



PHS4700
Physique pour les applications multimédia

PAGE COUVERTURE **OBLIGATOIRE** POUR TOUS LES DEVOIRS

Numéro du groupe : 01

Numéro de l'équipe : 11

Numéro de devoir : 3


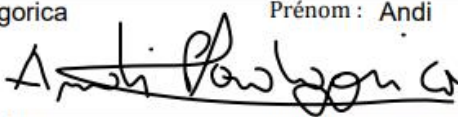
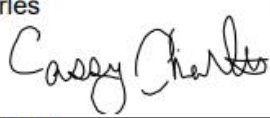
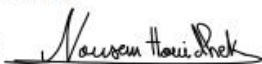
Nom: Yuan	Prénom: Ming Xiao	matricule: 1949477
Signature: 		
Nom: Podgorica	Prénom: Andi	matricule: 1955913
Signature: 		
Nom: Charles	Prénom: Cassy	matricule: 1947025
Signature: 		
Nom: Houdhek	Prénom: Noursen	matricule: 1949635
Signature: 		

Table des matières

Introduction	3
1. Théorie et équations.....	3
1.1 Méthodologie utilisée (Méthode des sphères)	3
1.2. Positions et vitesses ($r(t)$, $v(t)$, $\omega(t)$)	5
1.3. Coins du dé.....	5
1.4. Choix de Δt	6
1.5. Collision.....	6
1.5.1. Collision entre la sphère et la surface.....	6
1.5.2 Méthode des conditions initiales	6
1.5.2. Vitesses linéaires finales après la collision	7
1.5.3. Vitesses angulaires après la collision	8
1.5.4. Détection des collisions	9
2. Présentation et analyse des résultats.....	9
2.1 Simulation 1	10
2.2 Simulation 2	11
2.3 Simulation 3	11
2.3 Simulation 4	12
3. Conclusion.....	13

Introduction

Dans le devoir 3 du cours de PHS4700, Physique pour application multimédia, nous étudions les collisions. L'étude des collisions consiste à déterminer les forces qui sont impliquées ainsi que la résolution des équations du mouvement pour obtenir la trajectoire des solides durant la collision. Pour cela, il faut tenir en compte du déplacement linéaire et de rotation des solides. Aussi, il faut considérer le mouvement couplé des solides car chaque solide peut influencer le comportement des autres. Dans le chapitre 5 du cours, deux méthodes de traitements de collisions sont abordées, la méthode des forces ainsi que la méthode des conditions initiales.

La méthode des forces est l'analyse de la dynamique des forces de répulsion entre les objets impliqués dans la collision. Quant à la méthode des conditions initiales, cette dernière s'intéresse à l'avant et à l'après d'une collision et non à la collision en soi. Dans ce devoir, le but est de programmer une fonction Matlab (ou Octave) qui permet de simuler le lancement d'un dé sur le sol à partir d'une certaine hauteur. Le dé entre en contact avec le sol. Dans ce cas, la position et la vitesse du centre de masse sont déterminées ainsi que la vitesse angulaire après la collision. En bref, ce programme permet de tracer la trajectoire du centre de masse du dé lorsqu'il est lancé ainsi que son orientation finale. Afin de tracer la trajectoire du dé, nous utilisons la méthode des conditions initiales, car nous nous intéressons à l'avant et à l'après des collisions entre le dé et le sol. La simulation doit s'arrêter lorsque le dé n'a pas assez d'énergie pour changer le résultat du lancer.

À cet effet, nous avons utilisé des théories et des équations qui sont minutieusement élaborés à la section 1. Ensuite, la présentation de l'analyse des résultats obtenus pour les 4 différents lancers de dé est présentée dans la section 2. Enfin, la conclusion est abordée dans la section 3.

1. Théorie et équations

1.1 Méthodologie utilisée (Méthode des sphères)

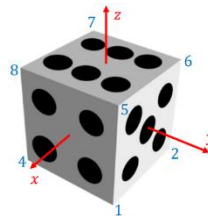


Figure 1: Visualisation de notre dé

Le référentiel du dé est choisi de façon à ce que le centre de masse se situe à son origine et que les faces numérotées de 1 à 6 du dé appartiennent aux plans suivants dans la figure 1. Pour débuter, nous englobons notre dé dans une sphère. Le rayon de la sphère est calculé à partir de la diagonale du dé (cube représenté dans la figure ci-dessous par le segment AG) qui correspond au diamètre de la sphère. En divisant cette diagonale par deux, le rayon est trouvé.

Sachant que notre $l = 0.04$ m, la diagonale de la sphère correspond à la distance du point 4 au point 6 dans la figure 1. Mais pour trouver cette distance, il est important de connaître la diagonale partant du point 4 au point 2. Avec le théorème de Pythagore, on a : $AG^2 = AC^2 + CG^2$

- 1) $Distance_{42} = \sqrt{Distance_{41}^2 + Distance_{12}^2} = \sqrt{l^2 + l^2}$
- 2) $Distance_{46} = \sqrt{Distance_{42}^2 + Distance_{26}^2} = \sqrt{Distance_{42}^2 + l^2}$
- 3) $Distance_{46} = \text{Diagonale}$
- 4) $\text{Rayon de la sphère} = \frac{Distance_{46}}{2}$

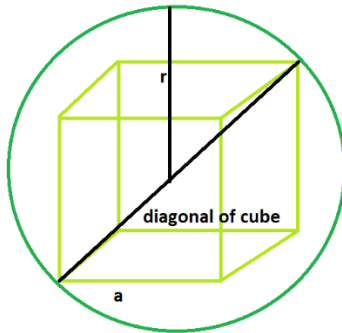


Figure 3: Sphère englobant le dé

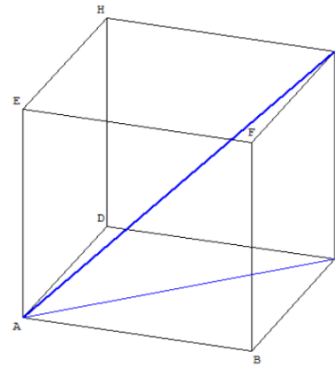


Figure 2: Diagonale du dé

Tant que le centre de masse de notre sphère est inférieur au rayon de la sphère, nous simulons la trajectoire de notre dé afin de déterminer la position $\mathbf{r}_{CM}(t)$, la vitesse $\mathbf{v}_{CM}(t)$ et la vitesse angulaire $\mathbf{w}_d(t)$ à chaque instant t . Cette partie nous permettra d'obtenir l'orientation et la position du dé lors de la collision.

De plus, lorsqu'une collision est détectée, il est nécessaire de connaître le point de contact du sommet avec la surface. Avec ce point et les vitesses lors de la collision, nous pouvons calculer les vitesses linéaires et angulaires générées après la collision tel qu'expliqué dans la méthode des conditions initiales. La simulation doit s'arrêter lorsque le dé n'a pas assez d'énergie pour changer le résultat du lancer soit :

$$\frac{1}{2}m\vec{v}^2 + \frac{1}{2}I\vec{\omega}^2 + mgz < \sqrt{2}mgl$$

Figure 4: Condition d'arrêt de la simulation

Ou $I = ml^2/6$ est le moment d'inertie du dé, m est la masse du dé, l est la longueur du côté et $g = 9.81\text{m/s}^2$.

1.2. Positions et vitesses ($r(t)$, $v(t)$, $\omega(t)$)

Pour simuler la trajectoire de notre dé, nous avons utilisé la résolution numérique Runge-Kutta d'ordre 4 en raison de sa précision. Cette méthode nous permet de trouver à chaque intervalle de temps Δt durant la simulation, la position $\mathbf{r}_{CM}(t)$, la vitesse $\mathbf{v}_{CM}(t)$ du centre de masse du dé avec la matrice de rotation initiale donnée.

L'équation à résoudre est tirée à partir des formules données en cours :

$$\vec{q}(t) = \begin{pmatrix} q_1(t) \\ q_2(t) \\ q_3(t) \\ q_4(t) \\ q_5(t) \\ q_6(t) \\ q_7(t) \\ q_8(t) \\ q_9(t) \\ q_{10}(t) \\ q_{11}(t) \\ q_{12}(t) \\ q_{13}(t) \\ q_{14}(t) \\ q_{15}(t) \\ q_{16}(t) \\ q_{17}(t) \\ q_{18}(t) \end{pmatrix} = \begin{pmatrix} v_{cx}(t) \\ v_{cy}(t) \\ v_{cz}(t) \\ x_c(t) \\ y_c(t) \\ z_c(t) \\ \omega_x(t) \\ \omega_y(t) \\ \omega_z(t) \\ R_{xx}(t) \\ R_{xy}(t) \\ R_{xz}(t) \\ R_{yx}(t) \\ R_{yy}(t) \\ R_{yz}(t) \\ R_{zx}(t) \\ R_{zy}(t) \\ R_{zz}(t) \end{pmatrix} = \begin{pmatrix} \vec{v}_c(t) \\ \vec{r}_c(t) \\ \vec{\omega}(t) \\ R(t) \\ \vec{\omega}R(t) \end{pmatrix} \quad \vec{g}(\vec{q}, t) = \begin{pmatrix} \vec{a}_c(t) \\ \vec{v}_c(t) \\ \vec{a}(t) \\ R(t) \\ \vec{\omega}R(t) \end{pmatrix}$$

La fonction Runge Kutta, pour tout $t > 0$, prend en paramètre l'intervalle de temps Δt , la dérivée de la matrice de rotation $\vec{\omega}R(t)$ et la fonction $\mathbf{g}(t)$ pour retourner $\mathbf{q}(t)$

$$= [rx(t) \ ry(t) \ rz(t) \ vx(t) \ vy(t) \ vz(t) \ \omega x(t) \ \omega y(t) \ Rxx(t) \ Rxy(t) \ Rxz(t) \ Ryx(t) \ Ryy(t) \ Ryz(t) \ Rzx(t) \ Rzy(t) \ Rzz(t)]$$

À $t=0s$, $q_0=q(t)$ avec $Rxx(t) \ Rxy(t) \ Rxz(t) \ Ryx(t) \ Ryy(t) \ Ryz(t) \ Rzx(t) \ Rzy(t) \ Rzz(t)$ étant la matrice de rotation initiale donnée r et v étant les vitesses et les positions initiales. À chaque instant $t > 0$, la seule force agissant sur le dé est la force gravitationnelle avec la formule :

$$\vec{F_g} = [0; 0; mg]$$

avec m représentant la masse du dé (0.02kg) et g , la valeur moyenne de l'accélération de pesanteur à la surface (9,81 m/s²). La force est négative en z car le sens de la force est négatif.

Selon la deuxième loi de Newton, l'accélération linéaire est donc donnée par la formule :

$$\mathbf{a} = \frac{\sum \mathbf{F}}{m} \text{ avec la somme des forces étant la force gravitationnelle et } m, \text{ la masse du dé.}$$

Les positions, vitesses linéaires et vitesses angulaires à l'instant $t=0s$, sont données dans les conditions initiales des simulations du travail, donc ne sont pas nulles.

1.3. Coins du dé

Nous connaissons les positions des sommets dans le référentiel local. Par exemple, le sommet 5, il est situé à $\frac{l}{2}\vec{i} + \frac{l}{2}\vec{j} + \frac{l}{2}\vec{k}$. Ensuite, nous ramenons chacun des sommets dans le référentiel global en appliquant la matrice de rotation au vecteur position des sommets par rapport au centre de masse.

Ainsi, lors d'une collision, à chaque instant $t > 0$, les positions des huit sommets du dé sont données à partir des formules suivantes :

Sommet1 = r_{CM} + matrice de rotation * [longueurCote/2; + longueurCote/2; - longueurCote/2];

Sommet2 = r_{CM} + matrice de rotation * [-longueurCote/2; longueurCote/2; - longueurCote/2];

Sommet3 = r_{CM} + matrice de rotation * [- longueurCote/2; - longueurCote/2; - longueurCote/2];

Sommet4 = r_{CM} + matrice de rotation * [longueurCote/2; - longueurCote/2; - longueurCote/2];

Sommet5 = r_{CM} + matrice de rotation * [longueurCote/2; longueurCote/2; longueurCote/2];

Sommet6 = r_{CM} + matrice de rotation * [- longueurCote/2; longueurCote/2; longueurCote/2];

Sommet7 = r_{CM} + matrice de rotation * [- longueurCote/2; - longueurCote/2; longueurCote/2];

Sommet8 = r_{CM} + matrice de rotation * [longueurCote/2; - longueurCote/2; longueurCote/2];

avec longueurCote, étant la longueur du coté du dé, $l=0,04m$.

1.4. Choix de Δt

Nous avons choisi notre Δt par essai erreur pour s'assurer de respecter la condition du nombre d'instants utilisés devant être compris entre 100 et 1000 demandé dans l'énoncé. Toutefois, notre Δt est recalculé pendant la collision, ce qui sera expliqué dans la section suivante.

1.5. Collision

1.5.1. Collision entre la sphère et la surface

Tel que mentionné dans la section 1.1, nous avons utilisé la méthode des sphères pour détecter la collision avec la surface. En effet, si la position du centre de masse sur l'axe z ($rz(t)$) est inférieur au rayon de la sphère (**rayonSphere**), notre dé est très proche de la surface et il y a risque de collision. Nous obtenons l'égalité suivante avec un coefficient d'erreur égal à 0.002.

$$rz(t) - rayonSphere < |0.002|$$

Si cette inégalité n'est pas respectée, le dé peut être en collision avec la surface ou non. On commence alors à calculer la position des huit sommets à chaque Δt pour vérifier si un sommet est inférieur à 0, donc enfoncé dans le sol. Si c'est le cas, notre code retourne à l'état précédent en reprenant la vitesse et la position d'avant à l'état $t(i-1)$ et le Δt est divisé par 2. De cette manière, il est plus facile d'avoir une meilleure approximation du contact du dé avec la surface et causer le rebondissement.

1.5.2 Méthode des conditions initiales

Lors de la collision, le comportement du dé entrant en contact avec le sol change. Effectivement, l'orientation de sa vitesse linéaire et de sa vitesse angulaire sera modifiée après la collision. Afin de trouver les conditions du dé après la collision, deux méthodes se proposent à nous, soit la méthode des forces ou bien la méthode des conditions initiales. Dans le cadre de ce devoir, la méthode des conditions initiales est choisie car cette méthode fait appel à l'état des objets avant et après une collision, mais pas en la collision elle-même. Comme nous connaissons la vitesse linéaire et angulaire des solides avant la collision ainsi que l'impulsion et le moment cinétique transférés entre le dé et le sol, nous pouvons déterminer la vitesse linéaire et angulaire du dé après la collision.

1.5.2. Vitesses linéaires finales après la collision

Premièrement, pour trouver les vitesses finales après la collision du dé et du sol, nous utilisons ces formules :

$$\begin{aligned}\vec{v}_a(t_f) &= \vec{v}_a(t_i) + \frac{\vec{J}}{m_a} \\ \vec{v}_b(t_f) &= \vec{v}_b(t_i) - \frac{\vec{J}}{m_b}\end{aligned}$$

Ces deux vitesses initiales $\vec{v}(t_i)$ correspondent respectivement à la vitesse du dé et à la vitesse du sol juste avant que la collision ne se produise. Vu que le sol n'a pas de vitesse, $\vec{v}_b(t)$ est égal à zéro. Dans le cadre du devoir, on travaillera principalement avec $\vec{v}_a(t)$. Le vecteur impulsion \vec{J} est donné par la formule suivante :

$$\vec{J} = \hat{n}j + \hat{t}j_t$$

- \hat{n} : représente la normale sortante de la surface donnée par le vecteur $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$
- $\vec{v}_a(t_i)$: est la vitesse du centre de masse calculé par runge kutta
- j : représente l'impulsion et est égal à $j = -\alpha(1 + \epsilon)v_r^-$
- \hat{t} : est donnée par la formule $\hat{n} \times \hat{u}$ avec $\hat{u} = \frac{\vec{v}_r^- \times \hat{n}}{|\vec{v}_r^- \times \hat{n}|}$
- j_t : pour sa part l'impulsion avec frottement est donnée par une condition :

$$j_t = \begin{cases} \alpha_t \mu_c (1 + \epsilon) v_r^- & \text{si } \mu_s (1 + \epsilon) |v_r^-| < |\hat{t} \cdot \vec{v}_r^-| \\ -\alpha_t |\hat{t} \cdot \vec{v}_r^-| & \text{sinon} \end{cases}$$

Dans notre situation, $\epsilon = 0.5$, $\mu_s = 0.5$ et $\mu_c = 0.3$.

Pour trouver α , nous avons utilisé les formules suivantes :

$$\alpha_t = \frac{1}{\frac{1}{m_a} + \frac{1}{m_b} + G_{a,t} + G_{b,t}} \quad \begin{aligned} G_{a,t} &= \hat{t} \cdot [(\mathbf{I}_a^{-1}(\vec{r}_{a,p} \times \hat{t})) \times \vec{r}_{a,p}] \\ G_{b,t} &= \hat{t} \cdot [(\mathbf{I}_b^{-1}(\vec{r}_{b,p} \times \hat{t})) \times \vec{r}_{b,p}] \end{aligned}$$

La masse du sol est considéré comme une valeur très grande, infinie, alors lorsqu'on divise 1 sur une masse infinie (m_b), on obtient la valeur de 0. Aussi, $\vec{r}_{b,p}$ et \hat{n} sont des vecteurs colinéaires alors leur produit vectoriel est aussi égal à 0, ce qui équivaut à $G_{b,t}=0$.

- $I^{-1} = \begin{bmatrix} \frac{6}{ml^2} & 0 & 0 \\ 0 & \frac{6}{ml^2} & 0 \\ 0 & 0 & \frac{6}{ml^2} \end{bmatrix}$ représente l'inverse de la matrice d'inertie du dé par rapport à son centre de masse ou $I = \begin{bmatrix} \frac{ml^2}{6} & 0 & 0 \\ 0 & \frac{ml^2}{6} & 0 \\ 0 & 0 & \frac{ml^2}{6} \end{bmatrix}$.
- $\vec{r}_{a,p} = \vec{r}_p - \text{sommetCollision}$ représente le vecteur entre le centre de masse (\vec{r}_p) et le point de collision (sommetCollision).

Il suffit maintenant de trouver la distance entre le centre de masse de notre dé et son point de collision donné par le vecteur \vec{v}_r^- dans la section 1.5.4.

Pour détecter les multiples collisions, il faut trouver la distance entre le centre de masse de notre dé et ce point de collision. Pour trouver la vitesse relative entre le sol et notre dé, nous utilisons la formule de \vec{V}_r^- :

$$\vec{v}_r^- = \vec{v}_{a,p}(ti) - \vec{v}_{b,p}(ti)$$

Pour trouver les vitesses relatives du point de collision, $\vec{v}_{a,p}(ti)$ et $\vec{v}_{b,p}(ti)$, nous utilisons ces équations en prenant le moment et les conditions juste avant que la collision ne se passe, soit ti :

$$\vec{v}_{a,p}(t) = \vec{v}_a(t) + (\vec{\omega}_a(t) \times \vec{r}_{a,p}(t))$$

$$\vec{v}_{b,p}(t) = \vec{v}_b(t) + (\vec{\omega}_b(t) \times \vec{r}_{b,p}(t))$$

Sachant que le vecteur $\vec{V}_{b,p}(t)$ est nul, nous obtenons alors :

$$\vec{v}_r^- = \vec{v}_{a,p}(ti) + \vec{\omega}_{a,p}(ti) \times \vec{r}_{a,p}$$

Avec $\vec{v}_{a,p}(ti)$, qui est la vitesse du centre de masse, $\vec{\omega}_{a,p}(ti)$, qui est le vecteur vitesse angulaire du cube à l'instant ti et $\vec{r}_{a,p}$, qui est le vecteur décrivant le point de contact avec le sol vers le centre de masse du dé.

1.5.3. Vitesses angulaires après la collision

Vu que notre collision se fait avec rotation, nous devons aussi calculer la vitesse angulaire finale de notre dé :

$$\vec{\omega}_a(tf) = \vec{\omega}_a(ti) + I_a^{-1}(\vec{r}_{a,p} \times \vec{J})$$

- $\vec{\omega}_a(ti)$ est la vitesse angulaire juste avant la collision du dé
- I_a^{-1} représente l'inverse de la matrice d'inertie du dé par rapport à son centre de masse
- $\vec{r}_{a,p}$ représente le vecteur entre le centre de masse et le point de collision
- \vec{J} est le vecteur impulsion par la formule montrée dans la section 1.5.2.

1.5.4. Détection des collisions

À chaque collision, le \vec{v}_r^- est négatif en z, ce qui signifie que notre dé est en contact avec le sol. Pour détecter les collisions suivantes, il est important de détecter quand \vec{v}_r^- redevient positif, car c'est ce qui nous indique que notre dé a rebondi. À sa prochaine itération négative, on note encore qu'une autre collision est détectée.

Aussi pour calculer quelle face du dé est vue du haut, nous avons calculé les 4 sommets les plus grands de notre dé en z avec leur index dans la matrice des sommets. Pour la suite, nous avons vérifié à quelle face dans le référentiel donné ce sommet correspondait.

Après un nombre de collisions déterminé par nous (100), notre dé commence à glisser jusqu'à ce qu'il s'arrête avec la condition d'arrêt de la simulation. Pour cela, nous avons utilisé la formule du frottement cinétique :

$$\vec{F}^f = -\mu_c N \frac{\vec{v}_{\text{glissement}}}{|\vec{v}_{\text{glissement}}|}$$

Avec la vitesse de glissement qui est la vitesse de notre centre de masse.

De là, notre nouvelle accélération devient la $\vec{a} = \frac{\sum \vec{F}}{m}$ avec \vec{F} étant la force de frottement, car avec la somme des forces, la normale et la force gravitationnelles s'annulent.

2. Présentation et analyse des résultats

Afin de mettre en application la théorie et de montrer le fonctionnement de notre programme, nous avons effectué 4 simulations avec différentes conditions initiales pour chaque lancer.

Tirs	Pos0 (m)	MatR0	V0 (m/s)	w0 (rad/s)
1	[0 ; 0 ; 1]	I	[0.5 ; 0 ; 0.5]	[0.1 ; 0.1 ; 0]
1	[0 ; 0 ; 1]	I	[0.6 ; 0 ; 0.5]	[0.1 ; 0.1 ; 0]
1	[0 ; 0 ; 1]	I	[0.5 ; 0 ; 0.5]	[0.2 ; 0.1 ; 0]
1	[0 ; 0 ; 1]	$\begin{pmatrix} 0.5 & 0.83205029 & 0.2401922 \\ 0.75 & -0.5547002 & 0.3602883 \\ 0.4330127 & 0 & -0.9013878 \end{pmatrix}$	[0.5 ; 0 ; 0.5]	[0.1 ; 0.1 ; 0]

Figure 5: Conditions initiales des simulations de lancer un dé

Nous allons analyser en détail les résultats de chacune des simulations tout en observant les graphiques illustrant la trajectoire du dé.

Dans notre code, notre condition d'arrêt donnée dans l'énoncé n'est pas respectée, ce qui cause une boucle à l'infini. Ce problème est dû au fait qu'une fois que notre dé ne rebondit plus et que la force de frottement s'applique, nos positions en x, y z ainsi que nos vitesses linéaires et angulaires ne changent pas. Nous avons essayé de détecter l'erreur mais sans issue. Nous avons donc ajouté une autre condition d'arrêt, faute de temps.

2.1 Simulation 1

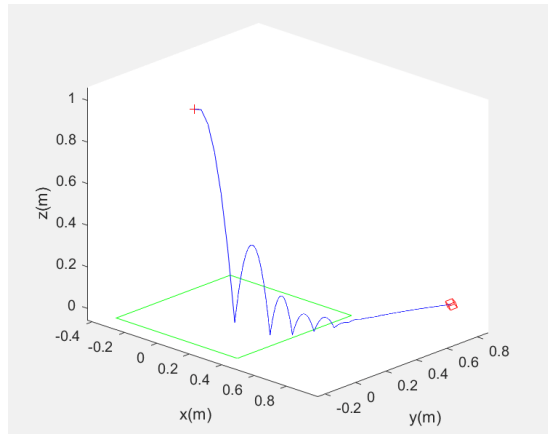


Figure 6: Simulation 1 du dé

Tableau 1: Conditions finales pour la simulation 1

Collision	$\Delta t(s)$	Face	$t_f(s)$	$r_f(m)$
100	0.082	1	2.613	[0.7966;0.8417;0.0362]

Pour cette simulation, la position finale de notre dé n'est pas logique en x et y car le dé dépasse la table. On pense qu'il y a une erreur au niveau des vitesses. Toutefois, nous pouvons observer les collisions et le glissement après. Comparativement à la position initiale qui est nulle en x et y, avec la vitesse angulaire, r_x et r_y sont alors pris en compte et deviennent respectivement 0.7966 et 0.8417. La vitesse diminue jusqu'à devenir négative, d'où notre condition d'arrêt du dé. Les vitesses finales sont nulles ou inférieures à 0, ce qui est logique si notre dé s'arrête. Aussi, la face supérieure du dé est la face 1.

2.2 Simulation 2

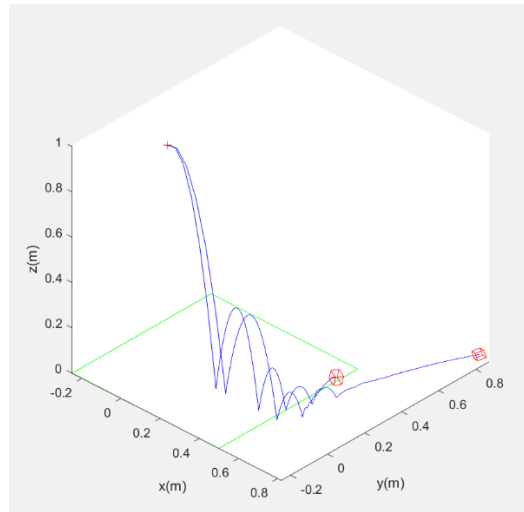


Figure 7: Simulation 2 du dé

Tableau 2: Conditions finales pour la simulation 2

Collision	$\Delta t(s)$	Face	$t_f(s)$	$r_f(m)$
100	0.082	4	2.30	[0.5177;0.3658;0.0331]

Pour cette simulation, comme notre simulation 1, notre dé est sur le coin de la table. La position finale en x et y est plus ou moins logique avec 0.5177 en x et 0.3658 en y car le dé ne déborde pas. Ce qui semble relativement normal. 100 collisions sont enregistrées avec un temps final de 2.30s. On peut les observer à travers la trajectoire dessinée à la figure 7. La face supérieure du dé est la face 4. Les vitesses finales sont nulles ou inférieures à 0, ce qui est logique si notre dé s'arrête.

2.3 Simulation 3

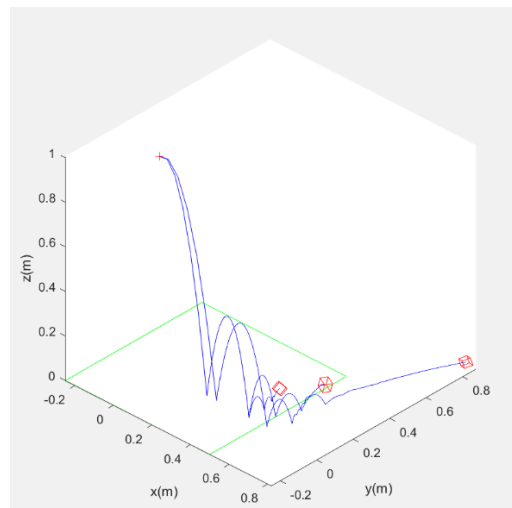


Figure 8: Simulation 3 du dé

Tableau 3: Conditions finales pour la simulation 3

Collision	$\Delta t(s)$	Face	$t_f(s)$	$r_f(m)$
100	0.082	1	2.1422	[0.4115;0.2274;0.0339]

Pour cette simulation, la position finale de notre dé est logique, car comme nous pouvons l'observer dans la figure 8, le dé reste sur la table avec une position finale de 0.4115 en x, 0.2274 en y. Ce qui semble relativement normal. Le temps pris pour rouler 100 collisions est de 2.1422 et la face de notre dé est la face 1. Les vitesses finales sont nulles ou inférieures à 0, ce qui est logique si notre dé s'arrête.

2.3 Simulation 4

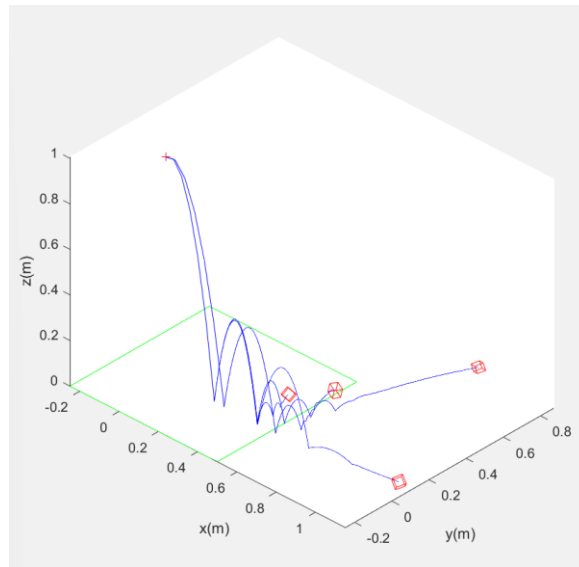


Figure 9: Simulation 4 du dé

Tableau 4: Conditions finales pour la simulation 4

Collision	$\Delta t(s)$	Face	$t_f(s)$	$r_f(m)$
100	0.082	6	1.9205	[1.1217;0.0716;0.0366]

Si la matrice de rotation initiale n'est pas égale à la matrice unitaire, cela veut dire que le dé a déjà une certaine inclinaison. Le dernier essai démarre donc avec une orientation. Alors, on part avec une position angulaire qui n'est pas égal à 0. Une fois de plus, notre dé laisse la table avec des positions peu logiques en x, 1.1217 et y, 0.0716. Le temps pris pour rouler 100 collisions est de 1.9205, inférieur

aux précédentes simulations et la face de notre dé est la face 6. Les vitesses finales sont nulles ou inférieures à 0, ce qui est logique si notre dé s'arrête.

3. Conclusion

En terminant, nous avons réussi à étudier la trajectoire d'un dé lancé et sa collision avec le sol. Nous avons utilisé la méthode des conditions initiales ainsi que Runge-Kutta pour simuler sa trajectoire.

Par ailleurs, nous avons utilisé la méthode des sphères pour englober notre dé et détecter le risque de collision. Dans le cas où le dé touche le sol ou s'enfonce dans le sol, nous retournons d'un pas de temps en le mettant plus petit pour avoir une meilleure approximation de la collision. Enfin, la méthode des conditions initiales a été utilisée pour trouver les conditions finales après la collision.

Nous avons rencontré de multiples difficultés lors de ce devoir. Une des grandes difficultés était de trouver l'instant où le cube touchait le sol. Au début, on ne détectait aucune collision entre le dé et le sol. Nous avons réussi à nous en sortir et avons réussi à appliquer les formules lorsqu'il y avait collision. Aussi, au départ nous avons oublié d'inclure les 9 éléments de la matrice de rotation dans le vecteur \vec{q} . Cela nous a causé des problèmes et des bogues, mais une fois que nous avons détecté notre erreur, notre simulation fonctionnait.