V. Degrooff, C. Heneffe, N. Tihon, J.-F. Remacle.

Instructions. *Les séances tutorées ont lieu les mardi et mercredi de 16h15 à 18h15. Veuillez vous rendre à la séance qui vous a été attribuée. Des étudiants moniteurs serront présents pour vous aider aussi bien au niveau théorique qu'au niveau de la programmation en langage C.*

1 L'algorithme de Newmark

On souhaite résoudre par une méthode numérique le système d'équations différentielles ordinaires du second ordre suivant :

$$M\ddot{\mathbf{q}} + C\dot{\mathbf{q}} + K\mathbf{q} = \mathbf{f}(t)$$

Ici, les lettres majuscules dénotent des matrices $K \in \mathbb{R}^{n \times n}$, les lettres en caractère gras désignent des vecteurs $\mathbf{q} \in \mathbb{R}^n$ et les lettes minuscules désignent des réels $t \in \mathbb{R}$.

Dans le cadre des éléments finis pour la dynamique du solide, K est est la matrice de raideur, C une matrice d'amortissement et M la matrice de masse. Il faut noter ici qu'alors que K et M sont bien définies dans le cadre de l'élastodynamique linéaire, la matrice d'amortissement n'est pas naturellement définie. On utilise souvent une combinaison linéaire du type $C = aM + bK$ (amortissement de Rayleigh). Le vecteur $\mathbf{q}(t)$ est le vecteur des déplacements nodaux de la structure.

Pour se ramener au cas des équations différentielles du premier ordre, on peut définir

$$\mathbf{z} = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{q} \end{bmatrix}$$

ce qui implique

$$\begin{bmatrix} M & 0 \\ 0 & D \end{bmatrix} \dot{\mathbf{z}} + \begin{bmatrix} C & K \\ -D & 0 \end{bmatrix} \mathbf{z} = \begin{bmatrix} f \\ 0 \end{bmatrix}$$

où D est inversible (on utilise habituellement $D = I$).

Cette façon de procéder n'est pas nécessairement optimale car on doit résoudre des systèmes deux fois plus grands et la matrice devant $\mathbf{z}$ n'est pas SDP alors que K et D pourraient l'être.

Le schéma de Newmark est la référence en dynamique des solides/structures. Imaginons que nous connaissions $\mathbf{q}(t_n)$ et que nous voulions calculer $\mathbf{q}(t_n + h)$.

Pasons maintenant en mode "discret". Appelons $\mathbf{q}_n = \mathbf{q}(t_n)$ et $\mathbf{q}_{n+1} = \mathbf{q}(t_n + h)$. Le pas de temps h peut être variable mais appelons le h. La méthode de Newmark suppose d'abord que l'accélération varie linéairement entre chaque pas de temps

$$\ddot{\mathbf{q}}_{n+1} = \ddot{\mathbf{q}}_n + h\mathbf{q}_n^{(3)}.$$

Les accroissements finis nous disent que

$$\dot{\mathbf{q}}_{n+1} = \dot{\mathbf{q}}_n + h\ddot{\mathbf{q}}_\gamma$$

avec $\ddot{\mathbf{q}}_\gamma$ évalué quelque part entre n et $n+1$. Si l'accélération varie linéairement entre les pas de temps,

$$\ddot{\mathbf{q}}_\gamma = (1-\gamma)\ddot{\mathbf{q}}_n + \gamma\ddot{\mathbf{q}}_{n+1}$$

pour $0 \leq \gamma \leq 1$ et donc

$$\dot{\mathbf{q}}_{n+1} = \dot{\mathbf{q}}_n + h\left[(1-\gamma)\ddot{\mathbf{q}}_n + \gamma\ddot{\mathbf{q}}_{n+1}\right]. \tag{1}$$

De même

$$\mathbf{q}_{n+1} = \mathbf{q}_n + h\dot{\mathbf{q}}_n + \frac{h^2}{2}\ddot{\mathbf{q}}_\beta$$

avec $\ddot{\mathbf{q}}_\beta$ évalué quelque part entre n et $n+1$. Si l'accélération varie linéairement entre les pas de temps,

$$\ddot{\mathbf{q}}_\beta = (1-2\beta)\ddot{\mathbf{q}}_n + 2\beta\ddot{\mathbf{q}}_{n+1}$$

pour $0 \leq 2\beta \leq 1$ et donc

$$\mathbf{q}_{n+1} = \mathbf{q}_n + h\dot{\mathbf{q}}_n + \frac{h^2}{2}\left[(1-2\beta)\ddot{\mathbf{q}}_n + 2\beta\ddot{\mathbf{q}}_{n+1}\right]. \tag{2}$$

Les équations (1) et (2) sont les équations de la méthode de Newmark. Elles dépendent évidemment de deux paramètres β et γ qui vont influencer la précison (erreur de troncature) et la stabilité de la méthode.

On suppose que $\mathbf{q}_0$ et $\dot{\mathbf{q}}_0$ sont connus et que l'on va calculer $\mathbf{q}_{n+1}$ et $\dot{\mathbf{q}}_{n+1}$ à partir de $\mathbf{q}_n$ et $\dot{\mathbf{q}}_n$ pour $t_{n+1} = t_n + h$.

Dans le cadre de ce devoir, on suppose des oscillations libres sans amortissement. Il existe donc une relation $M\ddot{\mathbf{q}} = -K\mathbf{q}$ qui lie $\mathbf{q}$ et $\ddot{\mathbf{q}}$. On doit donc résoudre un système linéaire à chaque itération temporelle

$$(M + h^2\beta K)\mathbf{q}_{n+1} = \left[M - \frac{h^2}{2}(1-2\beta)K\right]\mathbf{q}_n + h\mathbf{p}_n \tag{3}$$

$$\mathbf{p}_{n+1} = \mathbf{p}_n - hK\left[(1-\gamma)\mathbf{q}_n + \gamma\mathbf{q}_{n+1}\right], \tag{4}$$

où $\mathbf{p} = M\dot{\mathbf{q}}$.

2 Implémentation

L'algorithme de Newmark est largement documenté et il existe une version de cet algorithme i.e. un choix de β et γ qui est symplectique i.e. qui conserve exactement l'énergie (ce qui est la cas quand l'amortissement C est nul). Utilisez cette version de l'algorithme de Newmark pour résoudre le problème de dynamique des structures pour la pièce mécanique étudiée dans le devoir 2.

Vous pouvez utiliser votre version de l'algorithme de résolution sur matrice creuse ou utiliser le corrigé pour résoudre le système linéaire (3) à chaque itération.

Vous pouvez évidemment repartir du code fourni dans le devoir 2. Les matrices M et K y sont déjà construites sous format bande :

```
SymBandMatrix *K_band = model->K;
SymBandMatrix *M_band = model->M;
```

et peuvent être transformées au format `CSR` de votre choix. Soyez attentifs au fait que les entrées non-nulles de M sont strictement inclues dans celles de K.

Pour ce dernier devoir, nous vous laissons davantage de liberté dans l'implémentation de votre code. On vous demande :

1. De lire un fichier <initial.txt> qui contient les conditions initiales $\mathbf{q}_0$ et $\dot{\mathbf{q}}_0$. Ce fichier contient autant de lignes qu'il y a de nœuds dans le maillage et chacune contient 4 entrées :

   ```
   ux_i uy_i vx_i vy_i
   ```

2. De réaliser les itérations temporelles de l'algorithme de Newmark jusqu'à un temps final `T` donné, avec un pas de temps `dt` donné.
3. D'écrire dans un fichier <final.txt> le déplacement et la vitesse au temps `T`, au même format que le fichier <initial.txt>.
4. D'écrire dans un fichier <time.txt> le déplacement et la vitesse d'un nœud `I` à chaque itération temporelle. Chaque ligne du fichier contiendra donc 5 entrées, à savoir

   ```
   t_i ux_i uy_i vx_i vy_i
   ```

Les variables en virgule flottante seront écrites au format `.15le`.

Pour compiler et exécuter le code, nous utiliserons les commandes suivantes :

```
make
./deformation <model> <lc> <T> <dt> <initial.txt> <final.txt> <time.txt> <I>
```

3 Analyse des résultats

On vous demande ici de faire preuve d'imagination pour analyser vos résultats. Vous pouvez :

- créer une animation,
- faire une analyse *a posteriori* du déplacement en un noeud du maillage (FFT) et vérifier par exemple que les modes propres calculés en utilisant la méthode de la puissance inverse avec déflation correspondent à ce qu'on trouve en temporel
- vérifier que l'algorithme conserve exactement l'énergie,
- ...

Vous aurez compris que vous avez une grande liberté sur ce que vous faites avec ce dernier devoir. Pour l'évaluation nous valorisons la créativité, la rigueur dans la démarche et l'analyse, ainsi que le soin apporté à vos codes et votre rapport.

4 Questions théoriques

Le schéma de Newmark dépend de deux paramètres β et γ. On vous demande de répondre aux questions suivantes. Dans le cas qui nous occupe, le problème est exactement conservatif.

1. Montrez que le schéma de Newmark est consistant et que son erreur de troncature est au pire à l'ordre h.
2. Montrez que le schéma est globalement instable pour $\gamma < 1/2$.
3. Montrez que, si $\gamma \geq 1/2$ et $\beta \geq \frac{1}{4}(\gamma + 1/2)^2$, alors le schéma est inconditionnellement stable.
4. Montrez que, si $\gamma \geq 1/2$ et $\beta < \frac{1}{4}(\gamma + 1/2)^2$, alors le schéma est conditionnellement stable avec $h < h_c$.

5 Modalités

Pour le **16 mai 2025 à 23h59**, vous soumettrez par groupe de 2 sur Gradescope, **uniquement** :

— votre rapport réalisé avec LaTeX nommé `devoir_3.pdf`, et sa source `devoir_3.tex` où vous répondez aux questions théoriques et présentez vos analyses numériques,

— un dossier compressé `devoir_3.zip` (`< 50 Mo`), qui contient au moins : un Makefile, vos sources C, un script `.py` qui reproduit les figures de votre rapport. Ce `.zip` ne contiendra **PAS** le dossier `gmsh-sdk`.

Toutes les implémentations et les sources des rapports seront soumises à un logiciel anti-plagiat.

[28/04 Clarification $\dot{q}$, equations (3) et (4)]