



Table des matières

[Introduction 3](#_Toc120286296)

[1.1 Méthodologie utilisée 3](#_Toc120286297)

[1.2. Positions et vitesses (r(t), v(t), ) 3](#_Toc120286298)

[1.4. Choix de 3](#_Toc120286299)

[1.5. Collision 3](#_Toc120286300)

[2. Présentation et analyse des résultats 3](#_Toc120286301)

[2.1 Simulation 1 3](#_Toc120286302)

[2.2 Simulation 2 4](#_Toc120286303)

[2.3 Simulation 3 4](#_Toc120286304)

[2.3 Simulation 4 4](#_Toc120286305)

[3. Conclusion 4](#_Toc120286306)

# Introduction

Dans le devoir 4 du cours de PHS4700, Physique pour application multimédia, nous étudions l’optique. Le chapitre 6 porte sur les notions de l’optique nécessaires pour comprendre les modèles de la réflexion et de la réfraction.

Dans le présent devoir, nous allons utiliser ces notions d’optique afin de simuler la vue d’une boîte multicolore de forme parallélépipède se trouvant dans une sphère pleine transparente d’indice de réfraction . Cette sphère est placée dans un milieu d’indice de réfraction .

En bref, le but de ce devoir est de programmer une fonction Matlab (Octave) permettant de dessiner l’image de la boîte telle qu’observée par un observateur se trouvant à la position à l’extérieur de la sphère en utilisant la méthode des rayons. La simulation sera limitée à un maximum de 4 réflexions et/ou réfractions pour chaque rayon.

## Méthodologie utilisée

est le point de contact et est la position du centre de l’objet

Si alors il y a une réflexion totale interne, donc

Si alors il y a diffraction, donc

## 1.2. Positions et vitesses (r(t), v(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 durant la simulation, la position **rCM(t),** la vitesse **vCM(t)** du centre de masse du dé avec la matrice de rotation initiale donnée.

## 1.4. Choix de

Nous avons choisi notre 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 est recalculé pendant la collision, ce qui sera expliqué dans la section suivante.

## 1.5. Collision

# 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

## 2.1 Simulation 1

Tableau : Conditions finales pour la simulation 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Collision | ∆t(s) | Face | tf(s) | rf(m) |
| 100 | 0.082 | 1 | 2.613 | [0.7966;0.8417;0.0362] |

## 2.2 Simulation 2

Tableau : Conditions finales pour la simulation 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Collision | ∆t(s) | Face | tf(s) | rf(m) |
| 100 | 0.082 | 4 | 2.30 | [0.5177;0.3658;0.0331] |

## 2.3 Simulation 3

Tableau : Conditions finales pour la simulation 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Collision | ∆t(s) | Face | tf(s) | rf(m) |
| 100 | 0.082 | 1 | 2.1422 | [0.4115;0.2274;0.0339] |

## 2.3 Simulation 4

Tableau : Conditions finales pour la simulation 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Collision | ∆t(s) | Face | tf(s) | rf(m) |
| 100 | 0.082 | 6 | 1.9205 | [1.1217;0.0716;0.0366] |

# 3. Conclusion

En terminant, nous avons réussi à