Simulation d'un jeu de billard.
Compléments mathématiques et physiques.

Jean-Cédric Chappelier Laurent Colbois

version 1.2 (1er mai 2015)

Révisions

version 1.2	1er mai	**[mineure]** Re-rédaction de la section 3 pour une explication plus détaillée ;
version 1.1	29 mars	**[mineure]** Simplification de la gestion des chocs (section 4.2, page 8) : pour ce projet, il n'est pas nécessaire de recalculer les forces : déplacé en annexe C.3 ; **[mineure]** indication dans la gestion des chocs (section 4.2, page 8) que les α sont calculés de façon similaire à μ.

Table des matières

Notations

On notera par des lettres simples les grandeurs scalaires (e.g. x) et en **gras** surligné d'une flèche les grandeurs vectorielles (e.g. $\overrightarrow{\boldsymbol{x}}$).

Par convention, on notera x la norme du vecteur $\overrightarrow{\boldsymbol{x}}$ (e.g. $v = \|\overrightarrow{\boldsymbol{v}}\|$).

On notera par ailleurs $\left\{\overrightarrow{\boldsymbol{x}}\right\}_{\mathbf{1}}$ le vecteur unitaire colinéaire et de même sens que $\overrightarrow{\boldsymbol{x}}$, c.-à-d. :

$$\left\{\overrightarrow{\boldsymbol{x}}\right\}_{\mathbf{1}} = \frac{1}{x}\overrightarrow{\boldsymbol{x}}$$

Les composantes des vecteurs seront notées par un simple indice (et sont des scalaires) : x_1 est ainsi la première composante du vecteur $\overrightarrow{\boldsymbol{x}}$.

Enfin, la notation « $\longleftarrow$ » est une notation algorithmique signifiant « mise à jour », « devient ». Par exemple pour dire que la grandeur x augmente d'une valeur a, on notera : $x \longleftarrow x + a$.

1 Cadre général du projet

Le projet de cette année s'intéresse à la simulation numérique d'un jeu de billard : boules, et table, c.-à-d. parois (ou bandes) et tapis (ou sol) ; dans ce projet nous ne modéliserons pas la queue mais donnerons simplement des impulsions aux boules.

Le présent document ne constitue que le *complément* mathématique de la donnée du projet. Pour plus de détails sur la donnée précise, le cadre du projet lui-même, etc., voir la page Web du projet.

2 Modélisation des boules

2.1 Formulation générale

On s'intéresse ici à la simulation numérique d'objets mobiles à q degrés de liberté (voir annexe A pour la notion de degré de liberté). Pour un jeu de billard, les seuls objets mobiles seront les boules, pour lesquelles q vaudra 6 : 3 degrés de liberté de translation, les coordonnées usuelles (x, y, z) et 3 degrés de liberté de rotation, (α, β, γ).

De tels objets possèdent (au moins) :

1. un « vecteur d'état » $\overrightarrow{\mathbf{\Omega}} \in \mathbb{R}^q$ représentant ses degrés de liberté ($q = 6$ ici, mais nous souhaitons rester général, le moteur d'intégration que vous écrirez pouvant peut être vous servir plus tard...) ;
2. et une « équation d'évolution » f décrivant l'évolution temporelle de leur vecteur d'état :

$$\ddot{\overrightarrow{\mathbf{\Omega}}} = f(t, \overrightarrow{\mathbf{\Omega}}, \dot{\overrightarrow{\mathbf{\Omega}}}) \tag{1}$$

où f est une fonction de $\mathbb{R} \times \mathbb{R}^q \times \mathbb{R}^q$ dans $\mathbb{R}^q$, $\dot{\overrightarrow{\mathbf{\Omega}}}$ est le vecteur des dérivées premières des composantes de $\overrightarrow{\mathbf{\Omega}}$ par rapport au temps [1] et $\ddot{\overrightarrow{\mathbf{\Omega}}}$ celui des dérivées secondes.

Pour les boules de billard, l'équation d'évolution est simplement l'équation différentielle décrivant les lois newtoniennes du mouvement : si l'on connaît la somme $\overrightarrow{\boldsymbol{F}}$ des forces extérieures qui s'exercent sur la boule ainsi que leur moment $\overrightarrow{\boldsymbol{\mathcal{M}}}$ (lesquels pourraient dépendre du temps t, de la position ou de la vitesse), alors :

$$\begin{aligned} m\ (\ddot{x}, \ddot{y}, \ddot{z}) &= \overrightarrow{\boldsymbol{F}}(t, x, y, z, \alpha, \beta, \gamma, \dot{x}, \dot{y}, \dot{z}, \dot{\alpha}, \dot{\beta}, \dot{\gamma}) \\ J\ \left(\ddot{\alpha}, \ddot{\beta}, \ddot{\gamma}\right) &= \overrightarrow{\boldsymbol{\mathcal{M}}}(t, x, y, z, \alpha, \beta, \gamma, \dot{x}, \dot{y}, \dot{z}, \dot{\alpha}, \dot{\beta}, \dot{\gamma}) \end{aligned}$$

que l'on peut donc regrouper en [2] :

$$f(t, \overrightarrow{\mathbf{\Omega}}, \dot{\overrightarrow{\mathbf{\Omega}}}) = \left(\frac{1}{m}\overrightarrow{\boldsymbol{F}}(t, \overrightarrow{\mathbf{\Omega}}, \dot{\overrightarrow{\mathbf{\Omega}}}), \frac{1}{J}\overrightarrow{\boldsymbol{\mathcal{M}}}(t, \overrightarrow{\mathbf{\Omega}}, \dot{\overrightarrow{\mathbf{\Omega}}})\right)$$

où m est la masse de la boule et J son moment inertie, $J = \frac{2}{5} m R^2$, si R est le rayon de la boule.

2.2 Forces

Hors chocs (traités plus loin), les boules subiront comme forces :

— leur poids, $m\,\overrightarrow{\boldsymbol{g}}$;
— la réaction de la table, y compris des frottements (de translation et de rotation).

Pour les frottements, nous n'allons pas ici tenter une modélisation fine, mais simplement ajouter une petite force qui s'oppose aux mouvements de translation et de rotation. Soit

1. Nous utilisons ici ces notations pour éviter toute ambiguïté, puisque, pour un vecteur en général, sa dérivé $\dot{\overrightarrow{\mathbf{\Omega}}}$ n'est pas nécessairement égale au vecteur $\dot{\overrightarrow{\mathbf{\Omega}}}$ des dérivées de ses composantes ; en effet, dans le cas général $\dot{\overrightarrow{\mathbf{\Omega}}} = \dot{\overrightarrow{\mathbf{\Omega}}} + \overrightarrow{\boldsymbol{\omega_e}} \wedge \overrightarrow{\mathbf{\Omega}}$ où $\overrightarrow{\boldsymbol{\omega_e}}$ est le vecteur de rotation du repère dans lequel $\overrightarrow{\mathbf{\Omega}}$ est exprimé. Comme par la suite nous ne travaillerons que dans un repère fixe, ces deux notations seront finalement équivalentes.
2. La preuve est donnée en section 2.3.

$\overrightarrow{v}_C$ la vitesse de la boule au point de contact, i.e.

$$\overrightarrow{v}_C = \overrightarrow{v} + R\,\overrightarrow{n} \wedge \overrightarrow{\omega}$$

où $\overrightarrow{v}$ est la vitesse (du centre de gravité) de la boule, $\overrightarrow{\omega}$ sa vitesse de rotation, R son rayon et $\overrightarrow{n}$ la normale au point de contact.

Pour rappel, la condition de roulement sans glissement est donnée par le fait que cette vitesse est nulle [3].

S'il y a glissement, le frottement de glissement sera :

$$\overrightarrow{f} = -m\,g\,f_{\text{gl}}\,\left\{\overrightarrow{v}_C\right\}_{\mathbf{1}}$$

où f_{gl} est un coefficient à fixer, « coefficient (de frottement) de glissement » ;
et son moment sera, bien sûr, $\overrightarrow{f} \wedge \left(R\,\overrightarrow{n}\right)$;

S'il y a roulement sans glissement, le frottement de roulement sera :

$$\overrightarrow{f} = -m\,g\,f_{\text{r}}\,\left\{\overrightarrow{v}\right\}_{\mathbf{1}}$$

où f_{r} est un coefficient à fixer, « coefficient (de frottement) de roulement » ; attention, notez bien que c'est $\overrightarrow{v}$ ici, et non pas $\overrightarrow{v}_C$ (qui est nulle !) ;
le moment de frottement sera dans ce cas : $-m\,g\,\mu_{\text{r}}\,\left\{\overrightarrow{\omega}\right\}_{\mathbf{1}}$, où μ_{r} est un coefficient à fixer.

2.3 Démonstration des formules physiques

Note : vous pouvez bien sûr passer cette section si vous le souhaitez.

Soit $\mathcal{R}_O = (O, \overrightarrow{\imath}, \overrightarrow{\jmath}, \overrightarrow{k})$ le repère galiléen de référence.

Pour représenter les trois degrés de liberté angulaire du système, il est largement préférable pour notre simulation d'utiliser les trois angles de rotation par rapport au repère fixe de la table (voir figure 1) plutôt que les angles d'Euler usuels, qui sont dans notre cas peu pratiques pour exprimer les équations et ne sont pas stables numériquement (non continuité, non unicité, indéterminés dans certains cas). Nous utiliserons donc les angles :

- α, rotation autour de l'axe $\overrightarrow{\imath}$;
- β, rotation autour de l'axe $\overrightarrow{\jmath}$;
- γ, rotation autour de l'axe $\overrightarrow{k}$.

Note : ces angles sont bien sûr définis à 2π près, par exemple entre 0 et 2π (ou, au choix, entre $-\pi$ et π). Si l'on utilise pour chacun toute la plage de taille 2π, alors les représentations angulaires ne sont pas uniques (plusieurs triplets d'angles pour la même position) ;

3. Numériquement, bien sûr, on testera cette égalité à un petit seuil près.

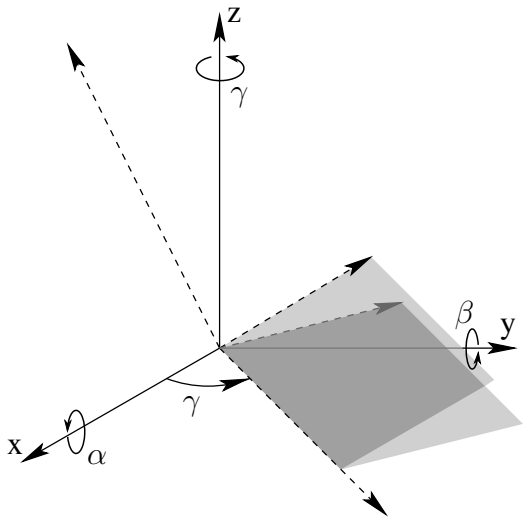


FIGURE 1 – Trois degrés de liberté angulaire, référencés ici par rapport au repère principal (fixe).

mais l'ambiguïté résultante n'étant pas un souci pour notre simulation, nous garderons ces plages afin d'éviter les discontinuités de calcul.

La rotation instantanée du solide est alors donnée par

$$\overrightarrow{\boldsymbol{\omega}} = \dot{\alpha}\,\overrightarrow{\boldsymbol{\imath}} + \dot{\beta}\,\overrightarrow{\boldsymbol{\jmath}} + \dot{\gamma}\,\overrightarrow{\boldsymbol{k}}$$

Par ailleurs, en notant $\overrightarrow{\dot{\boldsymbol{\omega}}}$ le vecteur dont les coordonnées sont les $\dot{\omega}_i$, on a :

$$\overrightarrow{\dot{\boldsymbol{\omega}}} = \ddot{\alpha}\,\overrightarrow{\boldsymbol{\imath}} + \ddot{\beta}\,\overrightarrow{\boldsymbol{\jmath}} + \ddot{\gamma}\,\overrightarrow{\boldsymbol{k}}$$

qui est égal au vecteur $\dot{\overrightarrow{\boldsymbol{\omega}}}$ puisque nous sommes dans un repère galiléen. Rappel : sinon on a, dans le cas général :

$$\dot{\overrightarrow{\boldsymbol{\omega}}} = \overrightarrow{\dot{\boldsymbol{\omega}}} + \overrightarrow{\boldsymbol{\omega_e}} \wedge \overrightarrow{\boldsymbol{\omega}}$$

en notant $\overrightarrow{\boldsymbol{\omega_e}}$ le vecteur de rotation du repère.

Comme dans le cas d'une sphère on a simplement (dérivée du moment cinétique) :

$$\dot{\overrightarrow{\boldsymbol{L}}} = J\,\overrightarrow{\dot{\boldsymbol{\omega}}} \ ,$$

en appliquant le théorème du moment cinétique ($\dot{\overrightarrow{\boldsymbol{L}}} = \overrightarrow{\boldsymbol{\mathcal{M}}}$), on trouve

$$\overrightarrow{\dot{\boldsymbol{\omega}}} = \frac{1}{J}\,\overrightarrow{\boldsymbol{\mathcal{M}}}$$

3 Evolution du système

Le système à simuler est composé :
- d'obstacles fixes (parois, tapis) ;
- et de boules, mobiles, subissant des forces et pouvant se heurter aux obstacles et aux autres boules.

Pour simuler un tel système, on utilisera l'algorithme suivant :

1. arrêter toutes les boules quasi immobiles : itérer sur les boules et tester si leur énergie cinétique est plus petite qu'un seuil et que la force subie est presque nulle ;
2. (re)calculer les forces qui s'appliquent à chaque boule : partir du poids et y ajouter la réaction de la table (voir section 2.2) ;
3. gérer toutes les collisions :
   - initialiser à 0 le « pas de temps effectivement effectué » (que je vais noter « $\mathrm{d}t_{\text{effectué}}$ »), ainsi que le fait que, pour le moment, nous n'avons pas de boule ni d'objet en collision ;
   - tant qu'on détecte une collision (c.-à-d. tant que l'on a une boule et un objet en collision, boucle *a posteriori*) :
     - parcourir toutes les boules et tous les objets « collisionnables »
       - si la boule n'est pas l'objet « collisionnable » considéré : chercher s'il y a collision et calculer le temps de collision (voir section suivante) ;
       - si il y a collision, et si celle-ci intervient avant la dernière collision détectée (ce tour ci), « noter » la boule et l'objet en collision (et leur temps de collision) ;
         on peut améliorer ce point en détectant les collisions multiples (même temps de collision) et mémorisant toutes les paires d'objets impliquées dans un choc à un même instant ; mais ceci est optionnel car un peu plus avancé ;
     - (une fois ces deux boucles terminées) si l'on a détecté une collision (il s'agit de celle se produisant « le plus tôt ») : effectuer un sous-pas de temps jusqu'à celle ci, puis gérez la collision (voir section suivante), et mettez à jour $\mathrm{d}t_{\text{effectué}}$;
4. finir le pas de temps : i.e. intégrer (voir plus loin, 5) sur le sous-pas de temps restant entre le temps de la dernière collision et le pas de temps de départ (par exemple si le pas de temps est de 0.1 et que la dernière collision a eu lieu à $t + 0.07$, il faut encore intégrer sur 0.03) ; en clair intégrer sur $\mathrm{d}t - \mathrm{d}t_{\text{effectué}}$ si $\mathrm{d}t_{\text{effectué}}$ est inférieur à $\mathrm{d}t$;
5. optionnel : séparer les boules qui s'interpénètrent.

Pour des remarques et améliorations possibles de cet algorithme, voir l'annexe C.

4 Gestion des chocs

Il y a deux aspects différents dans la gestion des chocs dans un programme :
- la détection des collisions ;
- le calcul des conséquences d'un choc.

4.1 Détection des collisions

Il existe de nombreuses techniques de gestion de collisions suivant deux grands principes :
- éviter les collisions (« *Continuous Colision Detection* ») : l'idée est d'estimer le prochain temps de contact et de n'effectuer le mouvement que jusqu'à ce moment, puis gérer le choc « juste avant qu'il ne se produise » ;
- laisser le mouvement se faire sur Δt, puis constater les chocs (objets interpénétrés) (« *Penetration Depth* ») et corriger le choc *a posteriori*.

Dans ce projet, nous allons appliquer la première technique en utilisant le « *Conservative Advancement* » dont l'idée consiste à adapter le pas de temps de calcul Δt de sorte à se mettre à la limite de la prochaine collision, que l'on gère ensuite (sous-section suivante).

L'algorithme pour calculer le temps de la prochaine collision entre deux objets *mobiles* de positions $\overrightarrow{x}_1$, $\overrightarrow{x}_2$, vitesses $\overrightarrow{v}_1$, $\overrightarrow{v}_2$ et rayons R_1, R_2, est le suivant :
- on calcule leur position au prochain pas de calcul Δt (à l'aide de l'intégrateur, cf section 5), disons $\overrightarrow{x}_1{}'$, $\overrightarrow{x}_2{}'$,
  et l'on va interpoler linéairement les trajectoires pour estimer le temps de contact (s'il y en a un) ;
- on résout pour cela (en τ) l'équation du second degré

$$A\,\tau^2 + B\,\tau + C = 0$$

  où : $A = \|\overrightarrow{b}\|^2$, $B = 2\,\overrightarrow{a}\cdot\overrightarrow{b}$, et $C = \|\overrightarrow{a}\|^2 - (R_1+R_2)^2$, avec $\overrightarrow{a} = \overrightarrow{x}_2 - \overrightarrow{x}_1$ et $\overrightarrow{b} = \overrightarrow{x}_2{}' - \overrightarrow{x}_1{}' - \overrightarrow{a}$
- si l'équation a deux solutions, on garde la plus petite des positives ;
- si la solution trouvée est négative ou plus grande que 1, c'est qu'il n'y a pas de collision ;
- si l'on a trouvé une solution positive, il y aura collision au temps $\tau\,\Delta t$.

Pour le choc entre un objet mobile et un objet immobile, la démarche est similaire mais l'équation à résoudre plus simple :
- on calcule la position $\overrightarrow{x}_1{}'$ de l'objet mobile au prochain pas de calcul Δt, ainsi que les écart (vectoriels) $\overrightarrow{D}$ et $\overrightarrow{D'}$ de l'objet mobile à l'obstacle (voir annexe B) avant et après le prochain pas de calcul ;
- si il y a collision, c.-à-d. si soit $\|\overrightarrow{D'}\| < R_1$ (trop près) soit $\overrightarrow{D}\cdot\overrightarrow{D'} < 0$ (passé de

l'autre coté), alors le temps de contact est donné par $\tau\,\Delta t$ avec :

$$\tau = \frac{R_1 - \overrightarrow{\boldsymbol{n}} \cdot \overrightarrow{\boldsymbol{D}}}{\overrightarrow{\boldsymbol{n}} \cdot \overrightarrow{\boldsymbol{D'}} - \overrightarrow{\boldsymbol{n}} \cdot \overrightarrow{\boldsymbol{D}}}$$

Note : dans le cas du sol, on ajoutera encore une condition qui plaque la boule au sol lorsque l'énergie cinétique est faible, de sorte à éviter de simuler plein de minuscules rebonds.

4.2 Choc avec objets sphériques à 6 degrés de liberté

Les paramètres de la théorie des chocs et des frottements solides (formules de Coulomb) sont :

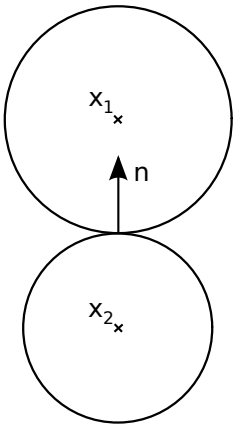
- le coefficient α de restitution (perte d'énergie dans les chocs) : $0 \leq \alpha \leq 1$; $\alpha = 1$ correspond aux chocs élastiques, $\alpha = 0$ à un choc totalement mou (écrasement) ;
- le coefficient μ de frottement entre corps ($\mu \geq 0$).
  Même si ce n'est pas très réaliste du point de vue physique, on supposera α et μ de la forme $\alpha_1 \cdot \alpha_2$ (et $\mu_1 \cdot \mu_2$) pour avoir différents types de contacts entre différentes surfaces.

Notations :

- $\overrightarrow{\boldsymbol{x}}_i$ position de l'objet i, m_i sa masse, etc.
- $\overrightarrow{\boldsymbol{n}}$ vecteur (unitaire) normal au point de choc, dirigé vers (i.e. dans) le corps 1 :

$$\overrightarrow{\boldsymbol{n}} = \left\{ \overrightarrow{\boldsymbol{x}}_1 - \overrightarrow{\boldsymbol{x}}_2 \right\}_{\mathbf{1}}$$

- $\lambda = (1+\alpha)\dfrac{m_2}{m_1 + m_2}$



Pour gérer le choc entre deux objets sphériques (de rayons respectifs R_1 et R_2, positions $\overrightarrow{\boldsymbol{x}}_1$ et $\overrightarrow{\boldsymbol{x}}_2$, et rotations $\overrightarrow{\boldsymbol{\omega}}_1$ et $\overrightarrow{\boldsymbol{\omega}}_2$), on utilisera l'algorithme suivant (lorsqu'il y a choc ! i.e. lorsque $\| \overrightarrow{\boldsymbol{x}}_1 - \overrightarrow{\boldsymbol{x}}_2 \| \leq R_1 + R_2$).

On considérera que le choc est instantané (i.e. se produit sur un temps nettement inférieur à Δt) de sorte que la mise à jour est vraiment un changement d'état (discontinuité) et non pas une équation d'évolution temporelle (i.e. pas de Δt dans ces équations) :

1. calculez

   $$v^* = (\overrightarrow{\boldsymbol{v}}_2 - \overrightarrow{\boldsymbol{v}}_1) \cdot \overrightarrow{\boldsymbol{n}}$$

   et la vitesse relative du point de contact

   $$\overrightarrow{\boldsymbol{v}}_c = \overrightarrow{\boldsymbol{v}}_1 - \overrightarrow{\boldsymbol{v}}_2 + v^* \overrightarrow{\boldsymbol{n}} + \overrightarrow{\boldsymbol{n}} \wedge \left(R_1 \overrightarrow{\boldsymbol{\omega}}_1 + R_2 \overrightarrow{\boldsymbol{\omega}}_2 \right)$$

2. si $7\mu(1+\alpha)v^* \geq 2v_c$:

$$\Delta\overrightarrow{\boldsymbol{v}} = \lambda v^* \overrightarrow{\boldsymbol{n}} - \frac{2m_2}{7(m_1+m_2)}\overrightarrow{\boldsymbol{v}}_c$$
$$\Delta\overrightarrow{\boldsymbol{\omega}} = \frac{5m_2}{7R_1(m_1+m_2)}\left(\overrightarrow{\boldsymbol{n}} \wedge \overrightarrow{\boldsymbol{v}}_c\right)$$

3. sinon ($\overrightarrow{\boldsymbol{v}}_c$ forcément non nul) :

$$\Delta\overrightarrow{\boldsymbol{v}} = \lambda v^* \left(\overrightarrow{\boldsymbol{n}} - \mu\left\{\overrightarrow{\boldsymbol{v}}_c\right\}_{\mathbf{1}}\right)$$
$$\Delta\overrightarrow{\boldsymbol{\omega}} = \frac{5}{2R_1}\mu\lambda v^* \left(\overrightarrow{\boldsymbol{n}} \wedge \left\{\overrightarrow{\boldsymbol{v}}_c\right\}_{\mathbf{1}}\right)$$

4. mise à jour :

$$\begin{aligned}
\overrightarrow{\boldsymbol{v}}_1 &= \overrightarrow{\boldsymbol{v}}_1 + \Delta\overrightarrow{\boldsymbol{v}} \\
\overrightarrow{\boldsymbol{\omega}}_1 &= \overrightarrow{\boldsymbol{\omega}}_1 + \Delta\overrightarrow{\boldsymbol{\omega}} \\
\overrightarrow{\boldsymbol{v}}_2 &= \overrightarrow{\boldsymbol{v}}_2 - \frac{m_1}{m_2}\Delta\overrightarrow{\boldsymbol{v}} \\
\overrightarrow{\boldsymbol{\omega}}_2 &= \overrightarrow{\boldsymbol{\omega}}_2 + \frac{m_1 R_1}{m_2 R_2}\Delta\overrightarrow{\boldsymbol{\omega}}
\end{aligned}$$

Pour les chocs avec les obstacles fixes (i.e. si $\|\overrightarrow{\boldsymbol{x}}_1 - \overrightarrow{\boldsymbol{x}}_2\| \leq R_1$, où cette fois $\overrightarrow{\boldsymbol{x}}_2$ est le point de l'obstacle le plus proche de la boule ; notez la différence de convention avec le cas précédent ; pour le calcul de ce point le plus proche pour différentes formes d'obstacles, voir l'annexe B) : reprendre les mêmes équations que ci-dessus, mais avec $\overrightarrow{\boldsymbol{v}}_2$ et $\overrightarrow{\boldsymbol{\omega}}_2$ nulles, et m_2 infinie (i.e. $m_1/m_2 = 0$, $m_2/(m_1+m_2) = 1$).

5 Intégration numérique

5.1 Introduction

La résolution mathématique exacte d'une équation telle que (1) page 4 est impossible dans la plupart des cas réels non triviaux. En fait, peu d'équations différentielles possèdent une solution calculable analytiquement. Pour les résoudre, il faut alors passer par l'analyse numérique, c.-à-d. le calcul approché, numérique, d'une solution particulière à l'équation différentielle pour des conditions initiales données. On parle d'« *intégration numérique* ». C'est un sujet délicat dont nous ne présentons ici que quelques solutions possibles, assez simples.

Le schéma général est le suivant :

$$\left.\begin{array}{c} \Delta t \\ \overrightarrow{\mathbf{\Omega}}(t) \\ \dot{\overrightarrow{\mathbf{\Omega}}}(t) \\ f \end{array}\right\} \longrightarrow \boxed{\text{Intégrateur}} \longrightarrow \left\{\begin{array}{l} \overrightarrow{\mathbf{\Omega}}(t+\Delta t) \\ \dot{\overrightarrow{\mathbf{\Omega}}}(t+\Delta t) \end{array}\right.$$

c.-à-d. que l'on fait appel à un module, l'« intégrateur », qui à partir de l'état courant de l'objet calcule son nouvel état un temps Δt plus tard.

Il existe de nombreux moyens d'intégrer numériquement une équation différentielle. Le moyen le plus simple, dont nous allons nous servir ici pour illustrer les principes, est la méthode dite « d'Euler ». Dans ce projet vous implémenterez dans un premier temps l'intégrateur d'Euler-Cromer (section suivante), et ensuite un autre intégrateur parmi ceux présentés plus loin. Mais commençons par le plus simple...

Imaginons que nous ayons une équation différentielle du second ordre, que l'on écrit sous la forme :

$$\ddot{\overrightarrow{\mathbf{\Omega}}} = f(t, \overrightarrow{\mathbf{\Omega}}, \dot{\overrightarrow{\mathbf{\Omega}}})$$

où $\overrightarrow{\mathbf{\Omega}} \in \mathbb{R}^q$ est un vecteur (les « degrés de liberté » en Physique, cf annexe A), $\ddot{\overrightarrow{\mathbf{\Omega}}} = \dfrac{\mathrm{d}^2\overrightarrow{\mathbf{\Omega}}}{\mathrm{d}t^2}$, $\dot{\overrightarrow{\mathbf{\Omega}}} = \dfrac{\mathrm{d}\overrightarrow{\mathbf{\Omega}}}{\mathrm{d}t}$ (dans un repère fixe) et f est une fonction vectorielle de $\mathbb{R} \times \mathbb{R}^q \times \mathbb{R}^q$ dans $\mathbb{R}^q$ (qui en Physique représente typiquement les équations du mouvement).

Imaginons également que nous connaissions des conditions initiales i.e. les valeurs $\overrightarrow{\mathbf{\Omega}}(T_0)$ et $\dot{\overrightarrow{\mathbf{\Omega}}}(T_0)$ de $\overrightarrow{\mathbf{\Omega}}$ et $\dot{\overrightarrow{\mathbf{\Omega}}}$ au temps T_0 . On peut alors utiliser un développement limité au premier ordre pour trouver la valeur de $\overrightarrow{\mathbf{\Omega}}$ à un temps $t + \Delta t$:

$$\overrightarrow{\mathbf{\Omega}}(T_0 + \Delta t) \simeq \overrightarrow{\mathbf{\Omega}}(T_0) + \Delta t \cdot \dot{\overrightarrow{\mathbf{\Omega}}}(T_0)$$

Par le même raisonnement, on peut écrire pour la dérivée :

$$\begin{aligned} \dot{\overrightarrow{\mathbf{\Omega}}}(T_0 + \Delta t) &\simeq \dot{\overrightarrow{\mathbf{\Omega}}}(T_0) + \Delta t \cdot \ddot{\overrightarrow{\mathbf{\Omega}}}(T_0) \\ &= \dot{\overrightarrow{\mathbf{\Omega}}}(T_0) + \Delta t \cdot f(T_0, \overrightarrow{\mathbf{\Omega}}(T_0), \dot{\overrightarrow{\mathbf{\Omega}}}(T_0)) \end{aligned}$$

On peut alors, ainsi de suite, en répétant les approximations précédentes de proche en proche, déterminer de cette manière la valeur de $\overrightarrow{\mathbf{\Omega}}$ et $\dot{\overrightarrow{\mathbf{\Omega}}}$ pour tout temps t de la forme $T_0 + n\Delta t$.

Cette méthode utilise un « *pas de temps* » Δt qu'il faut le choisir suffisamment petit pour que l'approximation faite dans le développement limité soit « raisonnable ». Ceci peut faire

l'objet de nombreuses études qui seront le sujet de votre cours d'analyse numérique. Dans notre cas, un choix arbitraire fixe (donné le moment voulu dans la donnée hebdomadaire) sera suffisant.

Nous présentons maintenant trois méthodes différentes, de la plus simple à la plus compliquée pour résoudre numériquement une équation différentielle du second ordre telle que :

$$\ddot{\overrightarrow{\mathbf{\Omega}}} = f(t, \overrightarrow{\mathbf{\Omega}}, \dot{\overrightarrow{\mathbf{\Omega}}}) \qquad (\overrightarrow{\mathbf{\Omega}} \in \mathbb{R}^q)$$

On notera $T_n = T_0 + n\Delta t$ et $\overrightarrow{\mathbf{\Omega}}^{(n)}$ le vecteur $\overrightarrow{\mathbf{\Omega}}(T_n)$.

Pour ceux que ça intéresse, les aspects (avancés) concernant la conservation de l'énergie des systèmes physiques ainsi simulés sont présentés en annexe D.

5.2 Schéma d'Euler-Cromer

Calculer $\overrightarrow{\mathbf{\Omega}}^{(n)}$ et $\dot{\overrightarrow{\mathbf{\Omega}}}^{(n)}$ comme suit :

$$\begin{aligned}
\dot{\overrightarrow{\mathbf{\Omega}}}^{(n)} &= \dot{\overrightarrow{\mathbf{\Omega}}}^{(n-1)} + \Delta t \cdot f(T_{n-1}, \overrightarrow{\mathbf{\Omega}}^{(n-1)}, \dot{\overrightarrow{\mathbf{\Omega}}}^{(n-1)}) \\
\overrightarrow{\mathbf{\Omega}}^{(n)} &= \overrightarrow{\mathbf{\Omega}}^{(n-1)} + \Delta t \cdot \dot{\overrightarrow{\mathbf{\Omega}}}^{(n)}
\end{aligned}$$

À noter la différence avec le schéma d'Euler classique présenté plus haut qui utilise l'*ancienne* version $\dot{\overrightarrow{\mathbf{\Omega}}}^{(n-1)}$ de la vitesse dans la seconde équation.

La méthode d'Euler-Cromer est meilleure que la méthode d'Euler classique car elle garantit la conservation de l'énergie (on parle d'« *intégrateur symplectique* », cf section D pour ceux que cela intéresse).

5.3 Méthode de Newmark

Cette méthode nécessite l'emploi de variables supplémentaires pour mémoriser les calculs intermédiaires : ξ, χ, χ'. Elle utilise aussi un paramètre ε pour contrôler la convergence à chaque pas de calcul (on parle de méthode « *implicite* », cf section D pour ceux que cela intéresse).

Initialisation :

- $\dot{\overrightarrow{\mathbf{\Omega}}}^{(n)} = \dot{\overrightarrow{\mathbf{\Omega}}}^{(n-1)}$
- $\overrightarrow{\mathbf{\Omega}}^{(n)} = \overrightarrow{\mathbf{\Omega}}^{(n-1)}$

— $\chi' = f(T_{n-1}, \overrightarrow{\mathbf{\Omega}}^{(n-1)}, \dot{\overrightarrow{\mathbf{\Omega}}}^{(n-1)})$

Répéter :

— $\xi = \overrightarrow{\mathbf{\Omega}}^{(n)}$

— $\chi = f(T_n, \overrightarrow{\mathbf{\Omega}}^{(n)}, \dot{\overrightarrow{\mathbf{\Omega}}}^{(n)})$

— $\dot{\overrightarrow{\mathbf{\Omega}}}^{(n)} = \dot{\overrightarrow{\mathbf{\Omega}}}^{(n-1)} + \frac{\Delta t}{2}(\chi + \chi')$

— $\overrightarrow{\mathbf{\Omega}}^{(n)} = \overrightarrow{\mathbf{\Omega}}^{(n-1)} + \Delta t \cdot \dot{\overrightarrow{\mathbf{\Omega}}}^{(n-1)} + \frac{\Delta t^2}{3}(\frac{1}{2}\chi + \chi')$

tant que $\|\overrightarrow{\mathbf{\Omega}}^{(n)} - \xi\| \geq \varepsilon$

5.4 Runge-Kutta d'ordre 4

Cette méthode nécessite huit variables intermédiaires : k_1, k_2, k_3, k_4, k'_1, k'_2, k'_3 et k'_4.

Calculer $\overrightarrow{\mathbf{\Omega}}^{(n)}$ et $\dot{\overrightarrow{\mathbf{\Omega}}}^{(n)}$ comme suit :

$$\overrightarrow{\mathbf{\Omega}}^{(n)} = \overrightarrow{\mathbf{\Omega}}^{(n-1)} + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$
$$\dot{\overrightarrow{\mathbf{\Omega}}}^{(n)} = \dot{\overrightarrow{\mathbf{\Omega}}}^{(n-1)} + \frac{\Delta t}{6}(k'_1 + 2k'_2 + 2k'_3 + k'_4)$$

avec :

$$k_1 = \dot{\overrightarrow{\mathbf{\Omega}}}^{(n-1)}$$
$$k'_1 = f(T_{n-1}, \overrightarrow{\mathbf{\Omega}}^{(n-1)}, \dot{\overrightarrow{\mathbf{\Omega}}}^{(n-1)})$$
$$k_2 = \dot{\overrightarrow{\mathbf{\Omega}}}^{(n-1)} + \frac{\Delta t}{2}k'_1$$
$$k'_2 = f(T_{n-1} + \frac{\Delta t}{2}, \overrightarrow{\mathbf{\Omega}}^{(n-1)} + \frac{\Delta t}{2}k_1, \dot{\overrightarrow{\mathbf{\Omega}}}^{(n-1)} + \frac{\Delta t}{2}k'_1)$$
$$k_3 = \dot{\overrightarrow{\mathbf{\Omega}}}^{(n-1)} + \frac{\Delta t}{2}k'_2$$
$$k'_3 = f(T_{n-1} + \frac{\Delta t}{2}, \overrightarrow{\mathbf{\Omega}}^{(n-1)} + \frac{\Delta t}{2}k_2, \dot{\overrightarrow{\mathbf{\Omega}}}^{(n-1)} + \frac{\Delta t}{2}k'_2)$$
$$k_4 = \dot{\overrightarrow{\mathbf{\Omega}}}^{(n-1)} + \Delta t\, k'_3$$
$$k'_4 = f(T_n, \overrightarrow{\mathbf{\Omega}}^{(n-1)} + \Delta t\, k_3, \dot{\overrightarrow{\mathbf{\Omega}}}^{(n-1)} + \Delta t\, k'_3)$$

Note : $T_n = T_{n-1} + \Delta t$.

Bibliographie

[1] *Mécanique*, Jean-Philippe Ansermet, PPUR 2009.
[2] *Mécanique générale*, Christian Gruber & Willy Benoit, PPUR 1998.

A Notion de degrés de liberté

En physique, le nombre de degrés de liberté d'un système correspond au nombre de déplacements *indépendants* qu'il peut effectuer :

— pour un point matériel libre de se déplacer dans tout l'espace, on compte 3 degrés de liberté, car il peut aller dans toutes les directions ;
— le centre de masse d'une boule qui reste sur un billard (i.e. on ne considère pas ici les rotations de la boule sur elle-même, et on suppose qu'elle ne saute pas) ne possède que deux degrés de liberté, car la boule est astreinte à se déplacer dans un plan ; la boule elle-même a cinq degrés de liberté (on ajoute 3 degrés de liberté de rotation), voire 6 si on autorise la boule à sauter ;
— un wagonnet de « grand-huit » ne possède qu'un seul degré de liberté car il est obligé de suivre les rails et ne peut aller qu'« en avant » ou « en arrière », même si en réalité il se déplace dans un espace à 3 dimensions.

Le nombre de degrés de liberté d'un système correspond ainsi au nombres de variables *indépendantes* qu'il faut pour décrire l'état de ce système. L'état d'un système à un seul degré de liberté peut se représenter à l'aide d'une seule variable réelle. Par exemple, pour décrire la position d'un wagonnet de « grand-huit », il suffit d'une seule variable, qui serait par exemple, la distance parcourue (suivant les rails) depuis la station de départ (abscisse curviligne).

Une telle variable n'est pas forcément une distance, il peut également s'agir d'un angle ou de tout autre moyen permettant de représenter de manière univoque toutes les positions que peut prendre l'objet étudié.

Au premier abord, on peut penser que l'on a inutilement compliqué la chose et perdu de l'information concernant la position « réelle » de l'objet dans l'espace. On n'a cependant rien perdu comme information, mais on a par contre gagné en simplicité puisque nous n'avons plus besoin de manipuler tous les vecteurs de toutes les composantes dans l'équation du mouvement, mais uniquement ceux qui sont nécessaires, *a priori* en nombre plus réduit. Le gain ne paraît pas important dans le cas d'un simple pendule, mais il l'est plus dans le cas de systèmes composés complexes ayant plusieurs degrés de liberté mais nettement moins que de composantes.

B Point le plus proche de différents obstacles

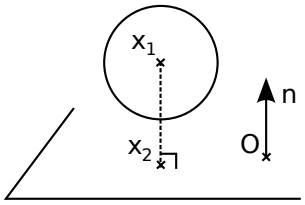
B.1 Sphère

Dans le cas d'une sphère de centre $\overrightarrow{O}$ et de rayon R, le point $\overrightarrow{x}_2$ le plus proche sur cette sphère de l'objet sphérique de centre $\overrightarrow{x}_1$ est donné par :

$$\overrightarrow{x}_2 = \overrightarrow{O} + R \left\{ \overrightarrow{x}_1 - \overrightarrow{O} \right\}_1$$

B.2 Plan infini

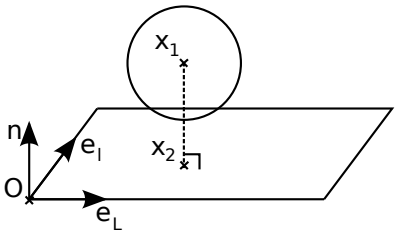
Dans le cas d'un plan infini défini par un point $\overrightarrow{O}$ et un vecteur normal $\overrightarrow{n}$ (unitaire), le point $\overrightarrow{x}_2$ le plus proche sur ce plan de l'objet sphérique de centre $\overrightarrow{x}_1$ s'exprime comme :



$$\overrightarrow{x}_2 = \overrightarrow{x}_1 + \left((\overrightarrow{O} - \overrightarrow{x}_1) \cdot \overrightarrow{n} \right) \overrightarrow{n}$$

B.3 Portion de plan

Dans le cas d'une portion rectangulaire de plan d'origine $\overrightarrow{O}$, de normale $\overrightarrow{n}$ (unitaire), de longueur L suivant la direction $\overrightarrow{e_L}$ (unitaire et orthogonal à $\overrightarrow{n}$) et de largeur l suivant la direction $\overrightarrow{e_l}$ ($\overrightarrow{e_l} = \overrightarrow{n} \wedge \overrightarrow{e_L}$), le point $\overrightarrow{x}_2$ le plus proche sur ce plan de l'objet sphérique de centre $\overrightarrow{x}_1$ se calcule comme suit :



1. Commencez par calculer le point le plus proche dans le plan infini comme précédemment :
$$\overrightarrow{x}_2 \longleftarrow \overrightarrow{x}_1 + \left((\overrightarrow{O} - \overrightarrow{x}_1) \cdot \overrightarrow{n} \right) \overrightarrow{n}.$$
2. Calculez ensuite ses coordonnées suivant $\overrightarrow{e_L}$ et $\overrightarrow{e_l}$:
$$x_{kL} = (\overrightarrow{x}_2 - \overrightarrow{O}) \cdot \overrightarrow{e_L},$$
$$x_{kl} = (\overrightarrow{x}_2 - \overrightarrow{O}) \cdot \overrightarrow{e_l}.$$

3. Si $x_{kL} > L$

$$\overrightarrow{\boldsymbol{x}}_2 \longleftarrow \overrightarrow{\boldsymbol{x}}_2 - (x_{kL} - L)\,\overrightarrow{\boldsymbol{e_L}},$$

sinon, si $x_{kL} < 0$

$$\overrightarrow{\boldsymbol{x}}_2 \longleftarrow \overrightarrow{\boldsymbol{x}}_2 - x_{kL}\,\overrightarrow{\boldsymbol{e_L}}.$$

4. Ensuite, si $x_{kl} > l$

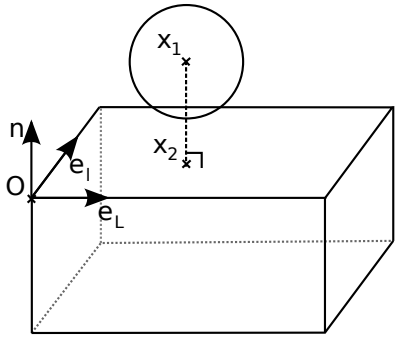
$$\overrightarrow{\boldsymbol{x}}_2 \longleftarrow \overrightarrow{\boldsymbol{x}}_2 - (x_{kl} - l)\,\overrightarrow{\boldsymbol{e_l}},$$

sinon, si $x_{kl} < 0$

$$\overrightarrow{\boldsymbol{x}}_2 \longleftarrow \overrightarrow{\boldsymbol{x}}_2 - x_{kl}\,\overrightarrow{\boldsymbol{e_l}}.$$

B.4 Brique (parallélépipède)

Dans le cas d'un parallélépipède rectangle (« brique ») d'origine $\overrightarrow{\boldsymbol{O}}$, de hauteur h suivant la direction $\overrightarrow{\boldsymbol{n}}$ (unitaire), de longueur L suivant la direction $\overrightarrow{\boldsymbol{e_L}}$ (unitaire et orthogonal à $\overrightarrow{\boldsymbol{n}}$) et de largeur l suivant la direction $\overrightarrow{\boldsymbol{e_l}}$ ($\overrightarrow{\boldsymbol{e_l}} = \overrightarrow{\boldsymbol{n}} \wedge \overrightarrow{\boldsymbol{e_L}}$), le point $\overrightarrow{\boldsymbol{x}}_2$ le plus proche sur l'objet sphérique de centre $\overrightarrow{\boldsymbol{x}}_1$ se calcule comme suit :



Calculez le point le plus proche à chacune des six faces (voir section précédente) et retenez au final le plus proche de $\overrightarrow{\boldsymbol{x}}_1$.

Attention pour cela aux caractéristiques de chacune des faces (je prends ici la convention que la normale sort de la brique et que la face est toujours du coté coordonnées positives de ses axes, ce n'est pas utile ici mais pourrait peut-être l'être ailleurs (distinction intérieur/extérieur)) :

face	origine	normale	longueur	« $\overrightarrow{\boldsymbol{e_L}}$ »	largeur	« $\overrightarrow{\boldsymbol{e_l}}$ »
1	$\overrightarrow{\boldsymbol{O}}$	$\overrightarrow{\boldsymbol{n}}$	L	$\overrightarrow{\boldsymbol{e_L}}$	l	$\overrightarrow{\boldsymbol{e_l}}$
2	$\overrightarrow{\boldsymbol{O}}$	$-\overrightarrow{\boldsymbol{e_l}}$	h	$-\overrightarrow{\boldsymbol{n}}$	L	$\overrightarrow{\boldsymbol{e_L}}$
3	$\overrightarrow{\boldsymbol{O}}$	$-\overrightarrow{\boldsymbol{e_L}}$	l	$\overrightarrow{\boldsymbol{e_l}}$	h	$-\overrightarrow{\boldsymbol{n}}$
4	$\overrightarrow{\boldsymbol{O}} + L\,\overrightarrow{\boldsymbol{e_L}}$	$\overrightarrow{\boldsymbol{e_L}}$	h	$-\overrightarrow{\boldsymbol{n}}$	l	$\overrightarrow{\boldsymbol{e_l}}$
5	$\overrightarrow{\boldsymbol{O}} + l\,\overrightarrow{\boldsymbol{e_l}}$	$\overrightarrow{\boldsymbol{e_l}}$	L	$\overrightarrow{\boldsymbol{e_L}}$	h	$-\overrightarrow{\boldsymbol{n}}$
6	$\overrightarrow{\boldsymbol{O}} - h\,\overrightarrow{\boldsymbol{n}}$	$-\overrightarrow{\boldsymbol{n}}$	l	$\overrightarrow{\boldsymbol{e_l}}$	L	$\overrightarrow{\boldsymbol{e_L}}$

C Améliorations optionnelles de l'algorithme de simulation

La version, simple, de l'algorithme de simulation présentée en section 3 n'est pas totalement parfaite. Ceux qui le souhaitent pourront essayer de l'améliorer comme suit, voire implémenter plusieurs types de moteur de simulation différents.

Note : Il s'agit bien là d'extensions optionnelles qui ne concernent éventuellement que les plus motivés d'entre vous (lesquels pourront utiliser ici le polymorphisme ou des pointeurs sur fonction de sorte à pouvoir choisir dynamiquement pendant l'exécution d'un même programme quelle méthode de simulation est utilisée. Mais ceci est vraiment pour ceux qui sont à un haut niveau !). J'insiste sur le fait qu'il serait catastrophique pour des étudiants ne se sentant pas vraiment à l'aise avec la programmation, d'essayer ici de grappiller inutilement des points inaccessibles et rendre au final un projet qui ne tient pas du tout la route !

C.1 Dissymétrie des interactions

L'algorithme n'est pas totalement exact car il rompt la symétrie des interactions (en introduisant une dépendance à l'ordre de mise à jour des objets).

Une version plus correcte, mais plus évoluée consisterait à ne faire aucune mise à jour dans l'étape 2 (mais simplement stocker ces mises à jour à faire) et n'effectuer que *toutes* les mises à jour (forces, vitesses, vitesses de rotation, et positions) dans l'étape 3. Cela suppose alors d'ajouter aux objets les attributs $\Delta\overrightarrow{\boldsymbol{F}}$, $\Delta\overrightarrow{\boldsymbol{v}}$ et $\Delta\overrightarrow{\boldsymbol{\omega}}$.

C.2 Calcul des forces de contact entre objets

Dans ce projet, nous faisons l'hypothèse que les boules sont soit en contact avec la table, soit en « chute libre », mais nous ne considérons pas des empilement de boules par exemple (forces de contact entre différents objets).

On pourrait considérer ce cas en ajoutant une boucle de calcul des forces dans la partie 1 de l'algorithme de simulation de la section 3 : pour toutes les boules en contact, il faudrait mettre à jour les forces s'exerçant sur elles (calcul des *réactions*) :

Soit $F_{n1} = \overrightarrow{\boldsymbol{F_1}} \cdot \overrightarrow{\boldsymbol{n}}$, la composante suivant la normale au choc des forces s'exerçant sur l'objet 1, et $F_{n2} = \overrightarrow{\boldsymbol{F_2}} \cdot \overrightarrow{\boldsymbol{n}}$ celle de l'objet 2.

Si $F_{n1} < 0$, alors cette partie de la force s'annule par réaction de l'autre objet :

$$\overrightarrow{\boldsymbol{F_1}} \longleftarrow \overrightarrow{\boldsymbol{F_1}} - F_{n1}\overrightarrow{\boldsymbol{n}}$$

Cette réaction s'ajoute par contre aux forces s'exerçant sur l'autre objet :

$$\overrightarrow{\boldsymbol{F_2}} \longleftarrow \overrightarrow{\boldsymbol{F_2}} + F_{n1}\overrightarrow{\boldsymbol{n}}$$

Refaire ensuite de même avec les forces de l'autre objet : si $F_{n2} = \overrightarrow{\boldsymbol{F_2}} \cdot \overrightarrow{\boldsymbol{n}} > 0$ (attention ici au changement de sens de l'inégalité !), alors

$$\overrightarrow{\boldsymbol{F_1}} \longleftarrow \overrightarrow{\boldsymbol{F_1}} + F_{n2}\overrightarrow{\boldsymbol{n}} \qquad \text{et} \qquad \overrightarrow{\boldsymbol{F_2}} \longleftarrow \overrightarrow{\boldsymbol{F_2}} - F_{n2}\overrightarrow{\boldsymbol{n}}$$

C.3 Synchronisation artificielle

Un autre problème de ce type de mise à jour, plus subtile, est que la boucle systématique exécutée dans le même ordre sur les objets induit une synchronisation artificielle du système avec une « horloge externe » (couplage).

Pour éviter ce type de phénomène, artefact de la simulation, les étapes de mise à jour devraient se faire dans un ordre aléatoire sur les objets : on choisit de tirer au hasard quel objet mettre à jour, puis on recommence ainsi de suite (en tirant éventuellement à nouveau le même objet). On parle alors de simulation Monte-Carlo, par opposition à une simulation déterministe qui considère toujours les objets dans le même ordre.

C.4 Meilleure gestion des chocs

Un troisième aspect qui n'est pas totalement satisfaisant dans l'algorithme de la section 3 résulte des erreurs de calcul (précision numérique) dans l'estimation des collisions. Il se peut en effet qu'en raison de la taille et de la vitesse des objets, la prise en compte du choc suivie de l'évolution n'empêche tout même pas de deux objets de s'interpénétrer. Normalement cela ne devrait pas se produire puisque justement on cherche à détecter le choc *avant* qu'il se produise et qu'un choc a justement pour conséquence d'éloigner les objets, mais peut être que ponctuellement ils ne s'éloignent pas assez l'un de l'autre et s'interpénètrent encore après la mise à jour.

Pour éviter cela, on peut ajouter encore une quatrième boucle qui redéplace artificiellement les objets qui sont encore interpénétrés après leur évolution. Il ne s'agit pas ici de les traiter comme de *nouveaux* chocs, mais bien seulement corriger les (petites) erreurs d'estimation de ces chocs !

Pour de tels chocs « mal traités », on déplacera simplement chacun des deux objets mobiles (resp. l'objet mobile, en cas de choc avec un obstacle, mais cela ne devrait pas arriver) d'une distance égale à la moitié de leur interpénétration.

En notant $L = R_1 + R_2$ la somme des rayons des 2 objets et e l'écart $\|\overrightarrow{\boldsymbol{x}}_1 - \overrightarrow{\boldsymbol{x}}_2\| - L$, lorsque $e < 0$ on écartera chaque objet de $e/2$ par rapport à leur milieu, i.e.

$$\overrightarrow{\boldsymbol{x}}_1 \longleftarrow \overrightarrow{\boldsymbol{x}}_1 + \frac{e}{2}\left\{\overrightarrow{\boldsymbol{x}}_2 - \overrightarrow{\boldsymbol{x}}_1\right\}_{\mathbf{1}}$$
$$\overrightarrow{\boldsymbol{x}}_2 \longleftarrow \overrightarrow{\boldsymbol{x}}_2 + \frac{e}{2}\left\{\overrightarrow{\boldsymbol{x}}_1 - \overrightarrow{\boldsymbol{x}}_2\right\}_{\mathbf{1}}$$

Attention cependant cette technique n'est pas la panacée : en déplaçant ainsi les objets on pourrait les faire pénétrer dans d'autres objets avec lesquels ils étaient aussi en choc (imaginez 2 boules en collision dans un coin de 2 murs). La seule bonne solution dans ce cas serait de réduire le pas de temps.

C.5 Encore plus loin ?

Pour encore améliorer les simulations on pourrait :

- dans la correction ci-dessus, ne pas corriger l'altitude z des boules qui sont sur la table pour ne pas mettre sous la table une boule qui interpénétreraient avec une autre lui tombant dessus ;
- dans la gestion de chocs proposée en section 4, nous gérons *le* prochain choc (calcul du temps de *la* prochaine collision) ; on commet donc l'erreur d'oublier toutes les autres collisions ayant lieu en même temps ; ceci est extrêmement improbable à cause des imprécisions numériques, mais ceux qui voudraient gérer ces cas devraient pouvoir traiter un *tableau* de collisions au lieu de n'en traiter qu'une seule.

D Compléments sur l'intégration numérique

D.1 Généralités

Cette section est totalement hors programme et certainement difficile pour des 1re années [4]. Je l'ai néanmoins ajoutée pour information pour ceux que le sujet intéresse et qui aimeraient en savoir un peu plus.

Les intégrateurs numériques sont par essence inexacts et calculent des valeurs approchées des solutions. Ils sont en cela caractérisés de trois façons :

- l'ordre de l'intégrateur, qui caractérise l'évolution (en fonction du temps de calcul) de l'erreur entre la solution calculée fournie et la solution exacte ;
  à ne pas confondre avec l'ordre de l'équation différentielle intégrée !
- l'aspect symplectique ou non de l'intégrateur, qui caractérise grosso-modo « les frottements numériques », c.-à-d. la perte d'énergie subie par le système simulé par rapport au vrai système en raison des approximations de calcul ;
  vous verrez plus tard en Physique que l'énergie d'un système est liée à ce que l'on appelle « l'hamiltonien » du système ; un intégrateur symplectique garde invariant l'hamiltonien du système simulé ;
- l'aspect « explicite » ou « implicite » : un intégrateur est « explicite » lorsque les calculs du pas suivant se font directement et il est « implicite » lorsque le calcul de chaque pas d'intégration est lui-même indirect, par approximations, et non pas exact, introduisant donc un deuxième niveau d'approximation dans le calcul.

Avant de détailler l'aspect simplectique des schémas du premier ordre, donnons un résumé synthétique des résultats connus :

- l'intégrateur d'Euler « classique » (section 5.1) est un intégrateur explicite du premier ordre non simplectique ;
- l'intégrateur d'Euler-Cromer (section 5.2) est également du premier ordre ; il est simplectique et explicite pour les hamiltoniens séparables ;
- une version d'Euler-Cromer existe pour les hamiltoniens non séparables, mais est implicite ;
- les méthodes de Runge-Kutta (section 5.4) sont non simplectiques, explicites pour les hamiltoniens séparables et existent en version implicite (non simplectique) pour les hamiltoniens non séparables ;
- la méthode de Newmark (section 5.3) est simplectique mais implicite.

4. On y parle notamment d'hamiltonien, de hessien, et autres joyeusetés que vous découvrirez plus tard dans votre scolarité.

D.2 De l'aspect symplectique des schémas « eulériens »

Soit une particule de masse m dans un potentiel $V = V(\overrightarrow{\boldsymbol{x}})$ ne dépendant que de la position ; l'accélération subie est $\overrightarrow{\boldsymbol{a}} = -\frac{1}{m}\overrightarrow{\boldsymbol{\nabla}}V$, où $\overrightarrow{\boldsymbol{\nabla}}V$ est le gradient de V.[5]

Considérons alors le schéma d'évolution suivant :

$$\overrightarrow{\boldsymbol{x}}_{n+1} = \overrightarrow{\boldsymbol{x}}_n + \Delta t\,\overrightarrow{\boldsymbol{v}}_n - \alpha\frac{\Delta t^2}{m}\overrightarrow{\boldsymbol{\nabla}}V_n \ , \qquad \overrightarrow{\boldsymbol{v}}_{n+1} = \overrightarrow{\boldsymbol{v}}_n - \frac{\Delta t}{m}\overrightarrow{\boldsymbol{\nabla}}V_n$$

où α est une constante qui vaut soit 0 (intégrateur d'Euler « classique »), soit 1/2 (développement de Taylor négligeant les dérivées de l'accélération), soit 1 (intégrateur d'Euler-Cromer).

D.2.1 Approche « à la main » sur l'énergie

Considérons alors l'évolution de l'énergie au second ordre :

$$E_{n+1} = E_n + \frac{\Delta t^2}{2}\left(\overrightarrow{\boldsymbol{v}}_n{}^T\nabla^2\mathbf{V}\,\overrightarrow{\boldsymbol{v}}_n + (1-2\alpha)\frac{1}{m}(\overrightarrow{\boldsymbol{\nabla}}V_n)^2\right) + \mathcal{O}(\Delta t^3) \ ,$$

où $\nabla^2\mathbf{V}$ est le hessien du potentiel V.

On voit bien que si $\alpha \leq \frac{1}{2}$ l'énergie ne fait qu'*augmenter*. Pour $\alpha \leq \frac{1}{2}$ l'intégrateur ne peut pas être simplectique (et donc ni Euler, ni « Taylor » ne sont simplectiques).

Illustrons cela de façon concrète sur l'oscillateur harmonique à une dimension. On a $V(x) = \frac{1}{2}\,k\,x^2$, et on connaît la solution analytique $x = x_0\cos(\omega t)$ et $v = -x_0\,\omega\sin(\omega t)$ où $\omega = \sqrt{k/m}$.

La variation d'énergie entre deux pas de calcul de l'intégrateur devient :

$$\begin{aligned}\Delta E &= \frac{\Delta t^2}{2}\left(k\,v^2 + (1-2\alpha)\frac{1}{m}k^2x^2\right) + \mathcal{O}(\Delta t^3)\\ &= \frac{\Delta t^2}{2}\frac{x_0k^2}{m}\left(\sin^2(\omega t) + (1-2\alpha)\cos^2(\omega t)\right) + \mathcal{O}(\Delta t^3)\end{aligned}$$

En intégrant sur une période entière de l'oscillation, on trouve $\Delta E \propto \pi(2-2\alpha)$, ce qui nous amène à conclure qu'il faut $\alpha = 1$ pour que l'énergie de l'oscillateur harmonique soit stable en moyenne.

5. Toute la suite peut en fait s'appliquer à un hamiltonien séparable $H(\overrightarrow{\boldsymbol{p}}, \overrightarrow{\boldsymbol{q}}) = T(\overrightarrow{\boldsymbol{p}}) + V(\overrightarrow{\boldsymbol{q}})$. Ici $\overrightarrow{\boldsymbol{p}} = m\overrightarrow{\boldsymbol{v}}$ et $\overrightarrow{\boldsymbol{q}} = \overrightarrow{\boldsymbol{x}}$.

D.2.2 Preuve formelle de simplecticité

La définition d'un intégrateur simplectique est l'orthogonalité du déplacement dans l'espace des phases par rapport au gradient de l'hamiltonien, ce qui ce traduit par l'équation suivante :

$$\left(\frac{\partial \overrightarrow{\boldsymbol{z}}_{n+1}}{\partial \overrightarrow{\boldsymbol{z}}_{n}}\right)^{T} J \left(\frac{\partial \overrightarrow{\boldsymbol{z}}_{n+1}}{\partial \overrightarrow{\boldsymbol{z}}_{n}}\right) = J$$

avec $\overrightarrow{\boldsymbol{z}} = (\overrightarrow{\boldsymbol{v}}, \overrightarrow{\boldsymbol{x}})$ (point de l'espace des phases), $J = \begin{pmatrix} 0 & \mathbb{1}_3 \\ -\mathbb{1}_3 & 0 \end{pmatrix}$, où $\mathbb{1}_3$ est la matrice identité de dimension 3, et $\left(\frac{\partial \overrightarrow{\boldsymbol{z}}_{n+1}}{\partial \overrightarrow{\boldsymbol{z}}_{n}}\right)$ la matrice jacobienne de $\overrightarrow{\boldsymbol{z}}_{n+1}$ (en tant que fonction de $\overrightarrow{\boldsymbol{z}}_{n}$).

Pour les schémas d'évolution considérés ici, on a :

$$\frac{\partial \overrightarrow{\boldsymbol{z}}_{n+1}}{\partial \overrightarrow{\boldsymbol{z}}_{n}} = \begin{pmatrix} \mathbb{1}_3 & -\frac{\Delta t}{m} \nabla^{\mathbf{2}}\mathbf{V} \\ \Delta t\, \mathbb{1}_3 & \mathbb{1}_3 - \alpha \frac{\Delta t^2}{m} \nabla^{\mathbf{2}}\mathbf{V} \end{pmatrix}$$

D'où :

$$\left(\frac{\partial \overrightarrow{\boldsymbol{z}}_{n+1}}{\partial \overrightarrow{\boldsymbol{z}}_{n}}\right)^{T} J \left(\frac{\partial \overrightarrow{\boldsymbol{z}}_{n+1}}{\partial \overrightarrow{\boldsymbol{z}}_{n}}\right) = J + (1-\alpha)\frac{\Delta t^2}{m} \begin{pmatrix} 0 & \nabla^{\mathbf{2}}\mathbf{V} \\ -\nabla^{\mathbf{2}}\mathbf{V} & 0 \end{pmatrix} .$$

Le schéma proposé n'est donc symplectique que pour $\alpha = 1$ (c.-à-d. Euler-Cromer).

D.2.3 Euler-Cromer pour les hamiltoniens non séparables

En fait, le schéma d'Euler-Cromer se généralise aux hamiltoniens non séparables, tout en restant simplectique. La généralisation est :

$$\overrightarrow{\boldsymbol{p}}_{n+1} = \overrightarrow{\boldsymbol{p}}_{n} - \Delta t \overrightarrow{\boldsymbol{\nabla}_{\boldsymbol{q}}} H(\overrightarrow{\boldsymbol{p}}_{n+1}, \overrightarrow{\boldsymbol{q}}_{n}) \ , \qquad \overrightarrow{\boldsymbol{q}}_{n+1} = \overrightarrow{\boldsymbol{q}}_{n} + \Delta t \overrightarrow{\boldsymbol{\nabla}_{\boldsymbol{p}}} H(\overrightarrow{\boldsymbol{p}}_{n+1}, \overrightarrow{\boldsymbol{q}}_{n}) \ .$$

Le problème avec ce schéma est que la première équation est *implicite* et nécessite alors souvent, en elle-même, une résolution numérique approchée (type « point fixe ») en $\overrightarrow{\boldsymbol{p}}_{n+1}$.