POLYTECHNIQUE MONTRÉAL

LE GÉNIE EN PREMIÈRE CLASSE

PHS4700

Physique pour les applications multimédia

Automne 2015

PAGE COUVERTURE **OBLIGATOIRE** POUR TOUS LES DEVOIRS

Numéro de devoir : 2

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 :				

Table des matières

1. Description du problème à résoudre (1 pt)

Le but de ce devoir est d'effectuer une simulation qui est d'apparence simple, mais qui nécessite l'utilisation de plusieurs concepts. Un de ses concepts est l'utilisation de méthode numérique pour résoudre des problèmes de simulations. Pour cet aspect, il faudra déterminer quelle méthode à utiliser ainsi que ses paramètres; ce choix va se faire en fonction de nos besoins de précisions que l'on détermine nécessaire. La compréhension seulement théorique ne sera pas suffisante, car il faudra l'appliquer dans un programme MATLAB. Un autre but de ce devoir est de simuler un objet en mouvement. Puis finalement, il faudra déterminer les moyens pour détecter la fin de la simulation.

Plus précisément, le contexte de ce devoir est de simuler un lancer de baseball. On doit considérer la trajectoire de la balle, mais aussi les conditions pour déterminer s'il s'agit d'une prise ou d'une balle (si elle atteint une cible imaginaire). Trois types de simulations sont demandés, chacune d'elle intégrant une difficulté supplémentaire. Le premier lancer va seulement considérer la force gravitationnelle, une force qui va être constante tout le long de son vol. Le second lancer intègre une force de frottement, une force qui dépend de la vitesse de la balle. Puis le dernier lancer intègre la force Magnus, qui dépend de la vitesse angulaire de la balle.

2. Les équations importantes pour les simulations (3 pts)

2.1 Équations du mouvement à résoudre (1 pt)

Notre fonction reçoit comme paramètre la vitesse initiale de la balle. On connait les forces possibles s'appliquant sur la balle le long de son parcours et l'on cherche à obtenir la position de la balle à plusieurs temps t. On connait aussi la position initiale et la masse de la balle. On cherche donc des équations qui mettent en relation x0, m et F et qui permettent de déterminer x.

2.1.1 Équation pour un mouvement rectiligne uniformément accéléré

Dans le cas d'une accélération constante, ce qui est le cas avec l'option où seulement la gravité est à considérer, il n'est pas nécessaire d'utiliser des méthodes numériques puisque les intégrales sont faciles à résoudre analytiquement. Il existe trois équations pour les MRUA

Variable	**Signification**
x(t)	Position de l'objet au temps t
x_0	Position initiale de l'objet (à t = 0)
v_0	Vitesse initiale de l'objet (à t = 0)
t	Temps écoulé depuis le temps initial (t=0)
a	Accélération de l'objet (soit être constante)

Ce qu'on cherche	**Équations**
La position de l'objet au temps t	$x(t) = x_0 + v_0 t + \frac{1}{2}at^2.$

2.1.2 Équations pour les accélérations non constantes

Pour les options de lancers 2 et 3, l'accélération n'est pas constante puisque la force appliquée sur la balle dépend de la vitesse de celle-ci.

Variable	**Signification**
v(t)	Vitesse au temps t
v(t_0)	Vitesse initiale
a(t)	Accélération au temps t
r(t)	Position au temps t

Ce qu'on cherche	**Équations**
La vitesse aux temps t	$\vec{v}(t) = \vec{v}(t_0) + \int_{t_0}^{t} \vec{a}(t')\,dt'$
La position aux temps t	$\vec{r}(t) = \vec{r}(t_0) + \int_{t_0}^{t} \vec{v}(t')\,dt'$

Or ici l'accélération ne dépend pas du temps, il faudra donc chercher d'autres équations.

2.1.3 Équations pour trouver la vitesse en fonction de la force

En premier lieu, il faudra être en mesure de transformer la force, chose qu'on connait, en une accélération, quelque chose dont on a besoin.

Variable	**Signification**
a(t)	Accélération au temps t
$F_{net}(t)$	Force net au temps t
m	Masse (kg)

Ce qu'on cherche	**Équations**
L'accélération au temps t	$a(t) = F_{net}(t)m$

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

2.1.4 Équations de la force dans notre devoir

Dans notre devoir trois types de forces vont s'appliquer à notre balle.

Force gravitationnelle

Variable	**Signification**
F_g	Force gravitationnelle
m_b	Masse de la balle

Ce qu'on cherche	**Équations**
Force gravitationnelle	$\vec{F_g} = m_b(0, 0, -9.8)^T$

Force de frottement

Variable	**Signification**
F_v	Force de frottement
d_b	Diamètre de la balle
p_{air}	Densité de l'air
C_v	Coefficient de frottement visqueux de l'air
v	Vitesse de la balle

Ce qu'on cherche	**Équations**
Force de frottement	$\vec{F}_v = -\frac{\pi d_b^2}{8}\rho_{\text{air}} C_v \lvert\vec{v}\rvert\vec{v}$

Force de Magnus

Variable	**Signification**
F_m	Force de Magnus
d_b	Diamètre de la balle
p_{air}	Densité de l'air
C_m	Coefficient de Magnus
ω	Vitesse angulaire

Ce qu'on cherche	**Équations**
Force de Magnus	$\vec{F}_M = \frac{\pi d_b^2}{8}\rho_{\text{air}} C_M(\lvert\vec{\omega}\rvert)\frac{\lvert\vec{v}\rvert}{\lvert\vec{\omega}\rvert}\vec{\omega} \times \vec{v}\ [\mathbf{N}]$

2.2 Équations qui contrôlent l'arrêt de la simulation (2 pts)

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 confondant. Dans ce devoir, il existe deux conditions d'arrêts. La première détermine si le lancer est une prise, c'est-à-dire que la balle est entièrement à l'intérieur de la zone de prise, un carré imaginaire au-dessus du marbre d'une taille prédéfinie. La deuxième condition vérifie si le lancer est une balle, c'est-à-dire, lorsque la balle touche le sol avant d'atteindre la zone de prise.

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	**Signification**
z_i	Position de la balle au temps i, par rapport à l'axe des z
r	Rayon de la balle
z_s	Position du sol, par rapport à l'axe des z
y_{cmin}	Position en y du côté gauche de la cible
v_i	Position de la balle au temps i, par rapport à l'axe des y
y_{cmax}	Position en ydu côté droit de la cible
z_{cmin}	Position en z du côté en bas de la cible
z_{cmax}	Position en z du côté en haut de la cible
x_i	Position de la balle au temps i, par rapport à l'axe des x
y_i	Position de la balle au temps i, par rapport à l'axe des y
x_c	Position de la cible par rapport à l'axe des x

Ce qu'on cherche	**Équations**
La balle traverse le sol	$z_i - 1 + r \geq z_s \geq z_i + r$
La balle traverse la position de la cible	$y_{cmin} \leq y_i + r \leq y_{cmax}$ $z_{cmin} \leq z_i + r \leq z_{cmax}$ $x_i - 1 + r \leq x_c \leq x_i + r$

3. Méthodes de résolution des équations

3.1 Équations numériques à résoudre

Puisque ce serait relativement difficile et long de résoudre analytiquement ces équations, nous allons employer des méthodes numériques qui consistent à trouver les valeurs de q(tn) basé sur la valeur de q(tn-1).

Variable	**Signification**
$\vec{q}_0$	Condition initiale (ici la position initiale de la balle)
$\vec{g}\left[\vec{q}(t), t\right]$	Différentielle de q au temps t. (ici la vitesse de la balle au temps t)
$\vec{q}(t_n)$	Valeur de q au temps t.
k_i	Approximation de $\vec{g}\left[\vec{q}(t), t\right]$ pour la méthode Runge-Kutta
$O(\Delta t)$	Erreur

Ce qu'on cherche	**Équations**
Méthode de Euler	$\vec{q}(t_n) = \vec{q}(t_{n-1}) + \vec{g}\left[\vec{q}(t_{n-1}), t_{n-1}\right]\Delta t + O(\Delta t^2)$
Méthode de Runge-Kutta	$\vec{q}(t_n) = \vec{q}(t_{n-1}) + \frac{1}{6}\left[\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4\right] + O(\Delta t^5)$

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

Variable	**Formule**
Vecteur ayant comme composant la vitesse et la position	$\vec{q}(t) = (v_x(t), v_y(t), v_z(t), x(t), y(t), z(t))$
Vecteur ayant comme composant l'accélération et la vitesse.	$\vec{g}(t) = (a_x[\vec{q}(t), t], a_y[\vec{q}(t), t], a_z[\vec{q}(t), t], v_x(t), v_y(t), v_z($

Donc on part du principe qu'en connaissant la vitesse et la position au temps t, nous sommes capables de trouver l'accélération à ce temps (selon 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.

Lors des simulations pour ce devoir, il est requis que pour chaque temps t fourni, il doit y avoir une précision de ±1mm pour la position en x, y ,z.

Dans le but d'avoir une meilleure précision, nous allons utiliser la méthode de Runge-Kutta plutôt que de la méthode d'Euler. Cette dernière a une erreur globale proportionnelle à delta-t et est donc moins précise du (delta-t)^4 de Runge-Kutta.

3.2 Description des mesures de vérifications de la precision (2 pts)

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

3.2.1 Contrainte sur la réponse finale de ±1mm

Variable	**Signification**
$\vec{q}(t)$	Réponse exacte de q(t)
$\vec{q}_a(t, \Delta t_m)$	Estimation de q(t), avec un pas de temps spécifié
$c(\Delta t_m)^4$	Erreur de l'estimation de q(t) utilisant le pas de temps spécifié

Ce qu'on cherche	**Équations**
Première estimation	$\vec{q}(t) = \vec{q}_a(t, \Delta t_m) + c(\Delta t_m)^4$
Deuxième estimation avec un pas de temps plus petit	$\vec{q}(t) = \vec{q}_b(t, \Delta t_{m+1}) + c(\Delta t_{m+1})^4 = \vec{q}_b(t, \Delta t_{m+1}) + c(\Delta t_m/2)^4$
Estimation de la constante c	$c = \frac{16}{15(\Delta t_m)^4}\left(\vec{q}_b(t, \Delta t_{m+1}) - \vec{q}_a(t, \Delta t_m)\right)$
Vecteur de la solution moyenne	$avg_{sol} = (q_b(t, t_m + 1) + q_a(t, t_m))/2$
Vecteur d'erreur de la solution	$err_{sol} = q_b(t, t_m + 1) - q_a(t, t_m)$
Évaluation de l'erreur relative	$Max_{Err} = \max(abs\left(\frac{err_{sol}}{avg_{sol}}\right))$

Donc en effectuant 2 simulations avec des pas de temps différents nous sommes capables d'estimer l'erreur et donc la précision de notre simulation. On calcule d'abord la solution moyenne en se basant sur les 2 premières simulations, puis l'erreur à chaque point. Ensuite, on estime que notre précision est égale à la plus grande erreur relative. Le principe est donc de rendre le pas de temps de plus en plus petit jusqu'à ce qu'on obtienne une erreur de moins de 1mm. L'avantage avec cette méthode est d'avoir un pas de temps constant. Cela nous évite de calculer à chaque pas, ce qui simplifie l'écriture du code. Le désavantage est que l'on doit souvent effectuer plusieurs fois la simulation, ce qui est couteux.

3.2.2 Contrainte pour ne pas manquer condition de terminaison

Comme il est possible de commencer avec des pas de temps initialement grand, nous devons avoir un mécanisme pour nous assurer que l'on ne passe pas tout droit autour des zones de terminaisons, c'est-à-dire la cible ou le sol. Le principe de notre solution est que lorsque l'on approche d'une de ses zones, nous diminuons le pas de temps pour faire en sorte de ne manquer pas la zone. Il y a deux problèmes à résoudre pour notre solution. Premièrement, il faut trouver ce que l'on veut dire par "approche de la zone" et de combien nous devons diminuer le pas de temps. On commence à diminuer le pas de temps si avec sa position, sa vitesse et le pas de temps courant, la balle se retrouverait de l'autre côté de la cible. Nous allons donc diminuer le pas de temps de façon à ce qu'avec sa vitesse courante, elle ne se déplace pas plus de son diamètre, de cette façon, la balle ne va jamais passer tout droit.

3.3 Justification du pas de temps (1 pt)

Nous allons donc avoir un pas de temps initial qui se raffine après chaque itération de simulation jusqu'à obtenir la précision désirée et un pas de temps d'approche, qui va être utilisé lorsque la balle va s'approcher des zones critiques (nous avons déjà abordé ce pas de temps dans la section 3.2.2). Pour ce qui est du pas de temps initial, il est important qu'il ne soit ni trop grand, car il faudra alors le raffiner trop souvent, ce qui est couteux et ni trop petit, car il exécuterait plus d'itérations et donc de calcul que nécessaire. Après plusieurs tests nous avons déterminé que le meilleur pas de temps initial idéal pour notre contexte est de 0,02s. Toutefois, même si nous utilisions un pas de temps plus élevé, notre programme se chargerait de le diminuer jusqu'au point où nous respectons la contrainte de précision (pour les options 2 et 3 qui sont résolues numériquement).

4. Description du logiciel MATLAB (4 pts)

Dans cette section, nous décrivons le fonctionnement de notre programme.

Contrairement au premier devoir, nous avons décidé de ne pas utiliser le paradigme de programmation orientée-objet puisque nous ne croyons pas que cela est pertinent à la résolution du problème. Nous avons donc décidé de diviser le programme en plusieurs fonctions, chacune contenant une partie de la fonctionnalité du programme.

test.m est la partie principale du programme. Dans cette partie, nous appelons la fonction Devoir2 nous permettant de calculer la position de la balle selon les trois tirs et selon les différents modèles utilisés (lignes 1 à 6, 53 à 56 et 153 à 155). Les lignes 8 à 48, 108 à 149 ainsi que 157 à 198 permettent l'affichage des résultats calculés par la fonction Devoir2.

La fonction Devoir2 calcule les résultats d'un tir selon l'option utilisée. Les variables initiales du problème sont définies aux lignes 2 à 11. Si l'on choisit de calculer la position de la balle en prenant seulement en compte l'accélération gravitationnelle, les calculs sont faits directement dans la fonction. Lors de ces calculs, nous utilisons FixDTZoneCrit ainsi que checkEnd. FixDTZoneCrit nous permet de modifier le pas utilisé lors du calcul de la position et checkEnd vérifie si la position de la balle vérifie une des conditions signifiant l'arrêt de la simulation. Pour les autres options, nous faisons appel à la fonction SOLRK4C pour les calculs en utilisant les autres forces s'appliquant sur la balle. Dans l'appel de fonction, nous spécifions la fonction g à utiliser lors de la méthode de Runge-Kutta.

La fonction FixDTZoneCrit permet de modifier le pas de temps lorsque la balle se rapproche du sol ou bien de la cible. On vérifie d'abord si la balle est entre la position de la cible ou du sol et deux fois plus loin que la distance qu'elle peut traverser en deux fois son pas de temps de base. Si oui, nous modifions le pas de temps courant afin que la distance parcourue avant la prochaine vérification soit plus petite que le diamètre de la balle, nous assurant que nous n'allons pas dépasser la cible.

La fonction checkEnd permet de détecter la condition d'arrêt de la simulation. Les lignes 2 à 5 sont une définition des différentes conditions de base du problème. Nous vérifions ensuite les conditions d'arrêt en implémentant les équations définies à la section 2.2.

La fonction SOLRK4C contient la méthode de résolution utilisée pour les options 2 et 3. Elle reçoit en paramètre la fonction utilisée pour calculer l'accélération et la vitesse de la balle en fonction de sa vitesse et position précédente (soit les fonctions Option1 et Option2). SOLRK4C appelle tout d'abord SOLRK4 pour avoir une première simulation de la trajectoire de la balle avec le pas de temps par défaut. Ensuite, SOLRK4 est rappelée avec un pas deux fois plus petit jusqu'à ce que deux simulations consécutives aient une erreur relative respectant nos contraintes. Par la suite, puisqu'il est possible qu'une collision avec la zone des prises n'ait pas été détectée, si le pas de temps était trop grand, une seconde simulation est effectuée à partir du point juste avant de dépasse le plan x = 0. Cette seconde simulation utilise un pas de temps bien plus petit pour faire en sorte qu'entre chaque point la balle se déplace d'au plus la moitié de son rayon. Si cette nouvelle simulation détecte une prise, ses points sont ajoutés à la liste de la première simulation et on retourne le résultat. Sinon, on effectue une troisième simulation similaire à la deuxième à partir du dernier point avant que la balle touche le sol, pour avoir une bonne précision sur la position et la vitesse de la bille à la collision.

La fonction SOLRK4 est essentiellement une boucle appelant la fonction SEDRK4 avec la fonctiong, permettant le calcul de la prochaine position de la balle et vérifiant si la balle atteint une condition d'arrêt avec la fonction checkEnd. Lorsqu'une condition d'arrêt est atteinte, la boucle s'arrête et on obtient les positions de la balle dans le temps.

La fonction SEDRK4 est l'implémentation de la méthode de Runge-Kutta d'ordre 4. En utilisant cette méthode, nous pouvons obtenir la position de la balle selon la fonction g et nous pouvons nous assurer d'obtenir la précision nécessaire au problème. La fonction appelle feval, permettant d'évaluer la fonction g.

Les deux différentes fonctions g possibles sont Option2 et Option3. La première fonction g calcule l'accélération due à la force visqueuse puis y rajoute l'accélération gravitationnelle de la balle. La deuxième fonction g calcul l'accélération de la force de Magnus ainsi que la force visqueuse et ajoute l'accélération de ces deux forces à l'accélération gravitationnelle.

5. Présentation et analyse des résultats (3 pts)

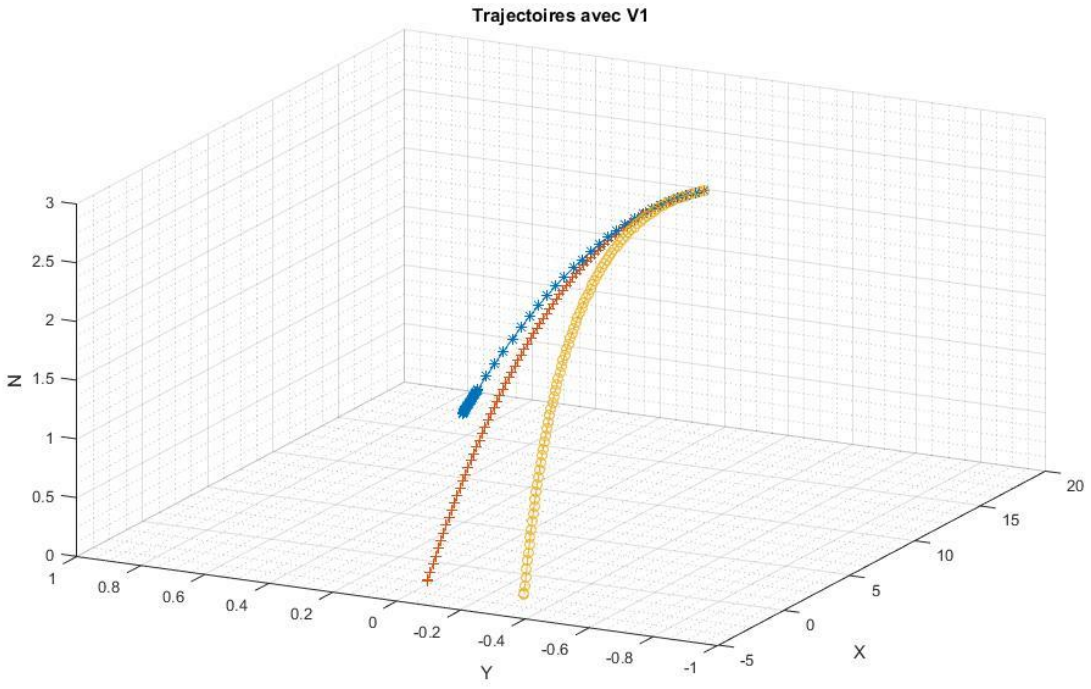
5.1 Tir #1

vi = (-120, 0, 4.55)km/h

Tableau des résultats:

	Résultat	Temps d'arrêt (sec)	Vitesse (m/s)	Position (m)
Option #1	Prise	0.55	(-33.33, 0, -4.15)	(-0.017, 0, 1.29)
Option #2	Balle	0.83	(-19.81, 0, -5.73)	(-2.71, 0, 0.032)
Option #3	Balle	0.83	(-19.81, -0.57, -5.73)	(-2.71, -0.30, 0.032)

Graphique



Analyse

En ne considérant que la force gravitationnelle, nous pouvons voir que la balle garde la vitesse initiale en X (33.3 m/s) et gagne une vitesse en Y due a la gravité. La balle a une vitesse nulle en Z.

Pour ce qui est de l'ajout de la force de frottement visqueuse avec l'air, nous pouvons observer que la balle ralentit beaucoup (dû à cette force) selon l'axe des X. Avec ce ralentissement, la balle subit plus longtemps la force gravitationnelle qu'avec l'option 1. Cela explique la plus grande vitesse selon l'axe des Y. Encore une fois, nous n'obtenons une vitesse nulle en Z.

Pour l'option 3, l'ajout de la force de Magnus influence encore une fois le comportement de la balle. Cette force s'ajoute aux deux autres déjà appliqués sur la balle. La balle tournant autour de l'axe des X, n'influence que la vitesse en Z. C'est ce que nous pouvons observer dans les résultats plus haut. La balle ne prend pas plus de temps à parcourir que l'option 2 (pas d'influence de la vitesse en X) et elle atteint le sol au même moment (pas de changement de vitesse en Y).

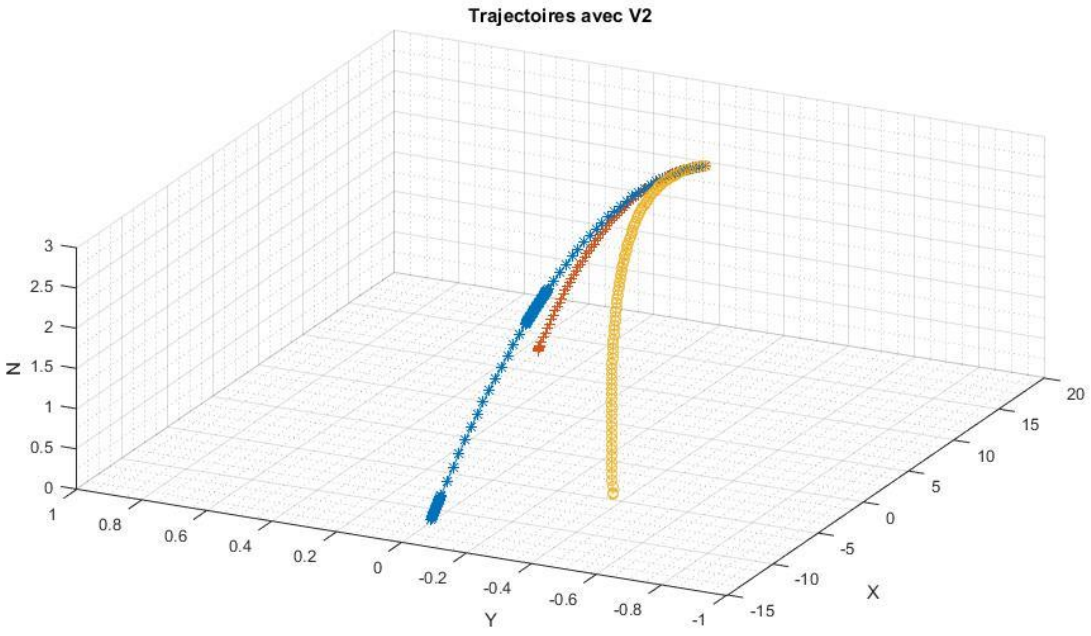
5.2 Tir #2

vi = (-120, 0, 7.79)km/h

Tableau des résultats:

	Résultat	Temps d'arrêt (sec)	Vitesse (m/s)	Position (m)
Option #1	Balle	0.91	(-33.33, 0, -6.71)	(-11.77, 0, 0.036)
Option #2	Prise	0.698	(-21.23, 0, -4.22)	(0.006, 0, 1.23)
Option #3	Balle	0.933	(-18.86, -0.59, -5.91)	(-4.68, -0.36, 0.036)

Graphique



Analyse

Dans ce tir, nous ajoutons une plus grande vitesse de tir de balle dans l'axe des Y (le lanceur lance plus fort vers le haut). Les résultats devraient beaucoup ressembler le premier tir au niveau de l'influence des différentes forces, mais le résultat final pourrait facilement être différent.

Avec l'option 1, nous pouvons voir que l'augmentation de la force initiale en Y influence beaucoup le temps d'arrêt de la balle (celle-ci est lancée plus fort vers le haut), même si la vitesse en X ne change pas entre les deux tirs. Au niveau du résultat, l'augmentation de la force initiale en Y fait passer la balle au-dessus de la zone de frappe, causant une balle.

Avec l'ajout de la force de frottement, nous pouvons encore voir que la garde sa vitesse en X. Par contre, ici la balle est ralentie assez par la force de frottement pour atteindre la zone de tir et ainsi causer une Prise.

Avec l'ajout de la force de Magnus, nous pouvons encore observer que la vitesse de la balle augmente selon l'axe des Z. Avec l'ajout de la force en Y, la balle peut atteindre la zone de frappe, comparer au tir 1.

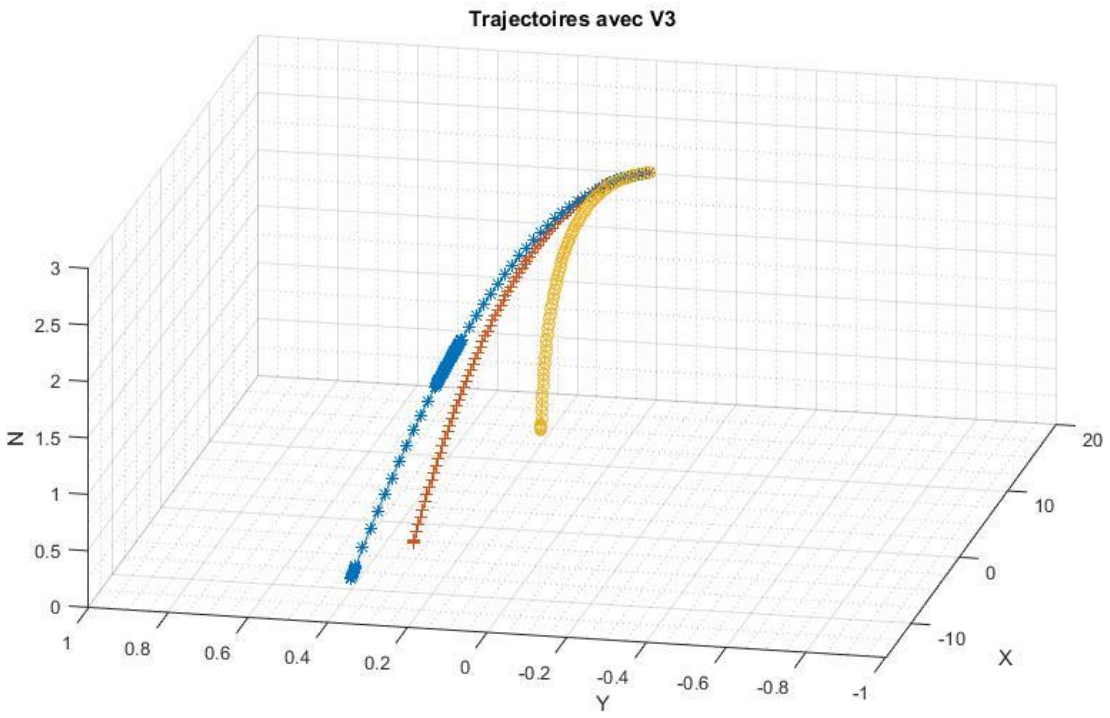
5.3 Tir #3

vi = (-120, 1.8, 5.63)km/h

Tableau des résultats:

	Résultat	Temps d'arrêt (sec)	Vitesse (m/s)	Position (m)
Option #1	Balle	0.828	(-33.33, 0.5, -6.54)	(-9.16, 0.41, 0.036)
Option #2	Balle	0.864	(-19.50, 0.29, -5.78)	(-3.35, 0.32, 0.036)
Option #3	Prise	0.698	(-21.23, -0.21, -4.60)	(0.006, 0.048, 0.895)

Graphique



Analyse

Dans ce tir, nous ajoutons une plus petite vitesse de tir de balle dans l'axe des Y, mais une vitesse en Z (le lancer est sur le côté).

Pour l'option 1, nous pouvons voir que l'ajout d'une vitesse initiale en Z influence la position finale et que la vitesse finale en X et Z est identique à la vitesse initiale (pas de modification par la gravité sur ces axes). Le lancer est ici aussi une balle comme il passe par dessus de la zone de frappe.

Pour l'option 2, nous pouvons voir que la force de friction influence la vitesse en X, Y et X, car le déplacement de la balle se fait selon ces 3 axes. L'ajout de la vitesse initiale en Z fait passer la balle hors de la zone de frappe.

Finalement, pour l'option 3, en ajoutant la force de Magnus, nous pouvons encore voir l'influence sur l'axe des Z. Avec l'ajout de la force initiale, la balle peut passer dans la zone de frappe.

6. Discussion des défis rencontrés

Avec le devoir #1, on a acquis une certaine compréhension du langage MATLAB, donc son utilisation cette fois-ci était plus facile. La difficulté de ce devoir concernait la résolution du problème lui-même, notamment avec les méthodes de résolution numérique et les conditions d'arrêts. L'utilisation des méthodes numériques demande une tout autre mentalité, car au lieu d'avoir des réponses exactes on a des estimations ce qui implique qu'on doit considérer l'erreur de l'estimation et donc trouver un mécanisme pour la contrôler. L'autre difficulté provient des considérations d'arrêts avec l'utilisation de méthodes discrètes, on devait penser à un mécanisme qui ferait en sorte qu'on ne manque pas la condition d'arrêt.