



PHS4700

## Physique pour les applications multimédia

Numéro du groupe : 02

Numéro de l'équipe: 1

Numéro du devoir: 3

Nom: Islam	Prénom: Mohammed Ariful	matricule: 1950221
signature:		
Nom: Bakkouri	Prénom: Ismail	matricule: 1954157
signature:		
Nom: Maliha	Prénom: John	matricule: 1984959
signature:		
Nom: Esse	Prénom: Dawut	matricule: 1956802
signature:		

## Introduction

Le but de ce devoir est de réaliser un programme capable de simuler le déplacement et les collisions d'un tir de boule qui peut rouler et glisser sur une table de carambole. Pour faire cela, nous allons utiliser 3 boules de forme sphérique avec une masse  $m = 210 \text{ g}$  et de rayon  $R = 3.1 \text{ cm}$ . Nous allons écrire une fonction Matlab ou octave (On a choisi de travailler avec Matlab) capable de calculer la trajectoire de 3 boules de couleurs différentes à partir d'un point initial (point de départ) jusqu'à ce que la boule principale (blanche) s'arrête, en gérant la possibilité de collision des billes entre elles-mêmes et le bord de la table. Bien sûr, si la bille blanche touche les deux autres billes, il faut prendre compte qu'un point vient d'être marqué ! La fonction prendra en paramètre d'entrée le centre de masse de la boule blanche ( $x_{yb}$ ), de la boule rouge ( $x_{yr}$ ), de la boule jaune ( $x_{yj}$ ) et la vitesse linéaire initiale du centre de masse de la boule blanche ( $V_{bo}$ ). Quatre simulations seront ensuite faites. Pour les trois premières, on utilisera les conditions initiales données et pour la quatrième, il faut déterminer une combinaison des vitesses linéaires et angulaires qui permettent de marquer un point à partir de la position initiale donnée. Les résultats seront analysés et commentés.

## Théorie et équations

### A) Roulement et glissement

Afin de réaliser les simulations, plusieurs notions théoriques et des équations de la dynamique sont nécessaires. En premier lieu, les boules peuvent avoir trois états : glisser sans rouler ; rouler et glisser ; et rouler sans glisser. Les forces qui s'appliquent sur les boules changent en fonction de cet état. Les forces à considérer sont les suivantes :

- La **force gravitationnelle**, sera toujours considérée :

$$\vec{F_g} = (0, 0, -mg)$$

- La **force normale**, opposée à la force gravitationnelle, sera également toujours considérée :

$$\vec{N} = (0, 0, mg)$$

- La **force de frottement**, sera considérée si l'objet **glisse** ou **glisse tout en roulant** :

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

Où  $N$  est la norme de la normale définie comme suit :

$$N = \frac{|\vec{N}|}{|\vec{N}|} ;$$

$\vec{v}_{\text{glissement}}$  est la vitesse de glissement définie comme suit :

$$\vec{v}_{\text{glissement}} = \vec{v}_{\text{cm}} - \vec{v}_{\text{roulement}} ;$$

Et  $\mu_c$  est le coefficient de frottement cinétique, et vaut 0,3.

- La **force de roulement**, si l'objet roule sans glisser (si l'objet roule et glisse, cette force est négligeable en comparaison à la force de glissement, elle sera donc négligée) :

$$\vec{F}_{\text{roulement}} = -\mu_r N \frac{\vec{v}}{v} = -\mu_r mg \frac{\vec{v}}{v}$$

Où  $\vec{v}$  est la vitesse du centre de masse ;

$N$  est la norme de la normale ;

Et  $\mu_r$  est le coefficient de résistance au roulement, et vaut 0,03.

## Transition roulement – glissement

Au début de la simulation, il n'y a pas de vitesse angulaire. La balle glisse donc sans rouler et  $\vec{v}_{\text{glissement}} = \vec{v}_{\text{cm}}$ . Cependant, comme  $\vec{F}_{\text{glissement}}$  est appliquée à la base de la boule, elle induit un moment de force défini comme suit :

$$\vec{\tau}_{\text{glissement}} = (\vec{r}_p - \vec{r}_c) \times \vec{F}^f = \vec{r}_{p,c} \times \vec{F}^f$$

$\vec{r}_{p,c}$  est le vecteur rayon défini comme:

$$\vec{r}_{p,c} = (0, 0, -R)$$

Et  $\vec{F}^f$  est la force de frottement définie plus haut.

Le moment  $\vec{\tau}_{\text{glissement}}$  induit une accélération angulaire qui induit à son tour une vitesse angulaire de plus en plus importante. Nous pouvons calculer la vitesse de roulement à chaque instant à l'aide de la vitesse angulaire induite par le moment de force :

$$\vec{v}_r = -\vec{\omega} \times \vec{r}_{p,c}$$

Ainsi, lorsque

$$\vec{v}_r = \vec{v}_{\text{cm}} ;$$

Nous avons :

$$\vec{v}_{\text{glissement}} = \vec{0} ;$$

À l'instant où il n'y a plus de glissement, la seule force influençant la vitesse du centre de masse est  $\vec{F}^{\text{roulement}}$ , ainsi l'accélération du cm sera :

$$\vec{a}_{\text{cm}} = \vec{F}^{\text{roulement}}/m ;$$

La boule continuera dans cet état jusqu'à l'arrêt ou jusqu'à une collision.

## B) Collisions

Nous avons deux types de collisions à considérer : Les collisions bille-bille et bille-bord, dans les cas, il s'agit de collision avec frottement et roulement.

Tel que demandé dans l'énoncé, afin de calculer nos collisions, nous sommes passé par la méthode des condition initiales. Cette méthode implique un calcul de l'impulsion que subissent les objets. Cette impulsion est ensuite utilisée pour recalculer la vitesse angulaire et linéaire des objets.

- Collision Bille - Bord

La vitesse linéaire après collision de la balle sera définie comme suit :

$$\vec{v}_a(t_f) = \vec{v}_a(t_i) + \frac{\vec{J}}{m_a}$$

Où  $\vec{v}_a(t_i)$  est la vitesse juste avant la collision ;

$\vec{J}$  est le vecteur impulsion (il sera détaillé plus bas);

$m_a$  est la masse de la boule ;

La vitesse angulaire après collision quant à elle sera définie comme :

$$\vec{\omega}_a(t_f) = \vec{\omega}_a(t_i) + \mathbf{I}_a^{-1} (\vec{r}_{a,p} \times \vec{J})$$

$\vec{\omega}_a(t_i)$  est la vitesse angulaire juste avant la collision ;

$\mathbf{I}_a^{-1}$  est l'inverse du moment d'inertie, défini comme :

$$\mathbf{I}^{-1} = \frac{5}{2mr^2} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$\vec{r}_{a,p}$  est le vecteur position du point où la force est appliquée (point d'intersection), avec le centre de masse de la boule comme origine ;

- Collision Bille - Bille

Dans le cas d'une collision bille – bille, les mêmes formules que pour la collision Bille – Bord s'appliquent, mais, nous aurons les équations de vitesses linéaire et angulaire pour chaque boule de sorte que :

$$\vec{v}_a(t_f) = \vec{v}_a(t_i) + \frac{\vec{J}}{m_a}$$

$$\vec{v}_b(t_f) = \vec{v}_b(t_i) - \frac{\vec{J}}{m_b}$$

$$\vec{\omega}_a(t_f) = \vec{\omega}_a(t_i) + \mathbf{I}_a^{-1} (\vec{r}_{a,p} \times \vec{J})$$

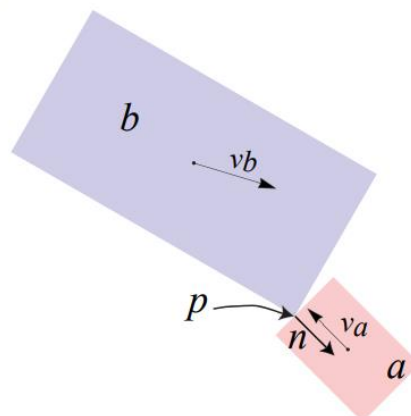
$$\vec{\omega}_b(t_f) = \vec{\omega}_b(t_i) - \mathbf{I}_b^{-1} (\vec{r}_{b,p} \times \vec{J})$$

- Calcul du vecteur impulsion

Le vecteur  $\vec{J}$  est défini comme suit :

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

1.  $\hat{n}$  est la direction normale à la surface au point de contact, définie comme :



2.  $\hat{\mathbf{t}}$  est la direction tangentielle au point de contact définie comme :

$$\hat{\mathbf{t}} = \hat{\mathbf{n}} \times \hat{\mathbf{u}}$$

- $\hat{\mathbf{u}}$  défini comme suit :

$$\hat{\mathbf{u}} = \frac{\vec{v}_r^- \times \hat{\mathbf{n}}}{|\vec{v}_r^- \times \hat{\mathbf{n}}|}$$

- $\vec{v}_r^-$  la différence entre les vitesses des objets juste avant la collision :

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

3.  $j$  est la composante de l'impulsion normale à la surface définie comme :

$$j = -\alpha(1 + \epsilon) v_-^r$$

- o  $\alpha$  est défini comme :

$$\alpha = \frac{1}{\frac{1}{m_a} + \frac{1}{m_b} + G_a + G_b}$$

$$G_a = \hat{\mathbf{n}} \cdot \left[ \left( \mathbf{I}_a^{-1} (\vec{r}_{a,p} \times \hat{\mathbf{n}}) \right) \times \vec{r}_{a,p} \right]$$

$$G_b = \hat{\mathbf{n}} \cdot \left[ \left( \mathbf{I}_b^{-1} (\vec{r}_{b,p} \times \hat{\mathbf{n}}) \right) \times \vec{r}_{b,p} \right]$$

- $\epsilon$  est le coefficient de restitution des vitesses.  
Il vaut 0,8 pour une collision bille-bord et 0,9 pour une collision bille-bille ;



4.  $\dot{j}_t$  est la composante de l'impulsion tangentielle à la surface définie comme :

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

- $\alpha_t$  est défini comme

$$\alpha_t = \frac{1}{\frac{1}{m_a} + \frac{1}{m_b} + G_{a,t} + G_{b,t}}$$

$$G_{a,t} = \hat{t} \cdot [(\mathbf{I}_a^{-1}(\vec{r}_{a,p} \times \hat{t})) \times \vec{r}_{a,p}]$$

$$G_{b,t} = \hat{t} \cdot [(\mathbf{I}_b^{-1}(\vec{r}_{b,p} \times \hat{t})) \times \vec{r}_{b,p}]$$

- $\mu_c$  est le coefficient de frottement cinétique entre les objets en collision.  
Il vaut 0.3 dans le cas d'une collision bille-bord et 0.1 dans le cas d'une collision bille-bille.
- $v_r^-$  est la composante de la différences des vitesses, normale au point de collision définie comme :

$$v_r^- = \hat{n} \cdot \vec{v}_r^-$$

- $\mu_s$  est le coefficient de frottement cinétique entre les objets en collision.  
Il vaut 0.3 dans le cas d'une collision bille-bord et 0.1 dans le cas d'une collision bille-bille.

## Procédure de simulation

- Déplacements

Afin de simuler le déplacement des billes, nous avons utilisé l'algorithme runge-kutta (même procédure que le 2<sup>ème</sup> devoir. Mais nous avons dû rajouter à l'équation les paramètres de calculs angulaires. Notre algorithme ressemble donc à :

$$\vec{q}(t) = \begin{pmatrix} v_{c,x}(t) \\ v_{c,y}(t) \\ v_{c,z}(t) \\ x_c(t) \\ y_c(t) \\ z_c(t) \\ \omega_x(t) \\ \omega_y(t) \\ \omega_z(t) \end{pmatrix} \quad \vec{g}(\vec{q}, t) = \begin{pmatrix} a_{c,x}(\vec{q}, t) \\ a_{c,y}(\vec{q}, t) \\ a_{c,z}(\vec{q}, t) \\ q_1(t) \\ q_2(t) \\ q_3(t) \\ \alpha_x(\vec{q}, t) \\ \alpha_y(\vec{q}, t) \\ \alpha_z(\vec{q}, t) \end{pmatrix}$$

A l'aide des forces, nous avons évalué l'accélération et les moments de forces pour obtenir la vitesse et position linéaires et angulaires à chaque instant à l'aide d'un petit delta t.

- Collisions

Pour détecter les collisions entre bille et bords, nous avons utilisé une condition simple. Dans la situation actuelle, les objets sont de forme sphérique il est donc simple de déterminer s'ils sont en collision sans avoir à passer par une méthode complexe comme les plans de divisions. Nous avons considéré une collision si :

$$|\vec{r}_1 - \vec{r}_2| \leq R_1 + R_2$$

$\vec{r}_1$  et  $\vec{r}_2$  sont la position du centre de masse de chaque boule  $R_1$  et  $R_2$  sont les rayons des boules.

Pour les collisions bille – mur, nous avons simplement considéré si la position du centre de masse de la bille additionnée à son rayon était supérieur à la position du mur à chaque pas itération de l'algorithme.

## Présentation et analyse des résultats

### Résultats

À la suite de la programmation du simulateur, nous avons réalisé les essais demandés avec les situations initiales suivantes :

Tirs	$(x_{bi}, y_{bi})$ m	$(x_{ri}, y_{ri})$ m	$(x_{ji}, y_{ji})$ m	$V_b$ (m/s)
1	(0.5, 0.3)	(1.5, 1.1)	(1.45, 1.31)	(1.7, 0.5)
2	(0.5, 0.3)	(1.5, 1.1)	(1.45, 1.31)	(1.5074, 1.3145)
3	(0.5, 0.3)	(1.5, 1.1)	(1.45, 1.31)	(0.4, 0.3)
4	(0.2, 0.2)	(1.5, 1.1)	(1.45, 1.31)	(1.6, 1.35)

Pour ces 4 tirs, nous avons obtenu les résultats suivants :

Tirs	$(x_{bf}, y_{bf})$ m	$(x_{rf}, y_{rf})$ m	$(x_{jf}, y_{jf})$ m	$T_r$ (roulement-glissement)	coll
1	(2.68, 1.03)	(1.5, 1.1)	(1.45, 1.31)	(0.17; 2.52)	0
2	(2.01, 1.24)	(1.5, 1.1)	(1.45, 1.31)	(2.01; 1.24)	1
3	(0.69, 0.44)	(1.5, 1.1)	(1.45, 1.31)	(0; 0.05)	0
4	(1.83, 1.44)	(1.5, 1.1)	(1.45, 1.31)	(0.20; 0.91; 1.09; 1.74)	2

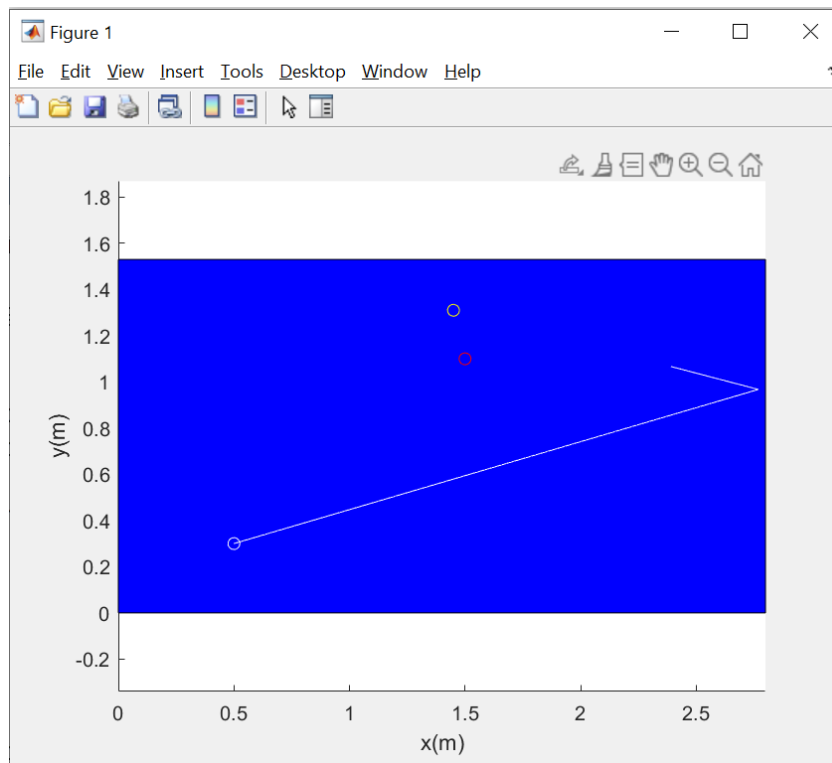
Comme on peut observer à la page suivante, la simulation 1 ne touche aucune bille, cette dernière va tout à droite et rebondi sur le mur. C'est dû à la faible vitesse initiale en  $y$ , l'angle de déplacement est donc trop réduit.

La deuxième simulation quant à elle a une vitesse en  $y$  beaucoup plus conséquente, la boule blanche touche donc la boule rouge et rebondi sur le mur du haut. Nous observons également qu'on transmet une vitesse à la boule rouge et qu'elle se met à se déplacer.

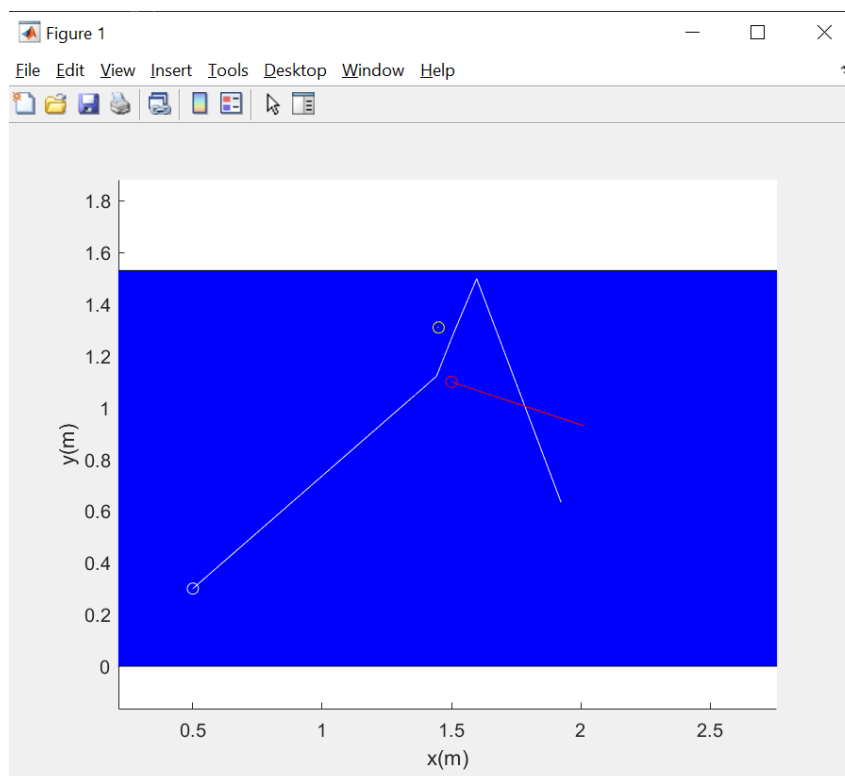
La troisième simulation a une vitesse beaucoup trop faible, elle s'arrête presque immédiatement et n'a pas le temps de gagner suffisamment de vitesse de roulement pour continuer.

Pour la dernière simulation, nous avons légèrement changé les valeurs de la deuxième simulation afin de permettre à la boule de toucher également la boule jaune.

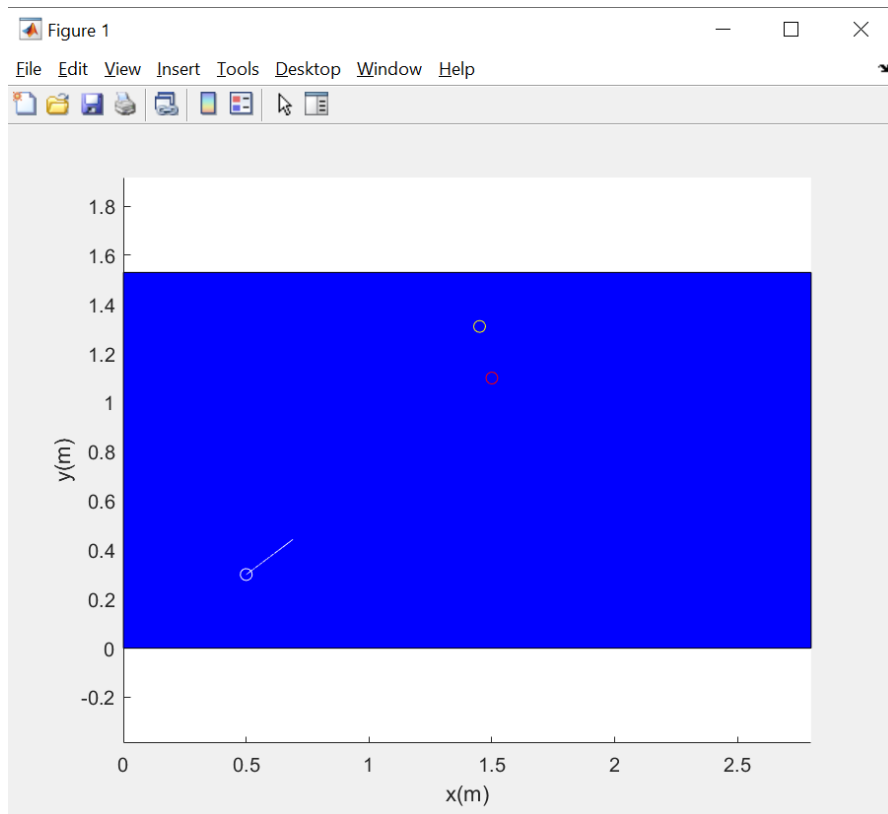
## Aperçu des simulations



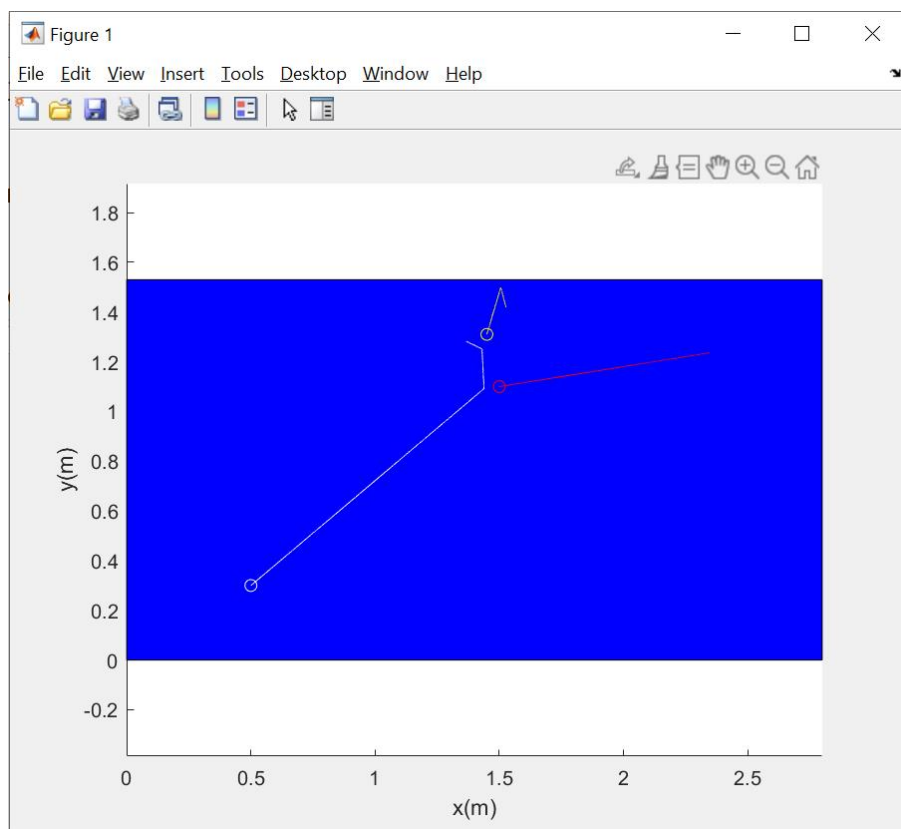
Simulation 1



Simulation 2



Simulation 3



Simulation 4

## Conclusion

Pour conclure, nous avons eu à appliquer des notions de la dynamique des collisions, des méthodes de forces et la dynamique des solides. De plus, on a été amené à appliquer la détection des collisions sur des sphères et des surfaces tels que le bord de la table de billard. Ces notions nous ont permis de calculer les forces agissantes sur les trois billes (1 blanche, 1 rouge, 1 jaune) ainsi que les collisions entre deux ou plusieurs billes et les collisions entre une bille et le bord de la table. Ensuite, on a été amené à calculer les différentes forces qui sont appliquées sur les billes à différents moments et d'évaluer les conséquences de ces forces sur les billes. Nous avons calculé la force gravitationnelle, la force normale de la table sur les billes, la force de roulement et la force de frottement de glissement. Après chaque collision, on doit recalculer la nouvelle vitesse linéaire et angulaire ainsi que son orientation pour chacune des billes. Ceci a été possible grâce à la méthode des conditions initiales. Nous avons rencontré quelques problèmes lors de la réalisation de ce lab. Premièrement, nous avons eu de la difficulté à transformer les concepts physiques et mathématiques en code Matlab. Par exemple, nous avons eu de la difficulté à implémenter la collision bille-bille. Pour régler cette situation, nous avons consulté le document de références du cours, ce qui nous a permis de mieux comprendre et de compléter le code. Finalement, nous avons découvert une erreur qui causait que les positions des billes après une collision étaient fausses. Pour remédier à cette situation, nous avons utilisé le débogage sur Matlab pour pouvoir observer les valeurs des variables. Après un certain temps, nous avons découvert une erreur de calcul dans le code qui causait ceci.