Bartolomeo Ryan 10208
Jeux et Sports

Modélisation de solides déformables

Plan

Présentation du problème et première approche

Motivation

Contraintes :
- Multi-joueurs
- Simulation en temps réel
- Machine parfois avec de faibles capacités

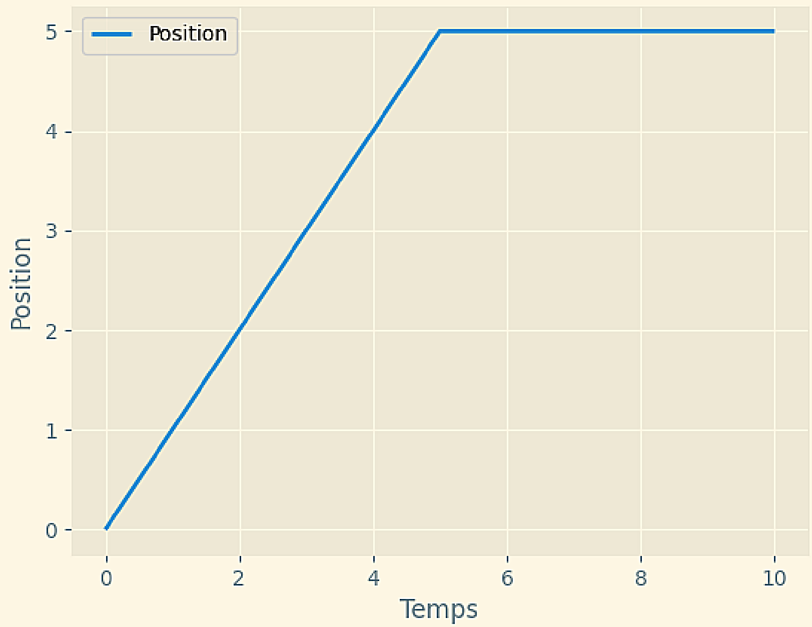
→ Comment minimiser le temps de calcul dans la simulation de solides déformable ?

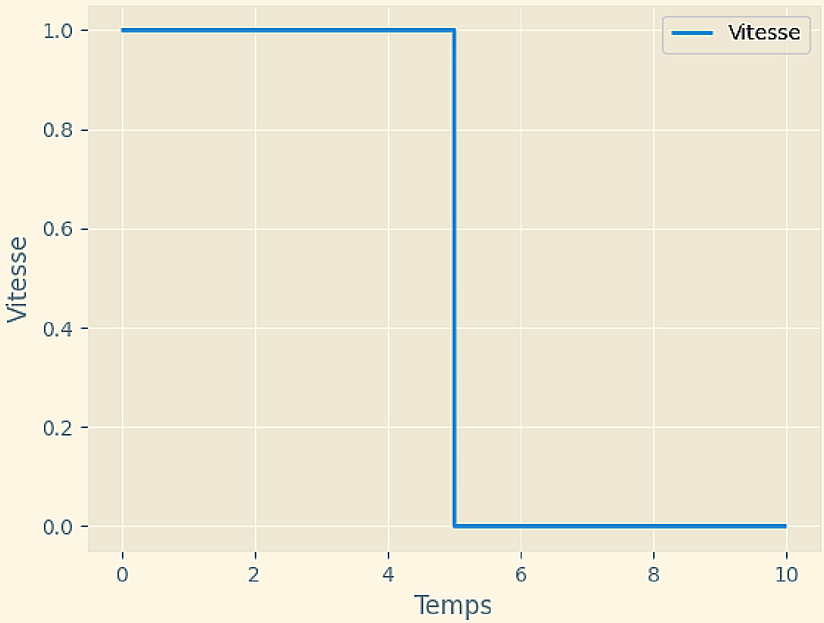
Modélisation réaliste d’un mouvement

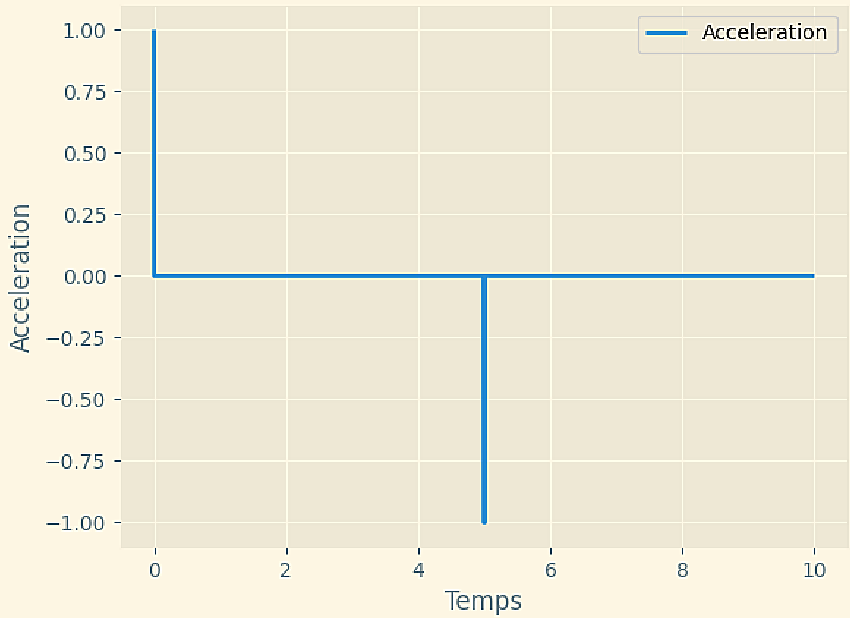
Cas simple

Modèle informatique simple

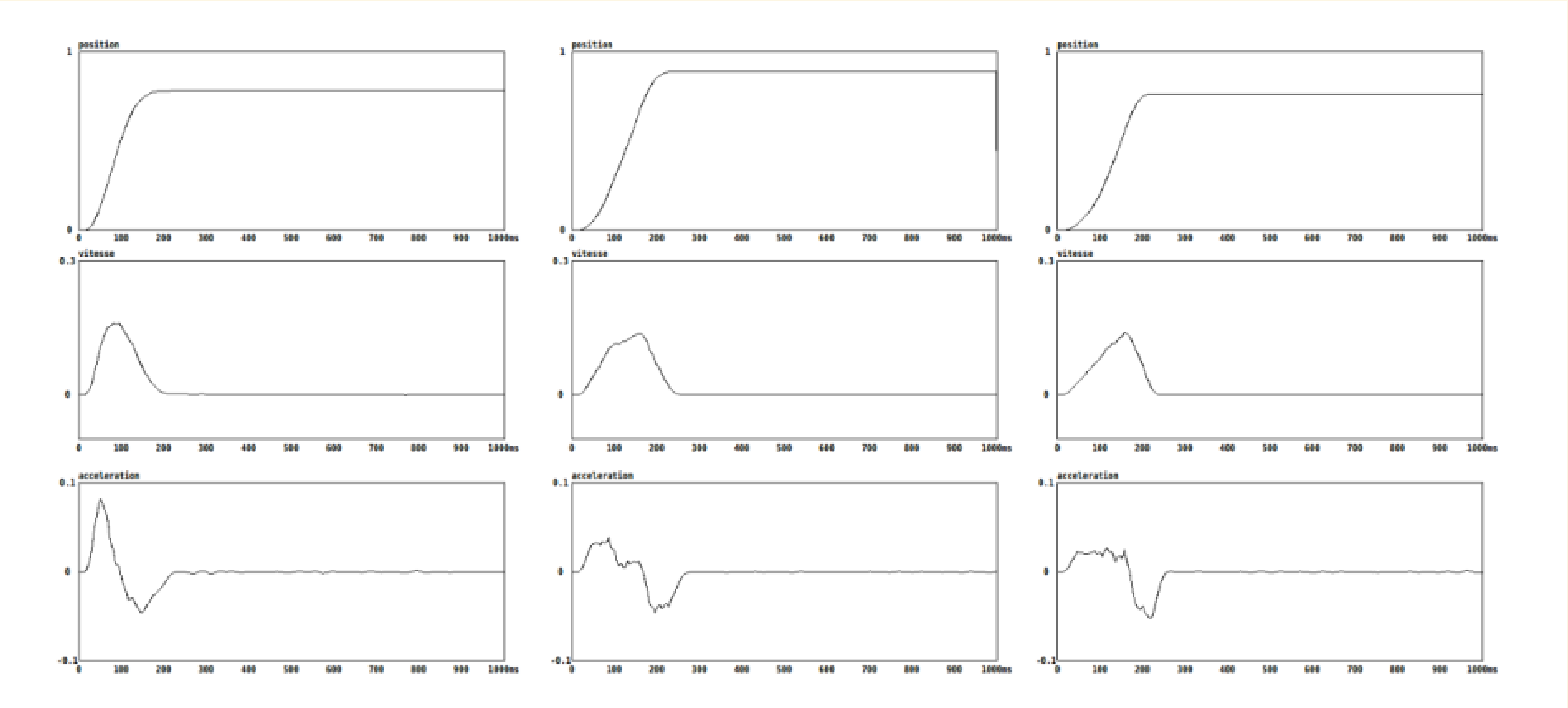
Profil de déplacement linaire







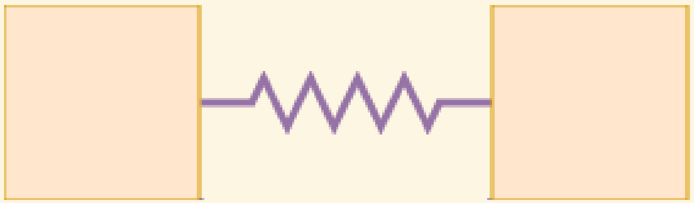
Mouvement naturel

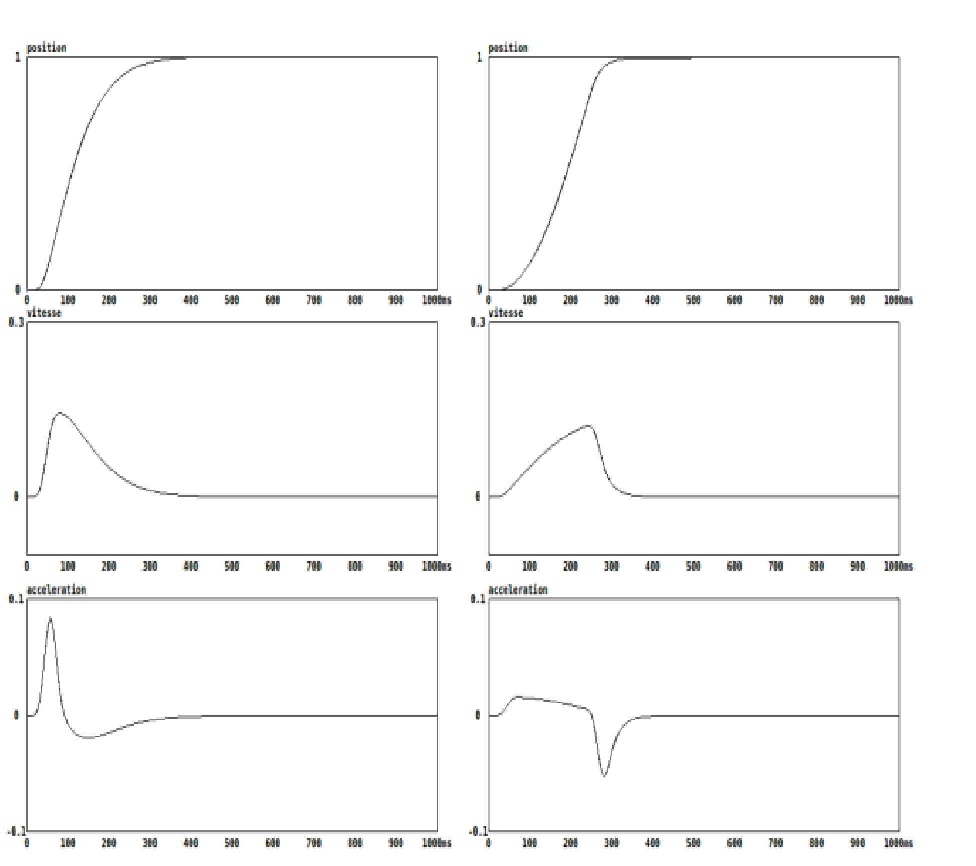


Mass-Spring-System model for real time expressive behaviour synthesis ~ Cyrille Henry

Mouvements type Système masse-ressort

Deux masses en mouvement l’une par rapport à l’autre



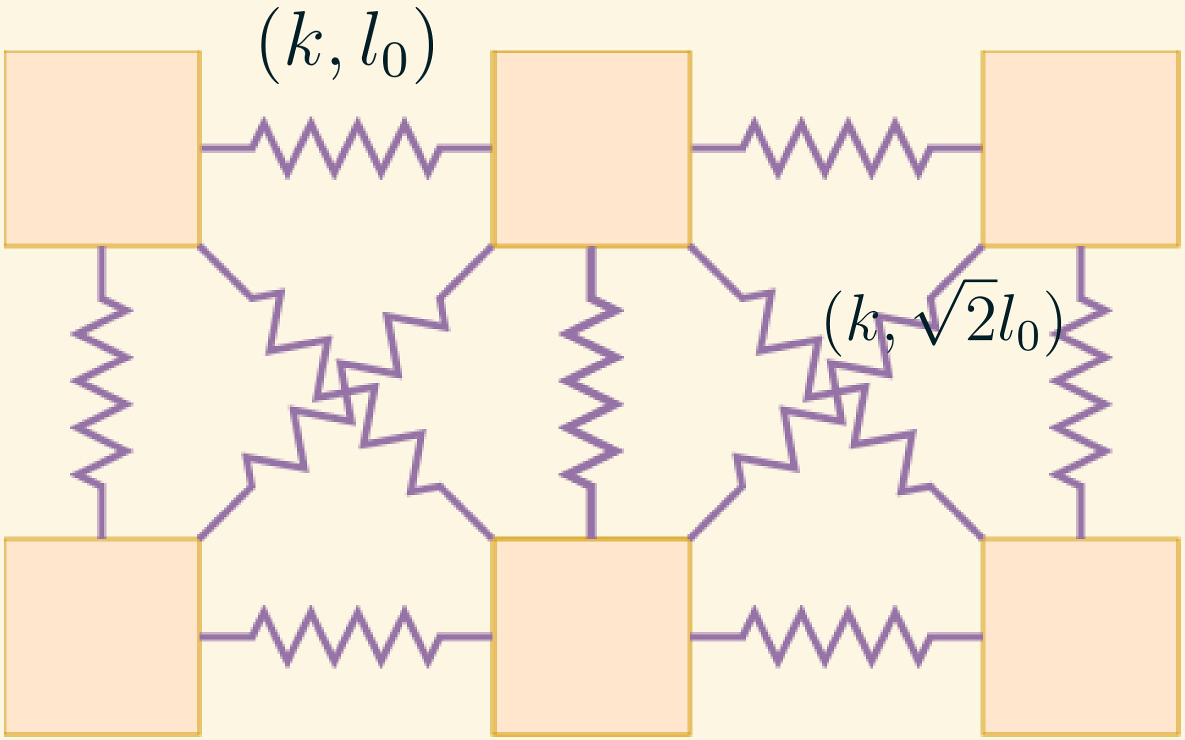


Mass-Spring-System model for real time expressive behaviour synthesis ~ Cyrille Henry

Problème à résoudre

II. Réalisation

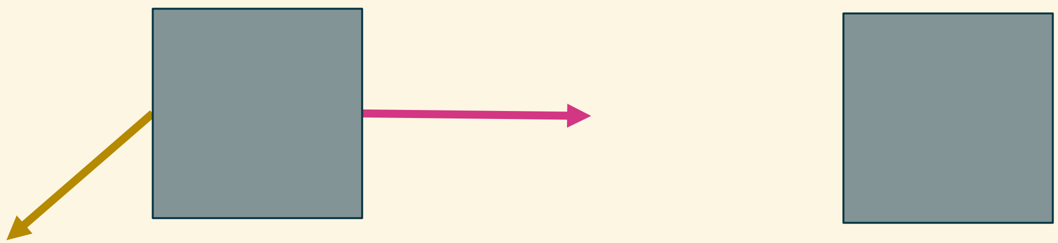
Structure du système



Méthode d’Euler explicite

$$\vec{R} = k * (l - l_0)\vec{e_x}$$

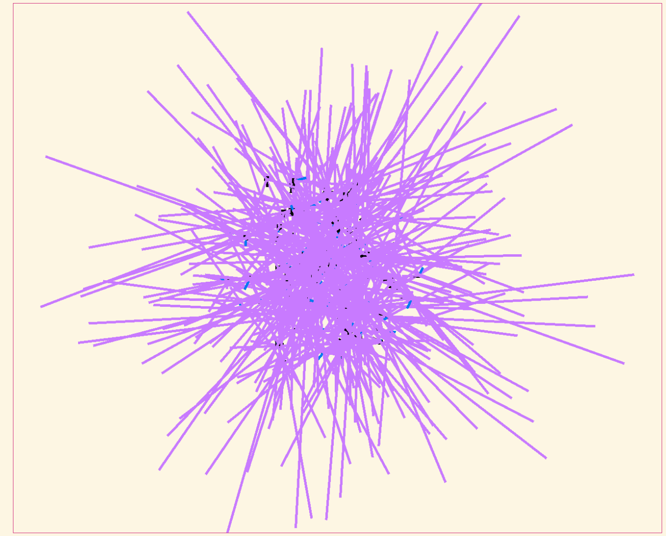
$$\vec{f} = -\alpha\vec{v}$$



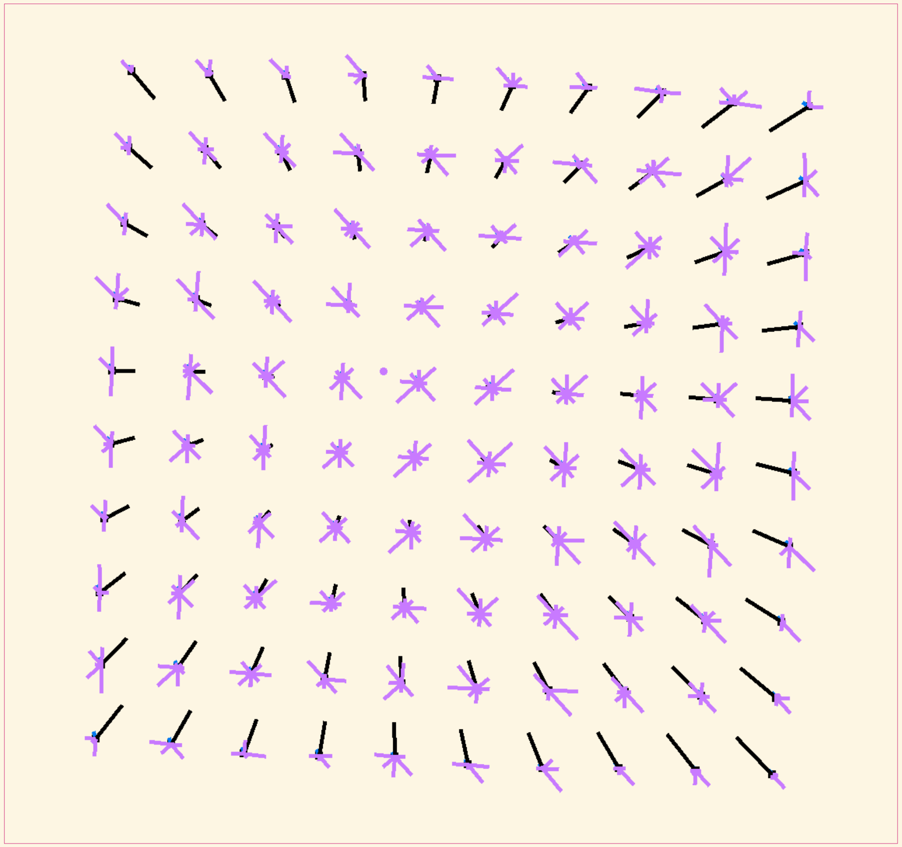
```
x[t + 1] = x[t] + v[t] * dt
v[t + 1] = v[t] + acceleration(t) * dt
```

Résultat de la simulation avec la méthode d'Euler

Compression sous l'effet de la force fictive



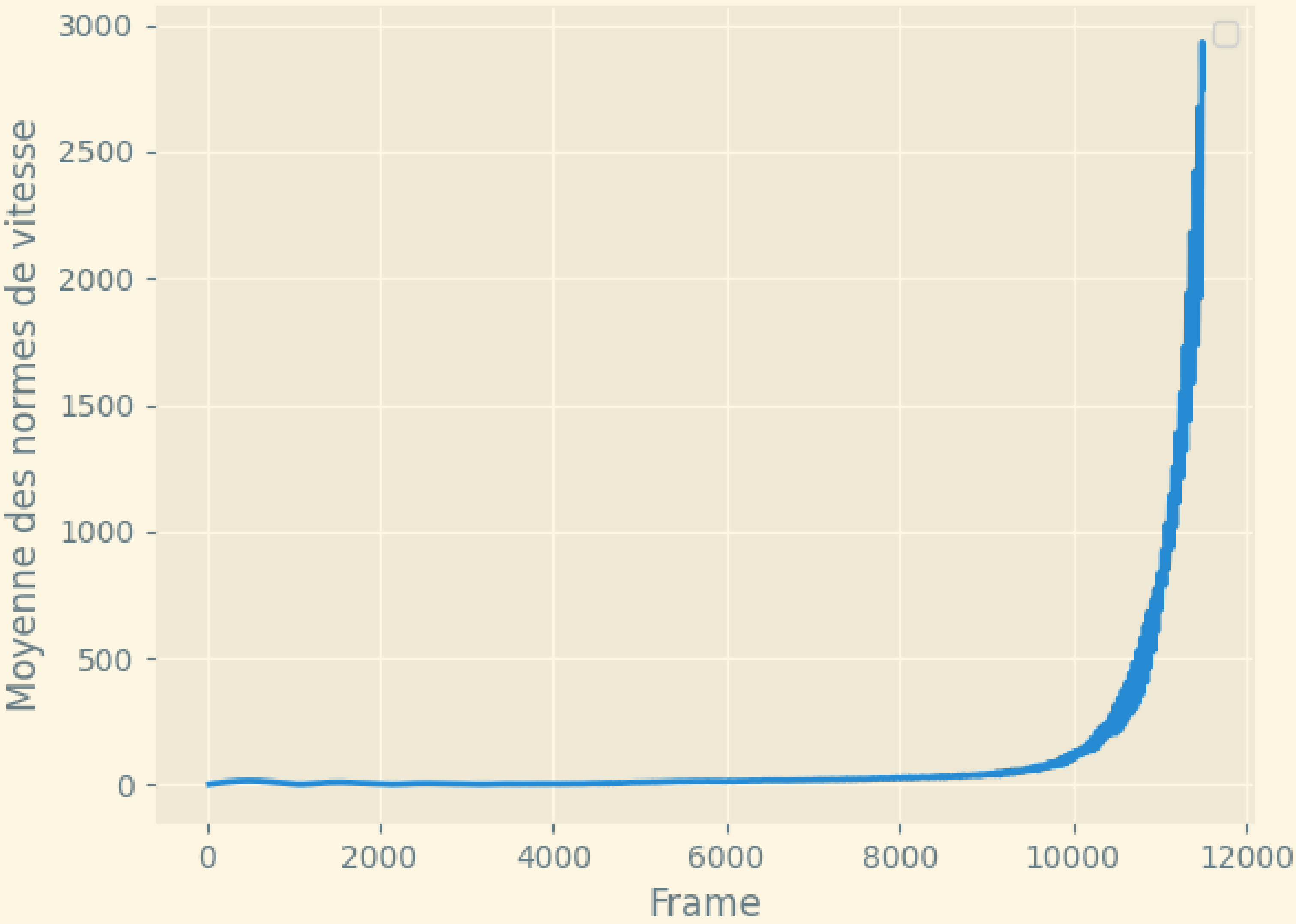
État initial



Explosion du système

Force élastique en violet
Force fictive en noir

Premier résultat



Méthode de Runge-Kutta

Approximation de Runge-Kutta

Problème de Cauchy à résoudre

$$\begin{cases} y'(t) = f(t, y) \\ y(t_0) = y_0 \end{cases}$$

Si z est solution alors :

$$t_{n,i} = t_n + c_i \times h$$

Tableau des coefficients utilisés

$$\begin{array}{c|cccc l} (c_i) & & & & & \\ 0 & 0 & 0 & 0 & 0 & (a_{i,j}) \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & \\ 1 & 0 & 0 & 1 & 0 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & (b_j) \end{array}$$

$$z(t_{n,i}) = z(t_n) + h \sum_{j<i} a_{i,j} f(t_{n,j}, z(t_{n,j}))$$

$$z(t_{n+1}) = z(t_n) + h \sum_{j\leq 4} b_j f(t_{n,j}, z(t_{n,j}))$$

Approximation de Runge-Kutta

Problème différentiel physique

$$\begin{cases} y'' = \mathrm{acc}(t, y, y') \\ y'(t_0) = y'_0 \\ y(t_0) = y_0 \end{cases}$$

On pose alors :

$$\mathrm{Y(t)} = \begin{pmatrix} y'(t) \\ y(t) \end{pmatrix}$$

$$\mathrm{Y(t_0)} = \begin{pmatrix} y'_0 \\ y_0 \end{pmatrix}$$

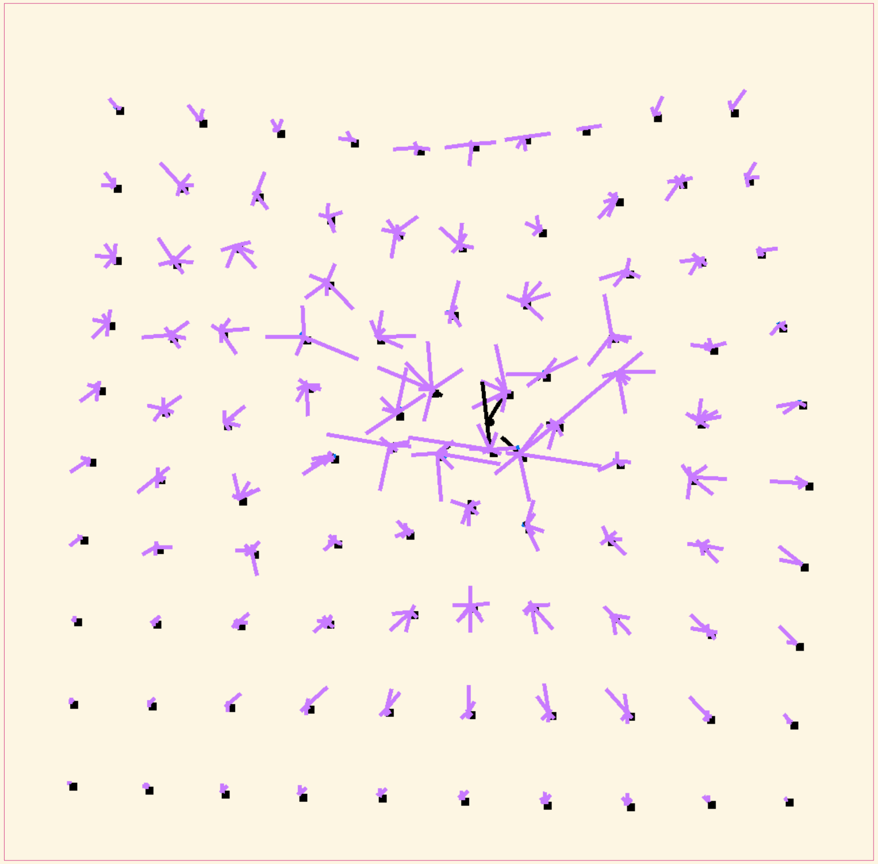
$$\text{d'où Y'(t)} = \begin{pmatrix} y''(t) \\ y'(t) \end{pmatrix} = \mathrm{F(t,Y)} = \begin{pmatrix} \mathrm{acc}(t, y, y') \\ y'(t) \end{pmatrix}$$

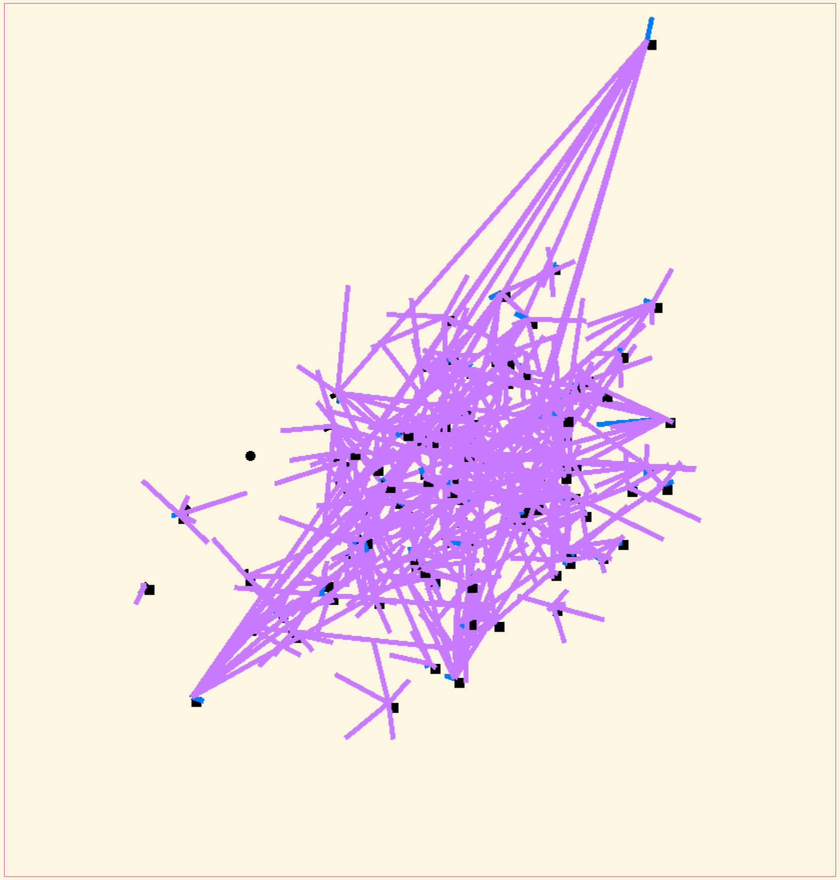
On a alors le problème de Cauchy d'ordre 1 :

$$\begin{cases} Y'(t) = F(t, Y) \\ Y(t_0) = Y_0 \end{cases}$$

Résultat d'algorithme

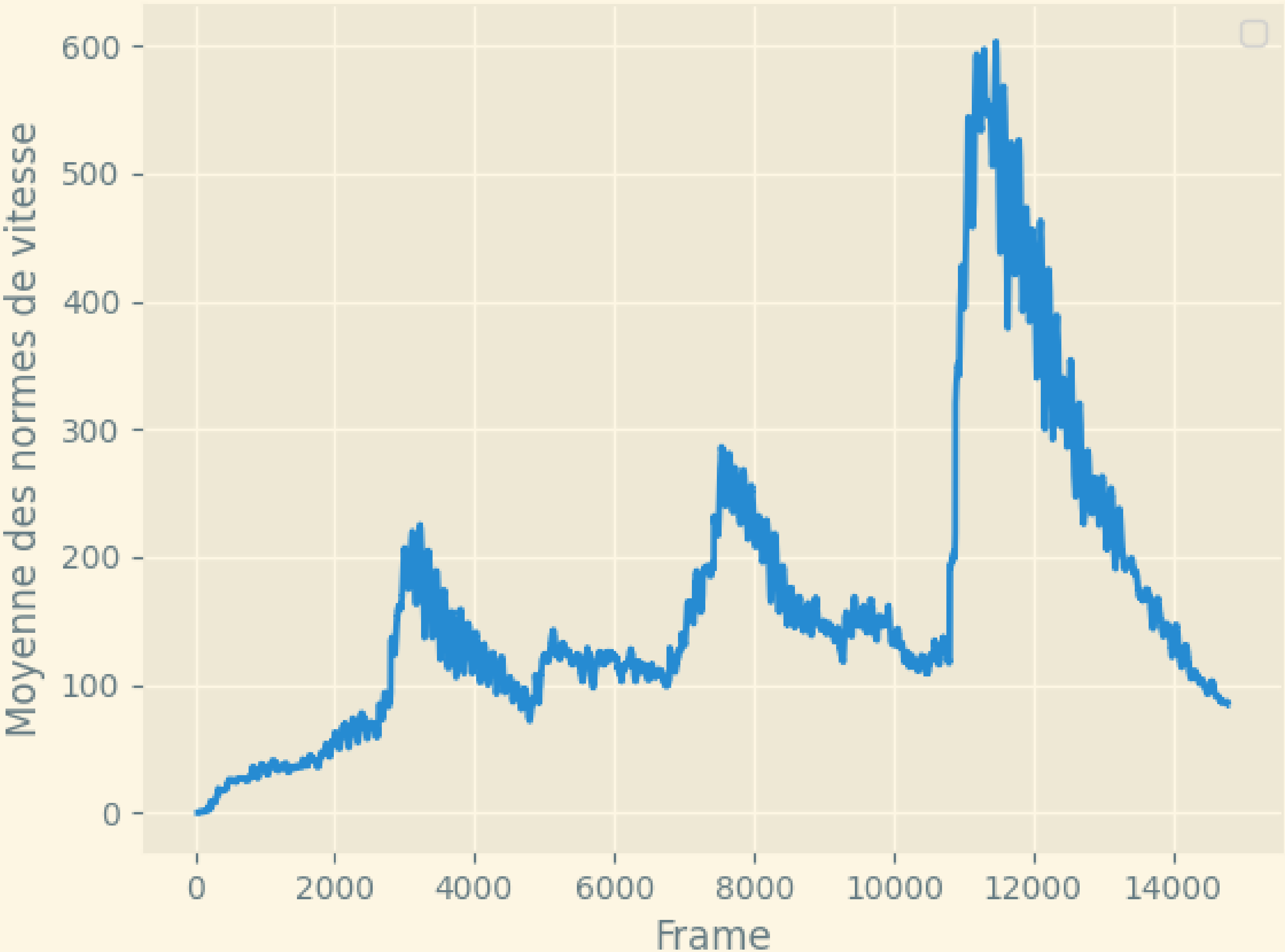
Déformation sous l'effet de la force fictive



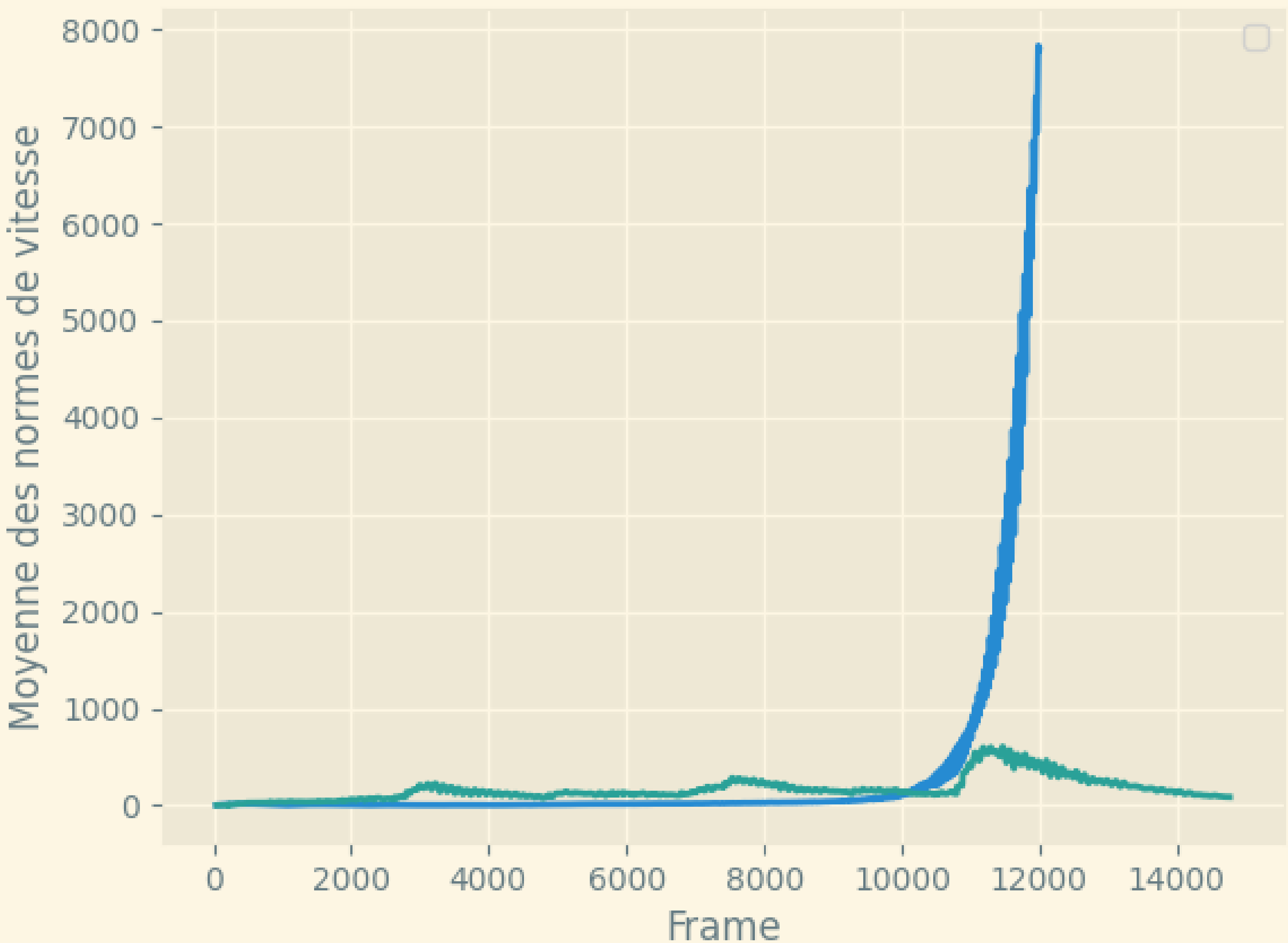


Désordre qui revient à une position d'équilibre

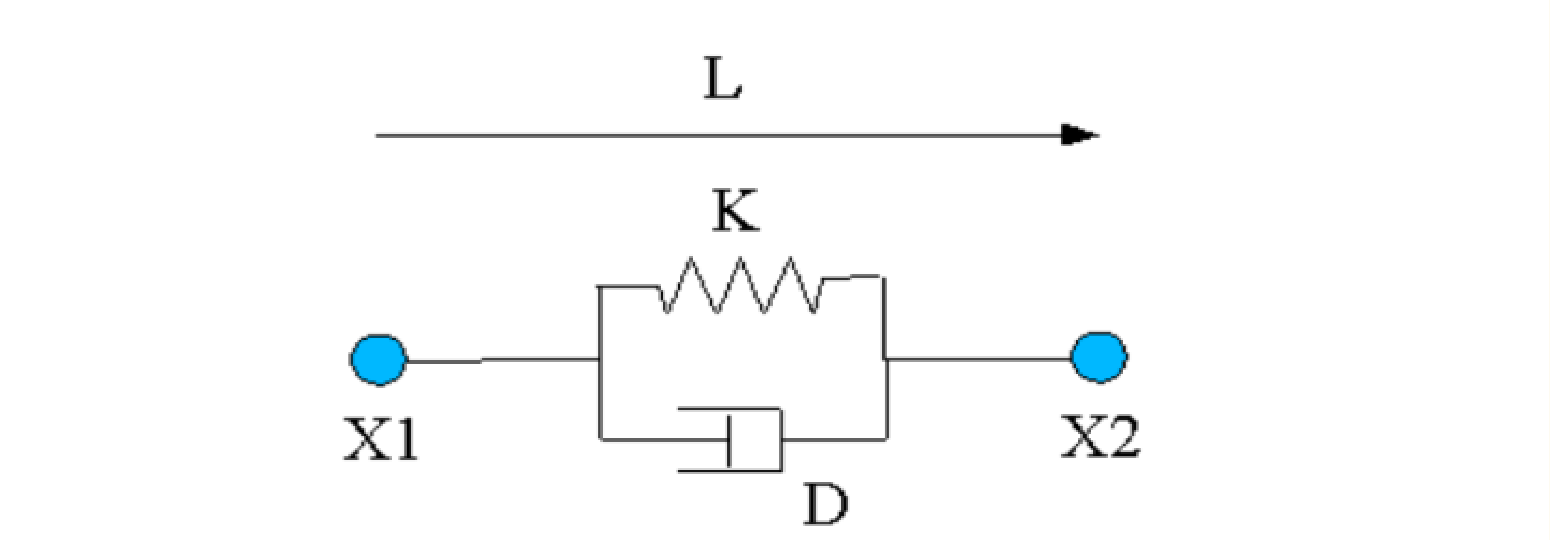
Résultat de la simulation avec RK4



Comparaison des résultats par rapport aux méthodes utilisées



Force d’amortissement pour stabiliser plus rapidement

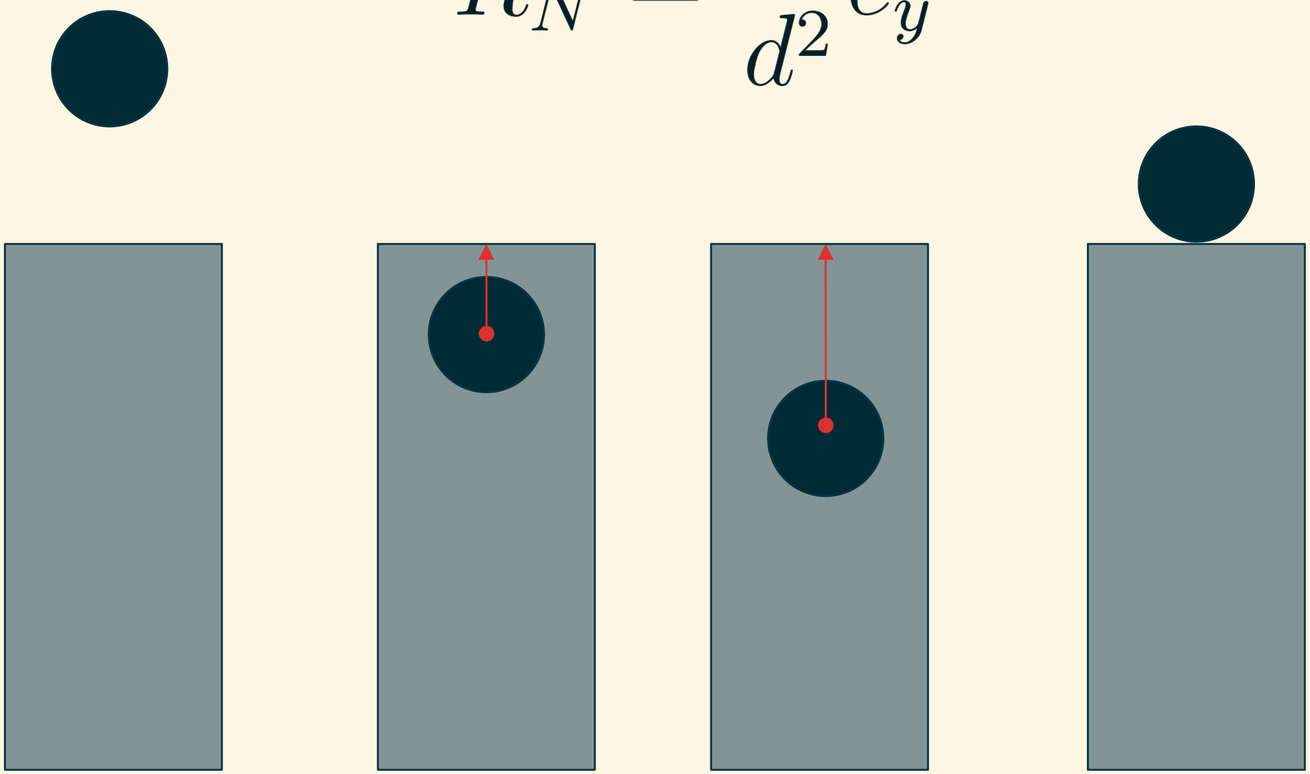


Mass-Spring-System model for real time expressive behaviour synthesis ~ Cyrille Henry

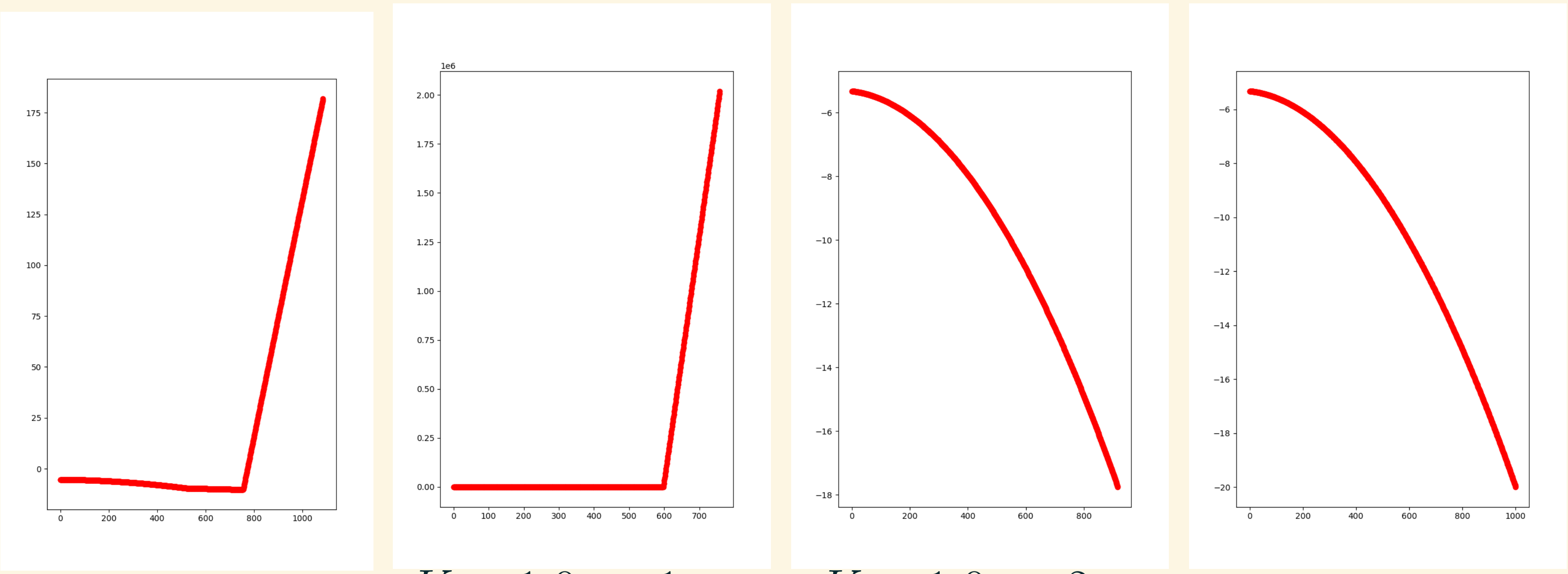
Collisions

Collision “magnétique”

$$\vec{R_N} = \frac{K}{d^2}\vec{e_y}$$



Résultats pour différentes valeurs de K

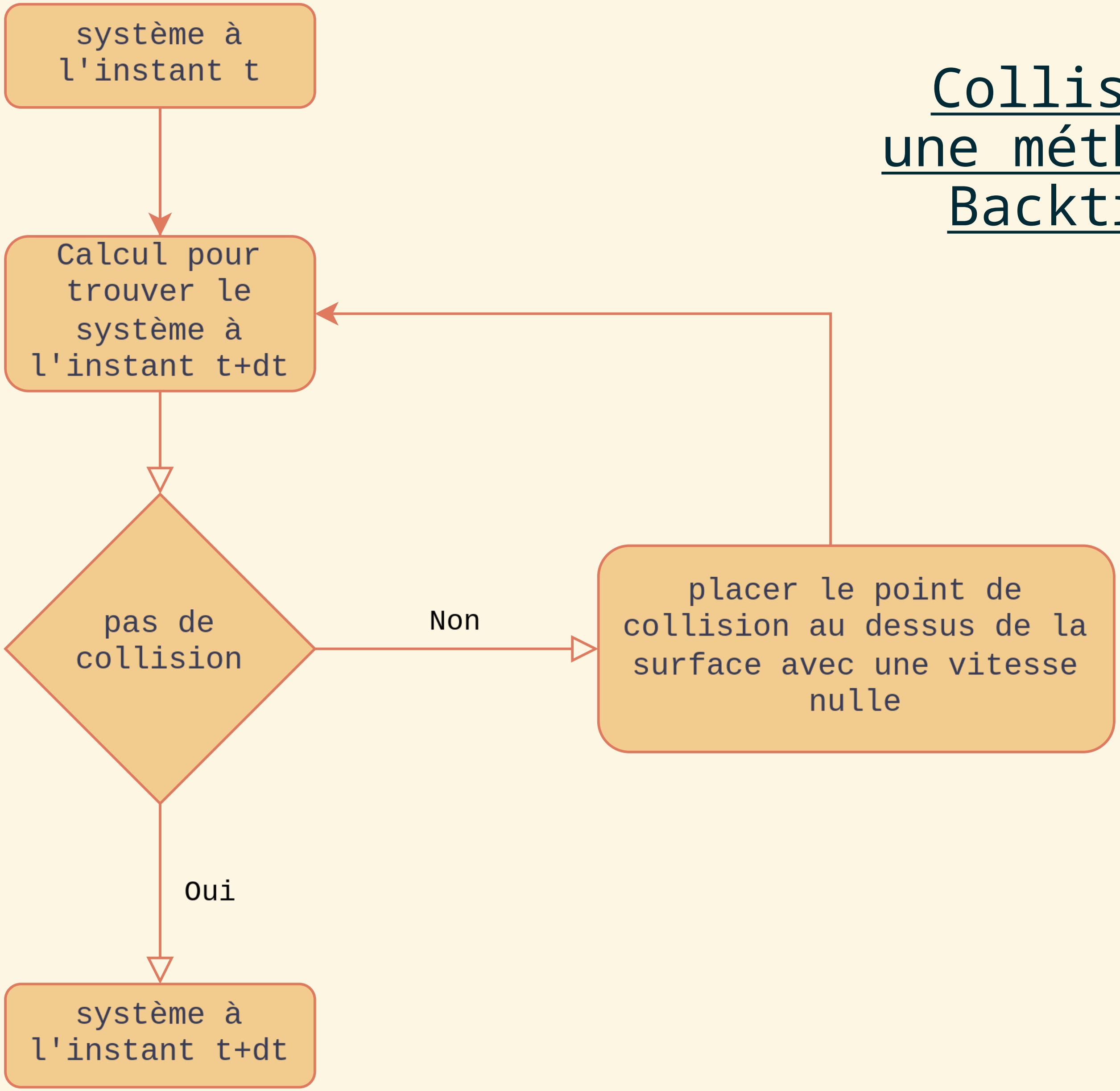


$K = 1.0$ $K = 1.0e-1$ $K = 1.0e-2$ $K = 1.0e-4$

Position moyenne verticale en fonction du temps

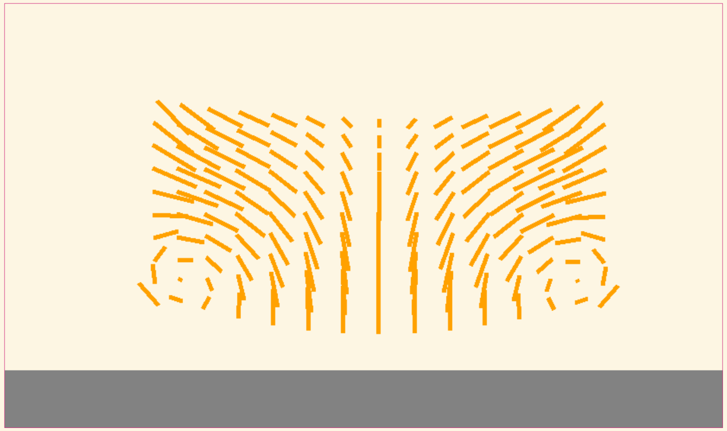
Résultats surprenants car :
→ Force discontinue
→ Force initialement très élevé lorsque d = 0

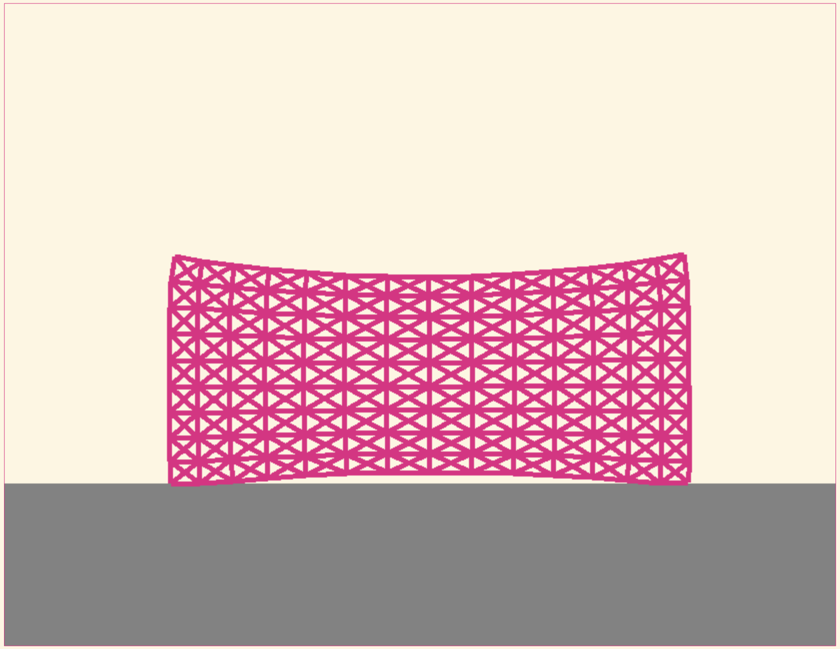
Collision via une méthode type Backtracking



Résultats

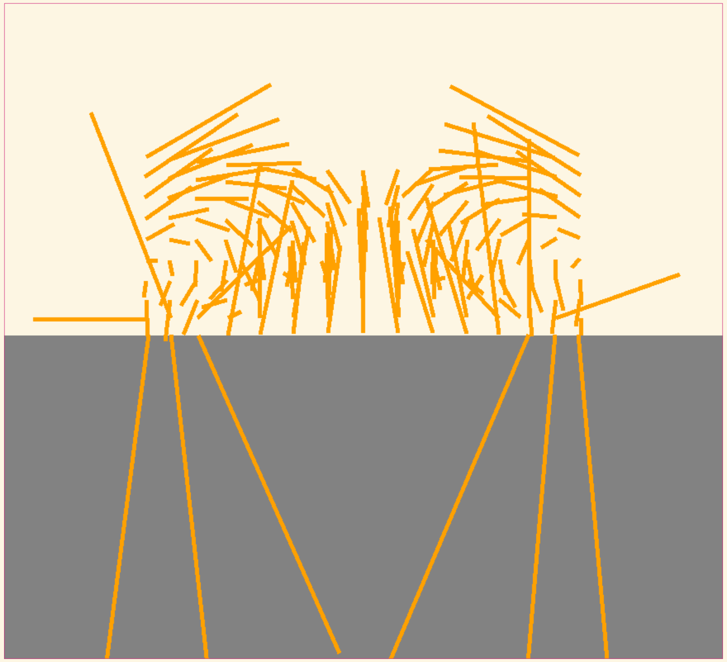
Champ d’accélération lorsque
le système rebondit en l’air





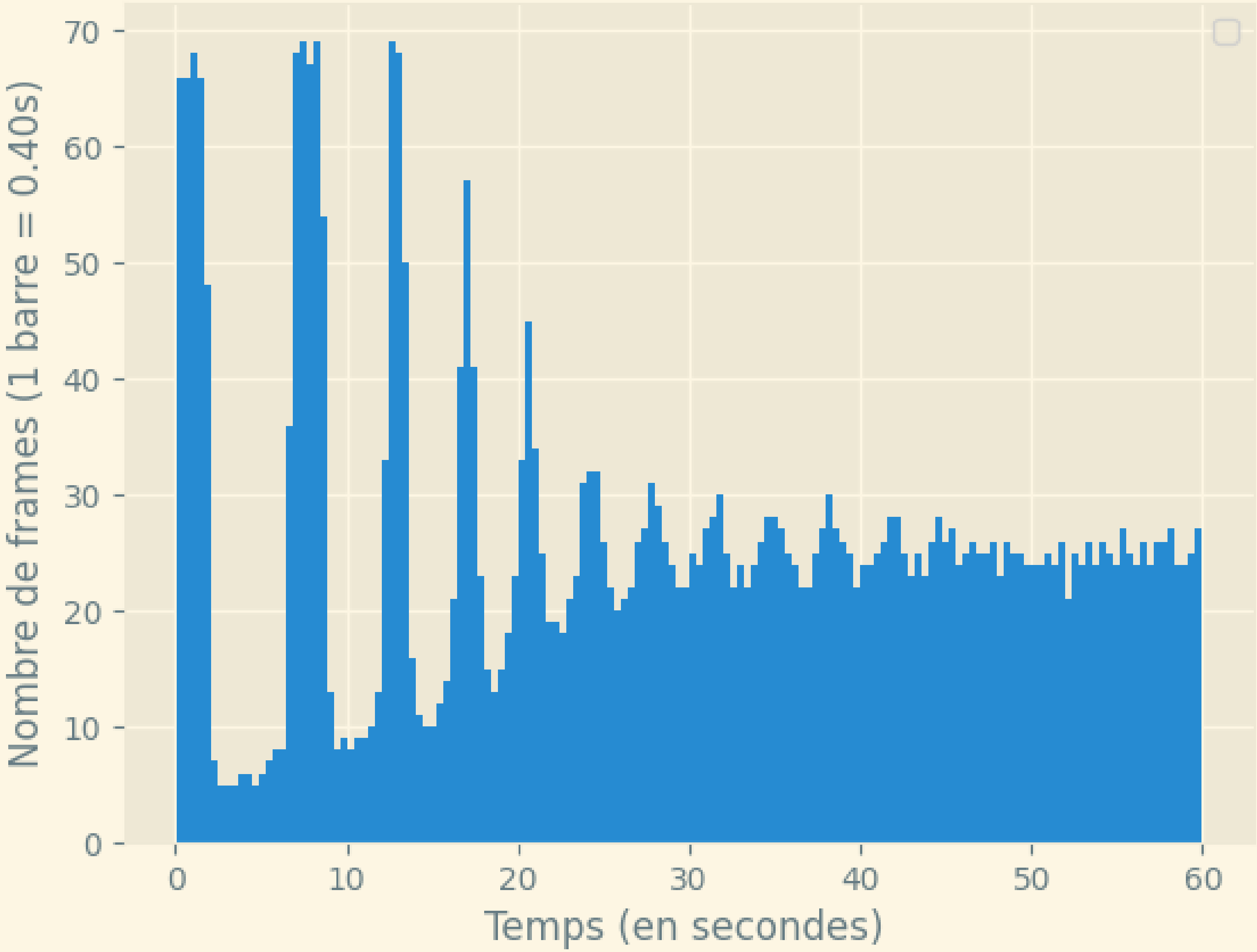
Liaisons des ressorts de la
position d’équilibre finale

Champ d’accélération lorsque le
système est au contact du sol



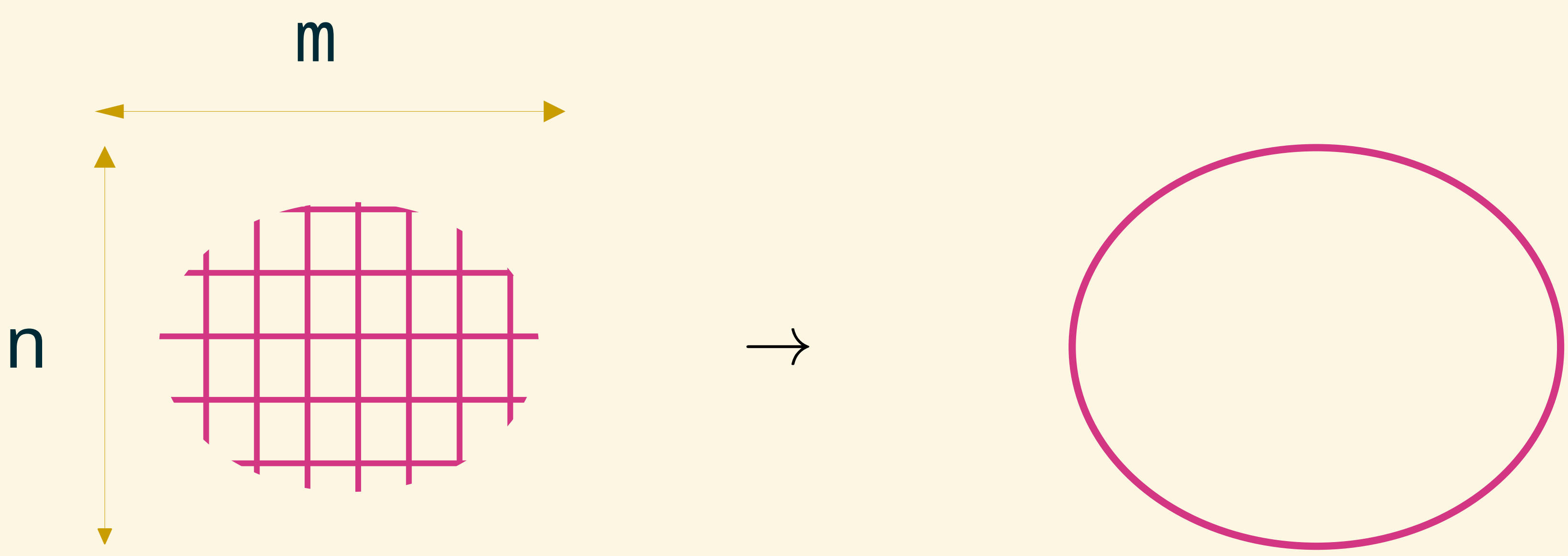
Résultats

Nombre de frames générées par période de 0,40s durant la simulation (sans affichage graphique)



III. Deuxième approche

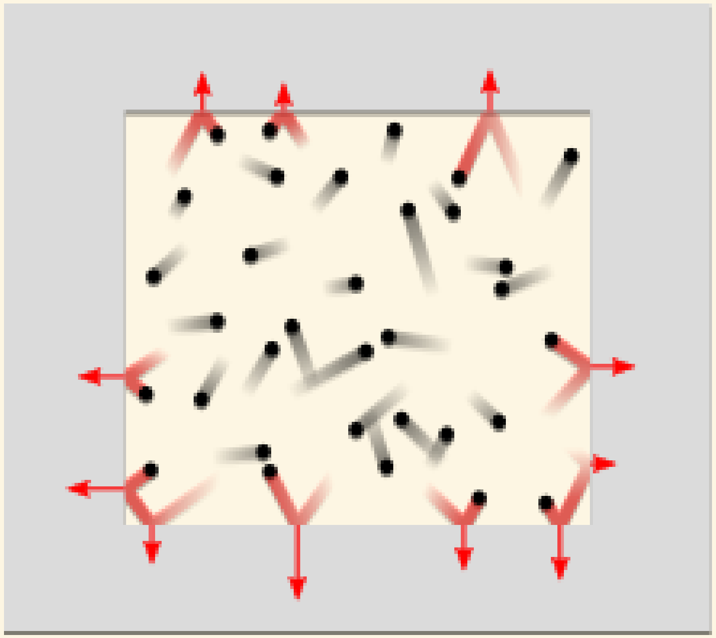
Modèle de la bulle de gaz parfait



Complexcité : $$\mathcal{O}(m * n) \rightarrow \mathcal{O}(m + n)$$

Avantages :
→ Moins de particules au total à simuler
→ Moins de particules au contact du sol

Modèle de la bulle de gaz parfait



$$\vec{F} = P d\vec{S}$$

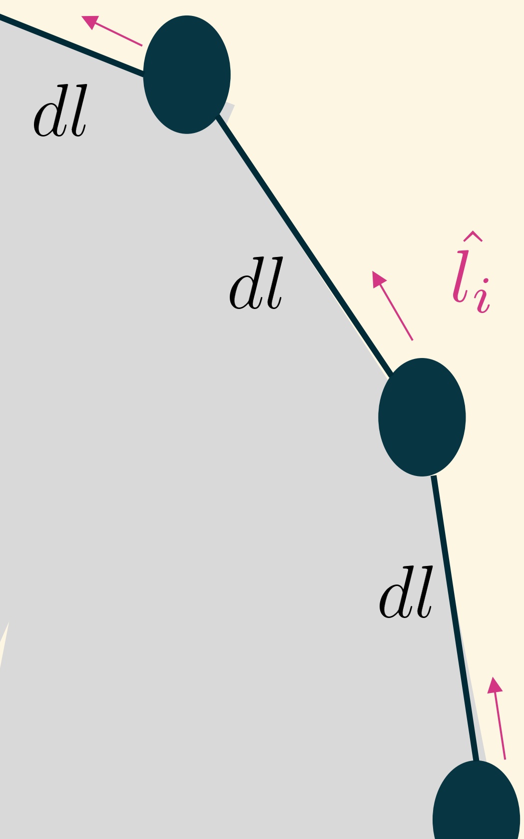
$$P = \frac{nRT}{V}$$

$$\boxed{\vec{F} = K_{nRT} \frac{1}{V} d\vec{S}}$$

Calcul du volume

Théorème de Stokes :

$$\iint_S \mathrm{div}\vec{F} \cdot d\vec{S} = \oint_C \vec{F} \cdot \vec{dl}$$



$$\vec{F} = x\vec{e_x}$$

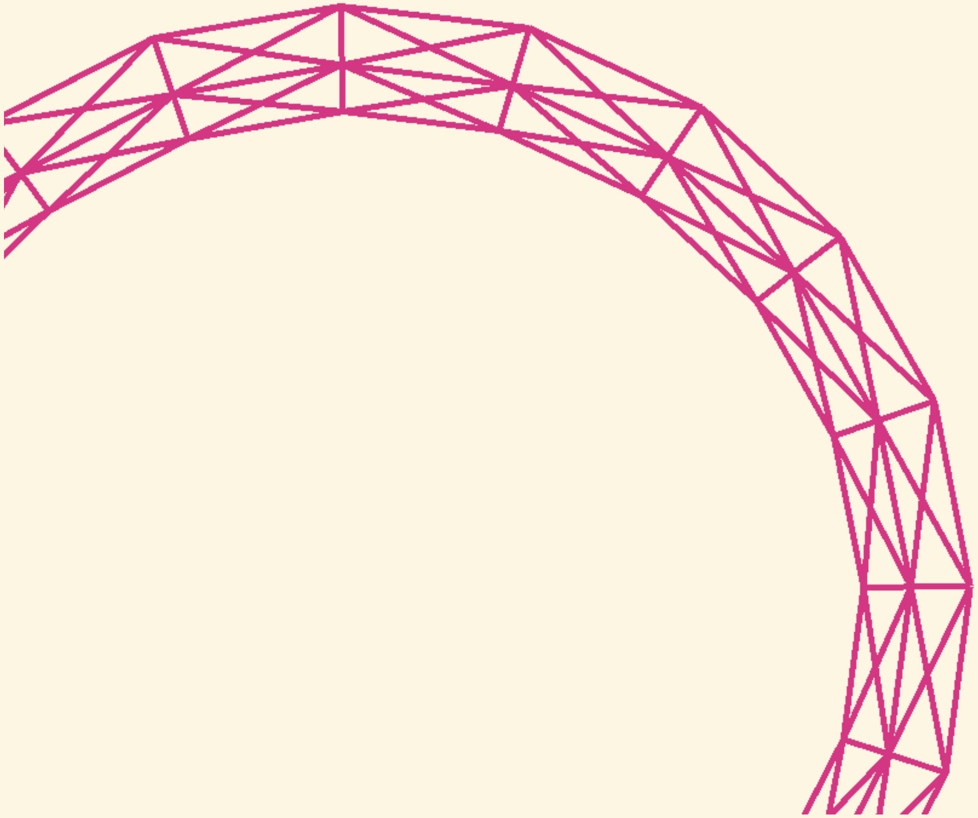
$$\mathrm{div}\vec{F} = 1$$

$$\iint_S \mathrm{div}\vec{F} \cdot d\vec{S} = S$$

$$\begin{aligned} &\vec{F} \cdot \vec{dl} \\ &= \vec{F} \cdot \hat{l} dl \\ &= x \cdot \hat{l}_x \cdot dl \end{aligned}$$

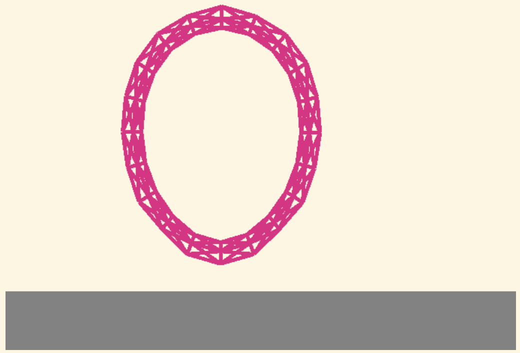
$$S \approx \sum x_i \cdot \hat{l}_{i,x} \cdot dl$$

Compromis ressort/gaz pour conserver la forme

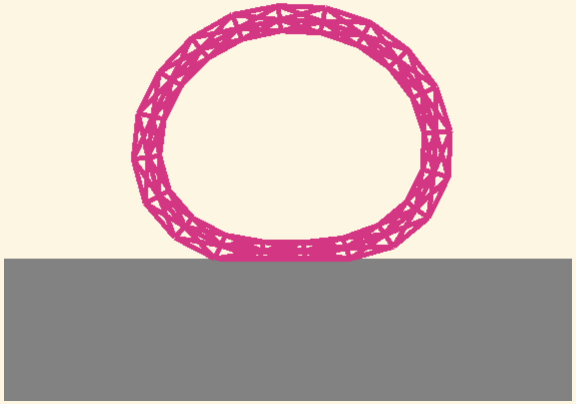


Résultat

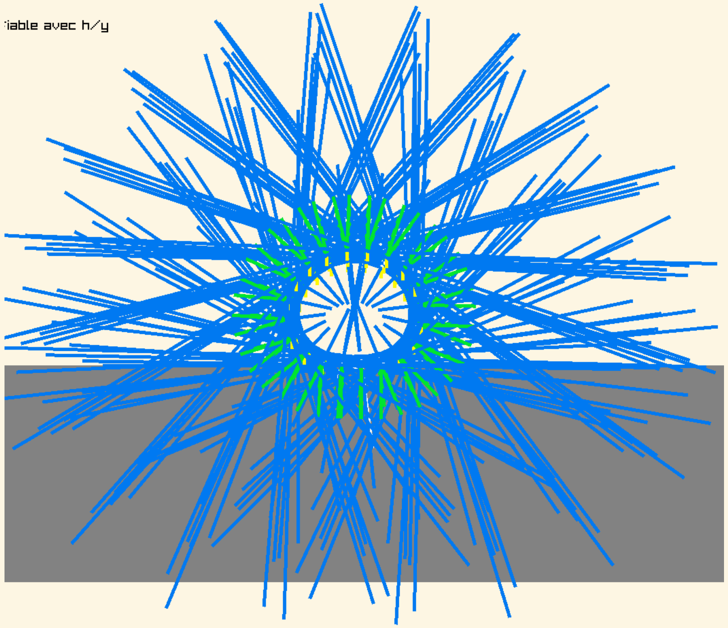
Déformation après rebond



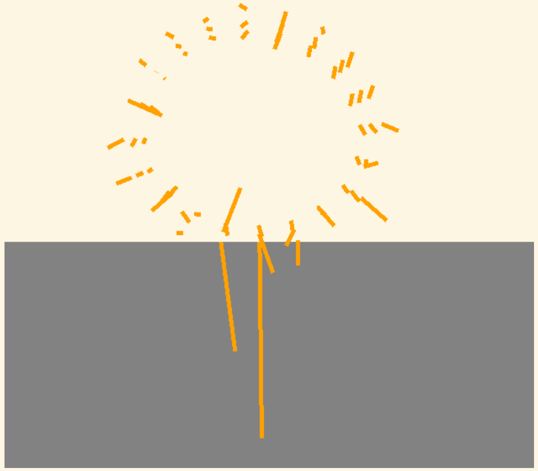
Position d'équilibre final



Forces appliquées sur les points

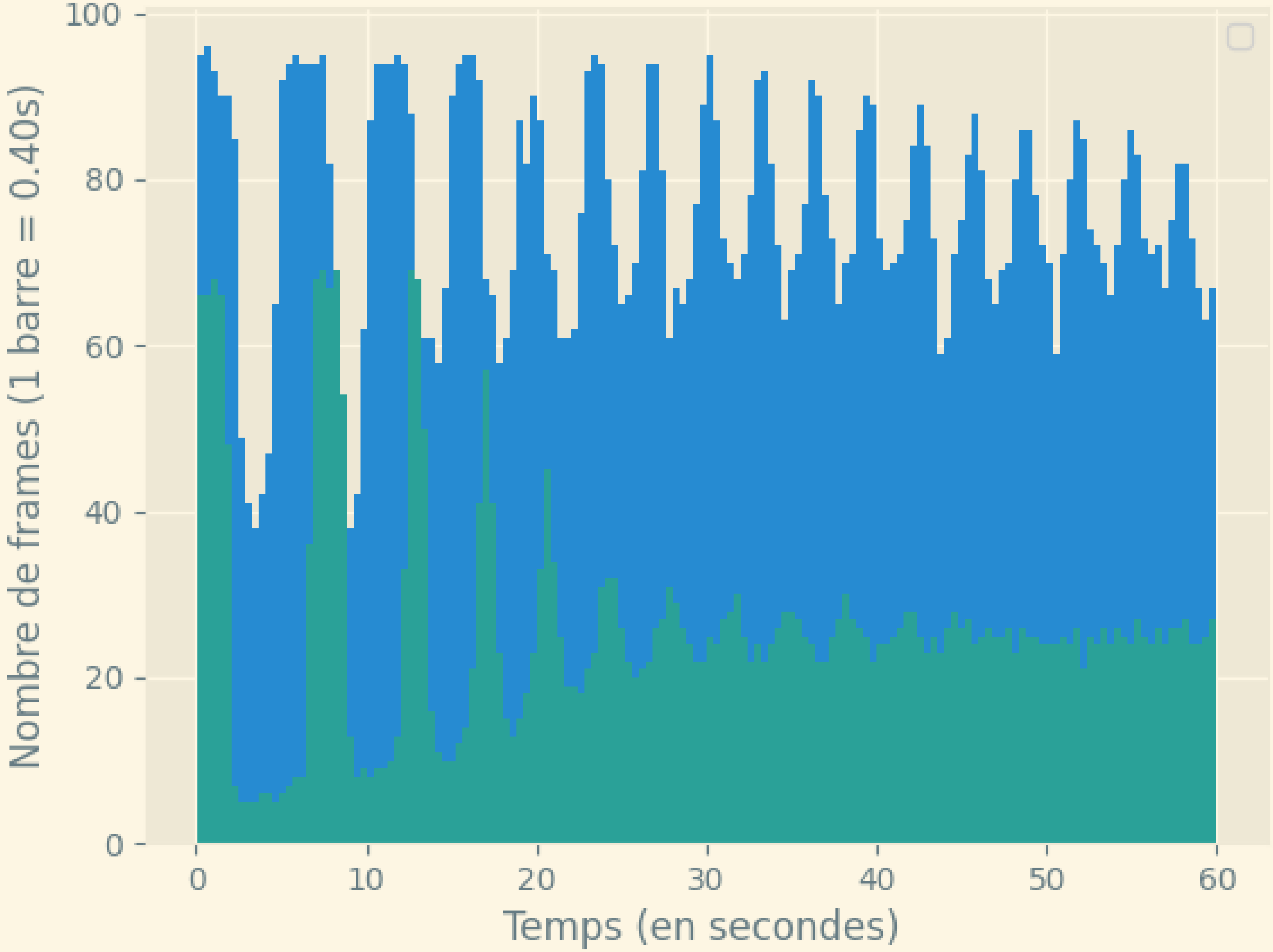


Profil d'accélération

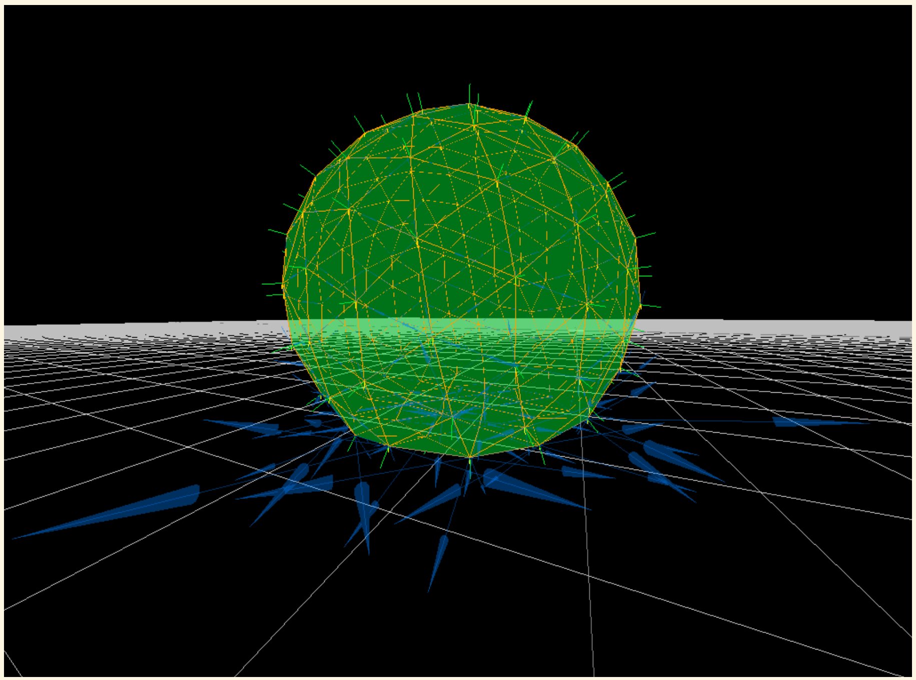


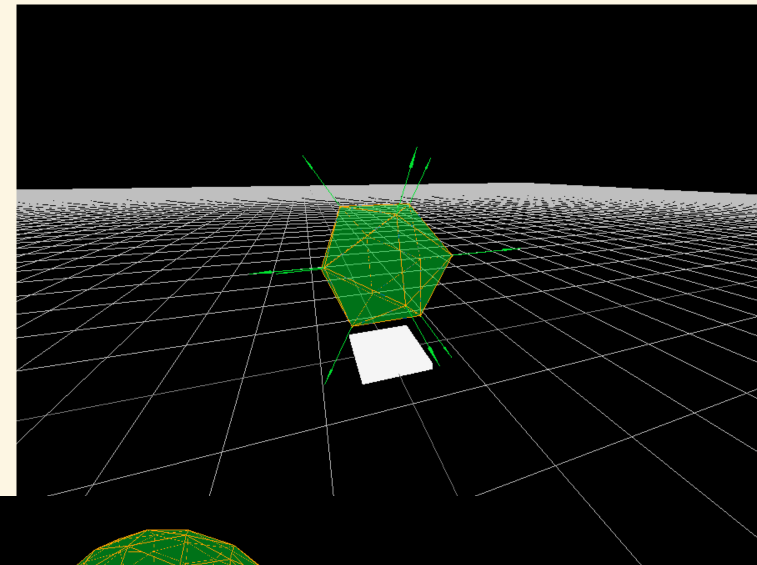
Résultats

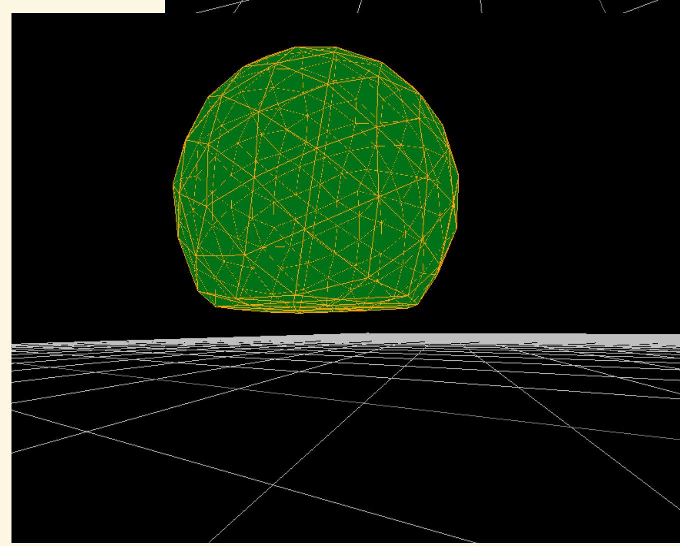
Comparaison du nombre de frames générés via les deux méthodes (pour un disque ayant la hauteur du cube)



Extrapolation en 3 dimensions





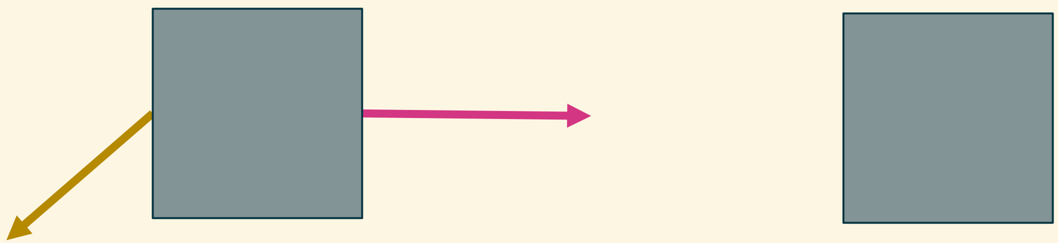


Annexe

Méthode d’Euler implicite

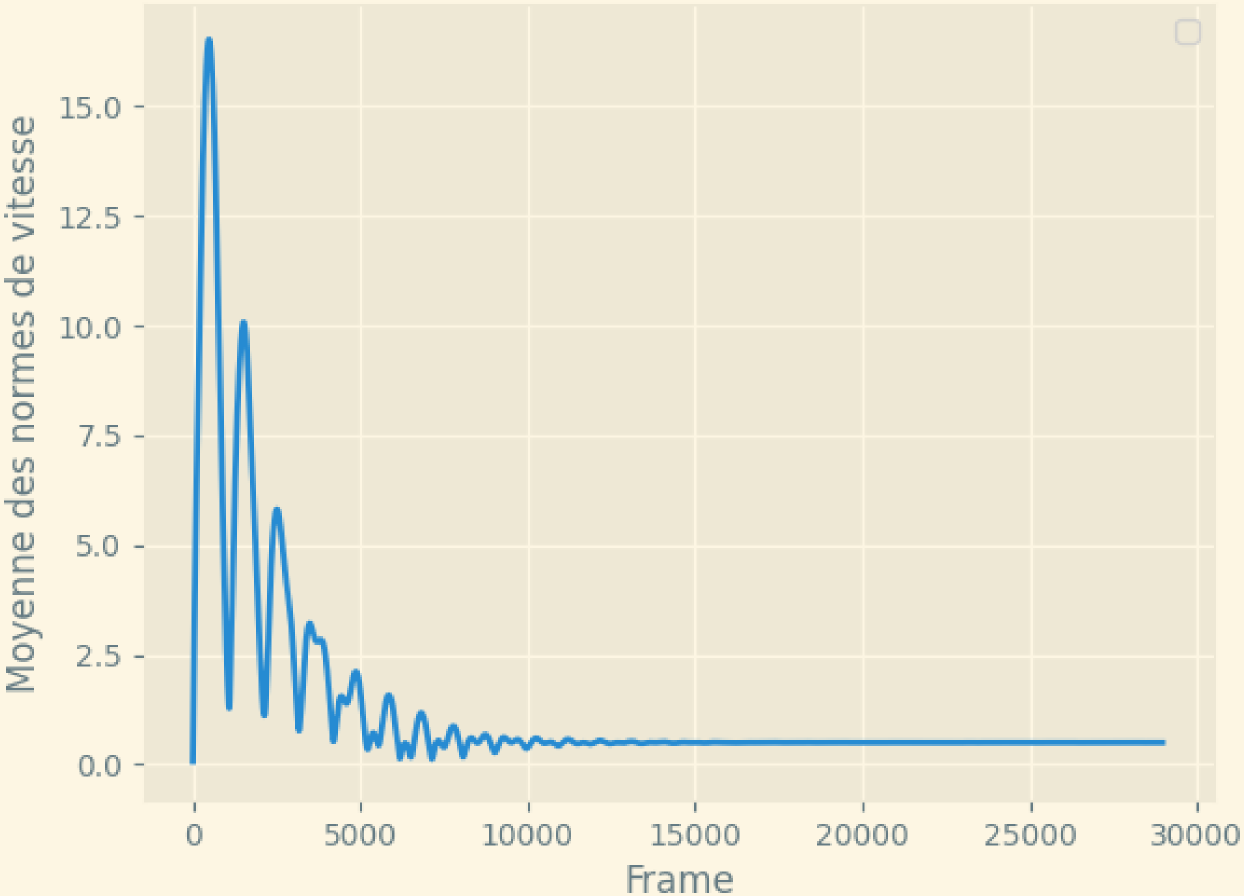
$$\vec{R} = k * (l - l_0)\vec{e_x}$$

$$\vec{f} = -\alpha\vec{v}$$

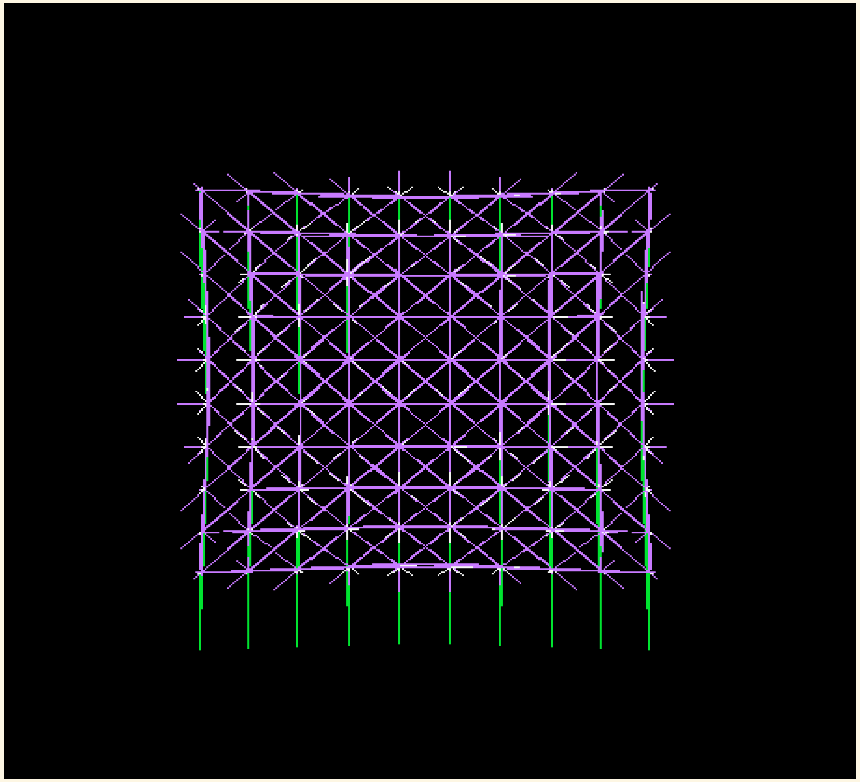


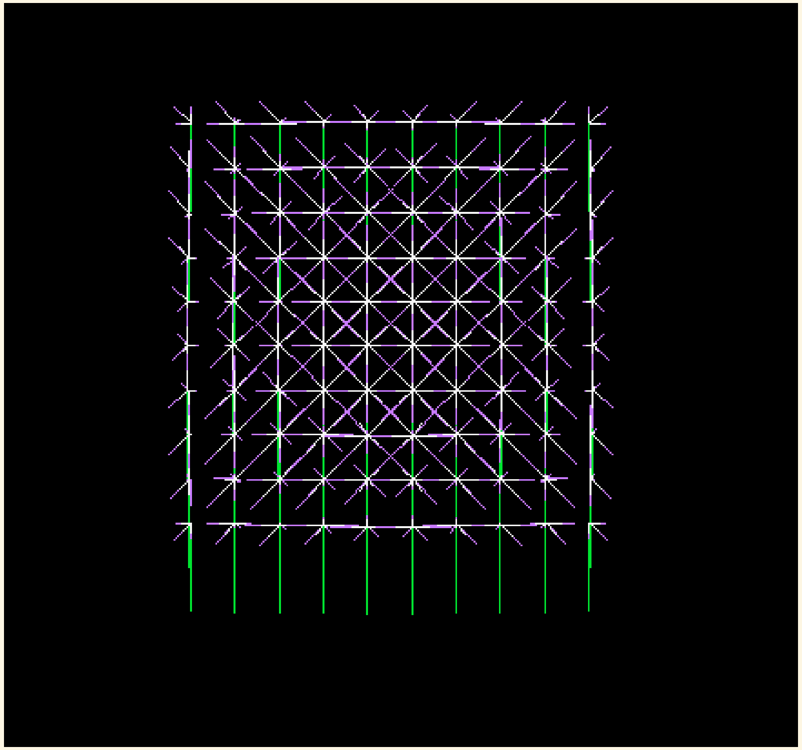
```
v[t + 1] = v[t] + acceleration(t) * dt
x[t + 1] = x[t] + v[t + 1] * dt
```

Méthode d’Euler implicite



Effet de respiration grâce au frottement





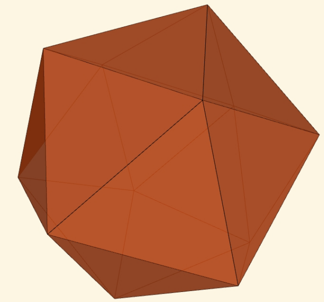
gravité

frottement amortisseur

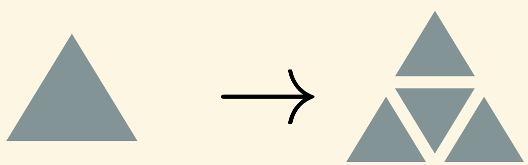
force de ressort élastique

Construction de l’icosphère

On démarre avec un icosaèdre (20 faces) :



Pour chacune des faces, on applique la transformation :



4 triangles

Puis on norme les vecteurs des sommets pour qu’ils soient distants de R avec le centre

On itère ce procédé pour diminuer la rugosité de la sphère

Il y aura ainsi 20×3^n sommets après n itérations