Simulation en temps réel du mouvement de solides soumis à des contraintes (en deux dimensions)

François Sillion

Francois.Sillion@imag.fr

La spécification du mouvement est l'un des problèmes les plus ardus pour la création d'animations réalistes en image de synthèse. Les applications sont multiples, de l'animation de dinosaures à la simulation du mouvement d'un robot.

Une possibilité serait bien sûr de spécifier complètement le mouvement, à chaque instant (du moins à chaque pas de temps pour lequel on calcule une image). Outre le coût évident de cette approche brutale, il est très difficile de trouver les positions et orientations des différents objets qui donneront à leur mouvement un aspect réaliste. Le talent des animateurs traditionnels est justement de savoir donner une impression de mouvement assez naturel, tenant compte des contraintes telles que les collisions entre objets ou les forces extérieures (gravité) [1].

Le but de ce projet est de réaliser un simulateur de mouvement pour une classe d'objets assez simples, représentés par un ensemble de solides soumis à certaines contraintes. On utilisera une méthode de calcul approchée, qui permet d'obtenir une approximation du mouvement avec un contrôle interactif de certaines contraintes, en temps réel.

1 Représentation des objets

On représente les objets par un repère lié à leur centre de gravité (position et orientation). On pourra également définir une forme géometrique dans ce repère propre, pour le tracé des objets (Figure 1).

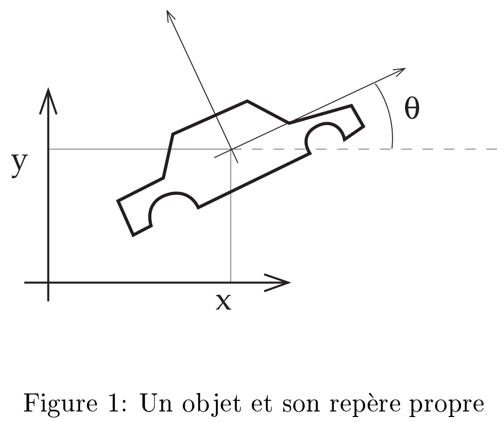

Figure 1: Un objet et son repère propre

Par ailleur on rappelle les équations du mouvement d'un solide en deux dimensions :

[1]En fait pour le dessin animé traditionnel on ne cherche pas à mimer la réalité, mais on prend au contraire certaines libertés avec les lois physiques pour accentuer certains mouvements ou donner un certain caractère aux personnages

$$\sum \vec{F} = m\ddot{\vec{r}} \qquad (1)$$

$$\sum \vec{M} = J\dot{\vec{\Omega}} \qquad (2)$$

$\vec{F}$ et $\vec{M}$ représentant les forces extérieures et leurs moments respectifs.

2 Simulation du mouvement sous contraintes

2.1 Principe de résolution

La méthode simple préconisée pour une première implémentation consiste à séparer l'action des forces extérieures et celle des forces internes qui assurent la vérification des contraintes. On calcule à chaque pas de temps les nouvelles positions de chaque solide en ignorant les contraintes, c'est-à-dire en intégrant simplement les équations du mouvement de façon indépendante. Les positions obtenues résultent en un certain nombre de violations des contraintes, et une correction est calculée *a posteriori* par une méthode itérative, qui modifie la position et la vitesse de chaque point de façon à vérifier les contraintes. Les corrections à effectuer pour chaque objet se font de façon indépendante pour chacune des contraintes : cela implique que le déplacement effectué pour une contrainte donnée peut exagérer la violation d'une autre contrainte, ce qui explique qu'une méthode itérative soit nécéssaire.

L'ensemble du processus de simulation peut donc être résumé comme suit : après avoir spécifié les conditions initiales (voir ci-dessous), les calculs suivants sont effectués pour chaque pas de temps, jusqu'à l'interruption du programme.

1. Calcul des positions des solides au temps suivant, en ne considérant que les forces extérieures.

2. Pour chacune des contraintes définies à l'instant courant, et si elle est violée, calculer une correction à apporter à chacun des objets mis en cause (voir ci-dessous). Cette étape est répeté jusqu'à ce que toutes les contraintes soient vérifiées, ou jusqu'à un nombre fixé d'itérations.

3. Calcul de la correction à apporter aux vitesses linéaires et angulaires pour assurer la cohérence du mouvement.

Remarquez que la convergence de l'algorithme proposé (point 2) n'a pas été prouvée ! par ailleurs le point 3 est nécessaire car la correction des positions et orientations des solides peut être considérée comme résultant de l'action d'une force de contrainte pendant l'intervalle de temps entre deux instants, et cette force a donc également un impact sur les vitesses. Notez cependant que cette force n'est jamais explicitement calculée.

Une condition importante pour conserver un comportement réaliste et aider la convergence est que les corrections apportées conservent la quantité de mouvement du système ainsi que son moment angulaire.

Pour commencer, on se limitera à des contraintes “point-à-point”, définissant des joints d'articulation entre solides.

2.2 Contraintes point-à-point

Une contrainte point-à-point est définie par la donnée de deux solides et d'un point pour chacun, repéré dans le système de coordonnées local.

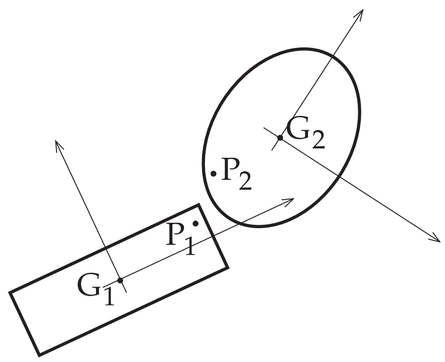

Figure 2: Calcul des corrections pour une contrainte point-à-point

Le calcul des corrections à effectuer sur le mouvement pour imposer une contrainte point-à-point se fait de la façon suivante :

On remarque tout d'abord que l'on pourrait se contenter de calculer une translation à apporter à chaque solide pour imposer la contrainte de coincidence des points. Pour cela il est facile de voir que si l'on veut conserver la quantité de mouvement il faut appliquer les déplacements suivants aux objets :

$$\Delta\vec{G}_1 = \frac{m_2}{m_1 + m_2} \vec{P_1P_2} \tag{3}$$

$$\Delta\vec{G}_2 = \frac{m_1}{m_1 + m_2} \vec{P_2P_1} \tag{4}$$

Cependant le mouvement résultant est très surprenant et ne paraît pas du tout naturel... il est préférable de combiner une rotation et une translation pour faire coincider les points.

Remarquez qu'une rotation seule ne peut suffire en général à ammener ces deux points en coincidence. Le choix de la combinaison rotation/translation à utiliser pourra faire l'objet de différentes études.

On fournit un choix possible, qui consiste à effectuer d'abord une rotation des deux objets (qui prend en compte leurs moments d'inertie respectifs pour assurer la conservation du moment), dont l'amplitude est obtenue en imaginant un modèle simple de force de contrainte s'appliquant entre les solides.

Supposons qu'une force constante (de module calculé comme celui d'un rappel élastique entre leurs positions "de départ") s'exerce entre P_1 et P_2 et que la masse des deux solides soit concentrée en ces points : on peut alors relier la raideur de ce ressort au temps nécessaire pour assurer la rencontre des deux points.

$$k = \frac{m_1 m_2}{m_1 + m_2} \frac{2}{\Delta t^2}$$

Et si l'on imagine maintenant que cette force constante s'exerce en P_1 et P_2 pendant un intervalle Δt, en considérant les solides fixés en leur centre d'inertie, on peut calculer la rotation correspondante de chaque solide

$$\Delta\vec{\Omega}_1 = \frac{m_1 m_2}{m_1 + m_2} J_1^{-1}(\vec{G_1P_1} \wedge \vec{P_1P_2}) \tag{5}$$

$$\Delta\vec{\Omega}_1 = \frac{m_1 m_2}{m_1 + m_2} J_1^{-1}(\vec{G_2P_2} \wedge \vec{P_2P_1}) \tag{6}$$

En pratique, on pourra donc appliquer cette rotation pour amener les points P_1 et P_2 en $P_1^{'}$ et $P_2^{'}$, suivie de la translation nécéssaire pour achever d'imposer la contrainte (donnée par les equations (3) et (4 avec $P_1^{'}$ et $P_2^{'}$).

2.3 Convergence

L'algorithme itératif proposé ne possède aucune garantie de convergence. En particulier puisque les contraintes sont traitées en toute indépendence, il se peut que l'un des solides se voie appliquer une correction très importante sous forme d'une somme de petites corrections. Pour éviter des problèmes de convergence on pourra considérer pour chaque contrainte le nombre de contraintes n_1 et n_2 auxquelles participe chacun des solides mis en cause, et diviser la correction calculée par $\max(n_1, n_2)$ (il est important de ne pas briser la symétrie entre les points, d'où l'emploi du max)

2.4 Mise à jour des vitesses

Après la convergence du processus itératif on ajuste les vitesses linéaires et angulaires en fonction de la correction de position (et orientation) calculée, en ajoutant simplement à la vitesse courante la correction divisée par le pas de temps.

2.5 Assemblage automatique d'une configuration initiale

Il est difficile de spécifier "à la main" une configuration de départ dans laquelle toutes les contraintes sont vérifiées. On aimerait disposer d'un algorithme permettant une mise en place des éléments à partir d'un ensemble de positions quelconques, et de la donnée des contraintes.

Pour cela on peut par exemple suivre (et visualiser) les étapes du processus d'imposition des contraintes évoqué ci-dessus, sans mettre à jour les vitesses lorsque les contraintes sont vérifiées. Autrement dit on imagine que la configuration de départ quelconque est le résultat de l'extrapolation des vitesses au "temps précédent", et on corrige les positions itérativement pour vérifier ces contraintes. Il est clair que dans ce cas les corrections peuvent être très importantes et donc que les corrections de vitesses seraient astronomiques...

On présentera un exemple de construction automatique d'une structure à partir d'une configuration quelconque. Imaginez par exemple qu'il faut assembler une station orbitale à partir d'un ensemble de composants largués en orbite par une fusée: bien entendu dans la pratique il faudrait être capable de calculer les forces à appliquer sur ces éléments pour contrôler des moteurs par exemple, mais ici on ne s'intéresse qu'au mouvement d'assemblage.

2.6 Extensions possibles

Au moins l'une de ces extensions devra être réalisée.

2.6.1 Pas de temps adaptatif

La convergence de l'algorithme n'est nullement garantie. Aussi on tient à repérer toute amorce de divergence et à adapter le pas de temps en conséquence (partant de l'hypothèse que l'algorithme convergera toujours pour des déplacements –et des violations de contraintes– assez petits).

Pour ce faire, on peut utiliser tout indicateur disponible pour évaluer la qualité du pas de temps choisi : par exemple on peut mesurer l'ampleur des corrections de positions apportées par le processus d'imposition des contraintes, et décider de retourner à la position précédente pour refaire le calcul avec un pas plus petit si les corrections sont importantes.

Par contre lorsque les corrections sont indignifiantes on peut allonger le pas de simulation.

2.6.2 Spécification de contraintes en temps réel

On autorisera l'intervention interactive lors de la simulation, par exemple en "saisissant" un point à la souris et en lui imprimant un mouvement.

2.6.3 Autres types de contraintes

On pourra réflechir à d'autres types de contraintes, telles qu'une contrainte de translation d'un point d'un solide le long d'un axe dans le repère d'un autre solide (contraintes "point-à-segment").

Pour cela il suffit de mettre au point un algorithme calculant la correction à effectuer (rotation + translation) adapté à ce cas de figure.

Une autre contrainte intéressante (pour l'animation de personnages par exemple) consiste à imposer une orientation donnée à un solide (verticale...)