

## Compte rendu semaine #13

Etudiant: *Roussel Desmond Nzoyem*

---

UE: *Stage M2* – Superviseur: *Pr. Stéphane Labbé*

Date: *28/4/2021 - 3/5/2021*

Cette semaine, j'ai cherché les raisons de la non-convergence du modèle en simulant le modèle de Dimitri en 2D. Avec l'aide de Dimitri, j'ai maintenant une idée de la source du problème, et de comment je pourrais le résoudre. J'ai aussi de nouvelles idées sur comment avancer par la suite.

### Tâches effectuées

1. Vérification du modèle et du code de calcul 2D (en pièce jointe). J'ai joué avec les signes dans la simulation afin de trouver une configuration qui marchait; je me suis aperçu que le signe du vecteur normal changeait beaucoup de choses, mais que ma configuration initiale (celle de la semaine passée) était la bonne; Pour déboguer, j'ai non seulement simulé les positions des nœuds, mais j'ai aussi dessiné le système masse ressort en Python.
2. La deuxième tâche était de relire le code et la thèse de Dimitri afin de trouver des nouvelles idées.

### Difficultés rencontrées

1. La première difficulté était de trouver pourquoi les simulations 1D n'aboutissaient pas; Mon schéma d'Euler explicite<sup>1</sup> à pas constant (bien que très faible) n'aboutit pas probablement parce qu'il n'est pas adapté à la situation. On devrait utiliser un schéma d'Euler semi-simplicte (semi-implicite) pour avoir une chance d'observer un comportement qui conserve le Hamiltonien du système. On peut aussi utiliser les fonctions de `scipy`, en locurence `odeint` et `solve_ivp` comme l'a fait Dimitri.
2. La deuxième grosse difficulté rencontrée est de faire fonctionner le code `simu-ressort` de Dimitri.

### Travail à venir

Après avoir discuter avec Dimitri, :

1. j'ai une meilleure compréhension de code, et du système linéarisé autour de la position d'équilibre dont il s'est servi pour ses simulations.
2. Une meilleure compréhension de sa thèse [Bal20], en particulier l'introduction du  $\varepsilon$  au Chapitre 6 qui permet non seulement d'augmenter la raideur et la dissipation du système, mais aussi d'augmenter la force appliquée sur la particule la plus à gauche dans le système masse-ressort à l'instant initial.

---

<sup>1</sup>La non-convergence est probablement due à une accumulation des erreurs à chaque pas de temps.

3. j'ai une idée de comment transformer le problème de percussion en un problème de Dirichlet sans passer par les vitesses après contact. En effet, il me suffit de calculer la force de contact, afin d'utiliser directement le code de Dimitri (qui fonctionne à présent). En se servant des 8 ou 9 valeurs propres non évanescentes (qui restent proches de 0 peu importe la raideur et dissipation du système), on peut calculer le déplacement maximal  $\max_{t \in \mathbb{R}^+} \|\mathbf{q}(t) - \mathbf{q}(0)\|$  de la particule percutée. Et c'est ce déplacement maximal<sup>2</sup> qui nous sert dans le calcul de l'initiation et de la propagation de la fracture.
4. j'ai appris que dans l'interface web du code de Dimitri, le système (dans l'ensemble) ne se déplace pas. C'est parce qu'une particule (celle en bas à droite) a été fixée. Il faudrait penser à résoudre ce problème plus tard en intégrant ce mouvement solide dans le calcul du déplacement maximal de la particule percutée.
5. j'ai appris qu'il faut faire des recherches sur les différents régimes qu'on observe lorsqu'on modifie la raideur et la dissipation du système masse-ressort (en percutant la particule la plus à gauche); en particulier le régime *visco-élastique*, et l'*évanescence*.

---

<sup>2</sup>On se contente du déplacement maximal parce que d'après les calculs de Dimitri, le système vibre indéfiniment malgré le dispositif visqueux.

## References

- [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>.