RF

Manuel de Poutrix 0.0.1

Introduction

Poutrix a pour ambition de devenir un logiciel de résolution numérique de problèmes physiques unidimensionnels aux dérivées partielles. Je me suis concentré ici à donner au programme un squelette comprenant l'import des résultats et leur export, en me restreignant aux équations de poutres dans les hypothèses d'Euler-Bernoulli.

Poutrix 0.0.1 est la version alpha de la version 0.1. Elle permet de résoudre des problèmes statiques sur poutres droites et en matériaux isotropes, avec des chargements uniquement aux noeuds des éléments.

Table des matières

1 Utilisation du code

1.1 Prérequis

Poutrix 0.0.1 nécessite Python 3.3.3, et ne fonctionne que sur Windows.

Les librairies python utilisées sont les suivantes : numpy, scipy, tkinter, sp, math et os.

On peut obtenir Python 3 et toutes ces librairies pour Windows en téléchargeant par exemple WinPython, qui comprend de plus Spyder, une interface python.

1.2 Fonctionnement

Dans la présente version du programme, la seule manière d'entrer des données pour le calcul est d'écrire un fichier assemblage au format texte qui suit la syntaxe imposée. Un fichier d'exemple est proposé. On peut sélectionner quel fichier sera traduit par Poutrix au moyen d'une interface graphique, en cliquant sur le nom du fichier à ouvrir. Tous les fichiers présents dans le dossier où se trouve le code Python sont proposés à l'interprétation, y compris ceux qui ne sont pas lisibles. L'interface graphique se lance si on lance "lancer_pb.py". Cette interface subira des modifications dans les versions suivantes.

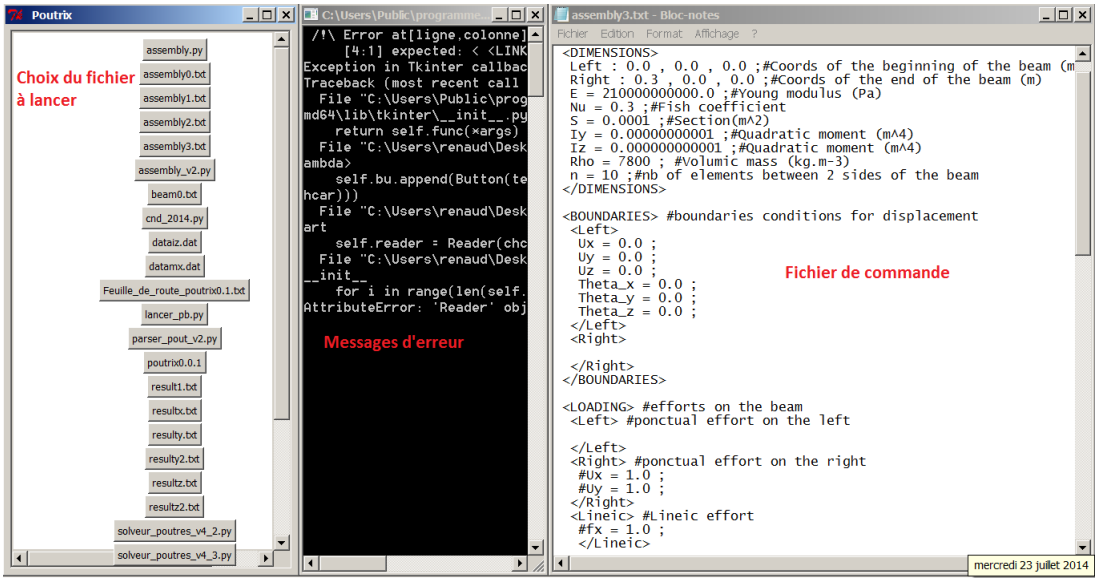

FIGURE 1 – Interface de Poutrix

On peut imposer un maillage pour une poutre donnée au travers d'un fichier .dat annexe, dans le cas contraire, le mailleur automatique intégré à poutrix divisera chaque partie de poutre entre deux liaisons (ou bords) en un nombre réglable d'éléments.

Poutrix sort un fichier texte contenant les valeurs du déplacement aux différents noeuds dans l'ordre des déclarations des poutres, et dans le repère global.

1.3 Performances

Poutrix utilise la bibliothèque d'algèbre linéaire de numpy, qui est très largement compilée. Cependant, les opérations d'assemblage des matrices sont interprétées. Il en résulte que les performances ne sont pas comparables avec un solveur codé en Fortran ou en C++. Un nombre de noeurs de l'ordre de quelques centaines (càd quelques milliers de ddl) représente déjà un problème conséquent pour Poutrix (mais ceci dépend évidemment des performances machine).

Compte tenu de la précision numérique des flottants Python, il existe une taille minimale de maille telle que les résultats soient fiables. Cette taille est de l'ordre de 10^{-6}. Des noeuds trop rapprochés seront considérés comme confondus par poutrix. Cette limitation sera mieux gérée dans les versions suivantes.

1.4 Fichier d'entrée

Le fichier d'entrée se divise en trois parties : tout d'abord les déclarations des poutres, avec leurs paramètres géométriques et matériaux, puis les déclarations des liaisons, et enfin des informations sur le solveur.

1.4.1 Considérations générales sur la syntaxe

Sauf indication contraire, les espaces et les sauts de ligne ne sont pas interprétés par Poutrix, ce qui signifie que l'on peut s'en servir pour organiser son fichier de commande.

L'une des lourdeurs de la syntaxe de Poutrix est que lors de l'affectation d'une valeur, il est indispensable de placer un espace avant et après le signe =. Ceci devrait être corrigé dans les prochaines versions.

Les commentaires doivent être précédés du symbole #, et peuvent être placés n'importe où sur une ligne. Aucun caractère placé sur une ligne après un # ne sera interprété.

Il est possible, lors de l'affectation de certaines valeurs sur l'ensemble de la poutre, d'écrire en argument non pas une valeur, mais le nom d'un fichier de données qui contient une liste de valeurs. Ce faisant, l'utilisateur impose aussi un maillage pour la poutre. Il est question de ceci dans la section "Fichier annexe de données".

L'ordre d'entrée des données est important et ne peut pas être perturbé sous peine d'erreur fatale pour le programme.

Pour entrer un réel, par exemple une grandeur matériau ou une dimension, il faut écrire un nombre à virgule au format : "0.0". Ni "0.", ni "0", ni encore ".0" ne sont tolérés. De plus, les réels négatifs sont interdits. Il est prévu de rendre ceci plus souple.

1.4.2 Les poutres

La déclaration d'une poutre commence par une balise de début de poutre, et finit par la balise de fin de poutre. Elle se divise elle-même en trois parties : la déclaration des caractéristiques (géométriques et matériaux), des conditions limites aux deux bords de la poutre (ddl imposé ou pas, nul ou non), et enfin les chargements mécaniques. Les chargements doivent être donnés dans le repère global.

En ce qui concerne le chargement linéique sur la poutre, la donnée entrée par l'utilisateur correspond à la valeur de l'effort qui sera placé sur chaque noeud du maillage. Ceci signifie que si le maillage n'est pas régulier, on n'aura pas un effort homogène. Ceci sera corrigé dans les versions ultérieures.

Toujours à propos du chargement linéique : si on pose un chargement linéique de f, compte-tenu de la façon dont le chargement est géré, il faut imposer aux deux extrémités de la poutre un chargement de $-\frac{f}{2}$.

1.4.3 Les liaisons

La déclaration d'une liaison comporte quatre affectations : le nom de la première poutre, l'abscisse curviligne du point de liaison sur la première poutre, le nom de la seconde poutre, et l'abscisse curviligne du point de liaison sur la deuxième poutre.

Il est à noter que dans la présente version, seules les liaisons complète sont prises en charge. Toute liaison déclarée liera donc tous les ddl des points liés entre eux. De plus, l'ordre de déclaration des poutres n'a aucune incidence sur le traitement de la liaison. Pour définir une liaison d'une poutre avec le bâti, il faut renseigner la partie "boundaries" de la poutre, et non pas définir une nouvelle liaison dans la partie "Links".

Remarque : il est possible dans Poutrix de lier deux points qui ne sont pas coincidents, leurs déplacements seront alors égaux, mais ceci ne les rendra pas coincidents pour autant.

1.4.4 Le solveur

Dans la partie solveur, deux affectations doivent être effectuées. En premier, la nature du problème. Dans Poutrix 0.0.1, seuls les problèmes statiques sont pris en charge. On donne aussi le nom du fichier de sortie (sans l'extension : le programme rajoutera .txt).

1.4.5 Liste des affectations facultatives

Certaines données sont facultatives et sont considérées comme nulles si elles ne sont pas données : c'est le cas des chargements ponctuels et linéiques. Pour ce qui est des ddl imposés, le fait de ne pas les exprimer conduit Poutrix à considérer que l'on a affaire à un bord libre.

1.5 Fichier donnée annexe

Un fichier annexe de données donne une liste de valeurs nodales en remplacement d'une seule valeur. Pour affecter un fichier annexe, l'utilisateur devra écrire son titre. Les fichiers de données doivent porter l'extension .dat. Par exemple, si on veut appeler "toto.dat", pour un module d'Young variable, il faut taper : "E = toto".

Les valeurs données dans un fichier annexe le sont aux noeuds du maillage considéré, et ce de la façon suivante : le fichier doit alterner un réel, donnant la valeur au noeud, et un entier donnant la longueur relative entre ce noeud et le suivant.

Ce format n'est pas destiné à rester tel quel et subira probablement des modifications dans les versions ultérieures de Poutrix.

2 Erreurs possibles

2.1 Retranscription des erreurs

Lors du lancement de "lancer_pb.py", en plus de la fenêtre de choix, une fenêtre apparaît dans laquelle sont listées les erreurs python d'exécution. Si l'erreur est une erreur de syntaxe poutrix, elle apparaît dans les premières lignes sous la forme suivante : "/!\ Error at [ligne,colonne]". C'est cette erreur qui donne ce que le parser attendait dans le fichier de commande, et qu'il n'a pas trouvé.

2.2 Liste des erreurs signalées

*Syntaxe : Nécessité d'écrire des espaces autour d'un signe =.

*Syntaxe : Nécessité d'écrire les réels au format "0.0" .

*Syntaxe : Les nombres négatifs sont interdits.

*Affichage : L'interface est un bug à elle toute seule.

*Modélisation : Lors de l'attribution d'une force linéique, celle-ci n'est pas constante si la taille de maille n'est pas constante sur la poutre.

*Solveur : Il se passe des choses suspectes dans le cas d'un assemblage avec couplage flexion-traction dans les parties faiblement connexes de l'assemblage. Sans-doute est-ce dû à un mauvais conditionnement à cause de la pénalisation.

*Et sûrement bien d'autres encore, qui n'attendent que d'être signalées...

F I N