

PI, le Pendule de Newton et le Billard Américain

18 février 2024

Résumé

1 Introduction

Dans le cadre de notre cours de modélisation mathématique, nous nous lançons dans un projet passionnant visant à modéliser le mouvement d'un pendule de Newton à l'aide du langage de programmation Processing. Notre objectif principal est de représenter de manière numérique le comportement dynamique du pendule en utilisant des formules que nous détaillerons au fil de ce projet.

Nous commencerons par explorer la modélisation des chocs élastiques, en examinant l'évolution en dehors des collisions ainsi que l'opérateur de collision. De plus, nous aborderons la simulation d'un billard, en élaborant un schéma numérique qui prend en compte le contact des boules les unes sur les autres lorsqu'on frappe la boule blanche avec la queue. Cette approche permettra une représentation réaliste des interactions dynamiques entre les boules sur la table de billard.

En continuant notre exploration, nous nous pencherons sur la modélisation du pendule de Newton, développant un schéma numérique précis pour représenter son mouvement. Nous mettrons en œuvre une application numérique pour visualiser et analyser le comportement du pendule, démontrant ainsi la pertinence de notre approche.

En parallèle, notre projet inclura également l'étude d'une approche permettant d'approximer la valeur de π . Nous mettrons en place un schéma numérique dédié à cette approximation et le testerons en l'appliquant au contexte des collisions de deux cubes. Cette démonstration illustrera comment les interactions entre les cubes peuvent converger vers la valeur de π .

Ainsi, notre projet combine la modélisation mathématique du pendule de Newton, la simulation réaliste d'un billard avec collisions, et une exploration

originale de l'approximation de π à travers des concepts mathématiques et numériques.

Ce rapport sera détaillé en plusieurs parties

Pour mettre à jour l'angle du pendule, la formule utilisée est :

$$\theta_{\text{new}} = \theta_{\text{old}} + \omega \cdot \Delta t$$

où θ_{new} est le nouvel angle, θ_{old} est l'angle précédent, ω est la vitesse angulaire et Δt est l'intervalle de temps entre les mises à jour.

La position (x, y) de la balle est calculée en utilisant les fonctions trigonométriques :

$$x = x_{\text{origin}} + L \cdot \sin(\theta)$$

$$y = y_{\text{origin}} + L \cdot \cos(\theta)$$

Calcul de l'accélération angulaire :

L'accélération angulaire est calculée en utilisant la formule :

$$\alpha = -\frac{g}{L} \cdot \sin(\theta)$$

où α est l'accélération angulaire, g est l'accélération due à la gravité et L est la longueur du pendule.

2 Modélisation des Interactions dans un Billard

L'étude des interactions au sein d'un jeu de billard implique la modélisation de la dynamique des boules, tant en l'absence de collisions qu'au moment de leur interaction. Cette section développe une approche mathématique et numérique pour simuler ces phénomènes, en s'appuyant sur les principes de la mécanique newtonienne.

2.1 Évolution des Boules en l'absence de Collision

Sans interaction, le mouvement de chaque boule sur le tapis de billard peut être considéré comme un mouvement rectiligne uniforme. Les positions futures des boules sont prédites à partir de leurs états actuels en utilisant les équations du mouvement linéaire :

$$x_i(t + \Delta t) = x_i(t) + v_{x,i} \cdot \Delta t, \quad (1)$$

$$y_i(t + \Delta t) = y_i(t) + v_{y,i} \cdot \Delta t, \quad (2)$$

où $(x_i(t), y_i(t))$ et $(v_{x,i}, v_{y,i})$ représentent respectivement la position et la vitesse de la boule i à l'instant t , et Δt est le pas de temps de la simulation.

La résistance du tapis est modélisée comme une force de frottement visqueux, conduisant à une décroissance exponentielle de la vitesse selon la relation :

$$v_{i,new} = v_{i,old} \times \exp(-\mu \Delta t), \quad (3)$$

où μ est le coefficient de frottement, simulant l'effet de ralentissement dû au tapis.

2.2 Gestion des Collisions : Opérateur de Contact

Les collisions entre boules sont des événements clés dans la simulation du billard, nécessitant une modélisation précise pour reproduire le comportement réel des boules. Lorsqu'une collision est détectée, les lois de conservation de l'énergie et de la quantité de mouvement sont appliquées pour calculer les nouvelles vitesses des boules impliquées.

La détection de collision entre deux boules i et j se fait en vérifiant la condition suivante :

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} < r_i + r_j, \quad (4)$$

où r_i et r_j sont les rayons des boules i et j .

En cas de collision, les nouvelles vitesses sont calculées en assumant une collision parfaitement élastique, ce qui se traduit par :

$$v_{i,new} = v_{i,old} + \frac{2m_j}{m_i + m_j} \left(\frac{(v_{j,old} - v_{i,old}) \cdot (x_i - x_j)}{\|x_i - x_j\|^2} \right) (x_i - x_j), \quad (5)$$

$$v_{j,new} = v_{j,old} + \frac{2m_i}{m_i + m_j} \left(\frac{(v_{i,old} - v_{j,old}) \cdot (x_j - x_i)}{\|x_j - x_i\|^2} \right) (x_j - x_i), \quad (6)$$

où m_i et m_j désignent les masses des boules i et j , et le terme \cdot représente le produit scalaire entre les vecteurs.

2.2.1 Implémentation de la Simulation de Billard

L'implémentation de la simulation du billard en Processing repose sur la modélisation des balles en tant qu'entités dynamiques dans un espace bidimensionnel. Cette section décrit les principales composantes de l'implémentation, illustrant la manière dont les interactions physiques entre les balles et avec l'environnement du billard sont simulées.

Initialisation des Balles et Paramètres du Billard : Chaque balle est caractérisée par sa position (x, y) , sa vitesse (v_x, v_y) , et sa masse m . Les balles sont initialisées avec des positions et des vitesses définies, et placées dans une configuration spécifique sur le billard de dimensions 2.54×1.27 mètres, avec un diamètre de balle de 57 mm.

$$x_{\text{init}}, y_{\text{init}}, v_{x,\text{init}}, v_{y,\text{init}} \quad \text{pour } i = 1, \dots, \text{nombreDeBalles} \quad (7)$$

Mise à Jour Dynamique : À chaque itération de la boucle `draw()`, les positions et les vitesses des balles sont mises à jour en tenant compte des forces appliquées, notamment la force de frottement simulant l'interaction avec le tapis du billard.

$$x_i \leftarrow x_i + v_{x,i} \Delta t, \quad (8)$$

$$y_i \leftarrow y_i + v_{y,i} \Delta t, \quad (9)$$

$$v_{x,i} \leftarrow v_{x,i} \times \text{coefficientFrottement}, \quad (10)$$

$$v_{y,i} \leftarrow v_{y,i} \times \text{coefficientFrottement}, \quad (11)$$

où Δt est le pas de temps de la simulation.

Gestion des Collisions : Les collisions entre les balles sont détectées par la vérification de la distance entre les centres des balles. Si cette distance est inférieure à la somme de leurs rayons, une collision est détectée et les vitesses post-collision sont calculées en conservant la quantité de mouvement et l'énergie cinétique.

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq r_i + r_j, \quad (12)$$

Les nouvelles vitesses $(v'_{x,i}, v'_{y,i})$ et $(v'_{x,j}, v'_{y,j})$ sont déterminées par :

$$v'_{x,i} = v_{x,i} - \frac{2m_j}{m_i + m_j} \frac{\langle v_i - v_j, x_i - x_j \rangle}{\|x_i - x_j\|^2} (x_i - x_j), \quad (13)$$

$$v'_{y,i} = v_{y,i} - \frac{2m_j}{m_i + m_j} \frac{\langle v_i - v_j, y_i - y_j \rangle}{\|y_i - y_j\|^2} (y_i - y_j), \quad (14)$$

et de manière similaire pour la balle j .

Interaction avec les Bords du Billard : Les collisions avec les bords du billard sont également gérées, reflétant les balles selon la loi de la réflexion, tout en considérant les trous comme des conditions limites absorbantes.

Visualisation : La visualisation des balles et du billard est réalisée en utilisant les fonctions graphiques de Processing, avec une mise à l'échelle appropriée pour adapter la simulation à la fenêtre de visualisation.

2.3 Interprétation et Analyse des Résultats

La simulation développée à partir de ces modèles offre une réplique numérique fidèle de la dynamique observée sur une table de billard réelle. L'évolution des boules sans collision montre des trajectoires linéaires amorties par le frottement du tapis, tandis que les interactions lors des collisions reflètent des changements de direction et de vitesse conformes aux principes de conservation. L'analyse des trajectoires et des interactions au sein de la simulation permet non seulement de valider l'approche de modélisation adoptée mais aussi d'approfondir notre compréhension des phénomènes physiques sous-jacents régissant le jeu de billard.

3 Simulation du Pendule de Newton

Le pendule de Newton, un dispositif emblématique illustrant la conservation de l'énergie et de la quantité de mouvement, offre un terrain fertile pour l'exploration des principes fondamentaux de la mécanique classique. Cette section détaille la construction d'une simulation numérique de ce système, depuis la modélisation théorique jusqu'à l'analyse des résultats simulés.

3.1 Développement d'un Schéma Numérique

La modélisation du pendule de Newton commence par l'approximation de chaque pendule comme un pendule simple, où la masse est concentrée dans une sphère de rayon r et suspendue à un fil inextensible de longueur l . Le mouvement de chaque pendule est alors régi par l'équation différentielle non linéaire suivante, dérivant de la seconde loi de Newton pour les systèmes rotatifs :

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l} \sin(\theta), \quad (15)$$

où θ représente l'angle du fil par rapport à la verticale, et g est l'accélération gravitationnelle.

3.1.1 Discrétisation Numérique

Pour résoudre cette équation de manière numérique, nous adoptons une discrétisation temporelle via la méthode d'Euler, qui actualise l'état du système à chaque pas de temps Δt :

$$\omega_{\text{new}} = \omega_{\text{old}} + \alpha \cdot \Delta t, \quad (16)$$

$$\theta_{\text{new}} = \theta_{\text{old}} + \omega_{\text{new}} \cdot \Delta t, \quad (17)$$

où ω représente la vitesse angulaire du pendule et α est l'accélération angulaire.

3.1.2 Implémentation

L'implémentation du modèle numérique dans un environnement de programmation (comme Processing) implique plusieurs étapes clés pour simuler fidèlement le comportement du pendule de Newton :

Initialisation des Pendules : Chaque pendule est initialisé avec une position angulaire θ_i et une vitesse angulaire initiale $\omega_i = 0$. Les masses m_i sont dérivées à partir du rayon des sphères, assumant une densité uniforme, ce qui nous donne $m_i = \frac{4}{3}\pi r_i^3 \rho$, où ρ est la densité du matériau.

Mise à Jour Dynamique : À chaque pas de temps Δt , les propriétés de chaque pendule sont mises à jour en utilisant les équations discrétisées de la dynamique pendulaire. Pour le pendule i , les mises à jour s'effectuent comme suit :

$$\theta_i \leftarrow \theta_i + \omega_i \cdot \Delta t, \quad (18)$$

$$\omega_i \leftarrow \omega_i + \alpha_i \cdot \Delta t. \quad (19)$$

Détection et Gestion des Collisions : La détection des collisions entre deux pendules i et j se fait en calculant la distance entre leurs sphères. Si cette distance est inférieure à la somme de leurs rayons, une collision est détectée :

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq r_i + r_j. \quad (20)$$

Lorsqu'une collision est détectée, les vitesses angulaires post-collisionnelles ω'_i et ω'_j sont calculées en conservant la quantité de mouvement et l'énergie cinétique.

Visualisation : La visualisation du pendule et de son mouvement est cruciale pour l'analyse de la simulation. Chaque pendule est représenté par une ligne (le fil) et une sphère (la masse pendulaire). La position de chaque sphère est calculée par :

$$x_i = x_{\text{origin}} + l \sin(\theta_i), \quad (21)$$

$$y_i = y_{\text{origin}} + l \cos(\theta_i), \quad (22)$$

où $(x_{\text{origin}}, y_{\text{origin}})$ est le point d'ancrage du fil du pendule.

3.2 Visualisation et Analyse Numérique du Mouvement

3.2.1 Visualisation

La simulation graphique est réalisée en dessinant les fils et les sphères représentant les pendules. Cette visualisation permet non seulement de suivre le mouvement

du système, mais aussi de visualiser le transfert d'énergie cinétique entre les pendules lors des collisions.

3.2.2 Analyse des Résultats

L'examen des données simulées révèle plusieurs phénomènes clés :

- **Conservation de l'énergie** : La somme de l'énergie cinétique et potentielle dans le système reste constante au fil du temps, confirmant la conservation de l'énergie dans le système isolé.
- **Transfert d'énergie** : L'énergie cinétique est transférée de manière presque intégrale d'un pendule à l'autre lors des collisions, illustrant la conservation de la quantité de mouvement et de l'énergie cinétique.
- **Effets non linéaires** : Pour des angles initiaux grands, des comportements non linéaires émergent, mettant en évidence les limites de l'approximation du pendule simple pour de grands écarts de la verticale.

Ces observations démontrent la capacité de la simulation à reproduire fidèlement la dynamique d'un pendule de Newton et à fournir des insights sur les principes physiques sous-jacents. En outre, l'analyse souligne l'importance d'une sélection judicieuse du pas de temps Δt pour équilibrer précision et performance computationnelle.

4 Approche d'Approximation de π par Collision de Cubes

Nous explorons une méthode innovante pour estimer la valeur de π en simulant numériquement les collisions entre deux cubes de masses différentes. Cette technique s'appuie sur les principes de conservation de la quantité de mouvement et de l'énergie cinétique.

4.1 Schéma Numérique pour l'Approximation

Le modèle consiste en deux cubes sur une surface parfaitement rigide et sans frottement. Le premier cube, de masse beaucoup plus importante, se déplace vers le second, initialement au repos. Les collisions sont considérées comme parfaitement élastiques.

4.1.1 Modélisation des Collisions

Les collisions entre les cubes sont modélisées en utilisant les lois de conservation de la quantité de mouvement et de l'énergie cinétique. Pour une collision entre deux cubes de masses m_1 et m_2 , avec des vitesses initiales v_1 et v_2 respectivement, les vitesses finales v'_1 et v'_2 sont déterminées par :

$$v'_1 = \frac{m_1 - m_2}{m_1 + m_2} v_1 + \frac{2m_2}{m_1 + m_2} v_2, \quad (23)$$

$$v'_2 = \frac{2m_1}{m_1 + m_2} v_1 + \frac{m_2 - m_1}{m_1 + m_2} v_2. \quad (24)$$

4.2 Implémentation Détaillée

L'implémentation de la simulation, par exemple en Processing, suit une série d'étapes précises pour simuler fidèlement les dynamiques de collision et compter le nombre total de collisions.

4.2.1 Initialisation

Au démarrage, les cubes sont initialisés avec leurs positions, vitesses et masses. Le cube de masse m_1 est positionné pour se déplacer vers le cube de masse m_2 , initialement au repos.

4.2.2 Boucle de Simulation

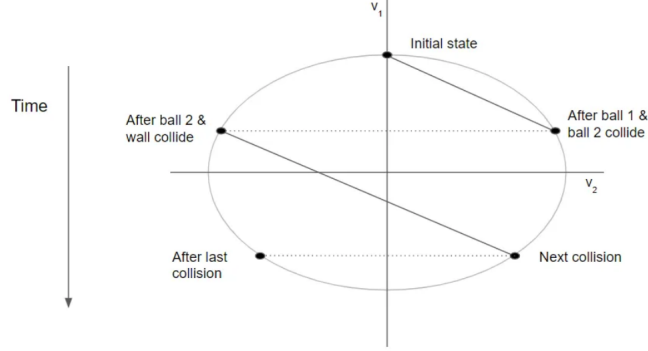
Dans chaque itération de la boucle de simulation, les actions suivantes sont effectuées :

- Mise à jour des positions des cubes en fonction de leurs vitesses actuelles.
- Détection et gestion des collisions entre les cubes et entre un cube et le mur, avec mise à jour des vitesses selon les équations de conservation.
- Incrémentation d'un compteur de collisions à chaque interaction détectée.

4.2.3 Gestion des Collisions

Les collisions sont gérées en inversant la vitesse du cube m_2 lorsqu'il entre en collision avec le mur, et en calculant les nouvelles vitesses des cubes après chaque collision entre eux selon les formes équations fournies plus haut.

Pour voir comment les vitesses dans le système évoluent au cours de l'interaction, représentons-le graphiquement dans l'espace des phases. Chaque point de l'espace des phases correspond à une position et un élan particuliers. Autrement dit, chaque point de l'espace des phases représente un état unique de notre interaction de bloc.



4.2.4 Calculer les collisions

Nous prendrons $x = v_2$ et $y = v_1$, tels que l'équation [25] devient $My + mx = M$, ce qui donne l'équation [27].

$$y = -\frac{m-M}{x} + 1, \quad (25)$$

Idême pour l'équation [26] qui est multipliée par $2/M$.

$$\frac{m}{M}x^2 + y^2 = 1, \quad (26)$$

Pour calculer le nombre de collisions, nous trouvons combien de fois la longueur des segments courbes est divisée en périmètre.

Pour transformer l'espace des phases, nous substituons $X = \sqrt{M/m}$ à x et Y à y de telle sorte que les équations [27] et [28] deviennent :

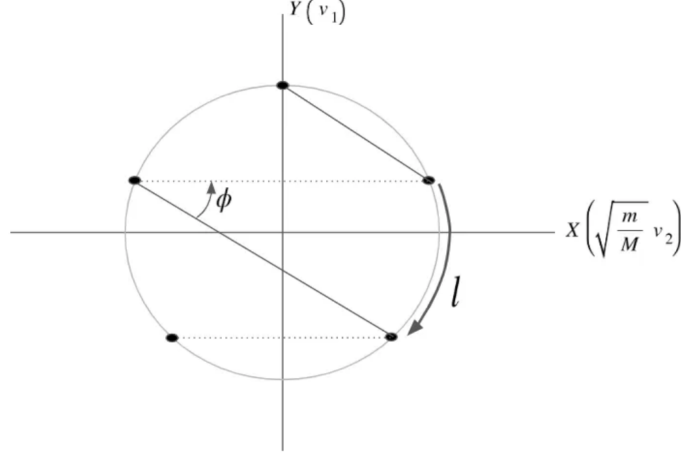
$$\begin{aligned} \frac{m}{M} \left(\frac{M}{m} \right) X^2 + Y^2 &= 1 \\ \text{donc } X^2 + Y^2 &= 1 \end{aligned} \quad (27)$$

et

$$\sqrt{\frac{M}{m}}X + \text{constante} \quad (28)$$

Le graphique de l'espace des phases est donc changé pour former un cercle de rayon 1 car d'un facteur de $\sqrt{M/m}$.

Ceci forme également un angle de ϕ sur la ligne imageant la vitesse des cubes en fonction de X et Y. Il forme aussi un arc de longueur l entre deux périodes de point (voir schéma ci-dessous) :



4.2.5 Approximation de π

D'après la définition de la tangente et l'équation [7] : $\tan \phi = \text{—pente—} = \sqrt{M/m}$, ce qui, réarrangé, nous donne : $\phi = \arctan \sqrt{M/m}$.

En se souvenant du théorème de l'angle inscrit (que l'angle inscrit est la moitié d'un angle central qui sous-tend le même arc) et en utilisant les formules de longueur d'arc pour la mesure d'angle de $l = 2\phi$ et la longueur de $l = R \cdot 2\phi = 2 \arctan \sqrt{M/m}$.

Une approximation aux petits angles donne $\arctan \theta \approx \theta$.

Il nous reste donc : $l \approx 2 \sqrt{M/m}$.

$$\pi \approx \frac{N}{\sqrt{\frac{m_1}{m_2}}}. \quad (29)$$

4.3 Analyse des Résultats

L'analyse de la simulation montre que le nombre de collisions N tend vers des multiples de la valeur de π lorsque le rapport de masse m_1/m_2 augmente, illustrant une méthode fascinante d'approximation de π à travers un système physique simple.

Cette approche d'approximation de π démontre l'interconnexion profonde entre la physique classique et les mathématiques, offrant une perspective unique sur l'exploration des constantes mathématiques à travers des simulations numériques.

5 Conclusion

À travers ce projet, nous avons entrepris de modéliser et de simuler le comportement de systèmes physiques complexes, notamment le pendule de Newton et le jeu de billard, ainsi que d'explorer une méthode innovante pour approximer la valeur de π à l'aide de collisions entre cubes. L'objectif était de démontrer l'efficacité des principes de la mécanique classique dans la modélisation de phénomènes physiques et dans la réalisation de simulations numériques précises.

Nos simulations ont produit des résultats significatifs, illustrant avec succès la conservation de l'énergie et de la quantité de mouvement dans le pendule de Newton, et capturant de manière réaliste les trajectoires et les collisions des boules de billard, y compris leurs interactions avec les bandes. L'approche utilisée pour approximer π via des collisions de cubes a également mis en évidence une méthode captivante et éducative de lier des concepts physiques à des constantes mathématiques fondamentales.

Néanmoins, une analyse critique de nos résultats soulève des questions sur les limites de nos modèles. Bien que la simulation du billard intègre les interactions avec les bandes, il pourrait être bénéfique d'affiner davantage ces interactions pour mieux représenter l'impact réel et les réflexions des boules. De même, le modèle du pendule de Newton pourrait être amélioré pour tenir compte des effets non linéaires importants à de grands angles.

Pour améliorer la précision et la robustesse de nos simulations, plusieurs améliorations sont envisageables. L'intégration de modèles dynamiques plus complexes pour simuler les effets du spin et les réflexions non élastiques des boules contre les bandes pourrait offrir une représentation plus fidèle de la réalité. L'adoption de méthodes numériques avancées pour résoudre les équations différentielles pourrait également améliorer la stabilité et la précision des simulations.

En conclusion, ce projet a réaffirmé la valeur de la modélisation mathématique et de la simulation numérique dans l'étude des phénomènes physiques. En continuant à peaufiner nos modèles et en explorant de nouvelles avenues de recherche, nous pouvons approfondir notre compréhension des lois naturelles et affiner nos outils pour résoudre des problèmes complexes dans le domaine de la physique.