Méthodes Numériques II
Polytech 4ème année Mécanique

TP Eléments finis

I. EXERCICES PRÉLIMINAIRES:

A. Création de matrices particulières

1. Ecrire en Python la fonction qui crée, en fonction de n, la matrice pentadiagonale A telle que:

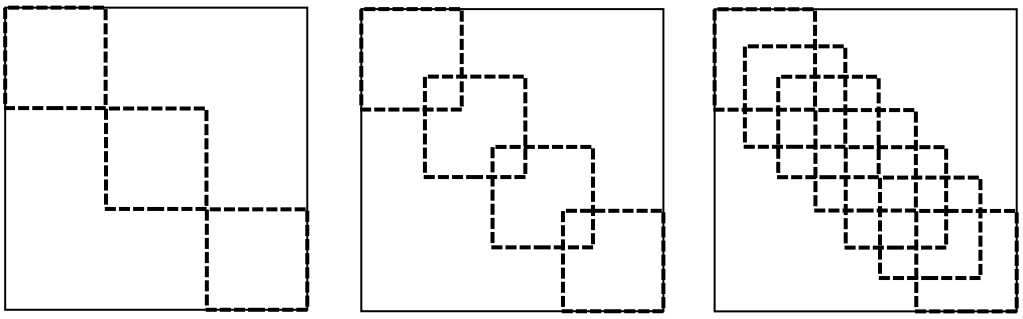
$$A_{ij}=\begin{pmatrix} 4, & si & i=j \\ -1, & si & i=j+1 \\ -1, & si & i=j-1 \\ 2, & si & i=j-n \\ 2, & si & i=j+n \end{pmatrix}$$

avec $i=1..n^2,\ j=1..n^2$.

On prendra n = 3. (On utilisera la fonction "diag" de la bibliothèque numpy.)

2. Créer une matrice 9x9 diagonale par blocs avec des sous-blocs 3x3 A=[2,1,1;1,2,1;1,1,2], telle que:

a- les blocs soient décalés de 3 lignes et de 3 colonnes à chaque itération,
b- les blocs soient décalés de 2 lignes et de 2 colonnes à chaque itération,
c- les blocs soient décalés d'une ligne et d'une colonne à chaque itération.



(a) (b) (c)

B. Tracé de graphiques

Tracer un nuage de points obtenus aléatoirement (fonction "random" de la bibliothèque random), en échelle linéaire puis en échelle logarithmique. Bien spécifier les légendes, titres, etc ...

C. Programmation de la méthode inversion de système linéaire par factorisation LU

Soit un système linéaire $AX=B$ avec A une matrice ($n \times n$), B un vecteur colonne de dimension n et X le vecteur solution recherché. La méthode de factorisation LU permet de réécrire le système linéaire sous la forme $LUX = B$. On a donc $A=LU$ avec L une matrice triangulaire inférieure de diagonale unitaire et U une matrice triangulaire supérieure. La résolution du système $AX=B$ revient alors à résoudre deux systèmes triangulaires :

$$AX=B, \quad \Rightarrow LUX=B \Rightarrow \begin{pmatrix} LY=B \\ UX=Y \end{pmatrix},$$

ce qui est moins coûteux en temps de calcul, en particulier lorsque l'on doit résoudre plusieurs systèmes linéaires avec différents seconds membres, mais la même matrice A.

Traduire l'algorithme de factorisation suivant en une fonction python dont l'entrée est la matrice A et les sorties les matrices L et U :

Initialisation:

$U_{ij} = 0$ pour $i = 1,n$ et $j = 1,n$ avec n=len(A)

$L_{ij} = \delta_{ij}$ pour $i = 1,n$ et $j = 1,n$ avec n=len(A)

Factorisation:

Pour j=1,n

Pour i=1,j

$$U_{ij} = A_{ij} - \sum_{k=1}^{i-1} L_{ik} U_{kj}$$

Fin

Pour i=j+1,n

$$L_{ij} = \frac{1}{U_{jj}} \left(A_{ij} - \sum_{k=1}^{j-1} L_{ik} U_{kj} \right)$$

Fin

Fin

Pour la résolution, on utilise donc successivement deux algorithmes : un algorithme dit de « descente » pour lequel on calcule le vecteur Y tel que $LY=B$ et un algorithme dit de « remontée » pour lequel on calcule le vecteur X tel que $UX=Y$.

Ecrire les deux fonctions Python correspondant à ces algorithmes.

Résolution:

1. Descente $LY=B$

Pour i=1,n

$$Y_i = B_i - \sum_{k=1}^{i-1} L_{ik} Y_k$$

Fin

2. Remontée $UX=Y$

Pour i=n,1

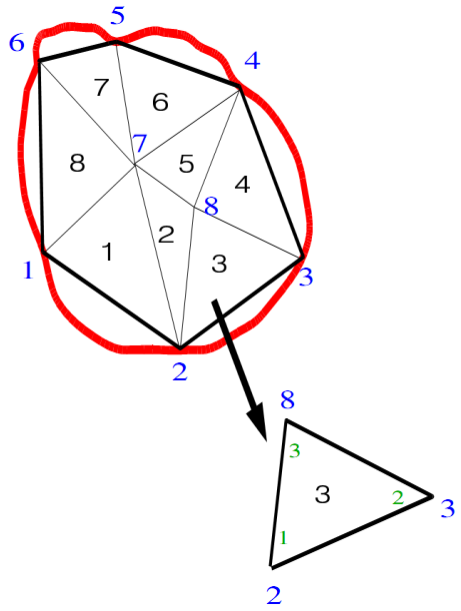
$$X_i = \frac{1}{U_{ii}} \left(Y_i - \sum_{k=i+1}^{n} U_{ik} X_k \right)$$

Fin

D. Principe de la structure des données pour les éléments finis:

1. La table de connexion:

Dans un maillage éléments finis, les éléments sont numérotés de 1 à N_e et les nœuds de 1 à N_n. Chaque nœud est défini par ses coordonnées globales répertoriées dans la table de coordonnées globales des nœuds géométriques. Chaque élément est défini par la liste de ses nœuds géométriques relevés dans le sens trigonométrique. Cette liste est stockée dans la table de connexion.



N° Noeud	*x*	*y*
1	0	0
2	2	-1.5
3	4	0
4	3	3
5	1	3.5
6	0	3
7	1.5	1.5
8	2.5	1

Table de coordonnées globales des noeuds

N° Triangle	s.l* n°1	s.l n°2	s.l n°3
1	1	2	7
2	2	8	7
3	2	3	8
4	3	4	8
5	4	7	8
6	4	5	7
7	5	6	7
8	1	7	6

* : sommet local

Table de connexion

En utilisant la fonction ***tables*** fournie, créez une table de connexion ainsi qu'une table de coordonnées avec n1 = 2 et n2 = 3. Observez leurs constructions.

2. Tracé de maillages

Faire une fonction Python qui trace un triangle à partir des coordonnées des trois sommets.

Faire une fonction qui trace un maillage avec la fonction précédente et les tables de connexion et de coordonnées précédemment calculées**.**

3. Principe général d'assemblage : algorithme d'assemblage avec la table de connexion.

Pour l'assemblage des matrices de résolution du problème éléments finis, on utilisera l'algorithme suivant:

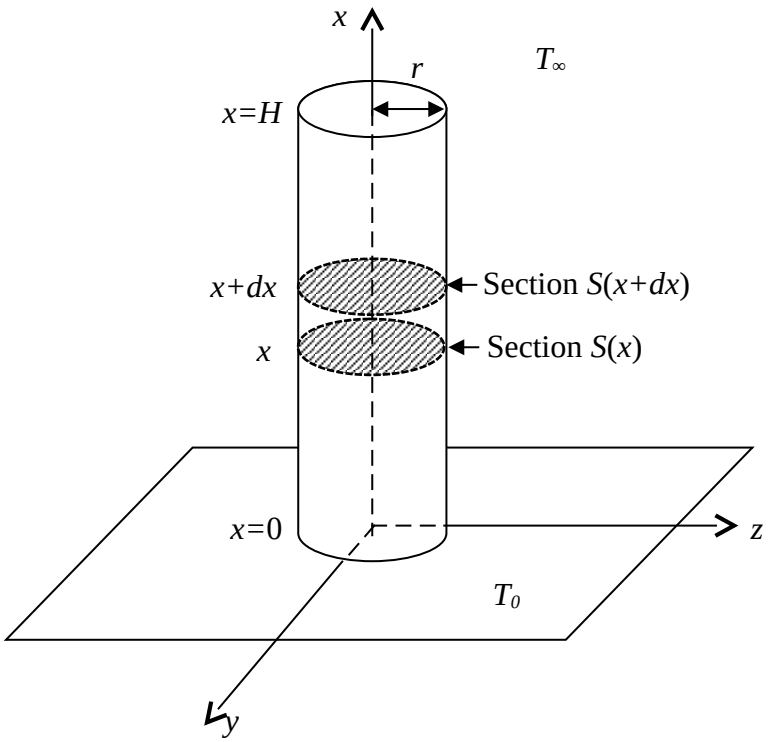
```
Pour k=1, Ne
    Pour i=1,len(Ke)
        Pour j=1,len(Ke)
            Ig = table_connexion(k,i)
            Jg = table_connexion(k,j)
            Kglobal (Ig,Jg) = Kglobal (Ig,Jg) + Ke(i,j)

        Fin Pour
    Fin Pour
Fin Pour
```

Traduire cet algorithme en une fonction Python « assemblage » dont l'entête sera:
`def assemblage(Ke,Ne,table_connexion)`
avec `Ke` la matrice élémentaire à assembler, `Ne` le nombre d'éléments du domaine et `table_connexion` la table de connexion.

II. Problème de convection / conduction en régime stationnaire

On cherche à obtenir la répartition de la température dans un barreau métallique (type ailette de refroidissement d'un processeur). Le barreau a une longueur H = 5 mm et le rayon de la section est r = 0.5 mm.
La géométrie du problème est présentée sur la figure suivante :



La température sur l'extrémité du barreau en x = 0 est imposée et égale à T_0 = 100°C. A l'autre extrémité, on a placé un matériau isolant (paroi adiabatique, flux thermique nul). L'ailette est refroidie par convection avec le milieu ambiant (h = 100 W.m^{-2}.°C^{-1}, T_∞ = 20°C).

Donnée : Conductivité thermique λ = 40 W.m^{-1}.°C^{-1}.

En écrivant que la somme des flux thermiques sur un petit élément du barreau d'épaisseur dx doit être nulle, déterminer l'équation d'équilibre du système et ses conditions aux limites.

ELÉMENTS FINIS UNIDIMENSIONNELS LINÉAIRES

On considère dans un premier temps que le problème est 1D.

1. Ecrire le résidu et la formulation faible de ce problème.

2. Déterminer l'expression des fonctions de base pour des éléments finis linéaires aussi appelés P1. En déduire la formulation faible discrète.

3. On fixe le nombre d'éléments Ne = 5. La longueur de l'intervalle de discrétisation est $dx=x_{i+1}$ - x_i. Calculer les matrices de raideur et de masse élémentaires.

4. Assembler la matrice globale à partir des matrices élémentaires.

5. Appliquer les conditions aux limites sur la matrice assemblée et le second membre.

6. Ecrire le programme Python pour résoudre ce problème. Tracer le profil de température.

7. La solution exacte est obtenue au moyen du calcul formel :

$$T_{ex}(x)=T_\infty+(T_0-T_\infty)\frac{\cosh(m(H-x))}{\cosh(mH)},\ avec\ m=\sqrt{\frac{hP}{\lambda S}}$$

et P le périmètre d'une section du barreau.

Etudier l'ordre de précision de la méthode des éléments finis P1 en traçant la courbe de l'erreur relative ε en fonction de dx (longueur d'un élément) dans un graphe en échelle logarithmique.

$$\varepsilon=\frac{\|U_{num}(x)-U_{ex}(x)\|}{\|U_{ex}(x)\|}=\frac{\sqrt{\left(\sum_{i=1}^{N_e+1}U_{num}(x_i)-U_{ex}(x_i)\right)^2}}{\sqrt{\left(\sum_{i=1}^{N_e+1}U_{ex}(x_i)\right)^2}}$$

On prendra
(voir la fonction `norm de la bibliothèque numpy.linalg`).

On retire désormais l'isolant thermique placé sur l'extrémité en $x = H$. La condition limite n'est donc plus adiabatique. Il y a échange avec le milieu extérieur par convection avec un coefficient d'échange $h' \neq h$. On a h' = 80 W.m^{-2}.°C^{-1}.

8. Exprimer la nouvelle condition aux limites et les modifications induites sur le système d'équations.

9. Ecrire le programme Python pour résoudre ce problème. Tracer le profil de température.

10. La solution exacte est:

$$T_{ex}(x)=T_\infty+(T_0-T_\infty)\frac{\cosh(m(H-x))+\frac{h'}{\lambda m}\sinh(m(H-x))}{\cosh(mH)+\frac{h'}{\lambda m}\sinh(m(H))},\quad avec\ m=\sqrt{\frac{hP}{\lambda S}}$$

.

Tracer la courbe de l'erreur relative ε en fonction de dx dans un graphe en échelle logarithmique et en déduire l'ordre de précision.

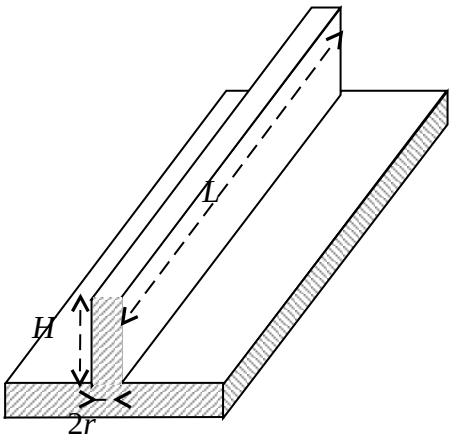
ELÉMENTS FINIS UNIDIMENSIONNELS QUADRATIQUES

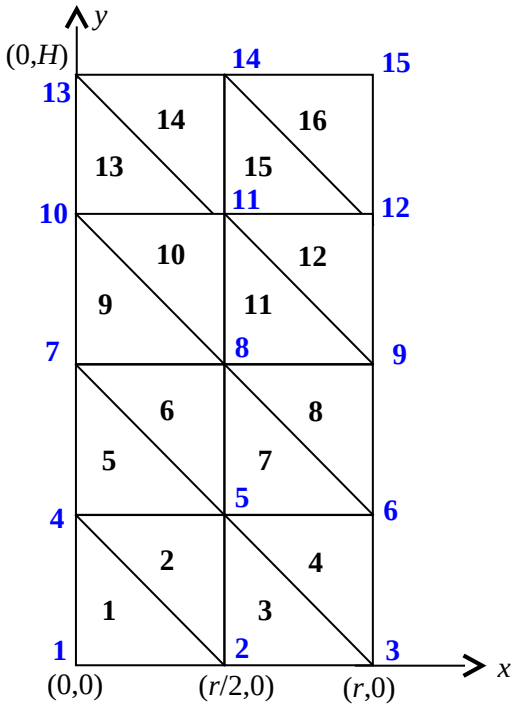
Reprendre le travail précédent en utilisant des éléments finis quadratiques. On pourra utiliser MAPLE pour calculer plus rapidement les composantes des matrices élémentaires. Quel est l'ordre de précision de la méthode des éléments finis P2?

ELÉMENTS FINIS BIDIMENSIONNELS LINÉAIRES:

B. Maillage triangulaire, éléments finis P1

On étudie maintenant le profil de température 2D dans une coupe d'une ailette de grande longueur L par rapport à sa hauteur H et largeur $2r$. Pour des raisons de symétrie, on limitera le domaine d'étude à $[0,r]\cup[0,H]$. On utilise le maillage triangulaire suivant :





Récupérer la table de connexion et la table de coordonnées avec la fonction ***tables***.

1. Ecrire l'équation d'équilibre du système, ses conditions aux limites et la formulation faible de ce problème 2D.

2. On utilise la méthode des éléments finis linéaires P1. En utilisant les résultats du TD, écrire la formulation faible discrète ainsi que les matrices de raideur élémentaires.

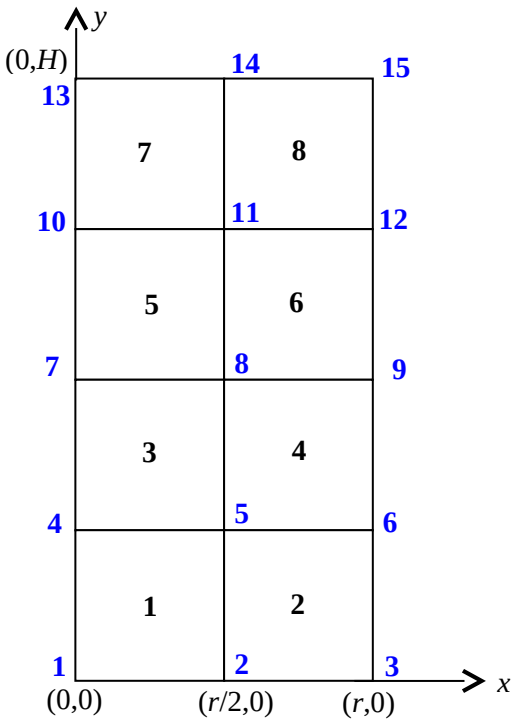
3. Ecrire le programme Python pour résoudre ce problème. Tracer le profil de température 2D. Pour le tracé du profil de température, on peut utiliser la fonction *contour* ou *contourf* de la bibliothèque matplotlib.pyplot.

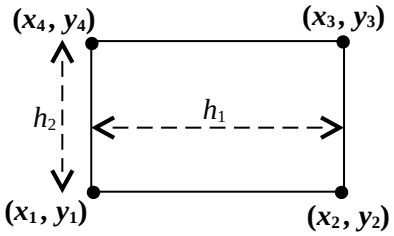
4. Généraliser le programme de manière à ce qu'il puisse fonctionner avec n'importe quel maillage.

C. Maillage quadrangulaire, éléments finis Q1

La seconde façon de mailler le domaine est d'utiliser des quadrangles. Les maillages quadrangulaires sont beaucoup utilisés, surtout en mécanique des solides car ils peuvent être plus précis. Cependant ils proposent des possibilités de maillage moins générales que les triangles.

1. En suivant la même méthode que pour le maillage triangulaire, écrire les tables de coordonnées et de connexion associées au maillage quadrangulaire ci-dessous. On considère ici que tous les quadrangles sont des rectangles de même hauteur h_2 et de même largeur h_1.





$$\Omega = Q_1 \cup Q_2 \cup \ldots \cup Q_{Nk}$$

avec Nk le nombre de quadrangles

Sur ce maillage, l'approximation est donnée par une somme de fonctions bilinéaires en x et y:

$$\widetilde{T}(x,y) = \sum_{k=1}^{Nk} \widetilde{T^{(k)}}(x,y)$$

avec

$$\widetilde{T}(x,y)=\begin{Bmatrix} T_1 H_1+T_2 H_2+T_3 H_3+T_4 H_4, & si(x,y)\in Q_k \\ 0, & sinon \end{Bmatrix}.$$

Les fonctions de forme sont de la forme:

$$H_i(x,y)=a_i+b_i x+c_i y+d_i xy, \quad i=1,4$$

et vérifient les propriétés suivantes:

$$H_i(x_j,y_j)=\begin{Bmatrix} 1, & si \ \ i=j \\ 0, & sinon \end{Bmatrix}$$

Dans ces conditions et en suivant le même développement que dans le cas d'un maillage triangulaire, on obtient les matrices de raideur suivantes dans le cas particulier d'un maillage parallèle avec les axes du plan :

$$K^x=\frac{h_2}{6h_1}\begin{bmatrix} 2 & -2 & -1 & 1 \\ -2 & 2 & 1 & -1 \\ -1 & 1 & 2 & -2 \\ 1 & -1 & -2 & 2 \end{bmatrix} \qquad K^y=\frac{h_1}{6h_2}\begin{bmatrix} 2 & 1 & -1 & -2 \\ 1 & 2 & -2 & -1 \\ -1 & -2 & 2 & 1 \\ -2 & -1 & 1 & 2 \end{bmatrix} \qquad \text{et} \qquad K=K^X+K^Y$$

2. Modifier le programme Python précédent pour résoudre le problème avec des éléments finis Q1. Tracer le profil de température 2D.

3. Généraliser ce code pour pouvoir faire évoluer le nombre de points de maillage suivant x et y et en déduire la valeur de la température obtenue en $(0,H)$. Tracer $\widetilde{T}(0,H)$ en fonction de h_1 en gardant constant le rapport h_1 / h_2. Que peut-on conclure?

III. CAS D'UNE EDO À COEFFICIENTS VARIABLES

On cherche à résoudre un problème aux conditions aux limites 1D par la méthode des éléments finis. Le problème s'écrit sous la forme de l'équation différentielle ordinaire suivante:

$$x^2\frac{d^2u(x)}{dx^2}+2x\frac{du(x)}{dx}+4=0 \qquad 1<x<4$$

associée aux conditions aux limite suivantes: $u(x=1)=1$ et $u(x=4)=0$.

1. Ecrire le résidu et la formulation faible de ce problème.

2. Déterminer l'expression des fonctions de forme pour des éléments finis linéaires.

3. On fixe le nombre d'éléments Ne = 5. La longueur de l'intervalle de discrétisation est $h=x_{i+1} - x_i$. Calculer les matrices de raideur et de masse élémentaires pour chaque élément.

4. Assembler la matrice globale à partir des matrices élémentaires.

5. Appliquer les conditions aux limites sur la matrice assemblée et le second membre.

6. Ecrire le programme Python pour résoudre ce problème.

7. La solution exacte est obtenue au moyen du calcul formel :

$$u_{ex}(x) = -4\ln(x) - \frac{\frac{32}{3}\ln(2) - \frac{4}{3}}{x} - \frac{1}{3} + \frac{32}{3}\ln(2).$$

Etudier l'ordre de précision de la méthode des éléments finis P1 en traçant la courbe de l'erreur ε en fonction de dx dans un graphe en échelle log-log.