

Simulation numérique de structure

Etienne Thomas

14 mai 2025

Table des matières

Introduction	2
0.1 Preamble	2
0.2 Résolution sur la vitesse	3
0.2.1 Changement d'ordre	3
0.2.2 Définition de l'inconnue	3
0.2.3 Mise en équation	4
0.2.4 Stabilisation du système	4
0.2.5 Extension a un système de plusieurs contraintes	4
0.2.6 Problèmes rencontrés	5
0.3 Résolution sur l'accélération	5
0.3.1 Changement d'ordre	5
0.3.2 Mise en équation	6
0.3.3 Stabilisation du système	6
0.3.4 Problèmes rencontrés	6
0.4 Mise en place d'un solveur hybride	6
0.4.1 Gain	7
0.5 Série de Taylor	7
0.6 Contraintes usuelles	7

notations

- q : position
- \dot{q} : vitesse
- $\dot{\underline{q}}$: vitesse avant correction
- \ddot{q} : accélération
- p : la quantité de mouvement
- \underline{p} : quantité de mouvement avant correction
- Δt : pas de temps
- M : matrice de masse (matrice carrée de meme dimension que q)
- W : l'inverse de la matrice de masse
- C : contrainte

- \dot{C} : dérivée de la contrainte
- J : jacobienne de la contrainte
- F : force
- $\mathcal{P}(F)$: travail d'une force
- λ : coefficient Lagrangien

Introduction

On cherche à simuler une structure solide (pendule, ponts, ect.) dans un fluide en mouvement. La simulation de fluide est traité dans le TIPE d'Alban Coadic. On s'intéresse ici à la simulation de la structure rigide et des forces qui s'exercent sur elle.

On retrace ici les différentes étapes de la mise en place de cette simulation.

0.1 Préambule

La simulation est discrétisée. On se place évident dans un référentiel galiléen. À chaque pas de temps (tick), le programme doit faire évoluer le système en respectant les lois de Newton :

1. Un corps soumis à aucune force a un déplacement rectiligne uniforme.
2. $\dot{\vec{p}} = \sum \vec{F}_{ext}$
3. Chaque action entraine une réaction égale et opposée.

Et doit respecter des interactions entre les corps (tige d'un pendule, rail d'un train), modélisé par des équations appelées contraintes.

Par exemple, la tige d'un pendule est modélisé par :

$$C : \|\vec{q}\| - l = 0$$

Où \vec{q} le vecteur position de la masse, l la longueur de la tige.

À chaque tick, on effectue les étapes suivantes :

- Intégrer les différentes grandeurs
- Le système est alors d'un état ou les contraintes ne sont plus vérifiées. Pour retrouver un état valide, on modifie le système en appliquant des forces qui ne travaillent pas.

0.2 Résolution sur la vitesse

0.2.1 Changement d'ordre

Résoudre à l'ordre 0 (la position) est un problème non linéaire. Travailler sur la vitesse (ordre 1) est plus simple. Le programme exécute dans l'ordre les étapes suivantes :

1. Intégrer la vitesse, via la méthode d'Euler implicite :

$$\underline{\dot{q}_{n+1}} = \dot{q}_n + \Delta t W F$$

2. Corriger la vitesse

3. Intégrer la position à partir de la vitesse précédemment corrigée :

Pour obtenir une contrainte sur la vitesse, on applique la règle de la chaîne :

$$\dot{C} : \frac{dC}{dt} = \frac{\partial C}{\partial q} \dot{q}$$

Notons $J = \frac{\partial C}{\partial q}$, ainsi $\dot{C} : J\dot{q} = 0$. J est la matrice jacobienne de la fonction associée à la contrainte.

Dans l'exemple du pendule, C est une fonction de deux variables, sa jacobienne est une matrice 1×2 , ($J = \frac{q^\top}{\|q\|}$)

0.2.2 Definition de l'inconnue

Les forces de corrections recherchées ne travaillent pas : $\mathcal{P}(F_c) = F_c \cdot \dot{q}$, Remarquons que :

$$\forall \lambda \in \mathbb{R}, (J^\top \lambda) \cdot \dot{q} = (J^\top \lambda)^\top \dot{q} = \lambda J \dot{q} = 0$$

Tout force colinéaire à J ne travaille pas, par conséquent, on cherche à isoler λ pour trouver la force de correction. λ est le coefficient Lagrangien.

0.2.3 Mise en équation

$\underline{\dot{q}}$ est la vitesse avant correction, \dot{q}_c est la correction recherchée.

$$\begin{aligned}
\dot{C} &\Leftrightarrow J\dot{q} = 0 \\
&\Leftrightarrow J(\dot{q}_c + \underline{\dot{q}}) = 0 \\
&\Leftrightarrow J\dot{q}_c = -J\underline{\dot{q}} \\
&\Leftrightarrow JW P_c = -J\underline{\dot{q}} \\
&\Leftrightarrow JW \Delta t F_c = -J\underline{\dot{q}} \\
&\Leftrightarrow JW \Delta t J^\top \lambda = -J\underline{\dot{q}} \\
&\Leftrightarrow JW J^\top \lambda' = -J\underline{\dot{q}}
\end{aligned}$$

En posant $\lambda' = \Delta t \lambda$, ainsi $P_c = J^\top \lambda'$.

$JW J^\top$ peut être interprété comme l'inverse de la masse perçu par la contrainte. Dans un système avec une unique contrainte, $JW J^\top$ est une matrice 1×1 , on se ramène donc à équation polynomiale du premier degré.

0.2.4 Stabilisation du système

Corriger uniquement la vitesse accumule les erreurs de positions, pour lutter contre, on utilise la stabilisation de Baumgarte, On pose : $\dot{C}' = \dot{C} + \frac{h}{\Delta t} \dot{C}$ où $h \in [0, 1]$ est un facteur d'amortissement. $h = 1$ correspond une correction en 1 tick. L'expression finale est :

$$JW J^\top \lambda' = -J\underline{\dot{q}} - \frac{h}{\Delta t} C$$

0.2.5 Extension a un système de plusieurs contraintes

Pour étendre la méthode à n objets et p contraintes, il faut considérer l'ensemble des vecteurs position comme un unique vecteur q de dimension $n \cdot d$, où d est la dimension de l'espace. Les contraintes sont alors rassemblées dans une application :

$$C : q \mapsto \begin{pmatrix} C_1 \\ \vdots \\ C_p \end{pmatrix}$$

— La jacobienne J devient une matrice $p \times (n \cdot d)$.

- $JWJ^\top \lambda' = -J\dot{q} - \frac{h}{\Delta t} \dot{C}$ est un système d'équations linéaires à p inconnues, que l'on peut résoudre par à l'aide du pivot de Gauss.
- JWJ^\top est inversible s'il n'y a pas de redondance entre les contraintes ou de contrainte nulle.
- On peut observer la 3^e loi de Newton dans les coefficients de J , une contrainte reliant deux corps entre eux appliquera une force sur les deux corps.
- Les contraintes affectent le déplacement des objets et donc les autres contraintes. Il est naturel de trouver un système linéaire les liant.

0.2.6 Problèmes rencontrés

Les simulations avec cette méthode ont tendance à perdre de l'énergie lors de forte inflexion dans l'accélération/période de forte correction du solveur. La correction de Baumgarte n'est pas physiquement réaliste, mais permet de conserver des liens rigides entre les corps.

0.3 Résolution sur l'accélération

Dans le but d'accroître la précision de la simulation, on a souhaité travailler à l'ordre 2, c'est-à-dire sur l'accélération. Les étapes du programme changent légèrement :

1. Corriger la force appliquée sur l'objet pour satisfaire la contrainte
2. Calculer la position et vitesse à l'aide de l'intégration de Loup Verlet, qui est une méthode d'intégration à l'ordre 4 :

$$q_{n+1} = 2q_n - q_{n-1} + (\Delta t)^2 \ddot{q}$$

$$\dot{q}_{n+1} = \frac{q_{n+1} - q_{n-1}}{2\Delta t}$$

0.3.1 Changement d'ordre

Pour obtenir une contrainte sur l'accélération, on dérive \dot{C}

$$\ddot{C} : \dot{J}\dot{q} + J\ddot{q} = 0$$

$\dot{J}\dot{q}$ est une écriture inutilement complexe, pour obtenir ce terme, il est plus facile de passer par l'égalité suivante :

$$\dot{J}\dot{q} = \frac{d^2 C}{dt^2} - J\ddot{q}$$

0.3.2 Mise en équation

De la même manière que précédemment, on cherche à isoler λ :

$$\begin{aligned}
\ddot{C} &\Leftrightarrow J\ddot{q} + \dot{J}\dot{q} = 0 \\
&\Leftrightarrow J\ddot{q} = -\dot{J}\dot{q} \\
&\Leftrightarrow JW(F_c + F) = -\dot{J}\dot{q} \\
&\Leftrightarrow JWF_c = -JWF - \dot{J}\dot{q} \\
&\Leftrightarrow JWJ^\top\lambda = -JWF - \dot{J}\dot{q}
\end{aligned}$$

0.3.3 Stabilisation du système

Indique à l'ordre 1, à l'exception qu'il y a désormais deux termes de stabilisation :

$$\ddot{C}' = \ddot{C} + \frac{h_v}{\Delta t}\dot{C} + \frac{h_p}{\Delta t^2}C$$

Ce qui nous donne l'équation finale :

$$JWJ^\top\lambda = -JWF - \dot{J}\dot{q} - \frac{h_v}{\Delta t}\dot{C} - \frac{h_p}{\Delta t^2}C$$

0.3.4 Problèmes rencontrés

Si effectivement cette méthode semble plus précise, elle est beaucoup plus soumise aux erreurs d'arrondis et d'intégration. Les erreurs de positions sont plus importantes, et la stabilisation de Baumgarte n'est pas suffisante pour corriger les erreurs. Des coefficients de stabilisation trop importants peuvent entraîner des oscillations dans le système, voir des divergences. À l'inverse, des coefficients trop faibles provoquent une mauvaise rigidité du système et des pertes d'énergie. Ce solveur donne de moins bons résultats que le solveur d'ordre 1.

0.4 Mise en place d'un solveur hybride

Le solveur d'ordre 2, bien qu'explosif, est plus endurant que le solveur d'ordre 1 quand toutes les stabilisations sont désactivées. De plus, ils partagent une structure similaire : ils nécessitent tous deux l'inversion de la matrice JWJ^\top .

Dans le but de diminuer les erreurs de corrections commises par le solveur d'ordre 1 lors de changements brutaux de la vitesse, on a souhaité mettre

en place un solveur hybride. Celui-ci chaine les deux solveurs, en utilisant le solveur d'ordre 1 pour les corrections de positions.

0.4.1 Gain

Ce solveur hybride est plus stable que les solveurs précédents, mais de l'ordre 1% pour le double du temps de calcul. Il est donc inutile en l'état actuel, car doubler la fréquence d'échantillonnage du solveur d'ordre 1 permet d'obtenir une simulation de meilleure qualité.

1. Corriger la force appliquée à l'aide du solveur d'ordre 2
2. Intégrer la vitesse à l'aide de la méthode d'Euler implicite
3. Corriger la vitesse à l'aide du solveur d'ordre 1
4. Intégrer la position à l'aide de la méthode d'Euler implicite. Utiliser une série de Taylor pour obtenir la position à l'ordre 2 provoque des explosions du système. Le solveur calcule déjà la précision nécessaire pour la position, il n'est pas nécessaire de l'intégrer à l'ordre 2. Il s'agit peut-être d'une accumulation avec la stabilisation de Baumgarte a l'ordre 1.

0.5 Série de Taylor

On utilise la série de Taylor pour approximer la position et la vitesse à l'ordre 2.

$$\begin{aligned}
 q(t + \Delta t) &= q(t) + \dot{q}(t)\Delta t + \frac{\ddot{q}(t)}{2}\Delta t^2 + O(\Delta t^3) \\
 \frac{q(t + \Delta t) - q(t)}{\Delta t} &= \dot{q}(t) + \frac{\ddot{q}(t)}{2}\Delta t + O(\Delta t^2) \\
 JWJ^\top \lambda' &= -J\dot{q} - \frac{1}{2\Delta t} \dot{J}\dot{q}
 \end{aligned}$$

0.6 Contraintes usuelles

Quelques contraintes usuelles en deux dimensions : $q = \begin{pmatrix} x \\ y \end{pmatrix}$

Pendule

— Contrainte sur la position : $C : \|q\| - l = 0$

— Contrainte sur la vitesse :

$$\begin{aligned}\frac{dC}{dT} = 0 &\Leftrightarrow \frac{d}{dt}(\sqrt{q^2}) = 0 \\ &\Leftrightarrow \frac{2q \cdot \dot{q}}{2\sqrt{q^2}} = 0 \\ &\Leftrightarrow \frac{q^\top}{\|q\|} \dot{q} = 0\end{aligned}$$

— Contrainte sur l'accélération :

$$\begin{aligned}\frac{d^2C}{dt^2} = 0 &\Leftrightarrow \frac{d}{dt} \left(\frac{q}{\|q\|} \cdot \dot{q} \right) = 0 \\ &\Leftrightarrow \frac{d}{dt} \left(\frac{q}{\|q\|} \right) \cdot \dot{q} + \frac{q}{\|q\|} \cdot \ddot{q} = 0 \\ &\Leftrightarrow \frac{\dot{q}\|q\| - q \left(\frac{q}{\|q\|} \cdot \dot{q} \right)}{\|q\|^2} \cdot \dot{q} + \frac{q}{\|q\|} \cdot \ddot{q} = 0 \\ &\Leftrightarrow \frac{\dot{q}\|q\|^2 - q(q \cdot \dot{q})}{\|q\|^3} \cdot \dot{q} + \frac{q}{\|q\|} \cdot \ddot{q} = 0 \\ &\Leftrightarrow \frac{\dot{q}^2 q^2 - (q \cdot \dot{q})^2}{\|q\|^3} + \frac{q}{\|q\|} \cdot \ddot{q} = 0\end{aligned}$$

— Jacobienne : $J = \frac{q^\top}{\|q\|}$

— Second terme ordre à l'ordre 2, $J\dot{q} = \frac{\dot{q}^2 q^2 - (q \cdot \dot{q})^2}{\|q\|^3} = \frac{(x\dot{y} - y\dot{x})^2}{(x+y)^{\frac{3}{2}}}$