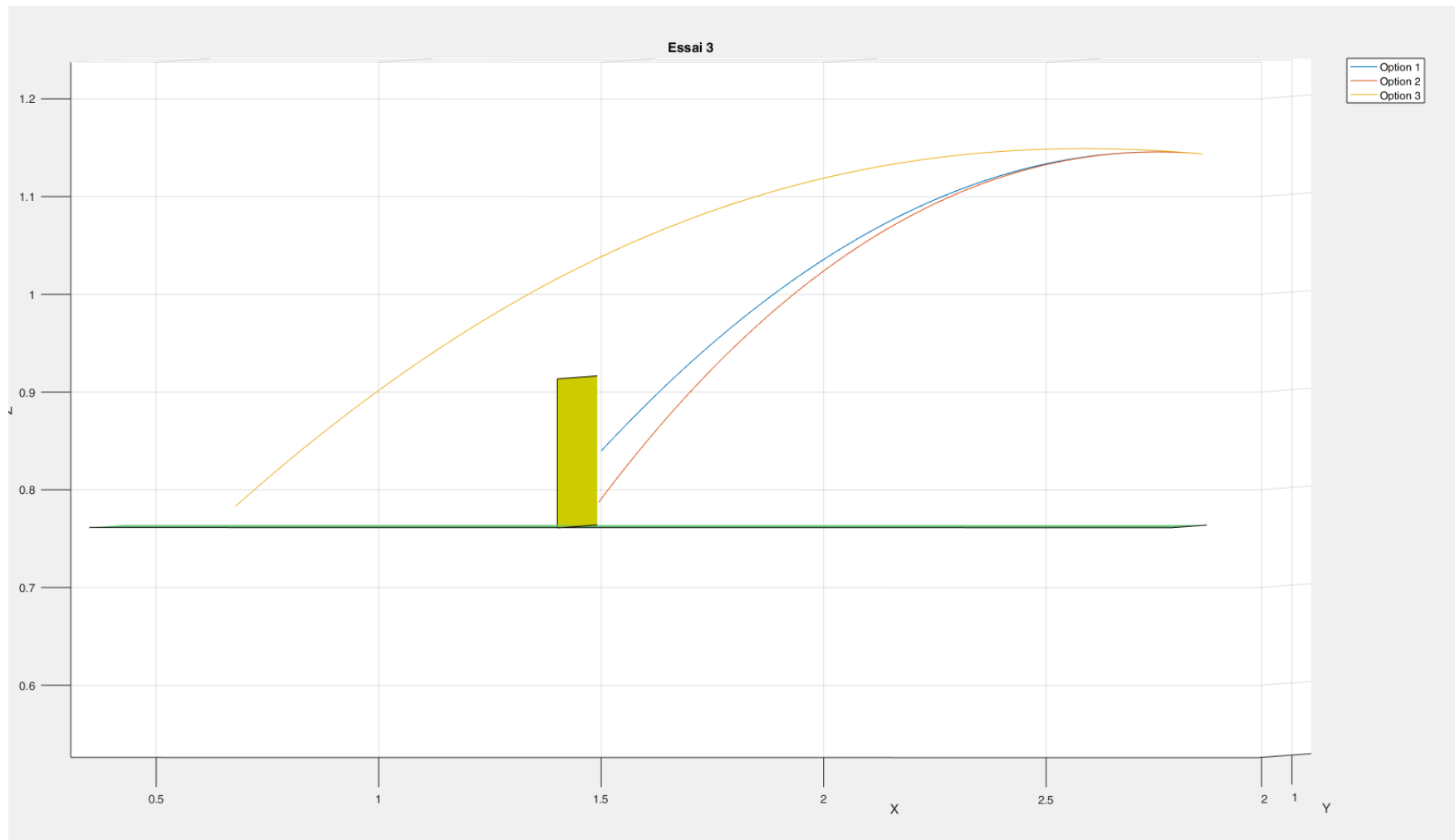


Figure 3 : Représentation graphique des coups de chacune des options lors de l'essai 3.

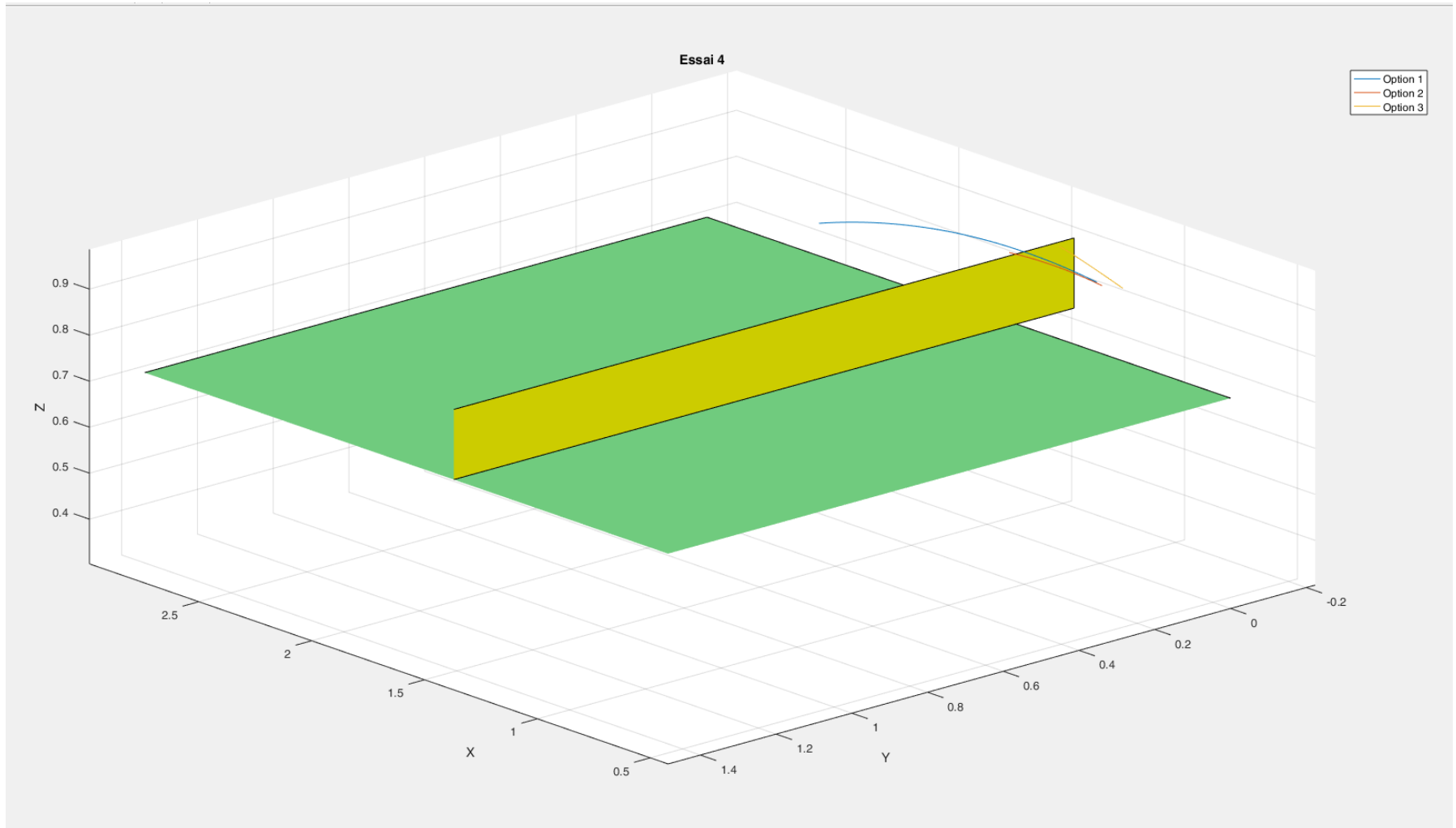


Figure 4 : Représentation graphique des coups de chacune des options lors de l'essai 4.

### Option 1 : Force gravitationnelle seulement

Les résultats obtenus dans l'option 1 représentent les coups d'une balle qui ne tient compte que de la force gravitationnelle. Elle suit donc une trajectoire qui subit une accélération vers l'axe des  $z$  négatif. De ce fait, la seule composante dont la vitesse a diminuée est celle en  $z$ . En effet, celles en  $x$  et en  $y$  demeurent constantes. C'est ce qui s'est en effet arrivé durant notre simulation. Il n'y a eu que la vitesse en  $z$  qui s'est retrouvée avec une vitesse finale négative due à l'accélération. Visuellement, on observe l'effet de gravité qui affecte la trajectoire de la balle en lui donnant la forme d'une parabole. Les essais 1 et 3 montrent que la balle s'est arrêtée au filet. Pour l'essai 2, la balle a atterri trop loin de la table. Enfin, l'essai 4 a traversé le filet et s'est arrêté sur la table.

### Option 2 : Force gravitationnelle et force visqueuse

L'option 2 prenait dans ses résultats, en plus de la gravité, la force de frottement visqueux causée par l'air. Le frottement devait influencer surtout la vitesse de la composante en  $x$ , car la balle se déplace surtout vers cette direction. On voit en effet que c'est en  $x$  qu'elle va le plus vite. Puisque l'équation de viscosité prend comme variable le vecteur de vitesse et sa norme, la composante qui subira le plus de frottement devrait être celle en  $x$ . Les autres devraient également diminuer de manière moins significative. La balle de ping-pong se rendrait ainsi moins loin sur le plan que pour l'option 1. Sa trajectoire serait la même. Néanmoins, la diminution de vitesse en  $x$  apportera un changement au déplacement en  $z$ , car l'effet de la gravité aura dans l'option 2 davantage d'impact. C'est en effet ce qui est arrivé : tous les essais se sont rendus moins loin par rapport à l'option 1 et la composante qui est la plus affectée est bien celle en  $x$ . L'essai 1 ne traverse pas le filet. L'essai 2 traverse le filet et atterrit sur la table. Enfin, les essais 3 et 4 ont atterri au filet.

### Option 3 : Force gravitationnelle, force visqueuse et force de Magnus

Cette dernière option a introduit l'effet de Magnus dans les résultats. Un effet attendu serait un changement dans la trajectoire. Cela affecte le frottement visqueux en créant une zone de haute pression en arrière de la balle (par rapport à la trajectoire) et une zone de basse pression en avant de la balle. Cette dernière crée un effet de vide ce qui fait courber la trajectoire de la balle.

Dans l'essai 1, la vitesse angulaire crée un *backspin* ce qui fait en sorte que la balle se rend plus loin que les autres options.

Dans l'essai 2, la vitesse angulaire crée une force de Magnus faisant en sorte que la balle va moins loin que les autres options (à cause de la composante  $z$  de la vitesse angulaire) et plus à droite que les autres options (à cause de la composante en  $y$  de la vitesse angulaire). Nous avons utilisé la règle de la main droite pour nous aider à visualiser ceci.

Dans l'essai 3, la balle vient de l'autre côté : cela fait en sorte qu'une valeur de composante plus petite indique une distance parcourue plus loin. Ici, nous avons encore un *backspin* : ainsi, nous nous attendions à ce que l'option 3 parcoure une plus grande distance que les autres options. Cela se reflète au niveau des graphiques obtenus ainsi que les positions finales obtenues.

Dans l'essai 4, on s'attend à ce que la balle courbe plus vers la droite que l'essai 2. En effet, l'addition vectorielle des deux composantes crée un vecteur qui pointe de manière plus prononcée vers le bas à droite en raison de la composante en  $z$  plus grande. Nous ne pouvons pas analyser la distance parcourue en raison de la collision avec le filet.

### 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  $\Delta t$  d'enregistrement des collisions. Cela aurait offert une meilleure estimation de la position.

## Conclusion

Après avoir eu un notre baptême du feu, suite au premier devoir, nous étions encore une fois en territoire inconnu : il fallait apprendre de nouvelles fonctions Matlab comme l'affichage de courbes. De plus, il était nécessaire de comprendre comment utiliser la méthode de Runge-Kutta. Cette tâche se révélait plus difficile que prévu puisque nous n'avions pas suivi le cours MTH1115 – Équations différentielles. En effet, ce cours n'est plus obligatoire à l'obtention du baccalauréat en génie logiciel. Finalement, nous avons réalisé qu'un exemple applicatif de la méthode se trouvait dans la boîte à outils du cours, alors cela nous a beaucoup aidés à comprendre la méthode.

Vérifier la validité des calculs était une tâche difficile. En effet, il y avait des calculs trop compliqués pour être faits à la main comme ceux de la méthode de Runge-Kutta. Cependant, nous avons fait des calculs pour l'option 1, la plus facile à évaluer, pour s'assurer que nous comprenions bien la base du problème. Puisque la force gravitationnelle revient dans les trois options à évaluer, nous trouvions cette validation pertinente à faire.

Finalement, se réunir en équipe pour ce devoir était difficile puisque nous devions attraper les balles que les différents autres cours nous lançaient en mi-session. Cela ne nous a tout de même pas empêché de terminer ce devoir.