

La harpe sympathique

1 - Conditions

Le programme implémente deux types de conditions initiales :

- ConditionsInitiales = 0 : corde **pincée** à la position $e/$
- ConditionsInitiales = 1 : corde **frappée** à la position $e/$

On retrouve deux types de conditions aux limites :

- ConditionsLimite = 0 : corde **encastrée** aux deux extrémités
- ConditionsLimite = 1 : corde sur **ressort** aux deux extrémités

Nous avons fait le choix de déclarer ces conditions en tant que **variables globales** afin de limiter le nombre d'arguments appelés par les fonctions du programme.

Les conditions (0,0) correspondent aux conditions d'*ExempleAssezFacile*.

Les conditions (1,1) correspondent aux conditions d'*ExempleMoinsFacile*.

Des amplitudes modales ont été déterminées pour le cas d'une corde pincée, montée sur ressorts aux deux extrémités (conditions (0,1)). Cependant, lors de la visualisation de u au cours du temps, on constate que les amplitudes obtenues ne sont pas correctes.

L'erreur provient probablement du choix de la fonction représentant le pincement en $S = e/$ à $t = 0$, c'est-à-dire des conditions initiales du problème.

Les amplitudes calculées sont disponibles dans le fichier *AmplitudeModale.m*.

2 - Affichage

Pour les différents affichages disponible dans notre programme on pourra retrouver :

- **ModePropre** : Aff = [x ~ ~ ~]
 - x=0 : Aucun affichage
 - x=1 : Visualisation des modes propres $n=1, n=2, n=3, n=Nw$
- **AmplitudeModale** : Aff = [~ x ~ ~ ~]
 - x=0 : Aucun affichage
 - x=1 : Visualisation des amplitude modales a_n et b_n
- **FctTemporelle** : Aff = [~ ~ x ~ ~]
 - x=0 : Aucun affichage
 - x=1 : Visualisation de $T(t)$ pour les modes propres $n=1, n=2, n=3, n=Nw$
- **FctDeplacement** : Aff = [~ ~ ~ x ~]
 - x=0 : Aucun affichage
 - x=1 : Visualisation de $u(s,t)$ en divers points de la corde et divers instants
 - x=2 : Visualisation de $u(s,t)$ au cours du temps
 - x=3 : Visualisation de $u(s,t)$ de $x=1$ et $x=2$ en même temps
- **DomaineModal** (Dans le cas ConditionsLimite = 1) : Aff=[~ ~ ~ ~ x]
 - x=0 : Aucun affichage
 - x=1 : Visualisations de la relation de dispersion

3 - Paramètres

Le paramètre *TypeCorde* déterminé dans le script *MAIN.m* permet de déterminer le **matériau** de notre corde : Acier (= 0) , fluorocarbone (= 1), aluminium (= 2), Nylon (= 3).

Nous avons fait le choix de regrouper tous les paramètres dans une même fonction *Parametre.m* au lieu de deux (*ParamInit.m* et *ParamInter.m*). Cette décision a en partie été prise car la fonction nous paraissait plus lourde à analyser lorsque divisée en deux.

4 - Fonctions définissant les différents domaines

Nous avons choisi de regrouper les fonctions définissant le domaine spatial et le domaine temporel au sein d'une seule et même fonction. Ce choix s'explique par le fait que ces deux fonctions sont relativement simples, et que leur regroupement permet de simplifier la structure du code tout en réduisant le nombre total de fonctions utilisées.

La fonction *DomaineModal.m* renvoie notamment la relation de dispersion, la longueur d'onde, la période et la fréquence de chaque mode.

Pour le cas ressort - ressort nous avons recours à la fonction *Transcendantale.m* qui calcule l'équation du même nom.

5 - Boucle if dans le programme principal

Dans le script *MAIN.m* on retrouve une première boucle if qui sert à bloquer le programme lorsque les conditions initiales "corde pincée" sont sélectionnées avec les conditions limites ressort-ressort (0,1).

Une seconde boucle if intervient lorsque nous sommes dans le cas ressort-ressort. Cette partie du programme gère la redimensionnalisation des domaines.

6 - Film et son

Lors de l'exécution du programme, un fichier vidéo *Corde.avi* est créé. Ce film permet de visualiser l'évolution au cours du temps du déplacement transverse de la corde le long de son domaine spatial. À chaque pas de temps, la configuration instantanée de la corde est tracée, puis enregistrée sous forme d'image, ce qui permet de reconstituer l'animation complète des vibrations de la corde au cours du temps.

Lors de l'exécution de la fonction *Son*, un signal sonore est construit à partir du déplacement de la corde en un point spatial donné. La fréquence d'échantillonnage est déterminée à partir du pas de temps de la simulation. Le signal obtenu, dont la durée initiale est d'environ 0,0046 s, est d'abord normalisé afin de limiter son amplitude à l'intervalle [-1, 1]. Afin d'obtenir un son audible de durée suffisante, ce signal est ensuite répété 1000 fois par concaténation, ce qui porte la durée totale à environ 4,6 s. Le signal final est alors enregistré dans le fichier audio *sound.wav*.