

GUIDE UTILISATEUR AVANCÉ

MAESSTRO

Benjamin Elie

23 décembre 2019

Table des matières

1	Introduction à MAESSTRO	3
2	Codes de calcul	3
2.1	Interface graphique	4
2.2	Python-MAESSTRO	4
2.3	Codes MATLAB	4
3	Installation	4
3.1	Installation de Python-MAESSTRO	4
3.2	Installation de Montjoie	5
3.3	Installation de MATLAB Runtime	5
3.4	Configuration de MAESSTRO	5
4	Utilisation de MAESSTRO en ligne de commande	6
4.1	Description des commandes	6
4.2	Exemples	6
5	Module du calcul des modes de table	6
5.1	Données normalisées d'entrée	7
5.2	Calcul des modes par la méthode de la plaque étendue (LVA)	8
5.3	Calcul des modes par éléments finis (Montjoie)	10
5.4	Données normalisées de sortie	11
6	Module du calcul de la dynamique de la corde	11
6.1	Données normalisées d'entrée	12
6.2	Modèles utilisés pour la simulation	12
6.2.1	Interaction marteau-corde	12
6.2.2	Non-linéarités de la corde	14
6.2.3	Mécanismes dissipatifs à l'intérieur de la corde	14
6.2.4	Homogénéisation de la corde	14
6.2.5	Réajustement de la position des points de couplage	15
6.2.6	Schémas numériques	16
6.2.7	Modèle d'étouffoir	16
6.3	Données normalisées de sortie	16
7	Module du calcul du mouvement de la table	17
7.1	Données normalisées d'entrée	17
7.2	Données normalisées de sortie	17

8	Module du calcul du rayonnement acoustique	17
8.1	Données normalisées d'entrée	17
8.2	Données normalisées de sortie	17
A	Gestionnaire de version Git	18

1 Introduction à MAESSTRO

MAESSTRO¹ est un projet de recherche financé par l'ANR (*Agence Nationale de la Recherche*), qui vise à renouveler les méthodes de conception des tables d'harmonie de piano. C'est un consortium impliquant plusieurs acteurs académiques et industriels, dont le LMS (*Laboratoire de Mécanique des Solides*, UMR 7649 de l'École Polytechnique), le LVA (*Laboratoire Vibrations Acoustique*, de l'INSA Lyon), le laboratoire NAVIER (UMR 8205), l'*Unité de Mécanique* de l'ENSTA Paris, et Stephen Paulello, facteur de piano de la société *SPCP*. Les objectifs et défis principaux du projet concernent l'approfondissement des connaissances sur la vibro-acoustique de la table de piano, le développement d'un logiciel de synthèse sonore pour la CAO de tables d'instruments de musique, et le développement de nouvelles architectures de tables de piano (structure sandwich, composites ou non) et des méthodes pour leur optimisation. Ce document détaille le fonctionnement du logiciel de synthèse sonore qui a été développé dans le cadre du projet.

Ainsi, le logiciel MAESSTRO² est un logiciel de synthèse sonore de sons de piano par modèle physique. L'objectif du logiciel est de fournir aux facteurs de piano des outils leur permettant de prédire et d'anticiper les impacts acoustiques de modifications structurelles de leur table d'harmonie sur le résultat sonore de l'instrument fini. L'intérêt de la modélisation physique fine à la base des calculs intégrés à MAESSTRO est de pouvoir rendre compte de réalités physiques : les variations acoustiques entre des notes de deux pianos différents reflètent de manière réaliste les différences géométriques et matérielles. Pour cela, MAESSTRO assemble et articule entre eux plusieurs modèles physiques et numériques permettant de simuler numériquement l'ensemble des phénomènes physiques à l'origine de la production des sons de pianos, de l'activation du marteau au rayonnement acoustique.

Ainsi, MAESSTRO présente plusieurs fonctionnalités permettant d'assister le facteur de piano durant les processus de fabrication, de modification, voire de restauration de pianos. Ces fonctionnalités sont :

1. une interface graphique (*cf.* paragraphe 2.1), permettant d'aider le facteur de piano à entrer les données géométriques et mécaniques de la table d'harmonie dans un format de données lisible par les codes de calcul MAESSTRO,
2. la création de fichiers audio des notes synthétisées à partir de fichiers MIDI,
3. un ensemble de modules de calcul scientifique qui simulent numériquement les phénomènes physiques responsables de la production des sons de piano,
4. des modules de post-traitement des sorties logicielles, afin d'analyser l'impact des modifications structurelles de la table d'harmonie sur son comportement vibroacoustique et sur son rendu sonore.

Ces fonctionnalités sont assurées par deux modules qui sont une librairie Python et une interface graphique pour la conception assistée par ordinateur (CAO sonore) de tables d'harmonie virtuelles. Ces modules ont été créés avec de l'aide de la société de développement informatique Logilab³.

La figure 1 présente un schéma de l'architecture logicielle de MAESSTRO. Elle consiste en une suite d'opérations qui fournit en sortie des sons de piano synthétisés à partir de données d'entrée fournies par l'utilisateur, incluant la géométrie et les matériaux de la table d'harmonie et des cordes virtuelles, ainsi que la suite de notes à synthétiser.

Afin de coordonner les modules de calculs entre eux, nous avons besoin de normaliser les données d'entrée et de sortie de ces modules. Nous appellerons ainsi données normalisées les données à fournir aux différents modules de sortie, et ce quelque soit le code utilisé.

2 Codes de calcul

Les codes de calcul sont entreposés sur divers entrepôts Git. Le lecteur peut se référer à l'annexe A pour plus de détails sur les fonctionnalités et les commandes nécessaires pour récupérer et gérer les codes source. Dans cette section, les commandes mentionnées sont à effectuer dans un environnement Linux (*par ex.* Ubuntu).

1. <https://www.maesstro.cnrs.fr/>

2. Pour la suite du document, MAESSTRO désigne le logiciel, et non le projet ANR.

3. www.logilab.fr

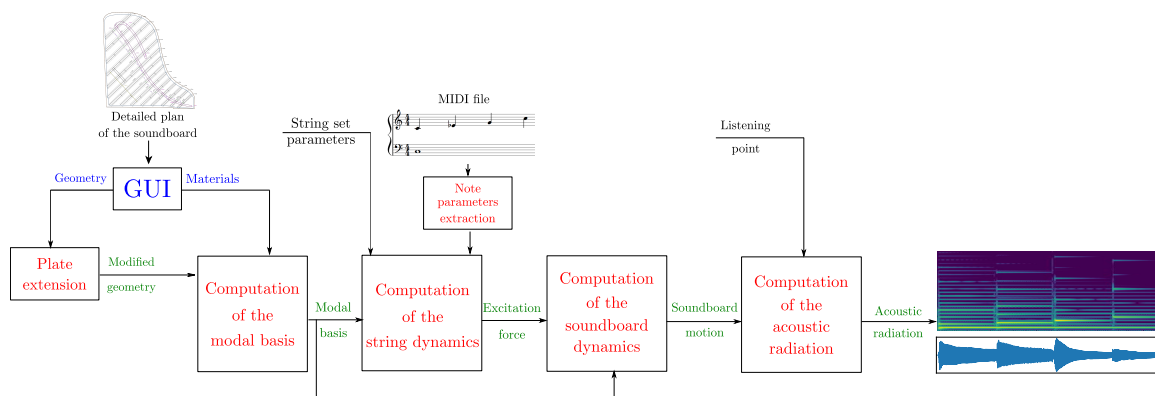


FIGURE 1 – Schéma bloc illustrant l’architecture logicielle de MAESSTRO. Les données d’entrée fournies par l’utilisateur sont représentées en noir. Les données fournies par l’utilisateur à l’aide d’outils graphiques sont en bleu. Les opérations effectuées par le logiciel sont en rouge, et les sorties du logiciel en vert.

2.1 Interface graphique

Les fichiers source de l’interface graphique sont à récupérer sur un entrepôt Git à l’adresse suivante <https://bitbucket.org/maestro/gui-maestro/src/master/>, à l’aide de la commande

```
git clone https://maestro@bitbucket.org/maestro/gui-maestro.git
```

2.2 Python-MAESSTRO

Python-MAESSTRO est un module Python permettant de lancer les différents modules de calcul de la physique du piano. Les fichiers source sont à récupérer sur un entrepôt Git à l’adresse suivante <https://bitbucket.org/maestro/maestro/src/master/>, à l’aide de la commande

```
git clone https://maestro@bitbucket.org/maestro/maestro.git
```

2.3 Codes MATLAB

Les codes sources MATLAB sont entreposés dans un entrepôt Git à l’adresse suivante <https://bitbucket.org/maestro/matlab-maestro/src/master/>. Ils peuvent être récupérés à l’aide la commande suivante :

```
git clone https://maestro@bitbucket.org/maestro/matlab-maestro.git
```

3 Installation

3.1 Installation de Python-MAESSTRO

Sur une machine UNIX, une solution simple pour installer python-MAESSTRO est d’utiliser le gestionnaire de paquets pip. S’il n’est pas installé sur la machine de l’utilisateur, celui-ci pourra le faire à l’aide la commande :

```
sudo apt install python3-pip
```

Ensuite, dans un terminal, il faut se placer dans le répertoire maestro, où se situe le fichier setup.py. Pour installer, il faudra lancer la commande :

```
python3 -m pip install -e .
```

Remarques :

- il faut bien vérifier que python3 est installé sur la machine,
- certains modules python peuvent être demandés. Il faudra alors les installer en tapant

```
pip install <nom_du_module>
```

3.2 Installation de Montjoie

Pour l'installation de Montjoie, qui est une librairie de calcul par éléments finis développés par Inria, l'utilisateur peut consulter la documentation en ligne, à l'adresse suivante : <http://montjoie.gforge.inria.fr/installation.php>.

3.3 Installation de MATLAB Runtime

Téléchargement et installation

Certains modules nécessitent l'exécution de codes MATLAB précompilés à l'aide du MATLAB Runtime Compiler, disponible gratuitement. La version utilisée est la 9.4 (R2018a), à télécharger à l'adresse suivante :

```
https://fr.mathworks.com/products/compiler/matlab-runtime.html.
```

Une fois téléchargé, extraire les fichiers et dossiers du .zip vers un dossier temporaire. Puis, à partir d'un terminal dans ce dossier temporaire, exécuter la commande suivante

```
./install -mode silent -agreeToLicense yes
```

Configuration de l'environnement d'exécution

Il est nécessaire de renseigner la variable d'environnement "LD_LIBRARY_PATH" pour que les bibliothèques MATLAB puissent être utilisées lors de l'exécution des codes. Pour cela, l'utilisateur peut ajouter les lignes suivantes au fichier "~/.bashrc" :

```
export MCR_PATH=/usr/local/MATLAB/MATLAB_Runtime/v94
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${MCR_PATH/runtime/glnxa64}:${MCR_PATH/bin/glnxa64}:${MCR_PATH/sys/os/glnxa64}
```

3.4 Configuration de MAESSTRO

Exécutables MATLAB

Pour compiler les codes MATLAB à l'aide du MATLAB Runtime Compiler, l'utilisateur peut directement exécuter le fichier "compile_module.m" contenu à la racine du dossier matlab-maesstro. Les exécutables sont stockés dans un dossier nommé matlab-builds.

Attention : si l'utilisateur ne modifie pas ce fichier, il créera e dossier matlab-builds à la racine du dossier personnel. Pour modifier l'emplacement de ces fichiers, il faut modifier la variable output_exe en entrant le chemin d'accès désiré.

Fichier de configuration

Afin de faire tourner correctement le logiciel MAESSTRO sur votre machine en local, il faut configurer les chemins d'accès aux différents codes source et compilés. Pour cela, il faut éditer le fichier config.ini, que l'on retrouve dans le dossier ~/.config/maestro. N'importe quel éditeur de fichier texte est suffisant. Si ce fichier n'est pas présent sur la machine de l'utilisateur, il pourra le créer à l'aide de la commande

```
maestro init-config
```

Plusieurs chemins d'accès sont demandés :

- **modes-table** : chemin d'accès vers l'exécutable MATLAB calculant les modes de table,
- **mouvement-table** : chemin d'accès vers l'exécutable MATLAB calculant le mouvement de la table,
- **rayonnement** : chemin d'accès vers l'exécutable MATLAB calculant le rayonnement acoustique,
- **runtime-path** : chemin d'accès vers le compilateur MATLAB runtime,
- **path** : chemin d'accès vers les codes sources Montjoie.

Généralement, le chemin d'accès des exécutables MATLAB est de la forme suivante :

```
$MATLAB_EXE/matlab-builds/<nom_du_module_linux>
```

avec \$MATLAB_EXE étant le chemin d'accès vers le dossier matlab-builds.

4 Utilisation de MAESSTRO en ligne de commande

4.1 Description des commandes

Le logiciel MAESSTRO peut se lancer en ligne de commande de syntaxe

```
maestro -option1 command -option2 <nom\_du\_module> <nom\_du\_fichier.ini>
```

Les différentes options et commandes sont :

- **option1** est vide ou vaut soit -version, soit -help, quelque soit la commande qui suit,
- **command** est le nom de la commande parmi les possibilités suivantes
 - init-config : crée un fichier texte qui définit les chemins d'accès aux codes de calcul, aux exécutables MATLAB, aux exécutables Montjoie, et au runtime MATLAB,
 - new : crée le fichier piano.ini qui contient les ponts d'accès aux fichiers de données d'entrée normalisées,
 - run : fait tourner le code spécifié par -option2 et <nom_du_module>,
- **option2** est vide ou vaut -help, ou alors :
 - -m pour lancer un module spécifique (défini par <nom_du_module>)
 - -f pour lancer une suite de modules à partir du module spécifié par <nom_du_module>, jusqu'au module rayonnement
- <**nom_du_module**> est le nom du module (ne marche qu'avec run -m ou run -f), à choisir parmi les possibilités suivantes
 - modes-table : lance le calcul de la base modale de la table,
 - marteau-corde : lance la règle StringImpedanceStiffNL de Montjoie pour simuler le déplacement de la corde correspondant aux notes voulues,
 - mouvement-table : lance le calcul des coordonnées modales de la table au cours du temps,
 - rayonnement : lance le calcul de la pression acoustique rayonnée,
- <**nom_du_fichier.ini**> est le nom du fichier de configuration .ini qui contient les points d'accès aux fichiers de données d'entrée normalisées.

4.2 Exemples

Aide générale

```
maestro --help
```

Aide de la commande run

```
maestro run --help
```

Lancer l'ensemble des modules

```
maestro run piano.ini
```

Lancer uniquement le module de calcul de la base modale

```
maestro run -m modes-table piano.ini
```

Lancer la suite de modules depuis le déplacement de la corde jusqu'au rayonnement acoustique

```
maestro run -f marteau-corde piano.ini
```

5 Module du calcul des modes de table

Cette section décrit le module mode-table qui permet le calcul de la base modale de la table d'harmonie à partir de sa géométrie issue de l'interface graphique et des propriétés mécaniques des matériaux. Ce module propose deux méthodes de calcul : une méthode dite semi-analytique, développée au LVA, qui utilise la méthode de la plaque étendue [1], et une méthode par éléments-finis, qui utilise la librairie de calculs Montjoie [2].

5.1 Données normalisées d'entrée

Le module mode-table nécessite trois fichiers d'entrée qui contiennent les données normalisées. Ces fichiers sont labélisés par les mot-clés *geometry*, *parameters*, et *material*.

Fichier *geometry*

Le fichier de géométrie est celui issu de l'interface graphique. Il est au format .json et contient les données normalisées relatives à la géométrie de la table d'harmonie, ainsi qu'aux matériaux qui la composent. Ces données sont celles du tableau 1.

Nom et description	Symbole	Unité	Type et taille
Panneau (<i>Panel</i>)			
Matériau du panneau de la table	<i>materialId</i>		chaîne de caractères
Angle d'orientation des fibres	θ_f (<i>orthotropicAngleDeg</i>)	degré	réel
Coordonnées des N_c points du contour de la table	$\{(x_{c_1}, y_{c_1}), \dots, (x_{c_{N_c}}, y_{c_{N_c}})\}$	mm	N_c couples de réels
Épaisseur de la table le long du contour	h_c (<i>thickness</i>)	mm	N_c réels
Chevalets (<i>Bridge</i>)			
Matériau du chevalet	<i>materialId</i>		chaîne de caractères
Coordonnées des N_b points de la ligne centrale du chevalet	$\{(x_{b_1}, y_{b_1}), \dots, (x_{b_{N_b}}, y_{b_{N_b}})\}$	mm	N_b couples de réels
Hauteur du chevalet au point de la ligne centrale	h_b (<i>height</i>)	mm	N_b réels
Épaisseur du chevalet au point de la ligne centrale	w_b (<i>width</i>)	mm	N_b réels
Point de référence	<i>isReference</i>		N_b booléens
Raidisseurs (<i>Ribs</i>)			
Matériau du raidisseur	<i>materialId</i>		chaîne de caractères
Coordonnées des points des extrémités du raidisseur	$\{(x_{start}, y_{start}), (x_{end}, y_{end})\}$	mm	2 couples de réels
Hauteur du raidisseur aux extrémités	h_r (<i>height</i>)	mm	2 réels
Épaisseur du raidisseur aux extrémités	w_r (<i>width</i>)	mm	2 réels
Position relative des N_r points intermédiaires du raidisseur	$\{p_1, p_2, \dots, p_{N_r}\}$ (<i>position</i>)		N_r réels
Hauteur du raidisseur aux points intermédiaires	h_r (<i>height</i>)	mm	N_r réels
Épaisseur du raidisseur aux points intermédiaires	w_r (<i>width</i>)	mm	N_r réels

TABLE 1 – Données normalisées contenues dans le fichier de géométrie

A noter que la plupart des données relatives à la géométrie sont données par des coordonnées dans le repère défini au début de l'utilisation de l'interface graphique, à l'exception de la position des points intermédiaires sur les raidisseurs. Ces positions sont définies en portion de segment entre la position de départ ($\{x_{start}, y_{start}\}$) et de fin ($\{x_{end}, y_{end}\}$), les N_r positions p_n sont donc comprises entre 0 et 1.

Fichier *parameters*

Le fichier de paramètres est au format .json et contient des valeurs de paramétrage des codes de calcul. Pour le calcul de la base modale, les paramètres concernés sont regroupés dans le tableau 2.

Nom et description [valeur par défaut]	Symbole	Unité	Type et taille
Fréquence maximale du calcul des modes [5000]	F_{max} (<i>max_freq</i>)	Hz	Réel
Méthode de résolution	<i>solver</i>		chaîne de caractères
Nombre de sinus nécessaires en x [82]	N_x (<i>n_xsin</i>)		Entier
Nombre de sinus nécessaires en y [63]	N_y (<i>n_ysin</i>)		Entier
Raideur des ressorts de blocage [2.5e6]	K_c (<i>spring_stiffness</i>)	N/m	Réel

TABLE 2 – Données normalisées contenues dans le fichier de paramétrage

La méthode de résolution est à choisir entre *LVA*, qui utilise la méthode de la plaque étendue (*cf.* section 5.2) ou *Montjoie*, qui utilise des éléments finis (*cf.* section 5.3). La chaîne de caractère à entrer pour le solveur est insensible à la casse des caractères. A noter également que la raideur des ressorts de blocage n'est utile que pour un solveur de type *LVA*.

Fichier *material*

Le dictionnaire de matériaux est au format .csv et contient les paramètres mécaniques d'un ensemble de matériaux prédéfinis. Ces paramètres sont décrits dans le tableau 3

Nom et description [valeur par défaut]	Symbole	Unité	Type et taille
Identifiant du matériau	<i>identifier</i>		chaîne de caractères
Masse volumique du matériau	ρ	kg/m ³	Réel
Module de Young selon x du matériau	E_x	GPa	Réel
Module de Young selon y du matériau	E_y	GPa	Réel
Module de cisaillement selon x et y du matériau	G_{xy}	GPa	Réel
Module de cisaillement selon x et z du matériau	G_{xz}	GPa	Réel
Module de cisaillement selon y et z du matériau	G_{yz}	GPa	Réel
Coefficient de Poisson selon x et y du matériau	ν_{xy}		Réel
Coefficient de Poisson selon y et x du matériau	ν_{yx}		Réel
Facteur de perte du matériau	η	%	Réel

TABLE 3 – Données normalisées contenues dans le dictionnaire des matériaux

5.2 Calcul des modes par la méthode de la plaque étendue (LVA)

Principe général

Cette méthode de calcul de la base modale s'appuie sur une représentation simplifiée de la géométrie de la table d'harmonie, permettant de calculer de manière semi-analytique les modes d'une plaque encastree orthotrope raidie de contour quelconque [1]. Le principe général est de considérer une plaque en appui simple ayant une orthotropie spéciale à circonscrite à la table d'harmonie. Cette plaque en appui simple est appelée *plaque étendue*. La surface de la plaque étendue non occupée par la table d'harmonie est artificiellement bloquée par une distribution de ressorts de blocage. La figure 2, tirée de [1] illustre le principe de la plaque étendue.

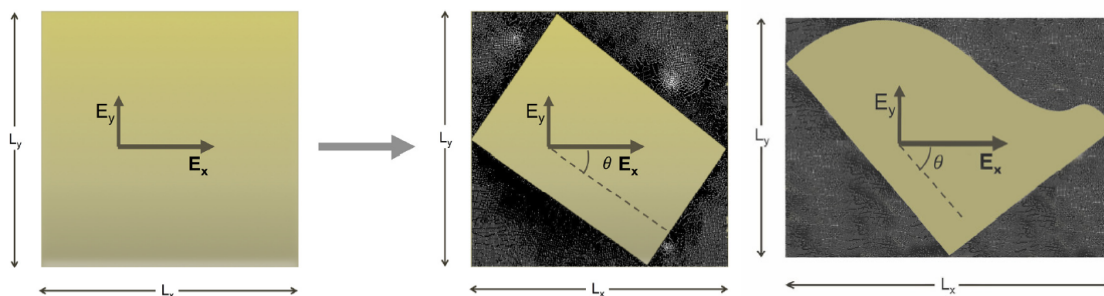


FIGURE 2 – Schéma de principe de la plaque étendue. À gauche, plaque étendue rectangulaire en appui simple, avec orthotropie spéciale. À droite, exemples de distribution de ressorts de blocage ponctuels qui permettent de dessiner des contours de table de géométrie quelconque. Figure extraite de [1].

Transformation des données normalisées pour faire exécuter ce code

Une étape importante du processus sera donc de modifier la géométrie de la table d'harmonie issue de l'interface graphique en une géométrie adaptée au calcul semi-analytique de la base modale. Cela s'effectue en plusieurs étapes :

1. rotation de la table d'harmonie afin d'orienter les fibres de manière parallèle à l'axe des x ,
2. recalage de l'orientation des raidisseurs pour qu'ils soient strictement perpendiculaires à l'axe des x ,
3. transformation de la géométrie des chevalets pour les rendre rectilignes et parallèles à l'axe des x ,
4. création de la densité de ressorts de blocage ponctuels.

La figure 3 illustre ces différentes étapes.

Rotation de la table d'harmonie :

Par convention, nous choisissons une rotation dans le sens trigonométrique direct, ainsi qu'un angle d'orientation des fibres, noté θ_f , positif. Dans ce cadre, l'ensemble des coordonnées des objets qui

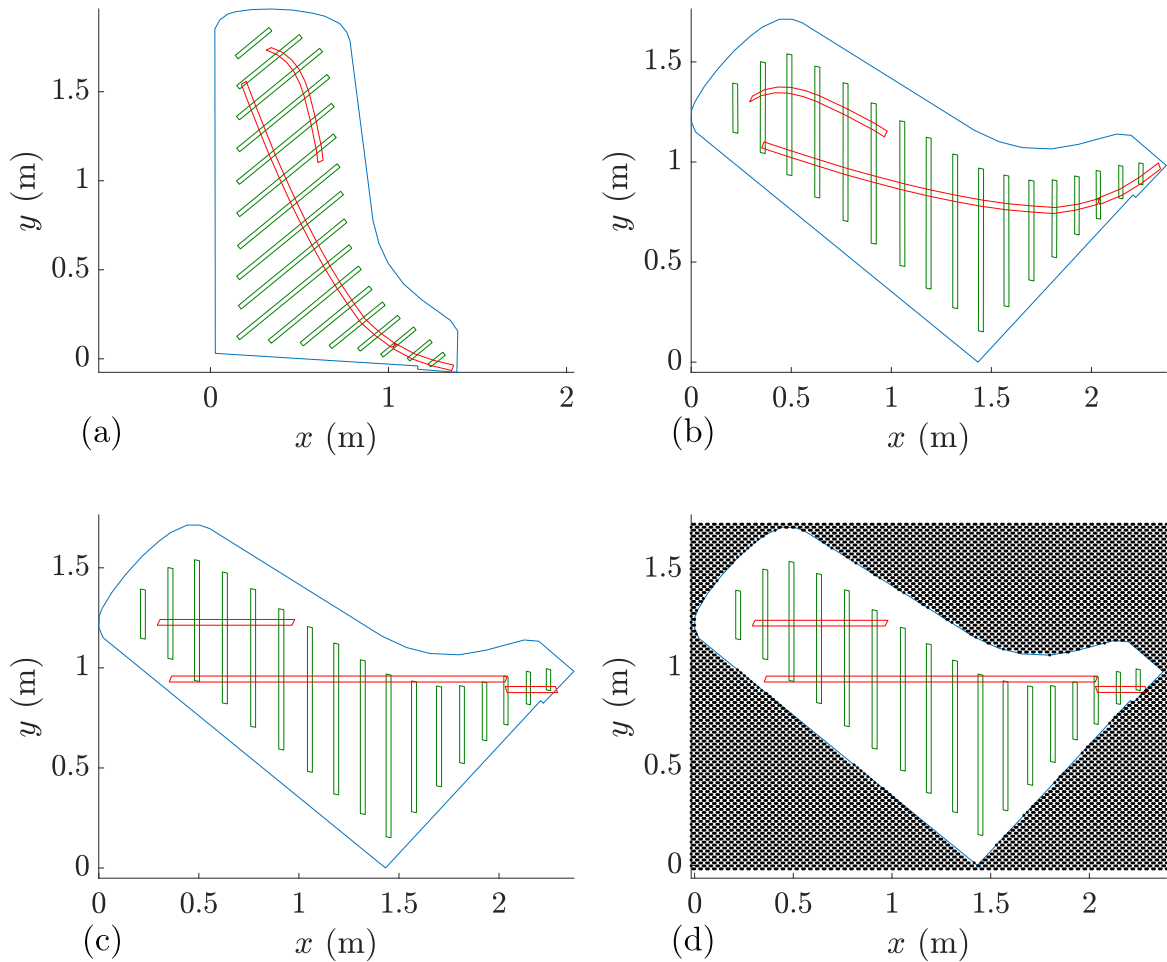


FIGURE 3 – Illustration des opérations permettant de passer d’une géométrie normalisée issue de l’interface graphique (a) à la géométrie de la plaque étendue avec distribution de ressorts de blocage ponctuels (d). La rotation et translation de la table d’harmonie est effectuée en (b), puis le recalage des raidisseurs et le redressement des chevalets sont effectués en (c). Enfin, le maillage des ressorts de blocage est effectué pour arriver à (d).

constituent la table d’harmonie subit une rotation d’un angle $\theta_r(\text{deg}) = 180 - \theta_f(\text{deg})$. Ensuite, l’origine du repère de la plaque étendue est décalé de manière à ce que l’abscisse minimale et l’ordonnée minimale de la table d’harmonie dans le repère de la plaque étendue soient nulles. Cela revient à prendre comme origine le coin inférieur gauche de la plaque étendue (*cf.* figure 2).

Recalage des raidisseurs :

Bien que les raidisseurs soient généralement orientés très proches de la perpendiculaire aux fibres, ce n’est jamais strictement le cas. Pour les recalier, nous gardons les ordonnées des points après rotation, tout en appliquant la contrainte sur les abscisses en les fixant à une valeur choisie comme la moyenne des abscisses des extrémités du raidisseur après rotation.

Redressement des chevalets :

Le calcul analytique de la base modale repose sur l’hypothèse que les superstructures sont constituées d’éléments rectilignes orientés, soit de manière parallèle à l’orientation des fibres, soit de manière perpendiculaire à l’orientation des fibres. Les chevalets étant généralement courbes et orientés indépendamment des fibres, il nous faut effectuer une opération de redressement des chevalets et d’ajustement de leur position. L’opération est alors analogue à celle du recalage des raidisseurs : les abscisses sont celles des extrémités du chevalet après rotation, et nous prendrons comme ordonnée la moyenne des ordonnées de la ligne centrale du chevalet après rotation.

Remarque : Dans la figure 3, le chevalet des aiguës est scindé en deux. En effet, il peut être utile de diviser le chevalet en plusieurs parties pour conserver sa forme courbée, ainsi que sa longueur,

lors du redressement. En revanche cela n'est pas fait automatiquement, et doit donc être réalisé lors du dessin de la table dans l'interface graphique.

Maillage de ressorts de blocage ponctuels :

Nous choisissons un maillage constitué de triangles équilatéraux régulièrement espacés dans la surface de la plaque étendue en dehors des contours de la table d'harmonie. La longueur l_c des côtés des triangles est choisie de sorte à ce qu'elle soit inférieure à un dixième de la plus petite longueur d'onde attendue. Cette longueur d'onde, note λ_{\min} est calculée à partir de la valeur de la fréquence maximale du calcul des modes F_{\max} , ainsi que des propriétés mécaniques du panneau de la table, soit

$$\lambda_{\min} = \sqrt{\frac{2\pi}{F_{\max}}} \left[\frac{D_3}{\rho_p h_p} \right]^{1/4}, \quad (1)$$

où ρ_p est la masse volumique du matériau utilisé pour la table d'harmonie, h_p est l'épaisseur de la table, et D_3 est son module de rigidité dans la direction perpendiculaire aux fibres (direction la moins raide). Nous obtenons alors

$$l_c = \frac{\lambda_{\min}}{10}. \quad (2)$$

5.3 Calcul des modes par éléments finis (Montjoie)

Principe général

Le logiciel MAESSTRO inclut la possibilité de calculer la base modale de la table d'harmonie à l'aide d'une méthode de type éléments finis. Pour cela, nous utilisons les codes du module Montjoie⁴.

Dans un premier temps, un maillage est généré automatiquement à partir du fichier de géométrie, et à l'aide du logiciel GMSH⁵. Pour réaliser ce maillage, nous considérons que :

- la table d'harmonie est d'épaisseur constante,
- les variations d'épaisseurs des superstructures sont discontinues, à savoir vues comme des "marches d'escalier",
- la longueur caractéristique des éléments du maillage est calculée de manière similaire à l'espacement maximal entre les ressorts de blocage de la méthode LVA (*cf.* Eqs. (1) et (2)). Étant donné que nous utiliserons un ordre de discrétisation supérieur à 1, la longueur caractéristique est augmentée de manière à être égale à $l_c = \frac{1}{4}\lambda_{\min}$.

Création des composantes physiques de la structure

Dans un premier temps, nous identifions l'ensemble des différentes surfaces qui composent la table d'harmonie et nous leur associons leur épaisseur et leurs propriétés mécaniques. Pour cela nous considérons que :

- l'épaisseur de la surface est la somme des épaisseurs des différentes superstructures qui se recouvrent sur cette surface, à savoir

$$h_{\text{tot}} = \sum_i h_i,$$

avec h_i l'épaisseur de la structure i ,

- la masse volumique est la somme des masses volumiques des différents matériaux pondérées par les épaisseurs, et divisée par l'épaisseur totale, à savoir

$$\rho_{\text{surf}} = \frac{1}{h_{\text{tot}}} \sum_i \rho_i h_i$$

- les superstructures sont supposées isotropes,
- si la surface contient un morceau de chevalet, la raideur du chevalet impose celle de la surface,
- sinon, si la surface contient un raidisseur (et pas de chevalet), la raideur du raidisseur impose celle de la surface.

4. *cf.* <https://www.math.u-bordeaux.fr/~durufle/montjoie/> pour le détail des codes de calcul

5. <http://gmsh.info/>

L'ensemble de ces informations est alors utilisé pour générer le fichier .geo qui servira à créer le maillage via GMSH et le fichier .ini qui sera créé pour lancer le calcul des modes de table via Montjoie.

Choix de l'ordre de discrétisation des éléments finis

Le choix de l'ordre de discrétisation est fait de manière à s'assurer que nous sommes au plus près de la solution exacte du problème tout en gardant un coût de calcul raisonnable. En effet, l'augmentation de l'ordre des éléments finis a pour conséquence de faire converger la solution vers la solution exacte, mais au prix d'un coût de calcul plus élevé. Nous utilisons donc une méthode de convergence pour choisir l'ordre. Cela consiste à calculer des fréquences de quelques modes propres (ici 5) à des fréquences élevés en incrémentant l'ordre à partir de 1. Dès que la différence en fréquences propres calculés avec l'ordre précédent devient suffisamment faible, nous considérons que nous sommes suffisamment proche de la solution exacte. Nous prenons comme métrique de différence la moyenne des écarts relatifs en fréquences. L'ordre de discrétisation choisi, nommé p , est donc la plus petite valeur \tilde{p} qui vérifie la relation suivante :

$$p = \operatorname{argmin}_{\tilde{p} \in \mathbb{N}} \left(\frac{1}{5} \sum_{k=1}^5 \left| \frac{f_k^{(\tilde{p}+1)} - f_k^{(\tilde{p})}}{f_k^{(\tilde{p})}} \right| < \epsilon \right), \quad (3)$$

avec $f_k^{(\tilde{p})}$ les fréquences modales calculées avec un ordre \tilde{p} et ϵ un seuil défini arbitrairement. Nous choisissons $\epsilon = 10^{-6}$.

5.4 Données normalisées de sortie

Les données normalisées de sortie sont stockées dans le fichier "SORTIES_MODES_TABLE.h5", au format hdf5. Celui-ci contient les ensembles de données (*datasets*) détaillés dans le tableau 4.

Nom et description [valeur par défaut]	Symbole	Unité	Type et taille
Dimensions de la base modale selon x et y	$\{N_x, N_y\}$ (<i>/basis_dim</i>)		2 entiers
Dimensions de la plaque considérée selon x et y	$\{L_x, L_y\}$ (<i>/soundboard_dimension</i>)		2 réels
Masses modales des N_m modes calculés au-dessous de F_{max}	m_n (<i>/masses_modales</i>)	kg	N_m réels
Raideurs modales des N_m modes calculés au-dessous de F_{max}	k_n (<i>/raideurs_modales</i>)	N/m	N_m réels
Amortissements modaux des N_m modes calculés avant F_{max}	c_n (<i>/amortissements_modaux</i>)		N_m réels
Coefficient des déformées dans la base des sinus des N_m modes calculés avant F_{max}	A_n (<i>/coefficients_deformees</i>)		Tableau de $(N_x \times N_y) \times N_m$ réels

TABLE 4 – Données normalisées de sortie du calcul de la base modale

Si le solveur utilisé est *Montjoie*, les valeurs L_x et L_y sont les dimensions du rectangle circonscrit à la table d'harmonie. Si le solveur est *LVA*, ces dimensions correspondent à celles de la plaque étendue. Dans ce cas, les coefficients des déformées sont alors également exprimés dans la base de la plaque étendue.

Toujours selon le choix du solveur plusieurs autres fichiers de sortie sont générés :

Le solveur est *LVA* :

Le fichier JSON, nommé "piano_simple.json", est un fichier de géométrie (lisible par l'interface graphique) correspondant à celui de la table simplifiée selon la méthode décrite dans la section 5.2.

Le solveur est *Montjoie* :

Des fichiers de géométrie ("geofile.geo") lisible par GMSH, ainsi que le maillage correspondant généré par GMSH ("geofile.msh") sont créés pour faire tourner le code d'éléments finis. Ils peuvent être lus par GMSH pour que l'utilisateur puisse vérifier la qualité du maillage généré. Le fichier de configuration utilisé lors de l'appel au code Montjoie est également créé et stocké. C'est le fichier "montjoie.ini" qui peut être appelé indépendamment de MAESSTRO avec la règle soundboard de Montjoie.

6 Module du calcul de la dynamique de la corde

Cette section décrit le module marteau-corde qui permet le calcul du déplacement d'une corde lorsque celle-ci est frappée par un marteau.

6.1 Données normalisées d'entrée

Le module marteau-corde nécessite quatre fichiers d'entrée qui contiennent les données normalisées. Ces fichiers sont identifiés par les mot-clés suivants : *geometry*, *parameters*, *jeu*, et *plan-cordes*.

En plus de ces fichiers, il est nécessaire de fournir en entrée la sortie du module "modes-table", à savoir le fichier "SORTIES_MODES_TABLE.h5".

Fichier *geometry*

Fichier identique à celui demandé pour le module modes-table. Celui-ci est requis pour une étape d'éventuelle recalage des points de couplage corde-chevalet sur la ligne centrale des chevalets, afin de s'assurer que le couplage s'effectue bien sur un chevalet, et non directement sur le panneau.

Fichier *parameters*

Fichier identique à celui demandé pour le module modes-table. Pour le calcul du mouvement de la corde, les paramètres concernés sont regroupés dans le tableau 5

Nom et description [valeur par défaut]	Symbole	Unité	Type et taille
Pas temporel de simulation [5e-6]	dt (<i>time_step</i>)	s	Réel
Nombre de points de discrétisation spatiale sur la corde [50]	N_{pts} (<i>nsteps</i>)		Entier

TABLE 5 – Données normalisées contenues dans le fichier de paramétrage pour le module marteau-corde

Fichier *jeu*

Ce fichier contient les informations sur les notes à synthétiser. Il peut être soit un fichier au format JSON qui contient les données normalisées détaillées dans le tableau 6, soit directement un fichier MIDI.

Nom et description	Symbole	Unité	Type et taille
Durée du morceau	T (<i>duration</i>)	s	Réel
Indices des N notes à synthétiser	i (<i>index</i>)		N entiers
Instants de début des N notes à synthétiser	t_0 (<i>start_time</i>)	s	N réels
Instants de relâchement des touches des N notes à synthétiser	t_{stop} (<i>stop_time</i>)	s	N réels
Vitesse du marteau lors de l'impact pour les N notes à synthétiser	v_0 (<i>initial_velocity</i>)	m/s	N réels

TABLE 6 – Données normalisées contenues dans le fichier de jeu du module marteau-corde

Fichier *plan-cordes*

Le plan des cordes est constitué d'un classeur Excel, composé de plusieurs feuillets, et qui contient l'ensemble des informations relatives aux cordes du piano virtuel. C'est ce fichier qui permettra de faire tourner les codes de simulation du mouvement des cordes pour la synthèse sonore de morceaux issus de fichiers MIDI. L'utilisateur pourra consulter le mode d'emploi utilisateur pour plus d'informations. Les données d'entrées normalisées sont décrites dans le tableau 7.

La position de l'impact est donné comme la fraction de la longueur totale de la corde depuis l'agrafe.

6.2 Modèles utilisés pour la simulation

6.2.1 Interaction marteau-corde

Nous reprenons le modèle à mémoire [3, 4] pour exprimer la force du marteau résultante en fonction de l'écrasement du feutre et de ses propriétés viscoélastiques. Celle-ci s'écrit

$$F^{\mathcal{H}} = K^{\mathcal{H}} e^p + R^{\mathcal{H}} \frac{d(e^p)}{dt}, \quad (4)$$

Nom et description	Symbole	Unité	Type et taille
Indices des N_s cordes du piano	i		N_s entiers, de 1 à N_s
Fréquence fondamentale des N_s notes	F_c	Hz	N_s réels
Nombre de cordes par choeur	N_c		N_s entiers
Abscisse des N_s points d'attache des cordes au chevalet	x_c	m	N_s réels
Ordonnées des N_s points d'attache des cordes au chevalet	y_c	m	N_s réels
Angle que forme les cordes par rapport au plan de la table	α	degré	N_s réels
Longueur des cordes	L_{str}	m	N_s réels
Matériau utilisé pour l'âme de la corde	MatAme		chaîne de caractères
Diamètre de l'âme	D_0	mm	N_s réels
Matériau utilisé pour le premier filage de la corde	MatFilInt		chaîne de caractères
Diamètre du premier filage	D_1	mm	N_s réels
Matériau utilisé pour le filage extérieur	MatFilExt		chaîne de caractères
Diamètre du second filage	D_2	mm	N_s réels
Position du point d'impact du marteau sur les cordes	λ_i		N_s réels
Masse volumique du matériau	ρ	kg/m ³	Réel
Module de Young du matériau	E	Pa	Réel
Module de cisaillement du matériau	G	Pa	Réel

TABLE 7 – Données normalisées contenues dans le fichier de plan de corde

où le symbole ${}^{\mathcal{H}}$ fait référence à un paramètre du marteau, K et R sont respectivement la dureté et l'amortissement du feutre, e est l'écrasement du feutre, et p est un paramètre tenant compte de la non-linéarité de l'écrasement du feutre.

Le modèle d'interaction avec la corde utilise la loi fondamentale de la dynamique en supposant que le marteau n'est soumis qu'à la force de contact avec la corde. On a donc

$$M^{\mathcal{H}} = \frac{d^2\xi(t)}{dt^2} = -F^{\mathcal{H}}(t), \quad (5)$$

où $M^{\mathcal{H}}$ est la masse du marteau et $\xi(t)$ la position de la pointe de la partie boisée du marteau sur son axe vertical.

Afin de prendre en compte une zone de contact non ponctuelle, le modèle utilisé introduit une fonction de répartition $\delta^{\mathcal{H}}$ qui vaut

$$\delta^{\mathcal{H}}(X) = \frac{\left[1 + w_x^{-\beta(X+w_x\beta/2)}\right]^{-1} - \left[1 + w_x^{-\beta(X-w_x\beta/2)}\right]^{-1}}{w_x\beta}, \quad (6)$$

où $X = x_x^{\mathcal{H}}$ avec $x_x^{\mathcal{H}}$ le point central de contact sur la corde, w_x est l'épaisseur du marteau dans l'axe de la corde, et β est un paramètre régularisant de $\delta^{\mathcal{H}}$.

Ces paramètres étant très difficiles d'accès pour le facteur de piano, nous choisissons des valeurs par défaut. Celles-ci sont issues du rapport technique de Chabassier et Duruflé [5]. Les auteurs proposent de déterminer les propriétés des marteaux à partir des relations suivantes :

$$\begin{cases} M^{\mathcal{H}} = -6.2348 \times 10^{-5}i + 0.0112 \\ p = 2.4295 \times 10^{-4}i^2 - 0.007703i + 2.337 \\ K^{\mathcal{H}} = 10^{(5.3097 \times 10^{-2}i + 7.6425)} \\ R^{\mathcal{H}} = K^{\mathcal{H}} \times 10^{(-0.04366i - 2.294)} \end{cases}, \quad (7)$$

où $i \in [1, 88]$ est l'indice de la corde sur le clavier. A noter que dans la version du rapport [5] disponible en ligne, la formule de $R^{\mathcal{H}}$ est incorrecte, la formule correcte étant celle de ce document.

Les autres paramètres sont considérés comme constants, avec pour valeurs :

$$\begin{cases} w_x = 2 \text{ cm} \\ w_y = 2 \text{ cm} \\ \beta = 2001 \end{cases}. \quad (8)$$

Ce sont des valeurs *ad hoc*. Elles peuvent être modifiés dans la fonction "write_montjoie_infile" du fichier "_montjoie.py".

6.2.2 Non-linéarités de la corde

Les calculs du déplacement de la corde prennent en compte certaines non-linéarités de la cordes, incluant la raideur, et les déformations géométriques locales. La raideur est prise en compte à l'aide d'un modèle de Timoshenko, alors que les déformations géométriques locales sont prises en compte à l'aide du modèle dit *géométriquement exact* [6]. L'effet de raideur introduit un décalage fréquentiel des fréquences propres de la corde par rapport à une corde idéale (fréquences harmoniques). Les fréquences propres ne sont donc pas harmoniques et sont appelés *partiels*. Ce phénomène spectral est appelé inharmonicité. Les fréquences des partiels sont données par la formule suivante [7] :

$$f_n = n f_0 \sqrt{1 + B n^2}, \quad (9)$$

où n est l'ordre du partiel, f_0 est la fréquence fondamentale de la corde, et B est appelé coefficient d'inharmonicité. Il vaut :

$$B = \frac{\pi^3 E d^4}{64 T_0 L^2}, \quad (10)$$

où E est le module de Young du matériau qui compose la corde, et d , T_0 , et L sont respectivement le diamètre, la tension, et la longueur de la corde.

6.2.3 Mécanismes dissipatifs à l'intérieur de la corde

De manière similaire aux marteaux, nous choisissons les valeurs pour les coefficients d'amortissement intrinsèque de la corde proposées par le rapport technique de Chabassier et Duruffé [5], à savoir :

$$\begin{cases} R_{u,i} = R_{\phi,i} = \alpha i + \beta \\ R_{v,i} = 0.5 \\ \eta_{u,i} = \eta_{\phi,i} = \gamma i + \delta \\ \eta_{v,i} = 10^{-9} \end{cases}, \quad (11)$$

où $\alpha = 5 \times 10^{-3} \text{ s}^{-1}$, $\beta = -0.015 \text{ s}^{-1}$, $\gamma = 2.78 \times 10^{-11} \text{ s}^{-1}$, and $\delta = 1.5274 \times 10^{-9} \text{ s}^{-1}$. R et η sont respectivement le coefficient d'amortissement fluide, exprimé en s^{-1} , et visqueux, exprimé en s^{-1} , et les indices \cdot_u , \cdot_v et \cdot_ϕ représentent respectivement la vibration transversale, la vibration longitudinale, et le cisaillement.

L'ensemble de ces valeurs peuvent être modifiés dans la fonction "write_montjoie_inifile" du fichier "_montjoie.py".

6.2.4 Homogénéisation de la corde

La méthode de calcul du mouvement de la corde est basé sur un modèle de corde homogène et uniforme : celles-ci sont entièrement décrites par leur longueur L , leur diamètre d , leur masse volumique ρ , et leur tension T_0 . Si l'on considère des cordes filées, qui sont par définition non homogènes, il faut alors passer par une étape d'homogénéisation. Celle-ci consiste à chercher une corde virtuel, de paramètres L_{eq} , d_{eq} , ρ_{eq} , et T_{eq} , dont le comportement est similaire à celui de la corde filée.

Nous suivons la méthode proposée par Chabassier [2]. Dans un premier temps, nous considérons la longueur équivalente comme identique à la longueur réelle de la corde ($L_{\text{eq}} = L$). Ensuite, nous calculons la masse linéique équivalente μ_{eq} . Celle-ci est calculée selon la formule suivante :

$$\mu_{\text{eq}} = \pi \rho_{\text{ame}} \left(\frac{d_{\text{ame}}}{2} \right)^2 + \frac{\pi^2 \rho_{\text{fil}_1}}{4} \left(\frac{d_{\text{fil}_1}}{2} \right)^2 + \frac{\pi^2 \rho_{\text{fil}_2}}{4} \left(\frac{d_{\text{fil}_2}}{2} \right)^2. \quad (12)$$

A partir de là, nous en déduisons la tension équivalente T_{eq} :

$$T_{\text{eq}} = \mu_{\text{eq}} (2 L_{\text{eq}} f_0)^2, \quad (13)$$

avec f_0 la fréquence fondamentale. Enfin, à partir de la formule (10) donnant le coefficient d'inharmonicité B , on en déduit le diamètre équivalent d_{eq} :

$$d_{\text{eq}}^4 = \frac{64 B T_{\text{eq}} L_{\text{eq}}^2}{\pi^3 E_{\text{ame}}}, \quad (14)$$

avec B le coefficient d'inharmonicité théorique, et E_{ame} le module de Young du matériau de l'âme de la corde.

A noter que le coefficient d'inharmonicité est difficile d'accès en pratique. Pour l'instant, la solution que nous avons trouvé est de définir des valeurs par défaut, issus du rapport technique de Chabassier et Duruflé [5]. Les valeurs sont stockés en dur dans le fichier `inharmoCoeff.py` pour un clavier de 88 notes, allant des graves vers les aigus.

6.2.5 Réajustement de la position des points de couplage

Pour diverses raisons (relevés imprécis, dessin imprécis dans l'interface graphique, orientation des repères mal définis...), il est possible que les positions des différents point de couplage de la corde avec le chevalet entrées par l'utilisateur ne coïncident pas exactement avec la ligne centrale du chevalet tel que dessiné dans l'interface graphique. Afin de s'assurer que chaque point de couplage soit sur la ligne centrale d'un chevalet, nous procédons à une opération de réajustement, qui consiste en deux étapes :

1. associer chaque point de couplage à un chevalet, en cherchant celui qui est le plus proche,
2. projection du point de couplage sur le point de la ligne centrale le plus proche.

Il est important de préciser que nous parlons ici des données normalisées. En effet, dans le cas où le *solver* choisi est *LVA*, il faut effectuer d'autres opérations. En effet, étant donné que les chevalets sont fortement modifiés lors du passage au repère de la plaque étendue, il se peut que cela modifie considérablement les points de couplage. Les étapes supplémentaires sont alors :

1. rotation et translation des points de couplage pour adaptation au repère de la plaque étendue,
2. calcul des abscisses curvilignes normalisés des points de couplage et des lignes centrales des chevalets,
3. repositionnement des points de couplage sur les lignes centrales des chevalets simplifiés de la plaque étendue en fonction de l'abscisse curviligne normalisée.

Nous appelons abscisse curviligne l'abscisse s_c définie par la relation de récurrence

$$s_{c+1} = s_c + \sqrt{(x_{c+1} - x_c)^2 + (y_{c+1} - y_c)^2},$$

où $s_0 = 0$. L'abscisse curviligne normalisée est alors s_c/l_b , avec l_b la longueur du chevalet. Elle est donc toujours comprise entre 0 et 1, où 0 et 1 correspondent aux deux extrémités du chevalet. Cela permet de s'assurer que le passage dans le repère de la plaque étendue conserve à la fois la distribution des points de couplage le long du chevalet, et également le fait qu'ils soient tous positionnés sur le chevalet.

La figure 4 illustre ces étapes de réajustement de la position des points de couplage.

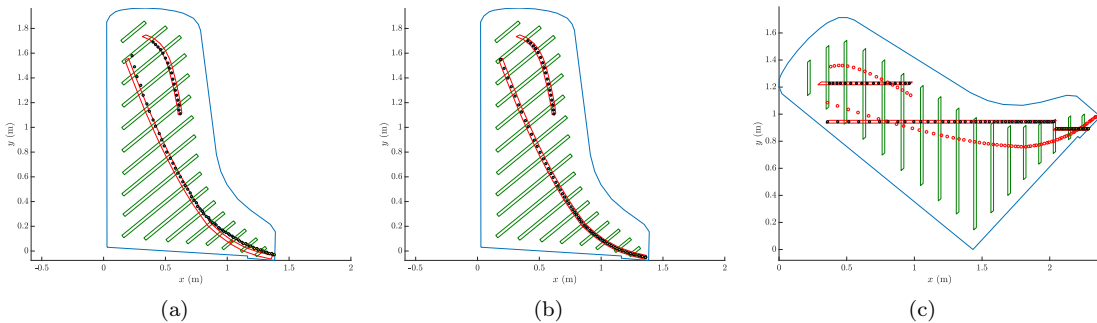


FIGURE 4 – Illustration des étapes de réajustement de la position des points de couplage. a) le relevé des points de couplage n'est pas assez précis : ils ne sont pas tous situés sur le chevalet. b) On les repositionne sur la ligne centrale par projection orthogonale. c) Repositionnement sur le repère de la plaque étendue par conservation de la distribution des points en abscisse curviligne le long des chevalets. Sur les figures, les points de couplage appartenant à un même chevalet partagent la même couleur. Dans c), les symboles 'o' représente la position des points de couplage dans le repère de la plaque étendue, avant l'étape de réajustement par distribution de l'abscisse curviligne.

6.2.6 Schémas numériques

6.2.7 Modèle d'étouffoir

Afin de gérer l'extinction des notes lors du relâchement des touches, nous ajoutons un modèle d'étouffoir de note. C'est un modèle de type *signal* qui consiste à prolonger les composantes sinusoïdales contenues dans le signal de forces en modifiant leur amortissement par une valeur pré-établie. Le coefficient d'amortissement de la partie étouffée est calculé en fonction de la fréquence fondamentale et la vitesse d'impact du marteau.

En effet, des mesures sur un piano réel montre qu'il existe une relation linéaire entre l'amortissement globale de la note étouffée est le logarithme de la fréquence fondamentale, à savoir :

$$\alpha_{\text{etouffee}} = a_0 \log(f_0) + a_1, \quad (15)$$

où a_0 (sans unité) et a_1 (en s^{-1}) sont des coefficients déterminés expérimentalement pour 5 nuances, pianissimo, piano, mezzoforte, forte, et fortissimo.

Ensuite, à partir des correspondances entre nuances et vitesse MIDI, puis en reprenant de la formule empirique de Goebel [8] suivante, reliant la vitesse MIDI V_{MIDI} à la vitesse d'impact du marteau V_{marteau} :

$$V_{\text{MIDI}} = -\frac{\frac{128}{V_{\text{marteau}}} - 604.5}{4.5}, \quad (16)$$

on en déduit la correspondance entre nuance et vitesse de marteau. Il est alors possible de déduire l'amortissement de la note étouffée à appliquer à partir de la connaissance de sa fréquence fondamentale f_0 et de la vitesse d'impact du marteau. Les valeurs sont détaillées dans le tableau 8.

Nuance	V_{MIDI}	V_{marteau} (m/s)	a_0	a_1 (s^{-1})
<i>pp</i>	35	0.29	10.9	-8.81
<i>p</i>	46	0.32	5.97	0.957
<i>mf</i>	69	0.44	5.50	0.499
<i>f</i>	83	0.55	3.91	3.71
<i>ff</i>	104	0.94	4.96	2.33

TABLE 8 – Coefficients du modèle d'étouffoir en fonction de la nuance

Ce modèle est amené à être amélioré, afin de prendre en compte de manière plus précise l'influence de la vitesse d'impact du marteau (idéalement de manière continue). La valeur des coefficients a_0 et a_1 , ainsi que les intervalles de valeurs de vitesse d'impact de marteau pour lesquels ils sont appliqués peuvent être modifiés dans la fonction `mufflingtone` du fichier "`_pianotone.py`".

6.3 Données normalisées de sortie

Les données normalisées de sortie sont stockées dans le fichier "`SORTIES_CORDE.h5`", au format hdf5. Ce fichier contient les ensembles de données (*datasets*) détaillés dans le tableau 9.

Nom et description [valeur par défaut]	Symbole	Unité	Type et taille
Vecteur temps et force exercée par la corde pour chaque note i	$t, f(t)$	s, N	Tableau de dimension $2 \times M$
Durée du morceau	T	s	Réel
Abscisses des N points de couplage dans le repère de la table	x_c	m	N réels
Ordonnées des N points de couplage dans le repère de la table	y_c	m	N réels
Instant de départ des N notes	t_0	s	N réels
Indices des N notes	i		N entiers

TABLE 9 – Données normalisées de sortie du calcul du déplacement de la corde

Dans le cas où l'utilisateur a fourni un fichier MIDI en entrée, un fichier nommé `tmp.json` est alors créé. Il contient la transcription du fichier MIDI vers les données normalisées dans le format du tableau 6.

7 Module du calcul du mouvement de la table

7.1 Données normalisées d'entrée

Ce module ne nécessite pas de fichier d'entrée fourni par l'utilisateur. En revanche, il nécessite les données normalisées de sortie des modules modes-table et marteau-corde, à savoir les fichiers "SORTIES_MODES_TABLE.h5" et "SORTIES_CORDE.h5".

7.2 Données normalisées de sortie

Les données normalisées de sortie sont stockées dans le fichier "SORTIES_MOUVEMENT_TABLE.h5", au format hdf5. Ce fichier contient les ensembles de données (*datasets*) détaillés dans le tableau 10.

Nom et description [valeur par défaut]	Symbole	Unité	Type et taille
Vecteur temps	t	s	M réels de 0 à T
Coordonnées temporelles des modes	$b_n(t)$	m	Tableau de $N_{mode} \times M$ réels

TABLE 10 – Données normalisées de sortie du calcul du mouvement de la table d'harmonie

8 Module du calcul du rayonnement acoustique

8.1 Données normalisées d'entrée

Le module rayonnement nécessite le fichier de paramètres (*parameters*), ainsi que les fichiers de sorties des modules modes-table et mouvement-table, à savoir "SORTIES_MODES_TABLE.h5" et "SORTIES_MOUVEMENT_TABLEs.h5".

Fichier *parameters*

Fichier identique à celui demandé pour les modules modes-table et marteau-corde. Pour le calcul du rayonnement, les paramètres concernés sont regroupés dans le tableau 11.

Nom et description [valeur par défaut]	Symbole	Unité	Type et taille
Vitesse du son [340]	c_s (<i>sound_velocity</i>)	m/s	Réel
Masse volumique de l'air [1.2]	ρ_a (<i>air_density</i>)	kg/m ³	Réel
Abscisse du point d'écoute dans le repère normalisé de la table [0]	x_{ac} (<i>x_ac</i>)	m	Réel
Ordonnée du point d'écoute dans le repère normalisé de la table [0]	y_{ac} (<i>y_ac</i>)	m	Réel
Élévation du point d'écoute dans le repère normalisé de la table [1.5]	z_{ac} (<i>z_ac</i>)	m	Réel

TABLE 11 – Données normalisées contenues dans le fichier de paramétrage pour le module rayonnement

8.2 Données normalisées de sortie

Les données normalisées de sortie sont stockées dans le fichier "SORTIES_RAYONNEMENT.h5", au format hdf5. Ce fichier contient les ensembles de données (*datasets*) détaillés dans le tableau 12.

Nom et description [valeur par défaut]	Symbole	Unité	Type et taille
Vecteur temps	t	s	M réels de 0 à T
Pression acoustique rayonnée au point d'écoute	$p_a(t)$	Pa	M réels

TABLE 12 – Données normalisées de sortie du calcul du rayonnement acoustique de la table d'harmonie

Références

- [1] B. Trévisan, K. Ege, and B. Laulagnet, "A modal approach to piano soundboard vibroacoustic behavior," *Journal of the Acoustical Society of America*, vol. 141, pp. 690–709, Feb. 2017.

- [2] J. Chabassier, A. Chaigne, and P. Joly, "Modeling and simulation of a grand piano," *The Journal of the Acoustical Society of America*, vol. 134, no. 1, pp. 648–665, 2013.
- [3] A. Stulov, "Hysteretic model of the grand piano hammer felt," *The Journal of the Acoustical Society of America*, vol. 97, no. 4, pp. 2577–2585, 1995.
- [4] A. Stulov, "Experimental and theoretical studies of piano hammer," in *Proceedings of the Stockholm Music Acoustics Conference*, vol. 485, 2003.
- [5] J. Chabassier and M. Duruflé, "Physical parameters for piano modeling," tech. rep., 04 2012.
- [6] P. M. Morse and K. U. Ingard, *Theoretical acoustics*. New York, McGraw-Hill, 1968.
- [7] H. Fletcher, "Normal vibration frequencies of a stiff piano string," *The Journal of the Acoustical Society of America*, vol. 36, no. 1, pp. 203–209, 1964.
- [8] W. Goebel, "Skilled piano performance : Melody lead caused by dynamic differentiation," in *Proc. of the 6th Int. Conf. on Music Perception and Cognition*, Citeseer, 2000.

A Gestionnaire de version Git

Installation

```
sudo apt install git
```

Configuration proxy

Il se peut que vous ayez des erreurs lors de tentatives de récupération des fichiers dans un entrepôt distant, de type : "unable to access" ou "Couldn't resolve host". Dans ce cas, il faut configurer le proxy.

La première étape consiste à connaître l'adresse du proxy en tapant

```
echo "$http_proxy"
```

Puis, il faut taper la commande

```
git config --global http.proxy <sortie_de_http_proxy>
```

Récupérer des dépôts

Pour récupérer un dépôt pour la première fois, il faut ouvrir un terminal à l'emplacement où l'utilisateur souhaite déposer le dossier contenant les codes source, puis taper :

```
git clone <adresse_du_depot>
```

L'adresse du dépôt dépend des codes sources à récupérer. L'utilisateur peut se référer au paragraphe 2 pour les connaître.

Récupérer la dernière version mise à jour

Pour récupérer la dernière version mise à jour, il faut se placer dans le dossier du dépôt et taper :

```
git pull origin master
```

Il est conseillé d'effectuer cette opération avant de travailler sur les codes sources, afin d'éviter des branches de travail parallèles.

Mettre à jour des modifications

Pour envoyer les modifications apportés par l'utilisateur, celui-ci devra effectuer les commandes suivantes :

```
git add .
git commit -m "message expliquant les modifications"
git push origin master
```

Attention : il est fortement conseillé de vérifier que les modifications apportées sont opérationnelles. De plus, en cas de changement de formats de fichiers d'entrée, il est important de clairement le mentionner, et de fournir des fichiers d'exemples opérationnels.