

# Simulation physique de corps rigides avec interaction

Merwan Achibet  
Université du Havre, 2011

# Table des matières

## Introduction

## Dynamique

- Composante linéaire

- Composante angulaire

## Collisions

- Détection

- Correction

- Réponse

## Le moteur

- Algorithme principal

- Démonstrations

- Perspectives d'évolution

## Conclusion

# Introduction

## Dynamique

- Composante linéaire

- Composante angulaire

## Collisions

- Détection

- Correction

- Réponse

## Le moteur

- Algorithme principal

- Démonstrations

- Perspectives d'évolution

## Conclusion

# Moteur physique ?

Moteur physique : système de simulation mécanique

Industrie, science, cinéma précis, lents

Jeu vidéo, réalité virtuelle approximatifs, temps réel

Ce projet :

- ▶ Moteur physique de base
- ▶ Corps rigides
- ▶ Corps convexes
- ▶ Temps réel

# Étude de cas

## 1. La chute

$$\vec{a} = \frac{1}{m} \sum_i \vec{F}_i$$

## 2. Le rebond

$$\vec{v}_1 = \gamma \vec{v}_2$$

## 3. Le repos

$$\vec{F}_{A/B} = -\vec{F}_{B/A}$$

$$\vec{F}_{A/B} + \vec{F}_{B/A} = 0$$

# Différentes tâches

- ▶ Dynamique
  - ▶ Composante linéaire
  - ▶ Composante angulaire
- ▶ Gestion des collisions
  - ▶ Détection
  - ▶ Correction
  - ▶ Réponse

## Introduction

## Dynamique

- Composante linéaire

- Composante angulaire

## Collisions

- Détection

- Correction

- Réponse

## Le moteur

- Algorithme principal

- Démonstrations

- Perspectives d'évolution

## Conclusion

# La composante linéaire

Entrée Forces environnementales

Sortie Changement de position

$$\vec{v} = \frac{\partial \vec{p}}{\partial t}$$



$$\vec{p} = \int \vec{v} \partial t$$

$$\vec{a} = \frac{\partial \vec{v}}{\partial t}$$

$$\vec{v} = \int \vec{a} \partial t$$



# Intégration de la composante linéaire

Intégration d'Euler :

$$x_{n+1} = x_n + x' \partial t$$

Appliquée à nos besoins :

$$\vec{a}_{t+\partial t} = \frac{1}{m} \sum_i \vec{F}_i$$

$$\vec{v}_{t+\partial t} = \vec{v}_t + \vec{a}_{t+\partial t} \partial t$$

$$\vec{p}_{t+\partial t} = \vec{p}_t + \vec{v}_{t+\partial t} \partial t$$

# Simplification grâce à l'élan linéaire

L'élan linéaire :

$$\vec{L} = m\vec{v} \qquad \sum_i \vec{F}_i = \frac{\partial \vec{L}}{\partial t} = \frac{\partial(m\vec{v})}{\partial t}$$

La nouvelle intégration :

$$\vec{L}_{t+\partial t} = \vec{L}_t + \sum_i \vec{F}_i \partial t$$

$$\vec{p}_{t+\partial t} = \vec{p}_t + \frac{1}{m} \vec{L}_{t+\partial t} \partial t$$

# Modélisation d'un corps

OK pour une particule, mais un objet plus complexe ?

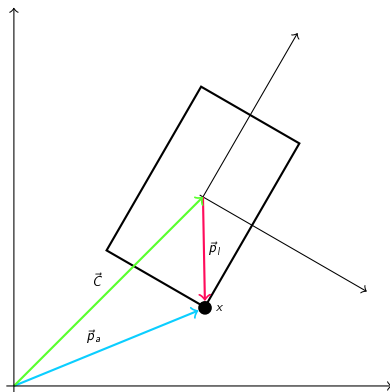
Une particule = un sommet Non

Une unique particule judicieusement placée Oui, le centre de masse

$$\vec{C} = \frac{1}{M} \sum_i m_i \vec{p}_i$$

# Le centre de masse

Centre de masse = origine du repère local



$$\vec{p}_l = \vec{p}_a - \vec{C}$$

# La composante angulaire

Il manque quelque chose... Les rotations !

**Matrice d'orientation** Un vecteur colonne = un axe du repère local

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

**Élan angulaire** Analogie à l'élan linéaire

$$\vec{A}_{t+\partial t} = \vec{A}_n + \sum_i \vec{\tau}_i \quad \vec{\tau}_i = (\vec{x} - \vec{C}) \times \vec{F}_i$$

# Quantités auxiliaires I

Passage de l'élan à la nouvelle orientation moins direct.

**Tenseur d'inertie local** Matrice représentant les efforts à fournir pour produire une rotation le long de chaque axe

**Tenseur d'inertie absolu** Pendant absolu du tenseur d'inertie local

$$I_a = R I_l {}^t R$$

**Vitesse angulaire**

$$\vec{\omega} = I_a^{-1} \vec{A}$$

## Quantités auxiliaires II

On définit l'opérateur  $*$  :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}^* = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}$$

On multiplie  $\vec{\omega}$  et chaque axe de  $R$  :

$$\frac{\partial R}{\partial t} = \begin{pmatrix} \vec{\omega}^* \begin{pmatrix} R_{xx} \\ R_{xy} \\ R_{xz} \end{pmatrix} & \vec{\omega}^* \begin{pmatrix} R_{yx} \\ R_{yy} \\ R_{yz} \end{pmatrix} & \vec{\omega}^* \begin{pmatrix} R_{zx} \\ R_{zy} \\ R_{zz} \end{pmatrix} \end{pmatrix}$$

Soit :

$$\frac{\partial R}{\partial t} = \vec{\omega}^* R$$

# Intégration de la composante angulaire

$$\vec{A}_{t+\partial t} = \vec{A}_t + \sum_i \vec{\tau}_i$$

$$I_a = R_t I_l^t R_t$$

$$\vec{\omega} = I_a^{-1} \vec{A}_{t+\partial t}$$

$$R_{t+\partial t} = R_t + \vec{\omega}^* R_t \partial t$$



## Introduction

## Dynamique

- Composante linéaire

- Composante angulaire

## Collisions

- Détection

- Correction

- Réponse

## Le moteur

- Algorithme principal

- Démonstrations

- Perspectives d'évolution

## Conclusion

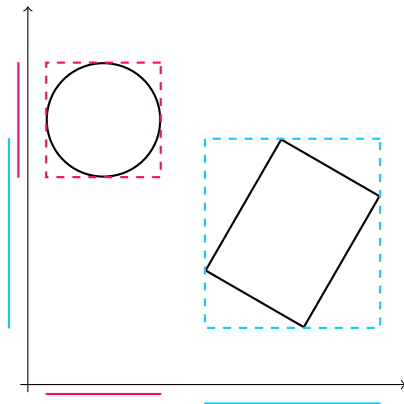
# Deux niveaux de précision

On teste les collisions entre paires de corps :  $\frac{n(n-1)}{2}$  tests

Beaucoup de tests, on veut accélérer le processus.

1. Détection grossière Économique, faux positif possible
2. Détection fine Précise, plus coûteuse

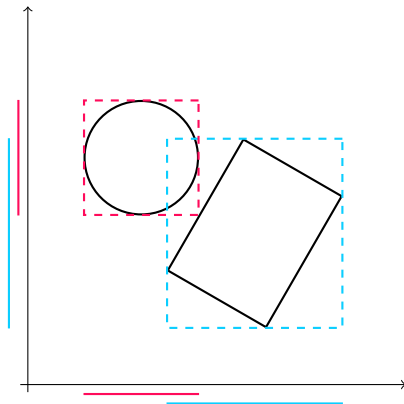
# Détection grossière



**Boîte englobante** Contient tous les sommets, donc tous les points

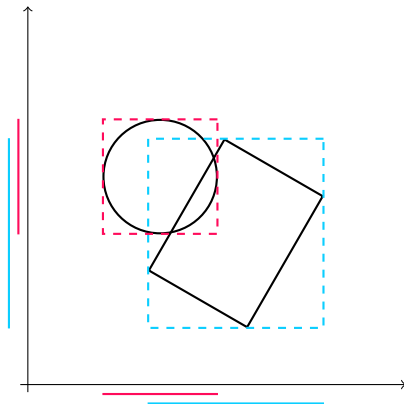
**SAT** Test rapide de collision entre boîtes

# Détection grossière



Faux positif La détection fine invalidera le résultat

# Détection grossière

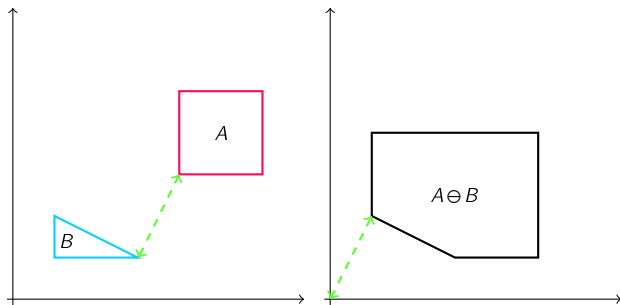


Collision détectée La détection fine validera le résultat

# Détection fine I

Somme de Minkowski  $A \oplus B = \{a + b \mid a \in A, b \in B\}$

Différence de Minkowski  $A \ominus B = A \oplus (-B)$



**Particularité** La plus petite distance de la différence de Minkowski à l'origine est la plus petite distance entre les corps  $A$  et  $B$

## Détection fine II

Comment calculer la plus petite distance entre  $M$  et l'origine ?

**Algorithme GJK** Expansion d'un simplex jusqu'à ce qu'il contienne le point le plus proche de l'origine.

**Simplex** Structure géométrique entièrement contenue dans  $M$  et liée à une dimension.

- 0 Sommet
- 1 Arête
- 2 Triangle
- 3 Tétraèdre

# Détection fine III

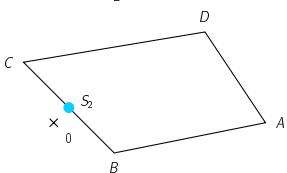
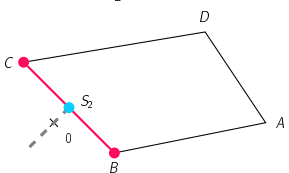
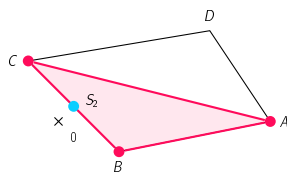
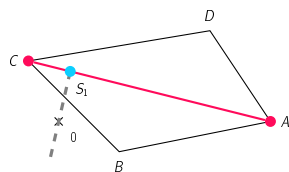
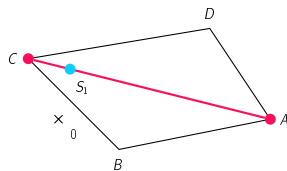
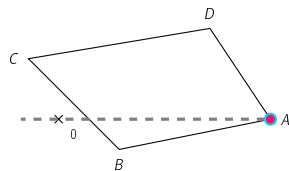
Comment guider la recherche ?

$S(\vec{d})$  Fonction de support renvoyant le sommet de  $M$  le plus extrême dans la direction  $\vec{d}$

**Avantage**  $S_{A \ominus B}(\vec{d}) = S_A(\vec{d}) - S_B(-\vec{d})$   
Inutile de calculer explicitement  $M$  !



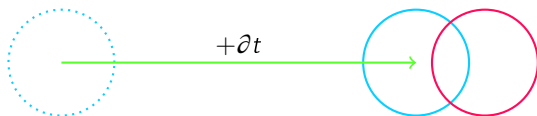
# détection fine IV



# Correction I

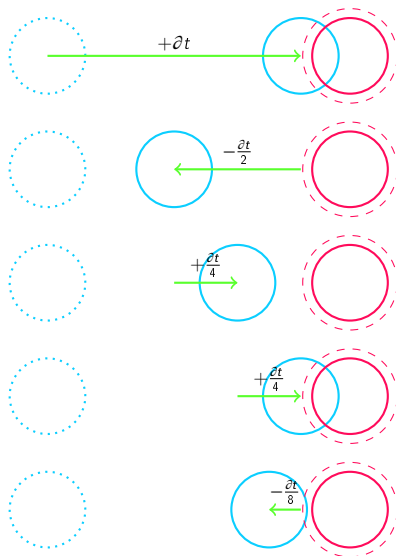
Intégration d'Euler Simulation discrète, pas de temps fixe

Problème Les collisions sont toujours pénétrantes



Solution Intégrer en arrière, par dichotomie

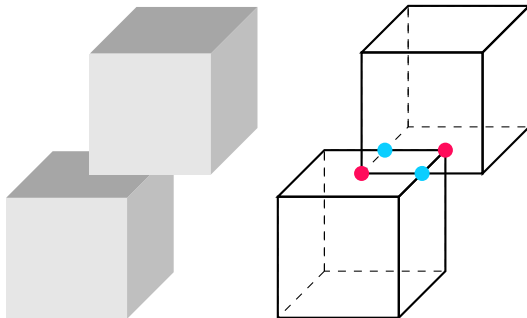
# Correction II



# Réponse 1

Corps rigide Défini par sommets, arêtes et faces

On s'intéresse uniquement aux contacts sommet-face et arête-arête.



## Réponse II

Un *contact* :

- ▶ Position
- ▶ Normale
- ▶ Temps

À chaque contact, une *impulsion* :

$$J = \vec{n} \frac{-(1 + \varepsilon) v_r}{\frac{1}{m_A} + \frac{1}{m_B} + \vec{n} (I_A^{-1} (\vec{r}_A \times \vec{n})) \times \vec{r}_A + (I_B^{-1} (\vec{r}_b \times \vec{n})) \times \vec{r}_B}$$

## Réponse III

Et pour les contacts de repos ?

On force un  $\varepsilon$  valant 0 pour produire une collision non-élastique.

**Problème** Les collisions continues font vibrer les corps

**Solution** On endort les corps dont l'énergie cinétique est faible pendant un laps de temps.

$$E_i = \frac{1}{2} m \vec{v}_i^2$$

$$E = \sum_i E_i$$

## Introduction

## Dynamique

- Composante linéaire

- Composante angulaire

## Collisions

- Détection

- Correction

- Réponse

## Le moteur

- Algorithme principal

- Démonstrations

- Perspectives d'évolution

## Conclusion

# Algorithme principal

---

## Algorithme 4 : Boucle principale

---

**Entrées :** Un pas de temps  $\partial t$

```
pour chaque paire de corps  $(A,B)$  faire
    si collisionGrossiere( $A,B$ ) alors
        si collisionFine( $A,B$ ) alors
             $(A,B) \leftarrow$  corrigerCollision( $A,B$ )
             $C \leftarrow$  detecterContacts( $A,B$ )
            pour chaque contact  $c \in C$  faire
                 $I \leftarrow$  calculerImpulsion( $c$ )
                appliquer( $I, A$ )
                appliquer( $-I, B$ )

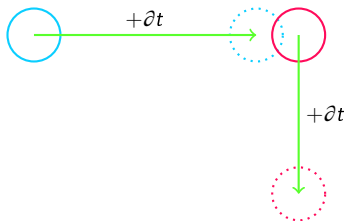
pour chaque corps  $A$  faire
    appliquerForcesEnvironnementales( $A$ )
    integrer( $A, \partial t$ )
```

---



# Défauts de cet algorithme

- ▶ Aucune cohérence temporelle
- ▶ L'ordre d'intégration des corps change l'issue de la simulation



# Algorithme principal amélioré

---

**Algorithme 5** : Boucle principale améliorée

---

**Entrées** : Un pas de temps  $\partial t$

$C \leftarrow \emptyset$

**pour chaque** paire de corps  $(A, B)$  **faire**

$(A_2, B_2) \leftarrow (A, B)$

    appliquerForcesEnvironnementales( $A_2$ )

    integrer( $A_2, \partial t$ )

    appliquerForcesEnvironnementales( $B_2$ )

    integrer( $B_2, \partial t$ )

**si** collisionGrossiere( $A_2, B_2$ ) **alors**

**si** collisionFine( $A_2, B_2$ ) **alors**

            corrigerCollision( $A_2, B_2$ )

$C \leftarrow C \cup \text{detecterContacts}(A_2, B_2)$

trierContacts( $C$ )

$\partial t_2 \leftarrow \min(\partial t, C[0].t)$

**pour chaque** corps  $A$  **faire**

    appliquerForcesEnvironnementales( $A$ )

    integrer( $A, \partial t_2$ )

**pour chaque** contact  $c \in C | c.t = \partial t_2$  **faire**

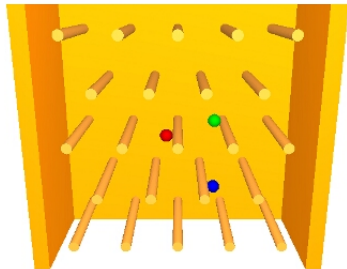
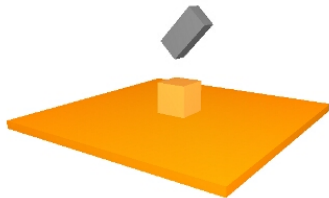
$I \leftarrow \text{calculerImpulsion}(c)$

    appliquer( $I, A$ )

    appliquer( $-I, B$ )

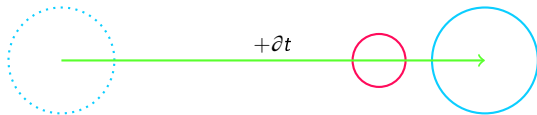
---

# Démonstrations



# Tunneling

**Problème** Les corps se traversent mais aucune collision détectée



# Tunneling

**Problème** Les corps se traversent mais aucune collision détectée



**Solution** Lancer de rayons

- + Économique
- Peut manquer les plus petits corps

# Tunneling

**Problème** Les corps se traversent mais aucune collision détectée



**Solution** Boîtes englobant les positions avant/après

- + Procédures de base déjà utilisées pour la détection grossière
- Certaines trajectoires peu avantageuses (diagonales)

## Introduction

## Dynamique

- Composante linéaire

- Composante angulaire

## Collisions

- Détection

- Correction

- Réponse

## Le moteur

- Algorithme principal

- Démonstrations

- Perspectives d'évolution

## Conclusion

# Conclusion

Différentes perspectives d'évolution :

- ▶ Partitionnement de l'espace
- ▶ Contraintes
- ▶ Plus de stabilité (empilement)

Une voie intéressante : résolution de systèmes linéaires pour les contact de repos.



# Questions