Rapport de stage

# Fracturation de floes de glace par percussion dans un modèle granulaire

**_Superviseur_**
Stéphane Labbé

**_Étudiant_**
Roussel Desmond Nzoyem

**_Enseignant référent_**
Christophe Prud'homme

*Stage effectué au Laboratoire Jacques-Louis Lions ;*
*du 03 février 2021, au 31 juillet 2021 ;*
*pour l'obtention du master 2 CSMI.*

Année académique 2020 - 2021

14/8/2021

# *Remerciements*

Avant tout développement sur cette expérience professionnelle, il apparaît opportun de commencer ce rapport de stage par des remerciements, à ceux qui mont appris des choses, et à ceux qui ont eu la gentillesse de faire de ce stage un moment agréable et profitable.

Ainsi, je remercie le Pr. Stéphane Labbé, mon maître de stage qui ma formé et accompagné tout au long de cette expérience avec beaucoup de patience et de pédagogie. Étant donné la situation sanitaire de COVID-19, il a su me transmettre tous les enseignements et les ressources (livres, reférences, etc.) nécessaires pour effectuer mes différentes missions (et bien plus encore), à distance comme en présentiel. Je vous en suis profondément reconnaissant.

J'éttends mes remerciements à mes illustre prédécesseurs Matthias Rabatel et Dimitri Balasoiui sans qui mon travail n'aurait pas eu lieu. Dimitri a su me guider dans les moments les plus difficile du stage. Les mots ne sauraient exprimer ma reconnaissance envers les visio-conférence organisées afin de me permettre de prendre en main de son travail.

Je remercie aussi l'ensemble du personnel du Laboratoire Jacques-Louis Lions qui ma permis deffectuer un stage scientifique très enrichissant dans les meilleures conditions possibles. J'addresses mes salutations aux doctorants et aux étudiants en séjour de recherche pour leur unique regard sur les difficultés auquelles fait face. En particulier, je remercie Madame Catherine Drouet de l'administration pour son assistance et ses conseils inestimables.

Enfin, je remercie mes proches, ma famille et mes amis pour leurs encouragements. Si un lecteur estime que son nom aurait du figurer ici de facon explicite, faite ceci : imprimer cette page, montrer la moi, et cela sera votre coupon pour une bièrre gratuite (ou un caffé, ou autre chose). Les trucs gratuits sont mieux qu'une mention, n'est-ce pas ?

# Table des matières

# Chapitre 1

# Problème 2D et percussion des floes

Dans cette section, nous étudierons le floe de glace en deux dimention (2D). Nous reprendrons le même chéminement qu'en 1D. En d'autres termes, nous partirons de l'approche approche par réseau de ressort introduite par Balasoiu pour modéliser dans un premier temps le déplacement d'un floe de glace contenant juste trois noeuds (masses), ensuite nous modéliserons la percussion inélastique sans rebond des noeuds de floes.

## 1.1 Dévelopement d'un modèle de déplacement des noeuds

Étudions le comportement d'un floe de glace 2D modélisé par un réseau de ressorts (3 ressort, 3 dispositif viseux, et 3 noeuds) (voir figure 1.1).

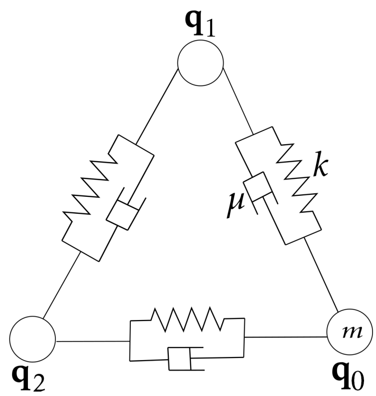



**Figure 1.1 –** Floe de glace 2D modélisé par un réseau de ressorts. Le floe est isolé de toutes forces extérieurs. Tous les neouds du réseau ont la même masse $m$, tous les ressorts on la même raideur $k$, et tous les dispositifs visquens ont le même coefficient $\mu$.

Comme nous l'avons présenté aux équations (1.4) et ($E$), le système de la figure 1.1 est régit par l' équation :

$$\forall i \in \mathbb{Z}/3\mathbb{Z}, \quad m\ddot{\mathbf{q}}_i = \sum_{j=i+1}^{i+2} C_{ij} \left[ k \left( \|\mathbf{q}_j - \mathbf{q}_i\| - L_{ij} \right) \mathbf{u}_{ij} - \mu \left\langle \dot{\mathbf{q}}_j - \dot{\mathbf{q}}_i, \mathbf{u}_{ij} \right\rangle \mathbf{u}_{ij} \right], \tag{1.1}$$

où $L_{ij}$ représente la longeur au repos du ressort entre les noeuds $i$ et $j$, et $C_{ij}$ indique si les noeuds $i$ et $j$ sont connectés ou non (pour ce modèle 2D simple, $C_{ij} = 1\ \forall 0 \leq i, j \leq 2$). Le vecteur unitaire $\mathbf{u}_{ij}$ vaut :

$$\mathbf{u}_{ij} = \frac{\mathbf{q}_j - \mathbf{q}_i}{\|\mathbf{q}_j - \mathbf{q}_i\|}.$$

**Simulation par un schéma d'Euler explicite.** On discrétise par un schéma de différence finies avec $N+1$ pas de temps, et pour une temps de simulations $T$ :

$$\forall i \in \mathbb{Z}/3\mathbb{Z},\ \forall n \in [\![0, N]\!], \quad t^n = n\Delta t = n\frac{T}{N}, \quad \mathbf{q}_i(t^n) \approx \mathbf{q}_i^n.$$

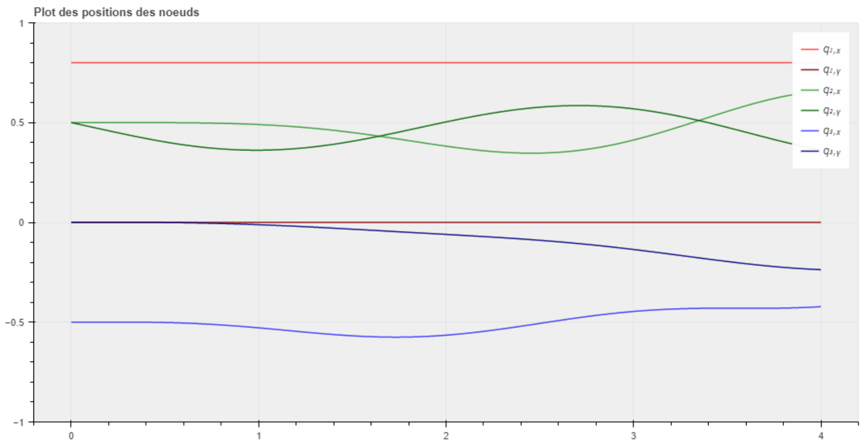
L'équation (1.1) devient :

$$m\frac{\mathbf{q}_i^{n+1} - 2\mathbf{q}_i^n + \mathbf{q}_i^{n-1}}{\Delta t^2} = \sum_{j=i+1}^{i+2} C_{ij}\left[k\left(\|\mathbf{q}_j^n - \mathbf{q}_i^n\| - L_{ij}\right)\mathbf{u}_{ij} - \mu\left\langle \frac{\mathbf{q}_j^n - \mathbf{q}_j^{n-1}}{\Delta t} - \frac{\mathbf{q}_i^n - \mathbf{q}_i^{n-1}}{\Delta t}, \mathbf{u}_{ij}\right\rangle \mathbf{u}_{ij}\right],$$

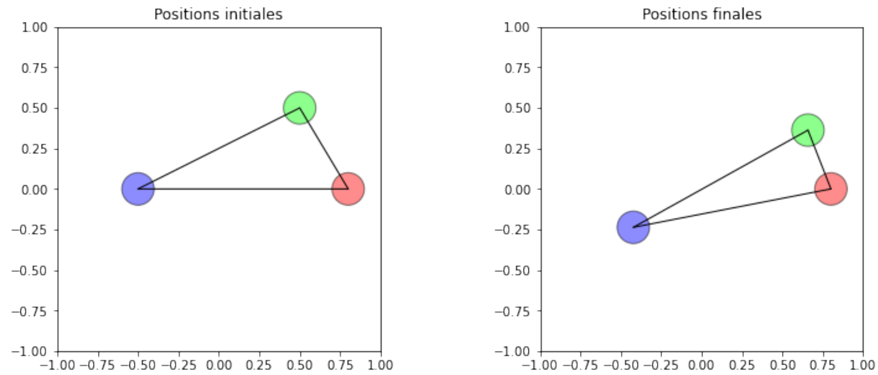
soit encore :

$$\mathbf{q}_i^{n+1} = 2\mathbf{q}_i^n - \mathbf{q}_i^{n-1} + \frac{\Delta t^2}{m}\sum_{j=i+1}^{i+2} C_{ij}\left[k\left(\|\mathbf{q}_j^n - \mathbf{q}_i^n\| - L_{ij}\right)\mathbf{u}_{ij} - \frac{\mu}{\Delta t}\left\langle \mathbf{q}_j^n - \mathbf{q}_j^{n-1} - \mathbf{q}_i^n + \mathbf{q}_i^{n-1}, \mathbf{u}_{ij}\right\rangle \mathbf{u}_{ij}\right]. \tag{1.2}$$

La simulation de ce modèle par un schéma d'Euler explicite à pas constant sur un intervalle de temps faible ($T = 4$) est présentée à la figure 1.2, ainsi que les positions des 2 noeuds au début et à la fin de la simulation. La simulation à la figure 1.3 permet d'observer le problème avec ce schéma ($T = 10$).




**(a)** Simulation des positions des noeuds.




**(b)** Illustration des positions initiales et finales des neouds.

**Figure 1.2** – Simulation du système 1.2 par un schéma d'Euler explicite avec $T = 4$ lorsqu'**un des noeuds est fixé** (le noeud rouge). La couleur rouge repésente le noeud $\mathbf{q}_1$, le vert le noeud $\mathbf{q}_2$, le blue le $\mathbf{q}_3$. Les paramètres utilisés ici sont les suivants : $m = 6.2$, $k = 23.3$, $\mu = 3$ ; à l'instant initiale, les trois noeuds perturbés avec des vitesses d'intensité respectives $v_1 = 0.3$, $v_2 = 0.1$, et $v_3 = 0.1$. Par rapport à l'axe des abcisses, ces vitesses ont sont orientées respectivement de $\theta_1 = 180^\circ$, $\theta_2 = 270^\circ$, et $\theta_3 = 240^\circ$ (voir `code/simu2D/Deplacement2D-1.ipynb`).

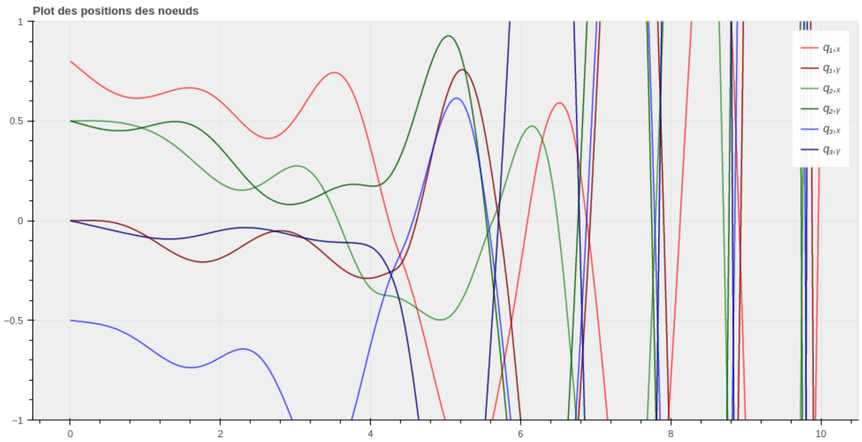



**Figure 1.3** – Simulation du système 1.2 par un schéma d'Euler explicite avec $T = 10$ lorsqu'**aucun des noeuds n'est fixé**. Cette figure utilise les mêmes paramtètres que la figure 1.2. On observe ici une divergence complète du système (voir `code/simu2D/Deplacement2D-2.ipynb`).

Les figures 1.2 et 1.3 permettent de constater que le schéma d'Euler explicite (peu importe son pas de temps), n'est pas adapté à ce problème. Nous étudierons donc d'autres alternatives.

**Simulation à l'aide des fonctions de la librarie Scipy.** À travers ses fonction telle que `odeint` et `solve_ivp`, `Scipy` offre une solution robuste et élégante pour simuler les systèmes d'ODE de la forme $Y' = AY$. Avec `Scipy`, nous résolvons numériquement nos EDO à l'ordre RK45. Autrement dit, l'erreur est controllée par un schéma de Runke-Kutta à l'ordre 4, et les pas de temps sont pris à l'ordre 5.

## 1.2 Dévelopement d'un modèle de percussion des floes

### 1.2.1 Présentation des travaux antérieurs

Les travaus antérieures sur le problème 2D ont été présentés dans la **??**. Particulièrement, nous avons adopté et amoliorer des résultats obtenus par D. Balasiou [Bal20]. Ces résultats concernent la simulation du comportement des floes de glace une fois une fois percuté par un object ponctuel ; et on été présentés à la **??** (voir aussi figure 1.4 pour une interface graphique web pour interagir avec le programme). Ici, nous explicitons les points sur lesquels nous nous sommes directement inspirés pour notre travail de stage.

### 1.2.2 Nouveaux travaux sur la percussion

Les floes de glace $\Omega_k$ et $\Omega_l$ sont modélisés par des systèmes masse-ressort (à grande raideur). Pour l'instant, nous considérons une moélisation simplifiée qui assimile un floe à un système de (trois) masses reliés par des ressorts (de constante de raideur $k$), et par des dispositifs visqueux de constante $\mu$. Nous désignerons par $n+1$ le nombre total de noeuds du floe $\Omega_k$, chaque noeud ayant pour masse $m$. De facon similaire, on définit les constantes $k'$, $\mu'$, $n'+1$, $m'+1$ pour le floe $\Omega_l$. Les positions des noeds de $\Omega_k$ seront noté $(q_i)_{0\leq i\leq n}$, tandis que ceux de $\Omega_l$ seront notés $(p_i)_{0\leq i\leq n'-1}$ (voir figure 1.5).

On définit la longeur à vide $L_{0j}$ et le vecteur unitaire $\mathbf{u}_{0j}$ (orienté de $\mathbf{q}_0$ vers $\mathbf{q}_j$) entre le noeud $\mathbf{q}_0$ et le noeud $\mathbf{q}_j$ et $u_{0j}$. Notons que ces définittions restent valable si nous remplacons le noeud $\mathbf{q}_0$ par un noeud $(\mathbf{q}_i)_{0\leq i\leq n-1}$. On définit aussi la matrice de connectivité $C$ :

$$\forall\, 0 \leq i < j \leq n-1, \quad C_{ij} = C_{ji} = \begin{cases} 1 \text{ si } \mathbf{q}_i \text{ est voisin de } \mathbf{q}_j\,, \\ 0 \text{ sinon }. \end{cases}$$

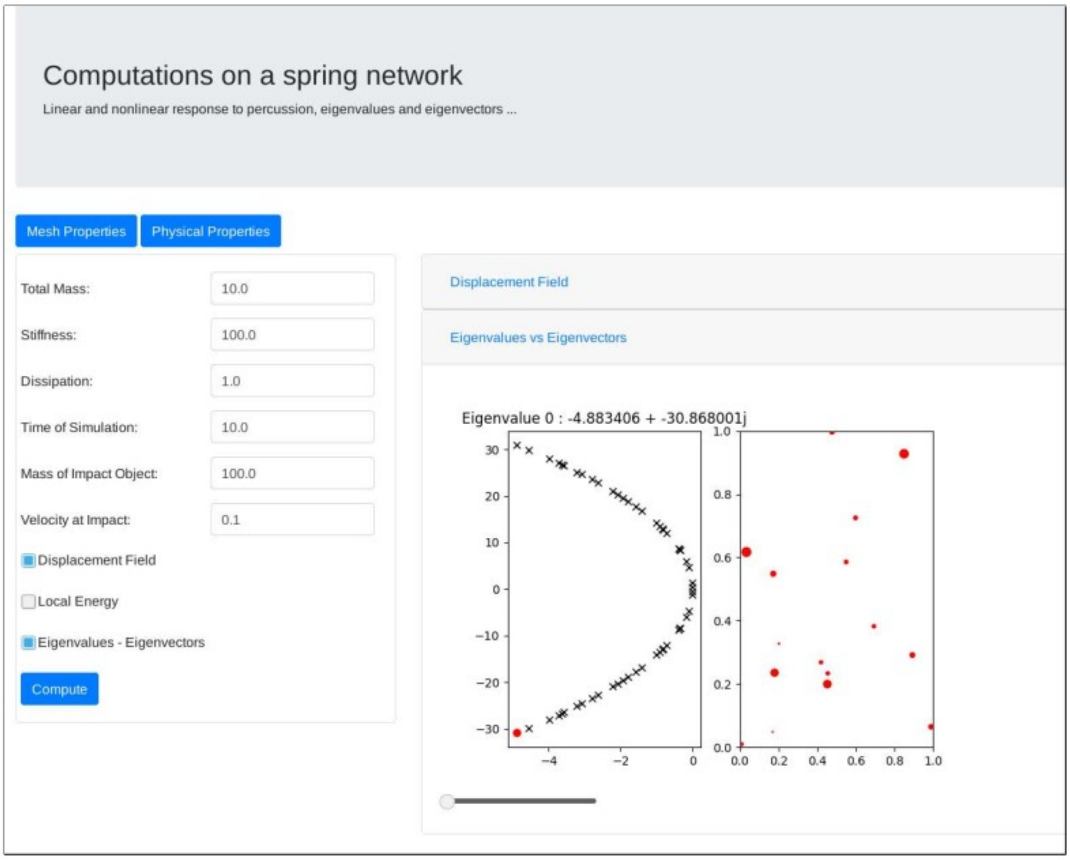



**Figure 1.4** – Client web `springslattice-web.py` développé dans [Bal20, p.197].

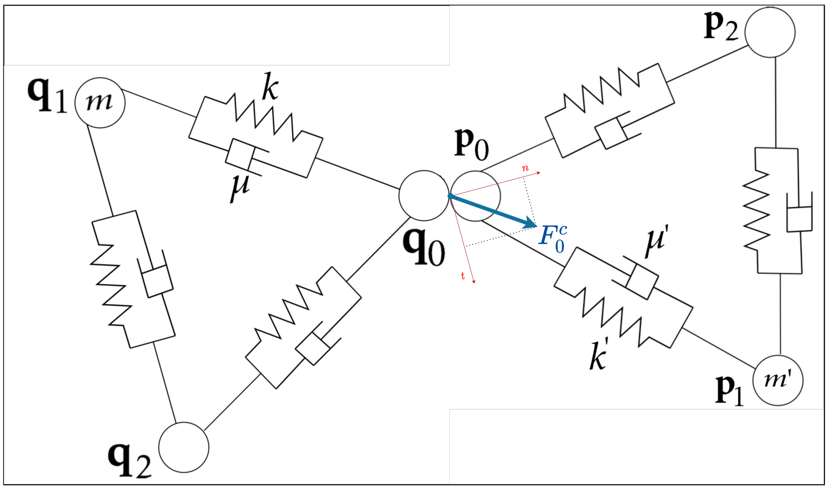



**Figure 1.5** – Contact entre deux floes aux points $\mathbf{q}_0$ et $\mathbf{p}_0$ respectivement.

Comme présenté dans les travaux [Bal20, p.186], le système différentiel qui modélise la percussion sécrit comme le couplage de deux sous-systèmes. Le premier, dit système intérieur (SI), est à évolution rapide et modélise la propagation des ondes élastiques dans le système masse-ressort. Ici, nous dérivons facilement et réutilisont le SI comme présenté par Balasoiu. Le second, dit système extérieur (SE), est à évolution lente et modélise la pénétration de lobjet solide dans le système masse-ressorts. Pour dériver le SE sur le floe $\Omega_k$, nous écrivons l'équation de Newton-Euler linéaire[1] au point de contact $q_0$ :

$$m\ddot{\mathbf{q}}_0 = \mathbf{F}_0 + \mathbf{F}_0^c\,, \tag{1.3}$$

où

$$\mathbf{F}_0 = \sum_{j=0}^{n} C_{0j}\left[\underbrace{k\left(\|\mathbf{q}_j - \mathbf{q}_0\| - L_{0j}\right)\mathbf{u}_{0j}}_{\text{Force de rappel}} - \underbrace{\mu\left\langle \dot{\mathbf{q}}_j - \dot{\mathbf{q}}_0\,, \mathbf{u}_{0j}\right\rangle \mathbf{u}_{0j}}_{\text{Force de dissipation}}\right], \tag{1.4}$$

représente la somme des forces de reaction et de disssipation exercées par le ressort et le dispositif visqueux sur le noeud $q_0$ ; et $\mathbf{F}_0^c(t)$ la force de contact durant la collison entre les deux particules. En supposnat qu'il existe un repère de contact $\mathcal{R}^c = \{q_0, \mathbf{n}, \mathbf{t}\}$ associé au floe $\Omega_k$ (voir figure 1.5), on peut écrire, pour $(\lambda, \beta) \in \mathbb{R}^2$ :

$$\mathbf{F}_0^c = \lambda\mathbf{n} + \beta\mathbf{t}\,. \tag{1.5}$$

Le système intérieur (SE) s'obtient facilement en combinant les équations équations (1.3) à (1.5). Le système intérieur (SI) s'obtient lui (pour les autres noeuds du réseau) en y supprimant la force de contact. On obtient au final :

$$\begin{cases} m\ddot{\mathbf{q}}_0 = \mathbf{F}_0 + \mathbf{F}_0^c\,, & \text{(SE)} \\ m\ddot{\mathbf{q}}_i = \mathbf{F}_i\,, \qquad \forall 1 \le i \le n-1\,. & \text{(SI)} \end{cases} \tag{$E$}$$

En ce qui concerne le floe $\Omega_l$, nous procédons de facons similaire et appliqons la 3ème loi de Newton (action-réaction) pour obtenir le système :

$$\begin{cases} m'\ddot{\mathbf{p}}_0 = \mathbf{F}'_0 - \mathbf{F}_0^c\,, & \text{(SE)} \\ m'\ddot{\mathbf{p}}_i = \mathbf{F}'_i\,, \qquad \forall 1 \le i \le n'-1\,, & \text{(SI)} \end{cases} \tag{$E'$}$$

où $(\mathbf{F}'_i)_{0\le i\le n'}$ sont définis de facon similaire à $\mathbf{F}_0$ (voir équation (1.4)) :

$$\mathbf{F}'_i = \sum_{j=i}^{n'} C_{ij}\left[k'\left(\|\mathbf{p}_j - \mathbf{p}_i\| - L'_{ij}\right)\mathbf{u'}_{ij} - \mu'\left\langle \dot{\mathbf{p}}_j - \dot{\mathbf{p}}_i\,, \mathbf{u}'_{ij}\right\rangle \mathbf{u}'_{ij}\right]\,. \tag{1.6}$$

Ensuite, on additionne membre à membre les équations des systèmes extérieurs (SE) de équations ($E$) et ($E'$) pour éliminer la force de contact. On obtient :

$$m\ddot{\mathbf{q}}_0 + m'\ddot{\mathbf{p}}_0 = \mathbf{F}_0 + \mathbf{F}'_0\,. \tag{1.7}$$

Remarquons que les positions relatives des noeuds $\mathbf{q}_0$ et $\mathbf{p}_0$ restent inchangées durant la collision. A l'instant initial, on note donc $\Delta_0 = \mathbf{q}_0(0) - \mathbf{p}_0(0)$, et $\dot{\mathbf{q}}_0(0) = \dot{\mathbf{p}}_0(0)$ ; idéalement, nous voudrions que :

$$\forall t \in \mathbb{R}^+\,, \quad \mathbf{q}_0(t) - \mathbf{p}_0(t) = \Delta_0\,. \tag{1.8}$$

1. La rotation du point matériel $q_0$ n'est pas prise en compte ici, d'où l'abscence de l'équation de Newton-Euler angulaire.

Pour satifaire cette condition, nous exhibons autant d'équations nécessaire pour que notre problème de percussion soit bien posé. Elles sont :

$$\begin{cases} (m + m')\ddot{\mathrm{q}}_0 = \mathbf{F}_0 + \mathbf{F}_0^{'} \,, & \text{(SE)} \\ \ddot{\mathrm{p}}_0 = \ddot{\mathrm{q}}_0 \,, \quad \dot{\mathrm{p}}_0 = \dot{\mathrm{q}}_0 \,, \quad \mathrm{p}_0 = \mathrm{q}_0 - \Delta_0 \,, & \text{(SE)} \\ m\ddot{\mathrm{q}}_i = \mathbf{F}_i \,, \qquad \forall 1 \leq i \leq n-1 \,. & \text{(SI)} \\ m'\ddot{\mathrm{p}}_i = \mathbf{F}_i^{'} \,, \qquad \forall 1 \leq i \leq n'-1 \,, & \text{(SI)} \end{cases} \tag{$\mathcal{P}$}$$

Quant à la visualisation des résultats, nous avons construit une interface graphique web pour interagir avec le programme. A travers cette interface, nous pouvons modifier tous les parametres du problèmes et lancer des simulations. Un tel exemple se trouve à la figure figure 1.6, et la simulation correspondant peut etre visualisée via ce lien SEAFILE.

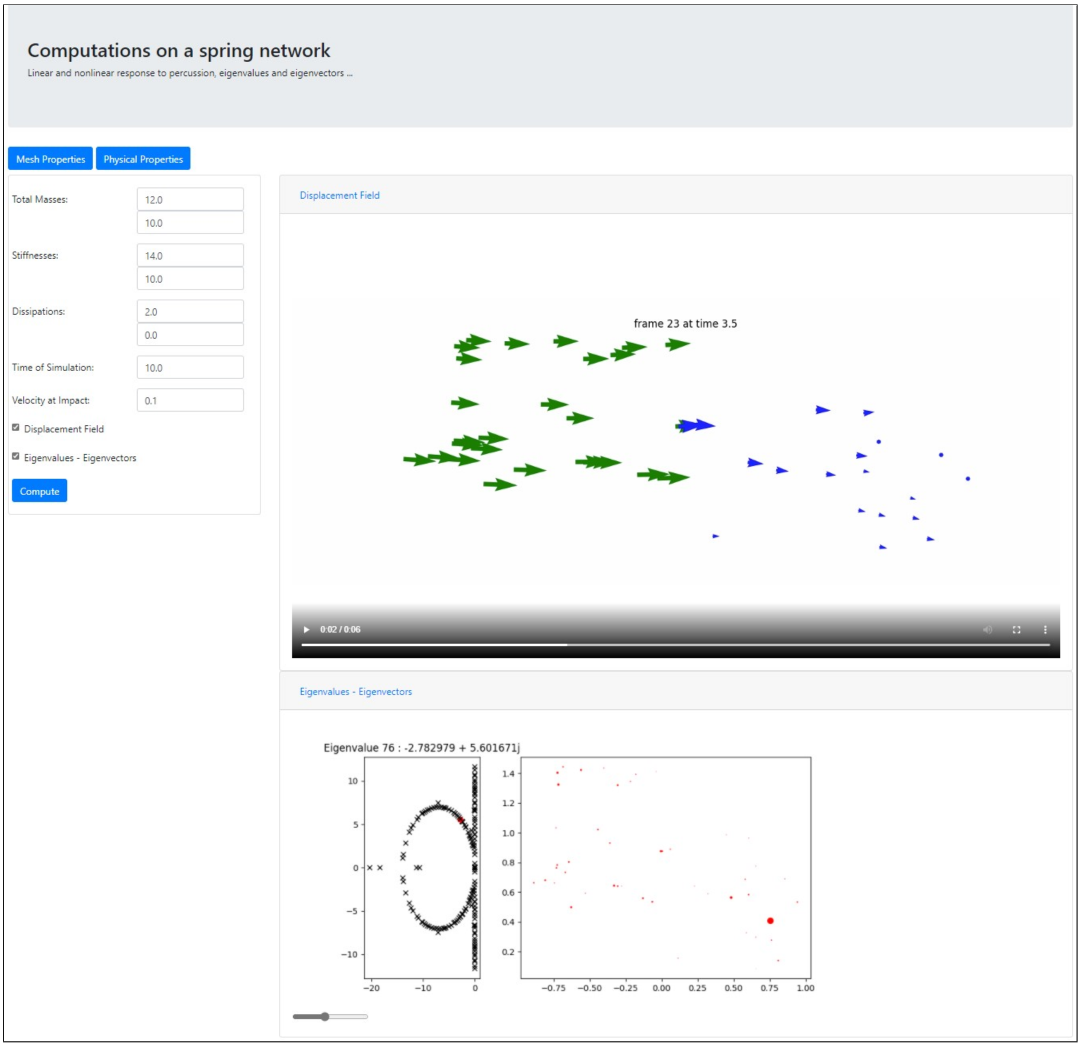



**Figure 1.6** – Client web développé durant le stage pour les percussion 2D. Comparativement à la figure 1.4, on constate la présence de plus de champ pour préciser les paramètre du second floe entrant en collision.

## 1.3 Code de calcul 2D

L'algorithme de calcul 2D est largement inspiré des travaux de Balasoiu. En effet, nous avons ajouté des modules Python à la librarie `springslattice` qu'il à développé pour traiter les réseaux de ressorts. Ces modules sont :

1. **multimesh** : pour la création d'un réseau de ressort consitué de deux floes de glace en contact, que nous considérerons comme un maillage.
2. **multisolver** : pour la création du solveur 2D suivant l'équation ($\mathcal{P}$) pour simuler les noeuds d'un objet `multimesh`. Une fois les caluls fait dans la fonction `Fhom`, nous les vérifions à l'aide d'un schéma symplectique (voir **??**).
3. **percussion-cli** : pour l'execution des simulations avec des paramètres donnés en ligne de commande (ou ceux insérer par défaut).
4. **percussion-web** : pour l'execution des simulations avec des paramètres donnés dans une interface web. Ici, on peut en plus observer les vecteur et les valeurs propres (toutes négatives) du système.

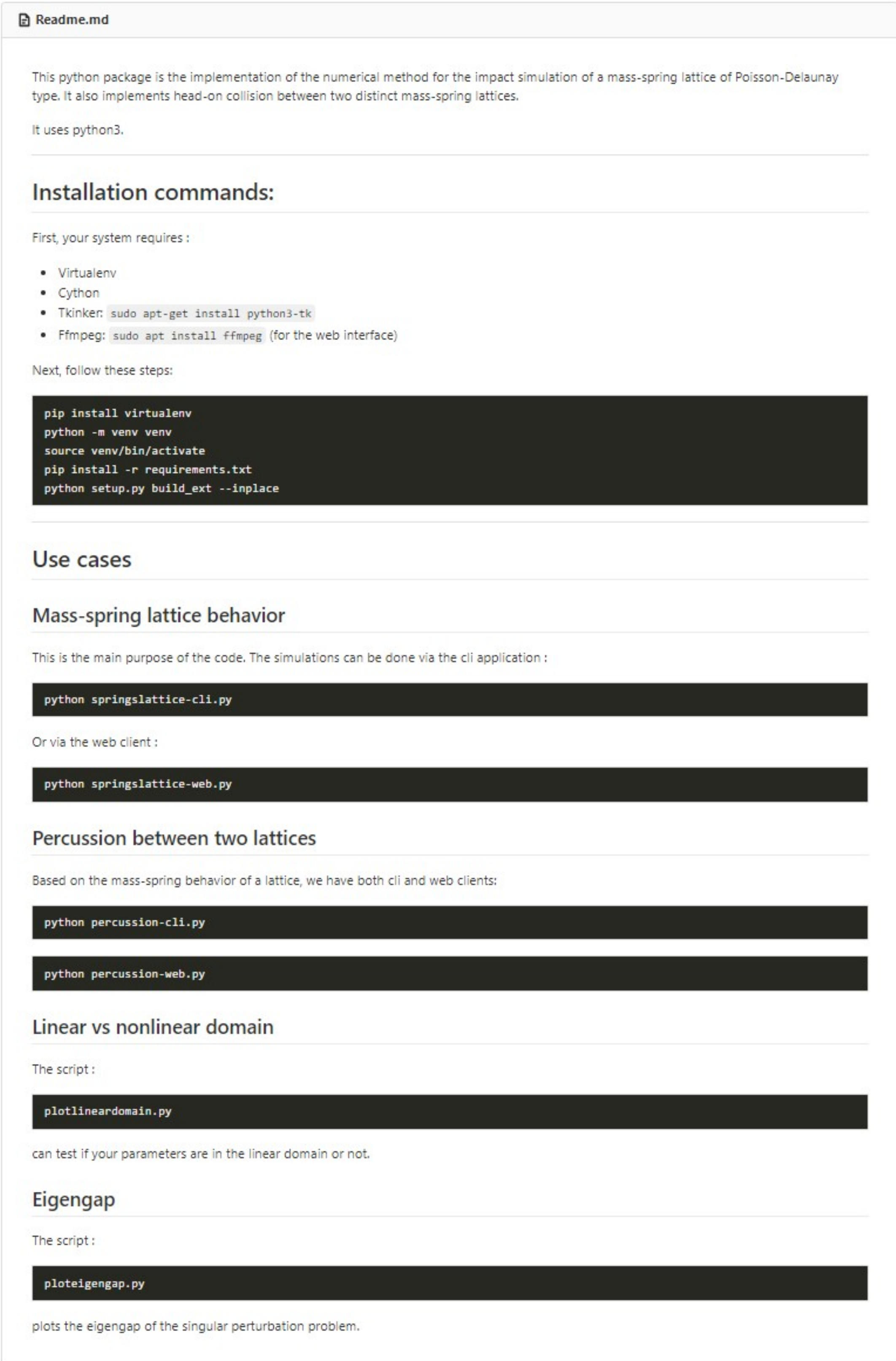
Nous présentons à la figure 1.7 le fichier README du dépot `springslattice` sur `Framagit` créer par Balasoiu dans ses travaux, et amélioré par nous durant ce stage. Tout comme le dépot principal ice-floes sur `GitHub`, ce dépot secondaire est privé.

## 1.4 Résumé des résultats obtenus

Bien que le modèle de fracture 1D soit facilement adaptable au calculs 2D, nous ne l'avons pas fait fait faute de temps. En particulier, nous avons :

— Modélisation du deplacement des noeuds d'un floe de glace 2D soumis à aucune force sur bord ;
— Modélisation de le collision (élastique) de deux floe de glace. Vu que le temps de la collision est connu (voir [Rab15]), nous pouvons limiter étendre ces travaux à la collision avec séparation des masses, comme nous l'avons fait en 1D.
— Création d'un interface web pour introduire les paramètres et obtenir les résultats d'une simulation.

Les travaux obtenus en 2D sont prometteurs, tout comme ceux obtenu en 1D. Ils ont été obtenu pendant un stage qui a demandé un discipline et une encadrement de taille. Au bout de ce travail, j'ai pu en tirer plusieurs bénéfices.



Readme.md

This python package is the implementation of the numerical method for the impact simulation of a mass-spring lattice of Poisson-Delaunay type. It also implements head-on collision between two distinct mass-spring lattices.

It uses python3.

## Installation commands:

First, your system requires :

- Virtualenv
- Cython
- Tkinker: `sudo apt-get install python3-tk`
- Ffmpeg: `sudo apt install ffmpeg` (for the web interface)

Next, follow these steps:

```
pip install virtualenv
python -m venv venv
source venv/bin/activate
pip install -r requirements.txt
python setup.py build_ext --inplace
```

## Use cases

### Mass-spring lattice behavior

This is the main purpose of the code. The simulations can be done via the cli application :

```
python springslattice-cli.py
```

Or via the web client :

```
python springslattice-web.py
```

### Percussion between two lattices

Based on the mass-spring behavior of a lattice, we have both cli and web clients:

```
python percussion-cli.py
```

```
python percussion-web.py
```

### Linear vs nonlinear domain

The script :

```
plotlineardomain.py
```

can test if your parameters are in the linear domain or not.

### Eigengap

The script :

```
ploteigengap.py
```

plots the eigengap of the singular perturbation problem.

**Figure 1.7 –** Appercu du dépot secondaire springslattice maintenu et amélioré durant le stage.

# Bibliographie

[Bal20] Dimitri Balasoiu. « Modélisation et simulation du comportement mécanique de floes de glace ». Theses. Université Grenoble Alpes [2020-....], oct. 2020. url : https://tel.archives-ouvertes.fr/tel-03116132.

[Rab15] Matthias Rabatel. « Modélisation dynamique d'un assemblage de floes rigides ». Theses. Université Grenoble Alpes, nov. 2015. url : https://tel.archives-ouvertes.fr/tel-01293341.