TD-TP Modélisation numérique : éléments finis

Exercice 1 : les fonctions d'interpolation.
But : comprendre comment les fonctions d'interpolations linéaires par morceaux interpolent linéairement une fonction quelconque
1.1 Montrer que la superposition des fonctions de formes linéaires définies en cours et appliquées à la fonction $y=f(x)$ entre x_j et x_{j+1} , donne bien l'équation d'une droite passant par (x_j, y_j) et (x_{j+1}, y_{j+1}) .
1.2 Créer une fonction d'interpolation linéaire ($y_{interp}=lin\_interp(x, etc)$), voir exo1_trou.m) utilisant les fonctions d'interpolations phi_0 et phi_1 vues en cours, et permettant de calculer *y= f(x)*, pour y variant linéairement entre (x_0, y_0) et (x_1, y_1) .
1.3 En utilisant cette fonction d'interpolation et 5 éléments de même longueur, interpoler la fonction suivante : $f(x)=x\sin(x)$ entre 0 et 5. Prendre des pas de 0.1.
1.4 Calculer l'erreur $\sum_{i=1}^{N}|y^i_{interp}-y^i|$, où N est le nombre de points de calculs c'est à dire la différence entre la fonction f(x) et la fonction l'interpolant par morceaux. Commencer par 5 éléments, puis passer à 10. Comment évolue l'erreur ?

Exercice 2 : Fonctions de formes et éléments finis
Soit une barre de section S, encastrée en x = 0 est soumise à son propre poids. On supposera que L = 5 m, ρ = 1000 kg/m3, E = 10e6 Pa, g = 9.8 kg/m3, S = 0.3 m² et on considérera les 2 cas de conditions limites suivantes :

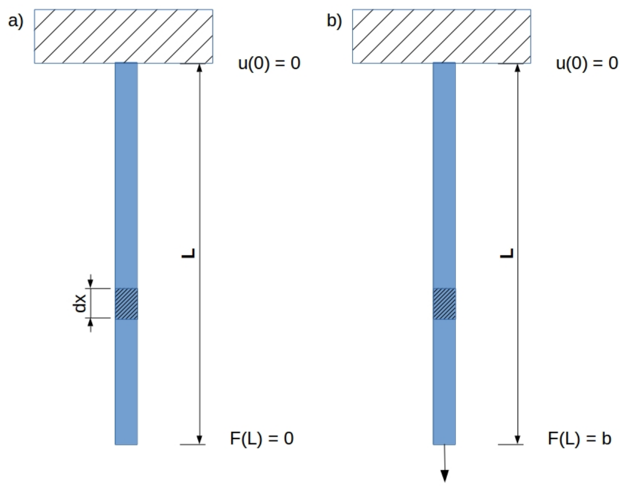

Illustration 1: Poutre encastrée a) extrémité libre en x = L. b) Force en x=L

1. Dériver l'équation aux dérivées partielles pour les déplacement de la poutre. La poutre est soumise à son poids. On écrira l'équilibre d'une petite tranche de poutre d'épaisseur dx , de section S , soumise à une force selon x, comme $F(x)=S\sigma_{xx}=SE\frac{du_x}{dx}$, où σ_{xx} est la contrainte selon x, u_x est le déplacement selon x, E est le module d'Young. On considérera 2 cas a) la poutre est libre en x = L F(b) = 0; b) Une force s'exerce en x = L, F(b) = 15000 N.

2. Dériver la solution analytique $u_x(x)$ pour ces 2 cas (Figure 1), la programmer et la dessiner.

3. Écrire la forme faible

4. Afin de discrétiser le problème, utiliser des fonctions de forme linéaires par morceaux. On supposera dans un premier temps qu'on a 3 éléments.

5. Ecrire l'approximation éléments finies de Galerkine. Écrire les matrices de raideur locales. On notera l = L/3.

6. Ecrire la matrice de raideur globale. Réduire le système sachant que u(0) = 0.

7. Programmer la solution analytique pour le cas F(l) = 0 et F(L) = b.

8. Programmer une fonction assemble_loc permettant le calcul de la matrice de raideur élémentaire et de la matrice de masse élémentaire .

9. Programmer le code éléments finis permettant de comparer le calcul analytique et numérique des déplacements de la barre à l'aide d'un script (exo2_trou.m) utilisant la fonction assemble_loc et une boucle « for » pour sommer la contribution de chaque élément aux matrices de raideur et de masse globales.

Commandes matlab à utiliser :
clear all
close all
clc
Définition des vecteurs *v = [1 2 3]* ou *v = [1 ; 2 ; 3]* ou *v = 1:1:3 ;*
Adressage de l'élément de la ligne i et colonne j de la matrice *A(i,j)*
fonctions
Boucle : *for*
Résolution de systèmes linéaire Ax=B, pour trouver x on fait *x = A\B*
Impression figure 2D : *plot*

10. Passer à 10 éléments. On notera l = L/10.
Que remarque t'on ?