M2 Image, Développement et Technologie 3D (ID3D) - UE Animation, Corps Articulés et Moteurs Physiques

Partie - Simulation par modèles physiques

Cours 4 - Dynamique des objets rigides

Florence Zara

LIRIS - Université Lyon 1

http://liris.cnrs.fr/florence.zara
E-mail: florence.zara@liris.cnrs.fr

Plan du cours

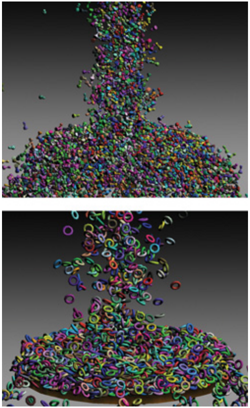
Dynamique des objets rigides

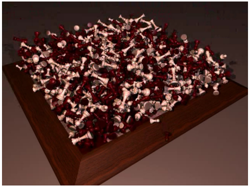
- Rappel sur la dynamique Newtonienne
- Concepts de la dynamique des objets rigides : espace objet, espace monde, vitesses, moment de la force, etc.
- Evolution de l'état de l'objet rigide

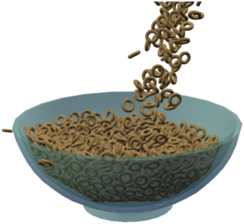
Point de vue animation 3D

- Structure de données de l'objet rigide
- Algorithmes

Simulation d'objets rigides - Exemples







Rappel sur la dynamique Newtonienne

Dynamique Newtonienne

- Position de l'objet au temps t : $x(t)$
- Vitesse de l'objet au temps t : $v(t) = \dot{x}(t) = \frac{d}{dt}x(t)$
- Etat de l'objet au temps t et sa dérivée :

$$X(t) = \begin{pmatrix} x(t) \\ v(t) \end{pmatrix},$$

$$\frac{d}{dt}X(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ v(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ F(t)/m \end{pmatrix}.$$

Rappel sur la dynamique Newtonienne

Dynamique Newtonienne

- Cas système décomposé en n particules :

$$X(t) = \begin{pmatrix} x_1(t) \\ v_1(t) \\ \vdots \\ x_n(t) \\ v_n(t) \end{pmatrix}, \ \frac{d}{dt}X(t) = \frac{d}{dt}\begin{pmatrix} x_1(t) \\ v_1(t) \\ \vdots \\ x_n(t) \\ v_n(t) \end{pmatrix} = \begin{pmatrix} v_1(t) \\ F_1(t)/m_1 \\ \vdots \\ v_n(t) \\ F_n(t)/m_n \end{pmatrix}.$$

- Valeurs à l'état initial t_0
- Boucle de simulation pour obtenir les positions à chaque pas de temps (avec méthode d'intégration)

Concepts de la dynamique du solide

- **Point matériel :** il n'est pas orienté dans l'espace (ce n'est pas une petite bille qui peut rouler, pivoter, tourner, etc.)
- **Corps matériel :** extension de la matière dans les 3 directions qui doit donc être orienté

$\rightarrow$ Utilise le même concept que pour la dynamique des particules, MAIS besoin de plus d'information pour l'état du solide

Concepts de la dynamique du solide

L'état d'un solide est défini par

- une translation (comme pour les particules) : $x(t)$
- mais également une rotation : $R(t)$

$$\rightarrow R(t) = \begin{pmatrix} R_{xx} & R_{yx} & R_{zx} \\ R_{xy} & R_{yy} & R_{zy} \\ R_{xz} & R_{yz} & R_{zz} \end{pmatrix} \qquad \text{matrice } 3 \times 3$$

On considère la rotation par rapport au barycentre du solide

Concepts de la dynamique du solide

- Etat de l'objet rigide au temps t :

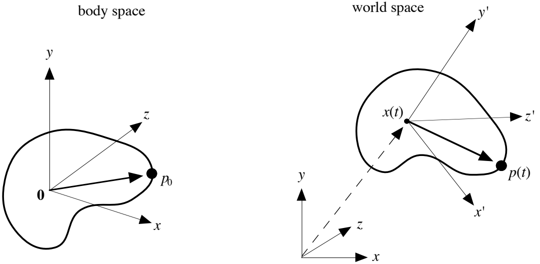
$$X(t) = \begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \text{Position} \\ \text{Orientation (ou rotation)} \\ \text{Quantité de mouvement} \\ \text{Moment cinétique} \end{pmatrix}$$

- Dérivée de l'état de l'objet rigide au temps t :

$$\frac{d}{dt}X(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} ? \\ \\ \\ \end{pmatrix}$$

Espaces objet et monde - Repères

- Origine du repère de l'objet = centre de masse (ou barycentre)
- Centre de masse devient le point $x(t)$ dans l'espace monde
- Axes x, y, z de l'espace objet transformés dans l'espace monde en x' = R(t) x, y' = R(t) y et z' = R(t) z
- Point fixé p_0 de l'espace objet est transformé au point $p(t) = R(t)\ p_0 + x(t)$ dans l'espace monde



Espaces objet et monde - Repères

- $x(t)$: position du centre de masse dans l'espace monde
- Axe x de l'espace objet a comme direction au temps t :

$$R(t)\begin{pmatrix}1\\0\\0\end{pmatrix} = \begin{pmatrix}R_{xx} & R_{yx} & R_{zx}\\R_{xy} & R_{yy} & R_{zy}\\R_{xz} & R_{yz} & R_{zz}\end{pmatrix}\begin{pmatrix}1\\0\\0\end{pmatrix} = \begin{pmatrix}R_{xx}\\R_{xy}\\R_{xz}\end{pmatrix}$$

→ Première colonne de R(t) donne la direction de l'axe x' dans l'espace monde

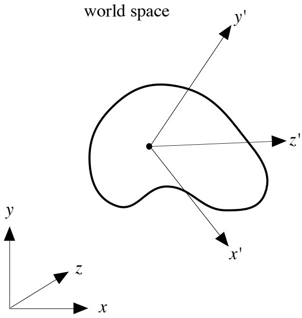
- Même raisonnement avec 2e et 3e colonne de $R(t)$ et axes y, z

→ 2e, 3e colonnes donnent direction axes y', z' dans l'espace monde

Espaces objet et monde - Matrice $R(t)$

Interprétation physique de la matrice de rotation $R(t)$

$R(t) = [\ x'\ \ y'\ \ z'\ ]$



Orientation de l'objet en fonction de son barycentre

Etat de l'objet rigide

Dynamique du solide

- Etat de l'objet rigide : position $x(t)$ et orientation $R(t)$
- Comment évolue cet état au cours du temps ?

$\rightarrow$ Formulation de $\dot{x}(t)$ et $\dot{R}(t)$?

Etat de l'objet rigide - Vitesse linéaire

Vitesse linéaire

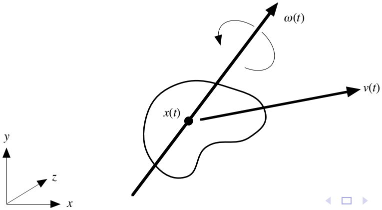
- Vitesse linéaire du centre de masse :

$$v(t) = \dot{x}(t) = \frac{d}{dt}x(t)$$

Etat de l'objet rigide - Vecteur vitesse angulaire

Vecteur vitesse angulaire

- Objet tourne également sur lui-même : vecteur vitesse angulaire $\vec{\omega}(t)$ qui donne axe et vitesse de la rotation
- Vecteur vitesse angulaire est un pseudovecteur :
    - direction de l'axe autour duquel objet tourne
    - vecteur normal au plan de rotation
    - amplitude $||\omega\vec{(t)}||$: angle rotation par rapport au temps



Etat de l'objet rigide - Vecteur vitesse angulaire

Relation entre $\dot{R}(t)$ et $\omega(t)$?

- Colonnes de $\dot{R}(t)$ doit décrire la vitesse à laquelle changent les axes x, y et z
- Relation entre $\dot{R}(t)$ et $\omega(t)$: $\dot{R}(t) = \vec{\omega}(t)^* R(t)$
- $\omega(t)^*$ équivalent à la forme matricielle de $-(\vec{\omega}(t)\times)$ (avec $\times$: produit vectoriel)

$$\rightarrow \dot{R}(t) = \begin{pmatrix} 0 & -\omega_z(t) & -\omega_y(t) \\ -\omega_z(t) & 0 & -\omega_x(t) \\ -\omega_y(t) & -\omega_x(t) & 0 \end{pmatrix} R(t)$$

Discrétisation de l'objet en n particules

Discrétisation

- Objet discrétisé en n particules de masses m_i, de positions constantes r_{0_i} dans l'espace objet
- Position de la particule i dans l'espace monde :

$$r_i(t) = R(t)\ r_{0_i} + x(t)$$

Masse de l'objet

- Masse totale de l'objet : $M = \sum_i m_i$

Discrétisation - Vitesse

Vitesse d'une particule

- Vitesse de la particule i :

$$\begin{aligned}
\dot{r}_i(t) &= \frac{d}{dt} r_i(t) = \frac{d}{dt}(R(t)\ r_{0_i} + x(t)) \\
&= \dot{R}(t)\ r_{0_i} + \dot{x}(t) \\
&= \omega(t)^* R(t)\ r_{0_i} + v(t) \\
&= \omega(t)^* (R(t)\ r_{0_i} + x(t) - x(t)) + v(t) \\
&= \omega(t)^* (r_i(t) - x(t)) + v(t) \\
&= \omega(t) \times (r_i(t) - x(t)) + v(t)
\end{aligned}$$

avec $\omega(t)^* a = \omega(t) \times a$ pour tout vecteur a

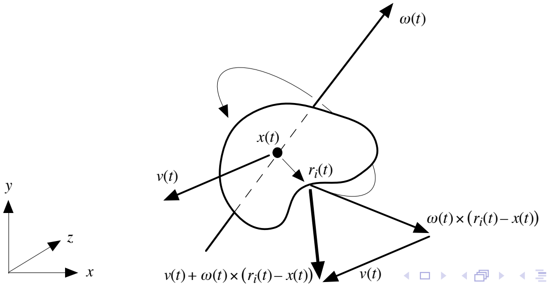
Discrétisation - Vitesse

Vitesse d'une particule

- Vitesse de la particule i :

$$\dot{r}_i(t) = \omega(t) \times (r_i(t) - x(t)) + v(t)$$

Décomposée en un terme angulaire et un terme linéaire



Discrétisation - Barycentre

Centre de masse de l'objet

- Centre de masse : $\frac{1}{M}\sum_i m_i r_i(t)$

Discrétisation - Barycentre

Centre de masse de l'objet

- Centre de masse est l'origine 0 du repère objet
  - particule de positions constantes r_{0_i} dans l'espace objet

$$\frac{1}{M}\sum_i m_i r_{0_i}(t) = \mathbf{0} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

- Centre de masse correspond à l'origine $x(t)$ du repère monde
  - particule de positions $r_i(t) = R(t)\, r_{0_i} + x(t)$ dans monde

$$\begin{aligned} \frac{1}{M}\sum_i m_i r_i(t) &= \frac{1}{M}\sum_i m_i \left(R(t)\, r_{0_i} + x(t)\right) \\ &= \frac{1}{M}\left(R(t)\sum_i m_i r_{0_i} + \sum_i m_i\, x(t)\right) = x(t) \end{aligned}$$

Discrétisation - Force

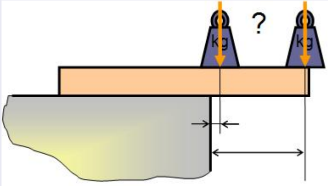
Forces appliquées sur l'objet

- $F_i(t)$: force appliquée à la particule i au temps t
- Force totale appliquée sur l'objet : $F(t) = \sum_i F_i(t)$

Moment de la force

Moment de la force (*torque*) par rapport à un point donné

- Moment de la force représente l'aptitude d'une force à faire tourner un système mécanique autour de ce point (pouvoir de basculement)



Moment de la force

Pouvoir de basculement dépend

- de l'intensité de la force
- de la position relative du point d'application de la force
- et du point de rotation (pivot)

Définition du moment

- Moment de la force $\vec{F}$ s'exerçant au point P par rapport au pivot O :

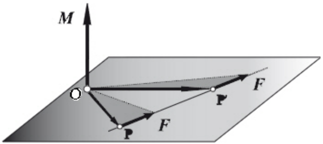
$$\overrightarrow{\mathscr{M}}_{\vec{F}/O} = \overrightarrow{OP} \times \vec{F} = (P - O) \times \vec{F}$$

Notation : $\times$ pour le produit vectoriel

Moment de la force

$$\overrightarrow{\mathscr{M}}_{\vec{F}/O} = \overrightarrow{OP} \times \vec{F}$$

- Le moment est normal au plan dans lequel se déroule la rotation que peut provoquer la force
- Il est colinéaire à l'axe de cette rotation et son sens donne le sens de rotation
- Il ne varie pas quand le point d'application P varie le long de sa ligne d'action



Discrétisation - Moment de la force

Moment de la force

- Moment de la force F_i s'exerçant sur la particule i de position $r_i(t)$ par rapport au barycentre (qui sert de pivot) de position $x(t)$:

$$\tau_i(t) = (r_i(t) - x(t)) \times F_i(t)$$

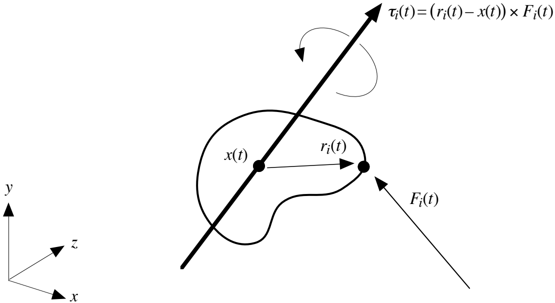
- Moment total des forces appliquées sur l'objet :

$$\tau = \sum_i \tau_i(t) = \sum_i (r_i(t) - x(t)) \times F_i(t)$$

Notation : $\times$ pour le produit vectoriel

Discrétisation - Moment de la force

Moment de la force $\tau_i(t)$ engendré par la force $F_i(t)$ agissant en $r_i(t)$ de l'objet



Discrétisation - Force et moment de la force

Différence entre force et moment de la force

- Force appliquée à la particule i indépendant de sa position
- Aucune info dans $F(t)$ sur l'endroit où les forces sont appliquées
- Moment de la force appliqué à la particule i dépend de la localisation relative $r_i(t)$ de la particule par rapport au centre de masse (pivot)
- τ comporte information sur la distribution des forces $F_i(t)$ sur l'objet

Discrétisation - Quantité de mouvement

Quantité de mouvement (*linear momentum*)

- Quantité de mouvement particule de masse m et de vitesse v :

$$\vec{p} = m\,\vec{v}$$

- Quantité de mouvement particule i de masse m_i et de vitesse $\dot{r}_i(t)$:

$$p_i = m_i \dot{r}_i(t)$$

Discrétisation - Quantité de mouvement

Quantité de mouvement

- Quantité de mouvement totale pour l'objet :

$$\begin{aligned} P(t) &= \sum_i m_i \dot{r}_i(t) \\ &= \sum_i m_i \left(\omega(t) \times (r_i(t) - x(t)) + v(t)\right) \\ &= \sum_i m_i \, \omega(t) \times (r_i(t) - x(t)) + \sum_i m_i \, v(t) \\ &= \omega(t) \times \sum_i m_i \, (r_i(t) - x(t)) + \sum_i m_i \, v(t) \end{aligned}$$

Discrétisation - Quantité de mouvement

Or on a :

$$\begin{aligned}\sum_i m_i \, (r_i(t) - x(t)) &= \sum_i m_i \, (R(t) \, r_{0_i} + x(t) - x(t)) \\ &= R(t) \sum_i m_i \, r_{0_i} = \mathbf{0}\end{aligned}$$

Quantité de mouvement

- Quantité de mouvement totale pour l'objet :

$$\Rightarrow P(t) = \sum_i m_i \, v(t) = M \, v(t)$$

$\rightarrow$ exprimée en fonction du barycentre

Discrétisation - Quantité de mouvement

Quantité de mouvement et sa dérivée par rapport au temps

$$\begin{aligned} P(t) &= M\ v(t) \\ \dot{P}(t) &= M\ \dot{v}(t) = F(t) \end{aligned}$$

Relations obtenues

$$\begin{aligned} v(t) &= \frac{P(t)}{M} \\ \dot{v}(t) &= \frac{\dot{P}(t)}{M} = \frac{F(t)}{M} \end{aligned}$$

Discrétisation - Moment cinétique en un point

Moment cinétique (*angular momentum*)

- Grandeur vectorielle qui est conservée
- Décrit l'état général de rotation d'un système physique
- Analogue à la quantité de mouvement pour la translation

Définition pour un point matériel

- Moment cinétique au point O d'un point matériel M de masse m = moment en ce point O de sa quantité de mouvement

$$\overrightarrow{L_O}(t) = \overrightarrow{OM} \times \overrightarrow{P}(t)$$

- Système de n points matériels M_i de masse m_i par rapport au point O

$$\overrightarrow{L_O}(t) = \sum_i \overrightarrow{OM_i} \times m_i \overrightarrow{v}(M_i)$$

Discrétisation - Moment cinétique du solide

Définition pour un solide

- Moment cinétique d'un solide : $L(t) = I(t)\,\omega(t)$
- $I(t)$: tenseur ou moment d'inertie (matrice 3×3)
  - décrit la répartition des masses dans l'objet par rapport à son barycentre
  - dépend de la rotation mais pas de la translation de l'objet
  - mesure la résistance d'un objet à sa mise en rotation

Dérivée du moment cinétique

- Egale au moment total des forces appliquées sur l'objet
- $\dot{L}(t) = \tau(t)$

Discrétisation - Dérivée du moment cinétique du solide

$$
\begin{aligned}
\frac{d}{dt}\overrightarrow{L_O}(t) &= \frac{d}{dt}\left(\sum_i \overrightarrow{OM_i} \times m_i \vec{v}(M_i)\right) \\
&= \frac{d}{dt}\left(\sum_i \overrightarrow{OM_i}\right) \times m_i \vec{v}(M_i) + \sum_i \overrightarrow{OM_i} \times \frac{d}{dt}\left(\sum_i m_i \vec{v}(M_i)\right) \\
&= \vec{v}(M_i) \times m_i \vec{v}(M_i) + \sum_i \overrightarrow{OM_i} \times \frac{d}{dt}\left(\sum_i m_i \vec{v}(M_i)\right) \\
&= \sum_i \overrightarrow{OM_i} \times \frac{d}{dt}\left(\sum_i m_i \vec{v}(M_i)\right) = \sum_i \overrightarrow{OM_i} \times \sum_i \frac{d}{dt}\left(m_i \vec{v}(M_i)\right) \\
&= \sum_i \overrightarrow{OM_i} \times \sum_i F_i(t) = \sum_i (\overrightarrow{OM_i} \times F_i(t)) \\
&= \sum_i (r_i(t) - x(t)) \times F_i(t)) = \sum_i \tau_i(t) = \tau(t)
\end{aligned}
$$

Discrétisation - Tenseur ou moment d'inertie

Tenseur ou moment d'inertie

- Déplacement de la particule i par rapport à $x(t)$: $r'_i = r_i(t) - x(t)$
- Tenseur d'inertie (matrice symétrique) :

$$I(t) = \sum \begin{pmatrix} m_i(r'^2_{iy} + r'^2_{iz}) & -m_i r'_{ix} r'_{iy} & -m_i r'_{ix} r'_{iz} \\ -m_i r'_{iy} r'_{ix} & m_i(r'^2_{ix} + r'^2_{iz}) & -m_i r'_{iy} r'_{iz} \\ -m_i r'_{iz} r'_{ix} & -m_i r'_{iz} r'_{iy} & m_i(r'^2_{ix} + r'^2_{iy}) \end{pmatrix}$$

Discrétisation - Tenseur d'inertie

Or $\quad {r'_i}^T r'_i = r'^2_{ix} + r'^2_{iy} + r'^2_{iz}, \qquad r'_i {r'_i}^T = \begin{pmatrix} r'^2_{ix} & -r'_{ix} r'_{iy} & -r'_{ix} r'_{iz} \\ -r'_{iy} r'_{ix} & r'^2_{iy} & -r'_{iy} r'_{iz} \\ -r'_{iz} r'_{ix} & -r'_{iz} r'_{iy} & r'^2_{iz} \end{pmatrix}.$

On en déduit : $I(t) =$

$$\sum m_i {r'_i}^T r'_i \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} m_i r'^2_{ix} & -m_i r'_{ix} r'_{iy} & -m_i r'_{ix} r'_{iz} \\ -m_i r'_{iy} r'_{ix} & m_i r'^2_{iy} & -m_i r'_{iy} r'_{iz} \\ -m_i r'_{iz} r'_{ix} & -m_i r'_{iz} r'_{iy} & m_i r'^2_{iz} \end{pmatrix}$$

$$\Rightarrow I(t) = \sum m_i \left(({r'_i}^T r'_i) \mathbf{1} - r'_i {r'_i}^T \right)$$

Discrétisation - Tenseur d'inertie

Or $r_i(t) = R(t){r_0}_i + x(t) \Rightarrow r'_i(t) = R(t){r_0}_i$ et $R(t)R(t)^T = \mathbf{1}$

$$\begin{aligned}
I(t) &= \sum m_i \left(({r'_i}^T r'_i)\mathbf{1} - r'_i {r'_i}^T \right) \\
&= \sum m_i \left((R(t){r_0}_i)^T (R(t){r_0}_i)\mathbf{1} - (R(t){r_0}_i)(R(t){r_0}_i)^T \right) \\
&= \sum m_i \left({r_0}_i^T R(t)^T R(t){r_0}_i \mathbf{1} - R(t){r_0}_i {r_0}_i^T R(t)^T \right) \\
&= \sum m_i \left({r_0}_i^T {r_0}_i \mathbf{1} - R(t){r_0}_i {r_0}_i^T R(t)^T \right) \\
&= \sum m_i \left(R(t)({r_0}_i^T {r_0}_i) R(t)^T \mathbf{1} - R(t){r_0}_i {r_0}_i^T R(t)^T \right) \\
&= R(t) \left(\sum m_i (({r_0}_i^T {r_0}_i)\mathbf{1} - {r_0}_i {r_0}_i^T) \right) R(t)^T
\end{aligned}$$

Discrétisation - Tenseur d'inertie

Tenseur d'inertie

- $I_{body} = \sum m_i \left((r_{0_i}^T r_{0_i})\mathbf{1} - r_{0_i} r_{0_i}^T \right)$ spécifié dans l'espace de l'objet (donc constant pendant la simulation, donc pré-calculé)
- $I(t) = R(t) I_{body} R(t)^T$
- Inverse du tenseur : $I^{-1}(t) = R(t) I_{body}^{-1} R(t)^T$

Dynamique du solide

Equation du mouvement de l'objet rigide

- Etat de l'objet rigide au temps t :

$$X(t) = \begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \text{Position} \\ \text{Orientation (ou rotation)} \\ \text{Quantité de mouvement} \\ \text{Moment cinétique} \end{pmatrix}$$

- $v(t) = \frac{P(t)}{M}$
- $I(t) = R(t) I_{body} R(t)^T$
- $\omega(t) = I(t)^{-1} L(t)$

Dynamique du solide

Equation du mouvement de l'objet rigide

- Dérivée de l'état de l'objet rigide au temps t :

$$\begin{aligned}\frac{d}{dt}X(t) &= \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ \omega(t)^* R(t) \\ F(t) \\ \tau(t) \end{pmatrix} \\ &= \begin{pmatrix} \text{Vitesse} \\ \omega(t)^* R(t) \\ \text{Force totale} \\ \text{Moment total des forces} \end{pmatrix}\end{aligned}$$

Point de vue animation 3D

Comment on implémente cela ?

Structure de données de l'objet rigide

Algorithm 1 Structure de données du solide

```
struct RigidBody {
/* Constant quantities */
double mass ;
matrix Ibody, Ibodyinv ;
/* State variables */
triple x, P, L ;
matrix R ;
/* Derived matrix */
matrix Iinv ;
triple v, omega ;
/* Computed quantities */
triple force, torque ;
};
```

Initialisation des objets solides

Etape 1 : Spécification de l'état de l'objet rigide $X(t)$

- Pour tous les solides de la scène :
  - Spécification de la masse : `mass`
  - Calcul de `IBody` et `IBodyinv`
  - Spécification de l'état $X(t)$: `x, R, P, L`

Algorithm 2 Initialisation des objets solides

```
RigidBody Bodies[NBodies] ;
#define StateSize 18
```

Calcul de $\frac{d}{dt}X(t)$

Etape 2 : Calcul de la dérivée de l'état $\frac{d}{dt}X(t)$

- Pour tous les solides de la scène :
  - Calcul des vitesses : $v(t) = P(t)/M$ (`v`)
  - Calcul de $I^{-1}(t) = R(t)I^{-1}_{body}R(t)^T$ (`Iinv`)
  - Calcul de $\omega(t) = I^{-1}(t)L(t)$ (`omega`)
  - Calcul de $\dot{R}(t) = \omega(t)^* R(t)$
  - Calcul de $F(t)$ (`force`) et de leurs moments $\tau(t)$ (`torque`)
    - Tient compte de toutes les forces : gravité, vent, interaction avec autres objets, etc.

Algorithm 3 Initialisation et calculs

```
RigidBody Bodies[NBodies] ;
#define StateSize 18
void ComputeForceAndTorque(double t, RigidBody *rb) ;
void DdtStateToArray(RigidBody *rb, double *xdot) ;
```

Calcul de $\frac{d}{dt}X(t)$

Algorithm 4 Calcul de $\frac{d}{dt}X(t)$

void Dxdt(double t, double x[], double xdot[]) {
/* Remplissage des variables de l'état du solide (x, R, P, L)
et calcul de $v, I^{-1}, \omega(t)$ */
ArrayToBodies(x) ;
for i = 0 to *NBodies* **do**
    ComputeForceAndTorque(t, &Bodies[i]) ;
    /* Remplissage de xdot ($v = \dot{x}, \dot{R}, F = \dot{P}, \tau = \dot{L}$)
    DdtStateToArray(&Bodies[i], &xdot[i * StateSize]) ;
end for
}

Calcul de $X(t+h)$

Etape 3 : Calcul du nouvel état de l'objet rigide : $X(t+h)$

- Solveur `solve` :
  - Allocation mémoire des structures employées (`xdot`, etc.)
  - Calcul de $\frac{d}{dt}X(t)$ en appelant `Dxdt`
  - Schéma d'intégration pour obtenir $X(t+h)$:

$\rightarrow$ Schéma d'Euler explicite : $X(t+h) = X(t) + h\ \frac{d}{dt}X(t)$

Boucle de simulation

Algorithm 5 Boucle de simulation

```
void RunSimulation() {
double x0[StateSize * NBodies], xFinal[StateSize * NBodies] ;
InitStates() ;
BodiesToArray(xFinal) ;
for t = 0 to 10 do
    /* copy xFinal back to x0 */
    for i = 0 to StateSize ∗ NBodies do
        x0[i] = xFinal[i] ;
        solve(x0, xFinal, StateSize * NBodies, t, t + 1./24., Dxdt) ;
        /* copy dX(t + 1/24 ) into state variables */
        ArrayToBodies(xFinal) ;
        DisplayBodies() ;
    end for
end for
}
```

Références

- Course notes SIGGRAPH 2001 - David Baraff - Physically Based Modeling - Rigid Body Simulation