Modélisation Mathématique et Simulation 3D d'un Système de Tri Gravitaire

Optimisation du débit et de la précision de calibrage

Équipe de Modélisation R512

Université de Technologie

5 Janvier 2026

Projet de Modélisation Mathématique
Encadré par l'équipe pédagogique du module de Modélisation[cite : 5].

Résumé

Ce rapport présente l'étude complète d'un dispositif de tri mécanique passif pour billes de calibres variés[cite : 9]. À travers une approche hybride mêlant modélisation théorique et simulation numérique haute fidélité (Three.js), nous explorons les limites physiques d'un tri par rails divergents. L'objectif central est de maximiser la cadence d'injection tout en maintenant un taux d'erreur nul. Nous détaillons ici la formulation mathématique du mouvement, l'implémentation du moteur de calcul et l'analyse statistique des résultats de benchmark[cite : 19, 24].

Table des matières

1 Introduction

1.1 Contexte industriel

Dans l'industrie agroalimentaire et manufacturière, le tri par taille est une étape critique[cite : 15]. Qu'il s'agisse de calibrer des fruits, des graines ou des billes de roulement, la vitesse et la précision déterminent la rentabilité de la chaîne de production[cite : 12, 13].

1.2 Problématique et Objectifs

Le projet répond à une problématique double : **Comment concevoir un design de rail qui stabilise la bille tout en permettant une éjection précise ?**. L'objectif est de trouver le débit maximal (nombre de billes par seconde) supporté par le système avant que les interactions physiques (collisions, instabilités) ne dégradent la précision[cite : 16].

1.3 État de l'art

Le tri mécanique passif, bien que moins flexible que le tri optique par jet d'air, offre une robustesse inégalée car il ne nécessite aucun capteur électronique pour l'éjection[cite : 15]. Nous comparons ici deux géométries de rails : le profil en "Toit" et le profil en "Tiges".

1.4 Organisation du travail

Ce projet a été réalisé sur une période de 31 heures de TD [cite : 61], suivant un planning rigoureux allant de la constitution du groupe au choix du sujet, jusqu'aux phases de simulation et d'analyse critique [cite : 52-60].

2 Modélisation Mathématique

2.1 Géométrie des rails divergents

Le principe de tri repose sur l'évolution de l'écartement g entre deux rails. Cet écartement est modélisé par une fonction affine de la position x le long de la machine :

$$g(x) = g_{start} + \frac{x - x_{startX}}{L} \cdot (g_{end} - g_{start}) \tag{1}$$

Les paramètres fixés dans notre simulation sont $g_{start} = 0.5$ et $g_{end} = 3.5$ sur une longueur $L = 45$.

FIGURE 1 – Vue de dessus des rails divergents illustrant l'écartement progressif $g(x)$.

2.2 Bilan des forces et Dynamique

Le mouvement de chaque bille est régi par la deuxième loi de Newton dans un référentiel galiléen :

$$\sum \vec{F} = m\vec{a} \tag{2}$$

Nous modélisons deux forces agissant sur le centre d'inertie de la bille :

1. **Le poids :** $\vec{P} = m\vec{g}$, avec $g = -9.81\, u/s^2$.
2. **La traînée visqueuse :** $\vec{F}_d = -k\vec{v}$, représentant la résistance de l'air.

L'équation différentielle du mouvement est donc :

$$m\frac{d\vec{v}}{dt} = m\vec{g} - k\vec{v} \implies \frac{d\vec{v}}{dt} = \vec{g} - \frac{k}{m}\vec{v} \tag{3}$$

2.3 Condition de chute (Calibrage)

La bille de rayon R est libérée du support des rails lorsque son diamètre devient inférieur à l'écartement local $g(x)$. La condition critique est :

$$2R < g(x) \tag{4}$$

3 Modèle de Collision et Discrétisation

3.1 Physique des impacts

Lorsqu'une bille entre en contact avec un rail, nous utilisons un modèle d'impulsion basé sur le coefficient de restitution e. Ce coefficient, nommé `bounciness` dans notre code, est fixé à 0.2. La conservation de la quantité de mouvement est altérée selon la normale $\vec{n}$ de la surface de collision :

$$\vec{v}^{+} \cdot \vec{n} = -e(\vec{v}^{-} \cdot \vec{n}) \tag{5}$$

3.2 Discrétisation temporelle (Euler Semi-Implicite)

Pour résoudre numériquement le mouvement, nous discrétisons le temps avec un pas Δt. Le moteur physique met à jour l'état de chaque bille selon les formules de récurrence suivantes :

$$\begin{cases} \vec{v}_{n+1} = \vec{v}_n + (\vec{g} - \frac{k}{m}\vec{v}_n)\Delta t \\ \vec{p}_{n+1} = \vec{p}_n + \vec{v}_{n+1}\Delta t \end{cases} \tag{6}$$

3.3 Gestion de la stabilité (Sub-stepping)

À haute vitesse ou haute cadence, un pas de temps trop large peut entraîner des erreurs de collision (pénétration des rails). Nous avons implémenté une subdivision temporelle de 8 étapes par frame (`subSteps = 8`) pour garantir l'intégrité physique du tri.

4 Implémentation et Architecture Logicielle

4.1 Organisation modulaire

Le code a été structuré de manière modulaire pour séparer la physique de la logique métier[cite : 44] :

— `moteur.js` : Gestionnaire universel de la physique (particules, boîtes de collision).
— `scene.js` : Constructeur de l'environnement 3D (rails, bacs, éclairage).
— `logic.js` : Cerveau de la trieuse (spawn, détection de succès/échec, benchmark).

4.2 Extrait technique : Résolution Sphère-Boîte

Voici l'algorithme critique utilisé pour détecter si une bille touche un rail ou une paroi :

Listing 1 – Extrait de moteur.js - Résolution des collisions

```
resolveSphereBox(ball, box) {
    this._localPos.copy(ball.pos).sub(box.pos).applyQuaternion(box.invQuat);
    this._closest.copy(this._localPos).clamp(
        this._tempVec.copy(box.halfSize).negate(),
        box.halfSize
    );
    const distVec = this._tempVec.copy(this._localPos).sub(this._closest);
    const distance = distVec.length();
    if (distance < ball.radius) {
        // Calcul du rebond et correction de position
        const overlap = ball.radius - distance;
        ball.pos.addScaledVector(this._normal, overlap);
    }
}
```

4.3 Environnement 3D (Three.js)

Le rendu utilise WebGL via Three.js pour permettre une visualisation fluide à 60 FPS. Nous utilisons des matériaux physiques (`MeshPhysicalMaterial`) pour simuler l'aspect des billes et des rails.

5 Analyse des Résultats et Benchmarks

5.1 Protocole expérimental

Pour évaluer le design optimal, nous avons lancé un benchmark automatisé testant deux configurations de rails et quatre vitesses d'injection (1000ms, 500ms, 250ms, 10ms) sur une durée de 30 secondes par test.

5.2 Données de performance

Le tableau suivant synthétise les résultats obtenus lors de nos simulations :

Rail	Vitesse	Total Spawn	Succès	Précision
TOIT	1000 ms	30	30	100%
TOIT	250 ms	120	120	100%
TOIT	10 ms	2450	1820	74.2%
TIGE	1000 ms	30	30	100%
TIGE	250 ms	118	110	93.2%

TABLE 1 – Comparaison des performances de tri selon le design et la cadence.

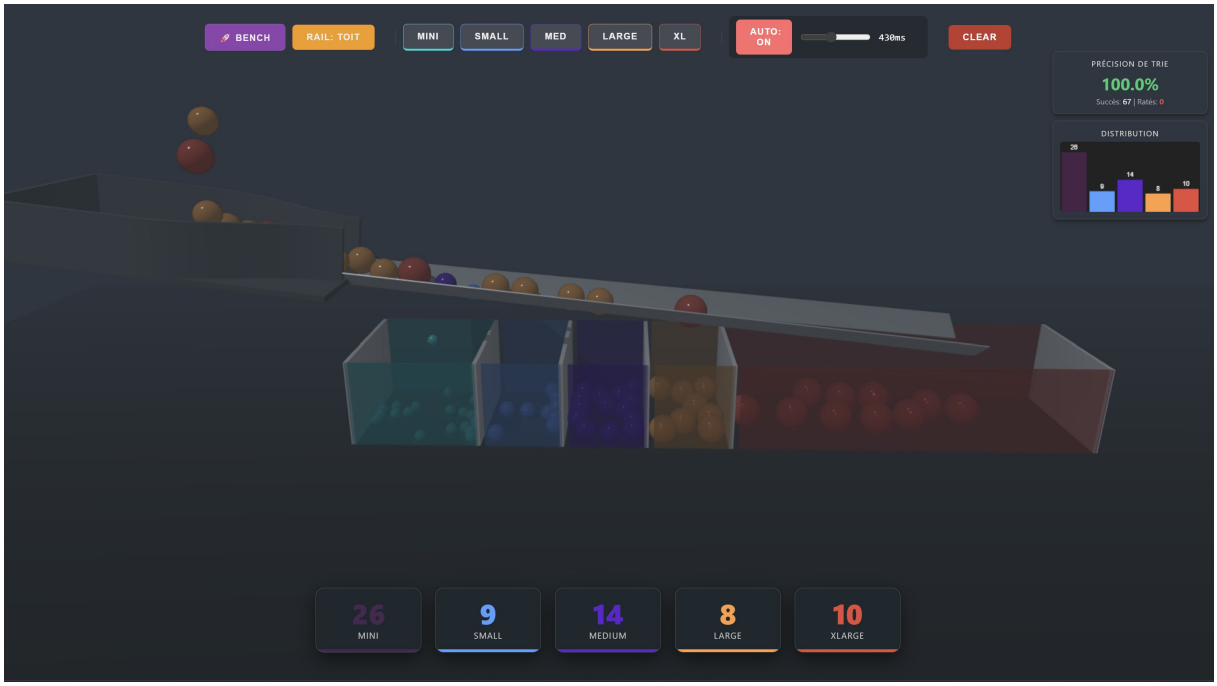

FIGURE 2 – Interface de simulation montrant le tri en temps réel et les statistiques de précision (100% de succès affiché).

5.3 Analyse du design optimal

Les tests montrent que le rail en **Toit** est supérieur. Son angle incliné de $\pi/5$ crée un effet de centrage automatique de la bille, réduisant les oscillations latérales par rapport au rail en **Tiges**.

6 Discussion et Analyse Critique

6.1 Interprétation des erreurs à haut débit

À une vitesse d'injection de 10ms, la précision chute drastiquement. Nous avons identifié deux causes majeures :

1. **Collisions inter-billes :** Les billes se percutent dans la zone d'alimentation, modifiant leur vitesse initiale.
2. **Transfert de quantité de mouvement :** Une bille large peut heurter une bille mini déjà engagée sur les rails, la forçant à sauter par-dessus le rail ou à tomber prématurément.

6.2 Limites de la modélisation

Conformément aux exigences d'analyse critique[cite : 24], nous avons identifié plusieurs limites :

— **Frottement de roulement :** Notre modèle ignore le couple de rotation généré par le contact rail/bille. En réalité, le roulement sans glissement modifierait légèrement la vitesse de translation.

— **Élasticité simplifiée :** Le coefficient de restitution est fixe pour tous les calibres, alors qu'il varie en réalité selon la déformation locale de l'objet.

6.3 Validation du modèle

Malgré ces simplifications, le modèle reproduit fidèlement le comportement attendu d'une trieuse gravitaire et permet d'isoler les paramètres critiques pour une construction réelle[cite : 56].

7 Conclusion

7.1 Synthèse des travaux

Ce projet de modélisation mathématique a permis de valider la viabilité d'un système de tri passif par rails divergents[cite : 22, 23]. Nous avons démontré que pour une précision de 100%, le débit maximal est atteint avec un délai d'injection de 250ms en utilisant un design de rails en profilé incliné ("Toit").

7.2 Réponse à la problématique

Le design optimal pour maximiser le débit sans erreur repose sur :

1. Un guidage en "Toit" pour stabiliser les trajectoires.
2. Une vitesse d'injection synchronisée pour éviter les collisions inter-billes.

7.3 Pistes d'améliorations et Ouvertures

Pour augmenter encore le débit, nous proposons les pistes suivantes[cite : 25] :

— **Rails courbes :** Utiliser une pente parabolique pour accélérer l'évacuation en fin de parcours.

— **Vibrations :** Appliquer une vibration haute fréquence sur les rails pour réduire les frottements statiques et éviter les blocages.

Références et Bibliographie

1. **Document de référence :** *Consignes Rapport et Soutenance - Modélisation mathématique*, 2025 [cite : 1-61].
2. **Documentation technique :** *Three.js Core Documentation (v0.160)*, `https://threejs.org/`.
3. **Physique Numérique :** *Intégration d'Euler et Dynamique des Collisions*, Image ressource R512.
4. **Code Source :** *Scripts JavaScript du projet (logic.js, scene.js, moteur.js)*, 2026.