

TP éléments finis

Compte rendu

UE-PL7017ME

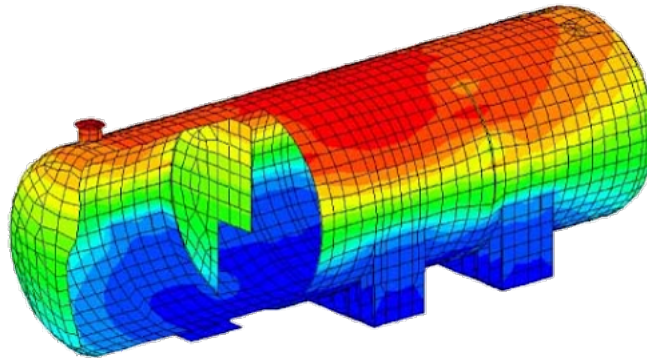


FIGURE 1 – Calculs d'éléments finis sur une cuve

Jérémy Archier
4A MECA
p2019441

Table des matières

I	Éléments finis unidimensionnels	3
1	Présentation du problème	3
2	Résolution analytique	4
2.1	Condition adiabatique	4
2.2	Condition non adiabatique	4
3	Résolution numérique	5
3.1	Éléments finis unidimensionnels linéaires	5
3.1.1	Fonctions de base	5
3.1.2	Condition adiabatique	6
3.1.3	Condition non adiabatique	7
3.2	Éléments finis unidimensionnels quadratiques	8
3.2.1	Fonctions de base	8
3.2.2	Condition adiabatique	10
3.2.3	Condition non adiabatique	11
4	Présentation des résultats	13
4.1	Problème de conduction-convection en régime stationnaire avec condition adiabatique	13
4.1.1	Éléments finis linéaires	13
4.1.2	Éléments finis quadratiques	15
4.2	Problème de conduction-convection en régime stationnaire avec condition non adiabatique . .	17
4.2.1	Éléments finis linéaires	17
4.2.2	Éléments finis Quadratiques	19
5	Conclusion	20
II	Éléments finis bidimensionnels	21
1	Présentation du problème	21
2	Résolution numérique	22
2.1	Éléments finis bidimensionnels linéaires	22
2.1.1	Fonctions de base	22
2.1.2	Formulation faible discrète	23
2.1.3	Écriture du système matriciel	24
2.1.4	Assemblage	25
3	Présentation des résultats	25
3.1	Paramètres	25
3.2	Graphe	26
4	Conclusion	26
A	Annexe	27
A.1	Fonction sousblock	27
A.2	Fonction pentadiagonale	27
A.3	La fonction triangle	28
A.4	La fonction tables	28

Résumé

Ce rapport réalisé dans le cadre du cours de Méthode numérique pour la mécanique II dispensé à Polytech Lyon par M. Hamda Ben Hadid. Il résume l'application de la méthode des éléments finis à un problème de convection / conduction en régime stationnaire.

On cherche à obtenir le champ de température dans un barreau métallique de type ailette. Pour cela on établit l'équation différentielle vérifiée par la température dans le domaine de l'ailette. Ensuite on va formuler dans un premier temps on utilisera les éléments finis linéaires unidimensionnels puis on reprendra la méthode en introduisant des éléments quadratiques. Un algorithme python permet l'assemblage de la matrice globale et la résolution du système matriciel. Ainsi on trace le champs de température dans l'ailette et l'erreur relative de la solution approché par rapport à la solution exacte obtenue par calcul analytique.

Mots clés : éléments finis, unidimensionnel, bidimensionnel, thermique, ailette

Introduction

On se propose ici d'utiliser la théorie des éléments finis avec des éléments linéaires puis quadratiques en 1D puis en 2D pour résoudre un problème de thermique traitant de la convection / conduction d'une ailette à l'aide du langage Python.

Première partie

Éléments finis unidimensionnels

1 Présentation du problème

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 :

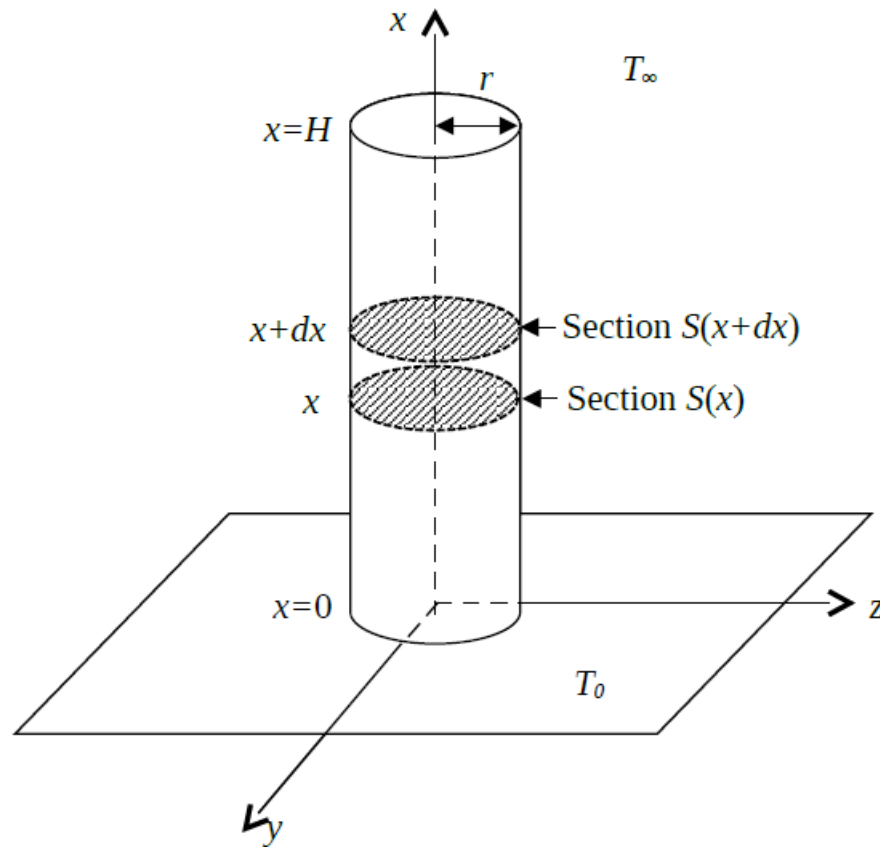


FIGURE 2 – Schéma de convection / conduction en régime stationnaire sur un barreau métallique

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 ($h = 100 \text{ W.m}^{-2}.\text{C}^{-1}$, $T_\infty = 20$ °C).

La conductivité thermique du barreau est $\lambda = 40 \text{ W.m}^{-1}.\text{C}^{-1}$

On fait un bilan thermique sur un élément de volume $V = Sdx$:

$$\phi_{cond}(x) - \phi_{cond}(x + dx) - \phi_{source} - \phi_{conv} = 0 \quad (1)$$

Ce qui peut se réécrire dans notre cas sans production de chaleur :

$$-\frac{\partial \phi_{cond}(x)}{\partial x} dx - \phi_{conv} = 0 \quad (2)$$

La loi de Fourier s'écrit :

$$\phi_{cond}(x) = -\lambda S \frac{\partial T}{\partial x} \quad (3)$$

La loi de Newton s'écrit :

$$\phi_{conv} = -hP(T(x) - T_{\infty}) \quad (4)$$

L'équation (2) peut se réécrire :

$$-\frac{\partial}{\partial x}(-\lambda S \frac{\partial T}{\partial x}) dx + hP(T(x) - T_{\infty}) dx = 0 \quad (5)$$

Finalement en considérant que le problème est 1D, il vient :

$$\frac{d^2 T}{dx^2} + \frac{hP}{\lambda S}(T(x) - T_{\infty}) = 0 \quad \forall x \in [0, L] \quad (6)$$

2 Résolution analytique

2.1 Condition adiabatique

Les conditions aux limites s'écrivent :

$$\begin{cases} T(0) = T_0 \\ -\lambda \frac{dT}{dx} \Big|_{x=H} = 0 \end{cases} \quad (7)$$

La solution exacte d'une telle équation différentielles s'écrit :

$$\boxed{T_{ex}(x) = T_{\infty} + (T_0 - T_{\infty}) \frac{\cosh(m(H-x))}{\cosh(mH)}} \quad \text{avec} \quad m^2 = \frac{hP}{\lambda S} \quad (8)$$

2.2 Condition non adiabatique

Les conditions aux limites s'écrivent :

$$\begin{cases} T(0) = T_0 \\ -\lambda S \frac{dT}{dx} \Big|_{x=H} = h'P(T(H) - T_{\infty}) \end{cases} \quad (9)$$

La solution exacte d'une telle équation différentielles s'écrit :

$$\boxed{T_{ex}(x) = T_{\infty} + (T_0 - T_{\infty}) \frac{\cosh(m(H-x)) + \frac{h'}{\lambda m} \sinh(m(H-x))}{\cosh(mH) + \frac{h'}{\lambda m} \sinh(mH)}} \quad \text{avec} \quad m^2 = \frac{hP}{\lambda S} \quad (10)$$

3 Résolution numérique

3.1 Éléments finis unidimensionnels linéaires

3.1.1 Fonctions de base

On considère la fonction d'essai linéaire sous la forme :

$$\tilde{T}(x) = a_1x + a_2 \quad (11)$$

On cherche à remplacer les valeurs de a_1 et a_2 par les valeurs de T aux noeuds T_i et T_{i+1} :



FIGURE 3 – Schéma de discrétisation

$$\tilde{T}(x_i) = a_1x_i + a_2 = T_i \quad (12)$$

$$\tilde{T}(x_{i+1}) = a_1x_{i+1} + a_2 = T_{i+1} \quad (13)$$

On va donc obtenir :

$$a_1 = \frac{T_{i+1} - T_i}{x_{i+1} - x_i} \quad (14)$$

$$a_2 = \frac{T_ix_{i+1} - T_{i+1}x_i}{x_{i+1} - x_i} \quad (15)$$

On peut donc réécrire (11) :

$$\tilde{T}(x) = H_1(x)T_i + H_2(x)T_{i+1} \quad (16)$$

où

$$H_1(x) = \frac{x_{i+1} - x}{h} \quad (17)$$

$$H_2(x) = \frac{x - x_i}{h} \quad (18)$$

$$h = x_{i+1} - x_i \quad (19)$$

Les équations (17) et (18) sont les fonctions de forme.

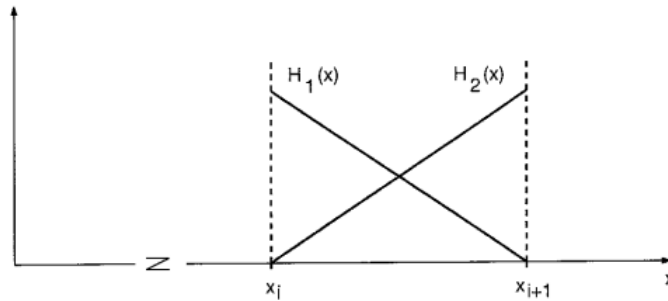


FIGURE 4 – Représentation des fonctions de forme

3.1.2 Condition adiabatique

Le résidu s'écrit en considérant la fonction d'essai \tilde{T} comme :

$$R(\tilde{T}) = \frac{d^2\tilde{T}}{dx^2} - m^2(\tilde{T}(x) - T_\infty) \quad (20)$$

On cherche à le minimiser :

$$||R(\tilde{T})|| = 0 \quad (21)$$

En introduisant une fonction de test (ou poids) ω telle que :

$$\omega(x) = \begin{bmatrix} \frac{\partial \tilde{T}_i}{\partial a_i} \\ \frac{\partial \tilde{T}_i}{\partial a_{i+1}} \end{bmatrix}$$

$$\omega(x) = \begin{bmatrix} H_1(x) \\ H_2(x) \end{bmatrix}$$

On obtient la formulation forte :

$$\int_0^H \left(\frac{d^2\tilde{T}}{dx^2} - m^2(\tilde{T}(x) - T_\infty) \right) \omega(x) dx = 0 \quad (22)$$

La formulation faible s'obtient grâce à la formule de Green :

$$\int_0^H \left(-\frac{d\omega}{dx} \frac{d\tilde{T}}{dx} - \omega \tilde{T} + m^2(\tilde{T}(x) - T_\infty) \omega(x) \right) dx + \left[\omega \frac{d\tilde{T}}{dx} \right]_0^H = 0 \quad (23)$$

Les conditions aux limites s'écrivent pour les fonctions de poids :

$$\begin{cases} \omega(0) = 0 \\ \omega(H) = 0 \end{cases} \quad (24)$$

Le terme de bord peut se simplifier :

$$\left[\omega \frac{d\tilde{T}}{dx} \right]_0^H = \omega(0) \frac{d\tilde{T}(0)}{dx} - \omega(H) \frac{d\tilde{T}(H)}{dx} = 0 \quad (25)$$

La formulation faible devient :

$$\int_0^H \left(-\frac{d\omega}{dx} \frac{d\tilde{T}}{dx} - \omega \tilde{T} + m^2(\tilde{T}(x) - T_\infty) \omega(x) \right) dx = 0 \quad (26)$$

On écrit la formulation faible discrète pour N_e éléments :

$$\sum_{i=1}^{N_e-1} \int_{x_i}^{x_{i+1}} \frac{d\omega}{dx} \frac{d\tilde{T}}{dx} + m^2 \tilde{T}(x) \omega(x) dx = \sum_{i=1}^{N_e-1} \int_{x_i}^{x_{i+1}} m^2 T_\infty \omega(x) dx \quad (27)$$

La formulation faible peut se réécrire :

$$\sum_{i=1}^{N_e-1} K^e \begin{bmatrix} T_i \\ T_{i+1} \end{bmatrix} + m^2 M^e \begin{bmatrix} T_i \\ T_{i+1} \end{bmatrix} dx = m^2 T_\infty B^e \quad (28)$$

$$\text{avec} \quad K^e = \frac{1}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad M^e = \frac{h}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \quad B^e = \frac{h}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

La matrice globale assemblée sera de la forme :

$$\begin{bmatrix}
K^{11} + m^2 M^{11} & -K^{11} + m^2 M^{12} & 0 & 0 & 0 & 0 \\
-K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 & 0 & 0 \\
0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 & 0 \\
0 & 0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 \\
0 & 0 & 0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} \\
0 & 0 & 0 & 0 & -K^1 + m^2 M^{11} & K^{11} + m^2 M^{11}
\end{bmatrix}
\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix}
=
\begin{bmatrix}
m^2 T_\infty B_1 \\
m^2 T_\infty (B_1 + B_2) \\
m^2 T_\infty (B_1 + B_2) \\
m^2 T_\infty (B_1 + B_2) \\
m^2 T_\infty (B_1 + B_2) \\
m^2 T_\infty B_2
\end{bmatrix}$$

Or, on a les conditions aux limites :

$$\begin{cases}
T(0) = T_0 \\
-\lambda \frac{\partial T}{\partial x} \Big|_{x=H} = 0
\end{cases} \quad (29)$$

La matrice assemblée en prenant en compte les conditions aux limites s'écrit :

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
-K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 & 0 & 0 \\
0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 & 0 \\
0 & 0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 \\
0 & 0 & 0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} \\
0 & 0 & 0 & 0 & -K^1 + m^2 M^{11} & K^{11} + m^2 M^{11}
\end{bmatrix}
\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix}
=
\begin{bmatrix}
T_0 \\
m^2 T_\infty (B_1 + B_2) \\
m^2 T_\infty (B_1 + B_2) \\
m^2 T_\infty (B_1 + B_2) \\
m^2 T_\infty (B_1 + B_2) \\
m^2 T_\infty B_2
\end{bmatrix}$$

3.1.3 Condition non adiabatique

On reprend la formulation faible générique de l'équation (23) :

$$\int_0^H \left(-\frac{d\omega}{dx} \frac{d\tilde{T}}{dx} - \omega \tilde{T} + m^2 (\tilde{T}(x) - T_\infty) \omega(x) \right) dx + \left[\omega \frac{d\tilde{T}}{dx} \right]_0^H = 0 \quad (30)$$

Les conditions aux limites s'écrivent :

$$\begin{cases}
\omega(0) = 0 \\
\frac{d\tilde{T}}{dx} \Big|_{x=H} = -\frac{h' P}{\lambda S} (\tilde{T}(H) - T_\infty)
\end{cases} \quad (31)$$

Le terme de bord peut se réécrire :

$$\left[\omega \frac{d\tilde{T}}{dx} \right]_0^H = \omega(0) \frac{d\tilde{T}(0)}{dx} - \omega(H) \frac{d\tilde{T}(H)}{dx} = \omega(H) \frac{h'P}{\lambda S} (\tilde{T}(H) - T_\infty) \quad (32)$$

La formulation faible devient en posant $m'^2 = \frac{h'P}{\lambda S}$:

$$\int_0^H \left(-\frac{d\omega}{dx} \frac{d\tilde{T}}{dx} - \omega \tilde{T} + m^2(\tilde{T}(x) - T_\infty)\omega(x) \right) dx + m'^2(\tilde{T}(H) - T_\infty)\omega(H) = 0 \quad (33)$$

On écrit la formulation faible discrète :

$$\sum_{i=1}^{N_e-1} \int_{x_i}^{x_{i+1}} \frac{d\omega}{dx} \frac{d\tilde{T}}{dx} + m^2(\tilde{T}(x) - T_\infty)\omega(x) dx + m'^2(\tilde{T}(H) - T_\infty)\omega(H) = 0 \quad (34)$$

La formulation faible peut se réécrire :

$$\sum_{i=1}^{N_e-1} K^e \begin{bmatrix} T_i \\ T_{i+1} \end{bmatrix} + m^2 M^e \begin{bmatrix} T_i \\ T_{i+1} \end{bmatrix} dx = m^2 T_\infty \begin{bmatrix} B_i \\ B_{i+1} \end{bmatrix} + m'^2 (T_\infty - T(H)) \begin{bmatrix} H_i(H) \\ H_{i+1}(H) \end{bmatrix} \quad (35)$$

avec $K^e = \frac{1}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$, $M^e = \frac{h}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$, $B^e = \frac{h}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Or, on a les conditions aux limites :

$$\begin{cases} T(0) = T_0 \\ \frac{\partial T}{\partial x} \Big|_{x=H} = -\frac{h'}{\lambda} (T(H) - T_\infty) \end{cases} \quad (36)$$

La matrice assemblée en prenant en compte les conditions aux limites s'écrit :

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 & 0 & 0 \\ 0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 & 0 \\ 0 & 0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 \\ 0 & 0 & 0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} \\ 0 & 0 & 0 & 0 & -K^1 + m^2 M^{11} & K^{11} + m^2 M^{11} + h'/\lambda \end{bmatrix}$$

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix} = \begin{bmatrix} T_0 \\ m^2 T_\infty (B_1 + B_2) \\ m^2 T_\infty (B_1 + B_2) \\ m^2 T_\infty (B_1 + B_2) \\ m^2 T_\infty (B_1 + B_2) \\ m^2 T_\infty B_2 + \frac{h'}{\lambda} \end{bmatrix}$$

3.2 Éléments finis unidimensionnels quadratiques

3.2.1 Fonctions de base

On considère la fonction d'essai linéaire sous la forme :

$$\tilde{T}(x) = a_0 + a_1 x + a_2 x^2 \quad (37)$$

Nous pouvons réécrire l'équation (37) :

$$\tilde{T}(x) = [1 \ x \ x^2] \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} \quad (38)$$

On cherche à remplacer les valeurs de a_0, a_1 et a_2 par les valeurs de T aux noeuds $T_i, T_{i+\frac{1}{2}}$ et T_{i+1} :

$$\begin{Bmatrix} u_i \\ u_{i+\frac{1}{2}} \\ u_{i+1} \end{Bmatrix} = \begin{bmatrix} 1 & x_i & x_i^2 \\ 1 & x_{i+\frac{1}{2}} & x_{i+\frac{1}{2}}^2 \\ 1 & x_{i+1} & x_{i+1}^2 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} \quad (39)$$

L'inversion de la matrice du système (39) donne :

$$\begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} x_{i+\frac{1}{2}}x_{i+1}^2 - x_{i+1}x_{i+\frac{1}{2}}^2 & x_{i+1}x_i^2 - x_ix_{i+1}^2 & x_ix_{i+\frac{1}{2}}^2 - x_{i+\frac{1}{2}}x_i^2 \\ x_{i+\frac{1}{2}}^2 - x_i^2 & x_{i+1}^2 - x_i^2 & x_i^2 - x_{i+\frac{1}{2}}^2 \\ x_{i+1} - x_{i+\frac{1}{2}} & x_i - x_{i+1} & x_{i+\frac{1}{2}} - x_i \end{bmatrix} \begin{Bmatrix} u_i \\ u_{i+\frac{1}{2}} \\ u_{i+1} \end{Bmatrix} \quad (40)$$

où

$$A = \frac{1}{2} \det \begin{bmatrix} 1 & x_i & x_i^2 \\ 1 & x_{i+\frac{1}{2}} & x_{i+\frac{1}{2}}^2 \\ 1 & x_{i+1} & x_{i+1}^2 \end{bmatrix} \quad (41)$$

On peut donc réécrire (37) :

$$\tilde{T}(x) = H_1(x)T_i + H_2(x)T_{i+\frac{1}{2}} + H_3(x)T_{i+1} \quad (42)$$

où

$$\begin{aligned} H_1(x) &= \frac{1}{2A} \left[(x_{i+\frac{1}{2}}x_{i+1}^2 - x_{i+1}x_{i+\frac{1}{2}}^2) + (x_{i+\frac{1}{2}}^2 - x_{i+1}^2)x + (x_{i+1} - x_{i+\frac{1}{2}})x^2 \right] \\ H_2(x) &= \frac{1}{2A} \left[(x_{i+1}x_i^2 - x_ix_{i+1}^2) + (x_{i+1}^2 - x_i^2)x + (x_i - x_{i+1})x^2 \right] \\ H_3(x) &= \frac{1}{2A} \left[(x_ix_{i+\frac{1}{2}}^2 - x_{i+\frac{1}{2}}x_i^2) + (x_i^2 - x_{i+\frac{1}{2}}^2)x + (x_{i+\frac{1}{2}} - x_i)x^2 \right] \end{aligned} \quad (43)$$

Les équations (43) sont les fonctions de forme.

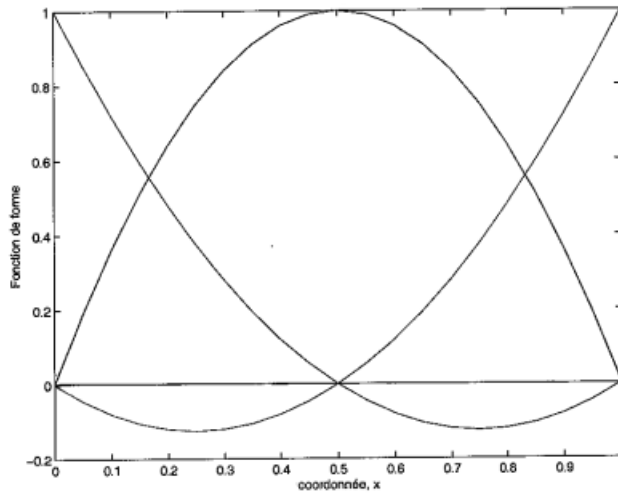


FIGURE 5 – Représentation des fonctions de forme en quadratique

3.2.2 Condition adiabatique

La formulation faible s'écrit comme pour les éléments linéaires :

$$\int_0^H \frac{dT}{dx} m^2 (T(x) - T_\infty) \omega(x) dx = 0 \quad (44)$$

On écrit la formulation faible discrète :

$$\sum_{i=1}^{N_e-1} \int_{x_i}^{x_{i+1}} \frac{d\omega}{dx} \frac{dT}{dx} + m^2 T(x) \omega(x) dx = \sum_{i=1}^{N_e-1} \int_{x_i}^{x_{i+1}} m^2 T_\infty \omega(x) dx \quad (45)$$

La formulation faible peut se réécrire :

$$\sum_{i=1}^{N_e-1} K^e \begin{bmatrix} T_i \\ T_{i+\frac{1}{2}} \\ T_{i+1} \end{bmatrix} + m^2 M^e \begin{bmatrix} T_i \\ T_{i+\frac{1}{2}} \\ T_{i+1} \end{bmatrix} dx = m^2 T_\infty B^e \quad (46)$$

avec $K^e = \frac{1}{3h} \begin{bmatrix} 7 & -8 & 1 \\ -8 & 16 & -8 \\ 1 & -8 & 7 \end{bmatrix}$, $M^e = \frac{h}{30} \begin{bmatrix} 4 & 2 & -1 \\ 2 & 16 & 2 \\ -1 & 2 & 4 \end{bmatrix}$, $B^e = \frac{h}{6} \begin{bmatrix} 1 \\ 4 \\ 1 \end{bmatrix}$

La matrice globale assemblée sera de la forme :

$$\begin{bmatrix} K^{11} + m^2 M^{11} & K^{12} + m^2 M^{12} & K^{13} + m^2 M^{13} & 0 & 0 & 0 \\ -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 & 0 & 0 \\ 0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 & 0 \\ 0 & 0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 \\ 0 & 0 & 0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} \\ 0 & 0 & 0 & 0 & -K^1 + m^2 M^{11} & K^{11} + m^2 M^{11} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_8 \\ T_9 \\ T_{10} \\ T_{11} \end{bmatrix}$$

$$= \begin{bmatrix} m^2 T_\infty \frac{h}{6} \\ m^2 T_\infty \frac{4h}{6} \\ m^2 T_\infty \frac{2h}{6} \\ m^2 T_\infty \frac{4h}{6} \\ m^2 T_\infty \frac{2h}{6} \\ m^2 T_\infty \frac{4h}{6} \\ m^2 T_\infty \frac{2h}{6} \\ m^2 T_\infty \frac{4h}{6} \\ m^2 T_\infty \frac{2h}{6} \\ m^2 T_\infty \frac{4h}{6} \\ m^2 T_\infty \frac{h}{6} \end{bmatrix}$$

Or, on a les conditions aux limites :

$$\begin{cases} T(0) = T_0 \\ -\lambda \frac{\partial T}{\partial x} \Big|_{x=H} = 0 \end{cases} \quad (47)$$

La matrice globale assemblée en prenant en compte les conditions aux limites s'écrit :

$$\begin{bmatrix}
 1 & 0 & 0 & 0 & 0 & 0 \\
 -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 & 0 & 0 \\
 0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 & 0 \\
 0 & 0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 \\
 0 & 0 & 0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} \\
 0 & 0 & 0 & 0 & -K^1 + m^2 M^{11} & K^{11} + m^2 M^{11}
 \end{bmatrix}
 \begin{bmatrix}
 T_1 \\
 T_2 \\
 T_3 \\
 T_4 \\
 T_5 \\
 T_6 \\
 T_7 \\
 T_8 \\
 T_9 \\
 T_{10} \\
 T_{11}
 \end{bmatrix}
 =
 \begin{bmatrix}
 T_0 \\
 m^2 T_\infty (B_1 + B_2) \\
 m^2 T_\infty (B_1 + B_2 + B_3) \\
 m^2 T_\infty (B_1 + B_2 + B_3) \\
 m^2 T_\infty (B_1 + B_2 + B_3) \\
 m^2 T_\infty (B_1 + B_2 + B_3) \\
 m^2 T_\infty (B_1 + B_2 + B_3) \\
 m^2 T_\infty (B_1 + B_2 + B_3) \\
 m^2 T_\infty (B_1 + B_2 + B_3) \\
 m^2 T_\infty (B_2 + B_3) \\
 m^2 T_\infty B_3
 \end{bmatrix}$$

3.2.3 Condition non adiabatique

La formulation faible s'écrit comme pour les éléments linéaires avec la condition non adiabatique :

$$\sum_{i=1}^{N_e-1} \int_{x_i}^{x_{i+1}} \frac{d\omega}{dx} \frac{d\tilde{T}}{dx} + m^2 (\tilde{T}(x) - T_\infty) \omega(x) dx + m'^2 (\tilde{T}(H) - T_\infty) \omega(H) = 0 \quad (48)$$

La formulation faible peut se réécrire :

$$\sum_{i=1}^{N_e-1} K^e \begin{bmatrix} T_i \\ T_{i+\frac{1}{2}} \\ T_{i+1} \end{bmatrix} + m^2 M^e \begin{bmatrix} T_i \\ T_{i+\frac{1}{2}} \\ T_{i+1} \end{bmatrix} dx = m^2 T_\infty B^e + m'^2 (T_\infty - T(H)) \begin{bmatrix} T_i \\ T_{i+\frac{1}{2}} \\ T_{i+1} \end{bmatrix} \quad (49)$$

Or, on a les conditions aux limites :

$$\begin{cases} T(0) = T_0 \\ \left. \frac{\partial T}{\partial x} \right|_{x=H} = -\frac{h'}{\lambda} (T(H) - T_\infty) \end{cases} \quad (50)$$

La matrice assemblée en prenant en compte les conditions aux limites s'écrit :

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
-K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 & 0 & 0 \\
0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 & 0 \\
0 & 0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} & 0 \\
0 & 0 & 0 & -K^1 + m^2 M^{11} & 2(K^{11} + m^2 M^{11}) & -K^1 + m^2 M^{11} \\
0 & 0 & 0 & 0 & -K^1 + m^2 M^{11} & K^{11} + m^2 M^{11}
\end{bmatrix}
\begin{bmatrix}
T_1 \\
T_2 \\
T_3 \\
T_4 \\
T_5 \\
T_6 \\
T_7 \\
T_8 \\
T_9 \\
T_{10} \\
T_{11}
\end{bmatrix}$$

=

$$\begin{bmatrix}
T_0 \\
m^2 T_\infty (B_1 + B_2) \\
m^2 T_\infty (B_1 + B_2 + B_3) \\
m^2 T_\infty (B_1 + B_2 + B_3) \\
m^2 T_\infty (B_1 + B_2 + B_3) \\
m^2 T_\infty (B_1 + B_2 + B_3) \\
m^2 T_\infty (B_1 + B_2 + B_3) \\
m^2 T_\infty (B_1 + B_2 + B_3) \\
m^2 T_\infty (B_1 + B_2 + B_3) \\
m^2 T_\infty (B_2 + B_3) \\
m^2 T_\infty B_3 + \frac{h'}{\lambda}
\end{bmatrix}$$

4 Présentation des résultats

4.1 Problème de conduction-convection en régime stationnaire avec condition adiabatique

4.1.1 Éléments finis linéaires

4.1.1.1 Graphe

On trace la solution obtenue à l'aide du programme python de la méthode des éléments finis unidimensionnels linéaires sur $N_e = 5$ éléments que l'on compare avec la solution exacte que l'on a calculé précédemment :

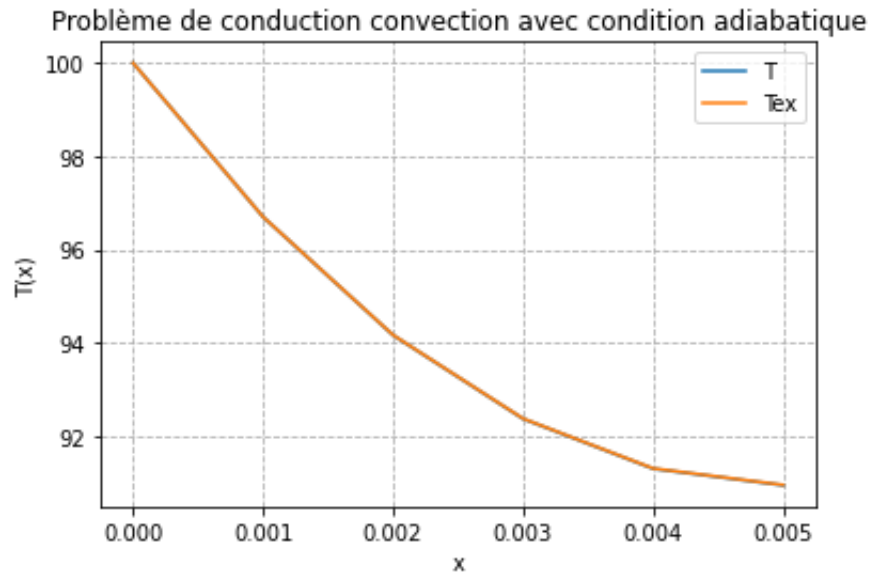


FIGURE 6 – Graphe de la solution éléments finis linéaire et la solution exacte du problème

On remarque que les solutions exacte et approchée sont tellement proche qu'elles se superposent. La méthode des éléments finis a donc bien approché la solution exacte. Cependant, on verra par la suite l'erreur commise grâce à la courbe de l'erreur relative.

On voit que l'évolution de la température dans le barreau subit une décroissance exponentielle ce qui est cohérent avec la forme de la solution exacte en cosinus hyperbolique.

4.1.1.2 Erreur relative

On étudie 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.

L'erreur relative s'écrit :

$$\epsilon = \frac{\|U_{num}(x) - U_{ex}(x)\|}{\|U_{ex}(x)\|} \quad (51)$$

On trace l'erreur relative pour la solution que l'on a calculé à l'aide des éléments finis :

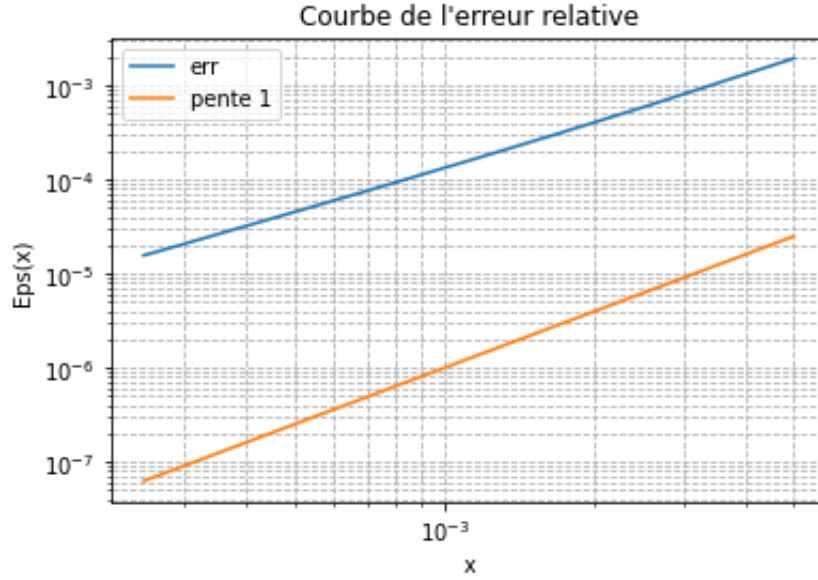


FIGURE 7 – Représentation de l'erreur relative en échelle logarithmique du problème adiabatique avec des éléments P1

La pente de la droite est de $\alpha = 1$, l'erreur relative est donc d'ordre 1. C'est à dire que si l'on réduit le pas d'intégration par 2, alors l'erreur commise est réduite par 10.

L'erreur relative est donc bien cohérente avec les éléments finis qui sont dit linéaire ou encore P1.

4.1.2 Éléments finis quadratiques

4.1.2.1 Graphe

On trace la solution obtenue à l'aide du programme python de la méthode des éléments finis unidimensionnels linéaires sur $N_e = 5$ éléments que l'on compare avec la solution exacte que l'on a calculé précédemment :

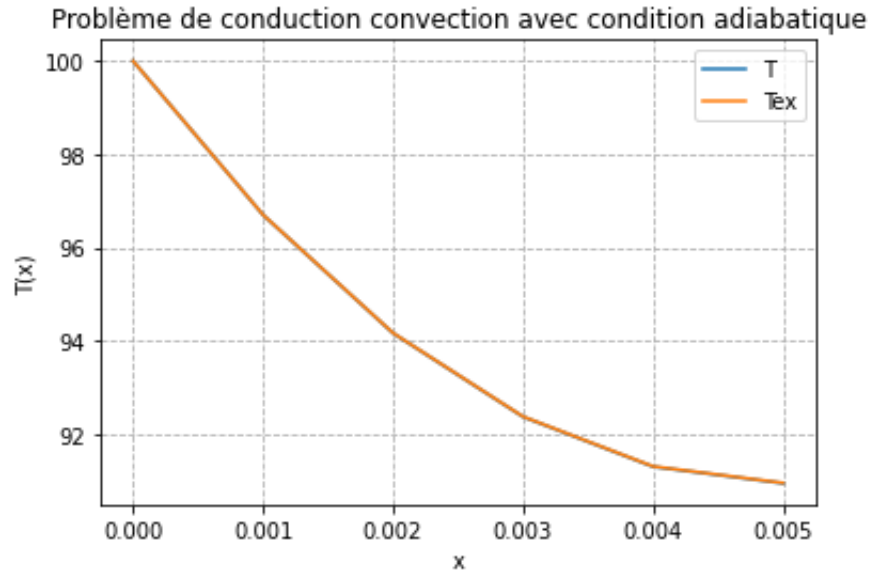


FIGURE 8 – Graphe de la solution éléments finis linéaire et la solution exacte du problème

On remarque que les solutions exacte et approchée sont tellement proche qu'elles se superposent. La méthode des éléments finis a donc bien approché la solution exacte. Cependant, on verra par la suite l'erreur commise grâce à la courbe de l'erreur relative.

On voit que l'évolution de la température dans le barreau subit une décroissance exponentielle ce qui est cohérent avec la forme de la solution exacte en cosinus hyperbolique.

4.1.2.2 Erreur relative

On étudie l'ordre de précision de la méthode des éléments finis P2 en traçant la courbe de l'erreur relative ϵ en fonction de dx (longueur d'un élément) dans un graphe en échelle logarithmique.

On trace l'erreur relative pour la solution que l'on a calculé à l'aide des éléments finis :

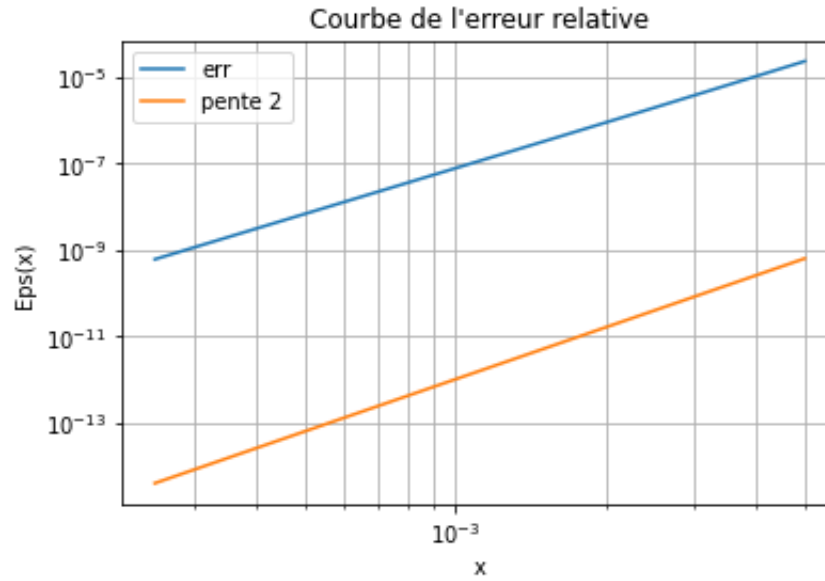


FIGURE 9 – Représentation de l'erreur relative en échelle logarithmique du problème adiabatique avec des éléments P2

La pente de la droite est de $\alpha = 2$, l'erreur relative est donc d'ordre 2. C'est à dire que si l'on réduit le pas d'intégration par 2, alors l'erreur commise est réduite par 100.

L'erreur relative est donc bien cohérente avec les éléments finis qui sont dit quadratique ou encore P2.

4.2 Problème de conduction-convection en régime stationnaire avec condition non adiabatique

4.2.1 Éléments finis linéaires

4.2.1.1 Graphe

On trace la solution obtenue à l'aide du programme python de la méthode des éléments finis unidimensionnels linéaires sur $N_e = 5$ éléments que l'on compare avec la solution exacte que l'on a calculé précédemment :

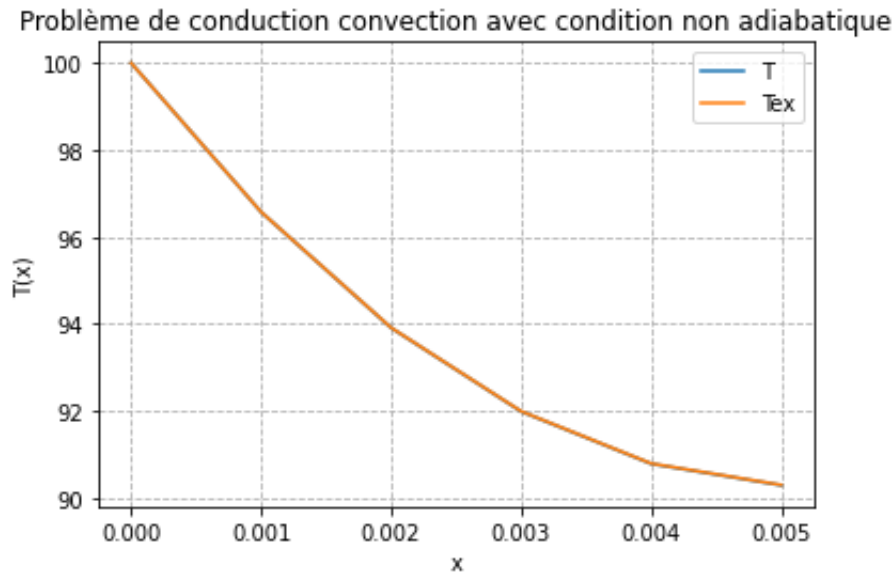


FIGURE 10 – Graphe de la solution éléments finis linéaire et la solution exacte du problème

On remarque qu'une nouvelle fois les solutions exacte et approchée sont tellement proche qu'elles se superposent. La méthode des éléments finis a donc bien approché la solution exacte. Cependant, on verra par la suite l'erreur commise grâce à la courbe de l'erreur relative.

L'évolution de la température dans le barreau subit encore une décroissance exponentielle ce qui est cohérent avec la forme de la solution exacte qui est une combinaison linéaire de cosinus et sinus hyperbolique.

L'évolution de la température est similaire à celle avec une condition adiabatique bien que la température au bout du barreau est plus basse pour la condition non adiabatique.

4.2.1.2 Erreur relative

On étudie 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.

On trace l'erreur relative pour la solution que l'on a calculé à l'aide des éléments finis :

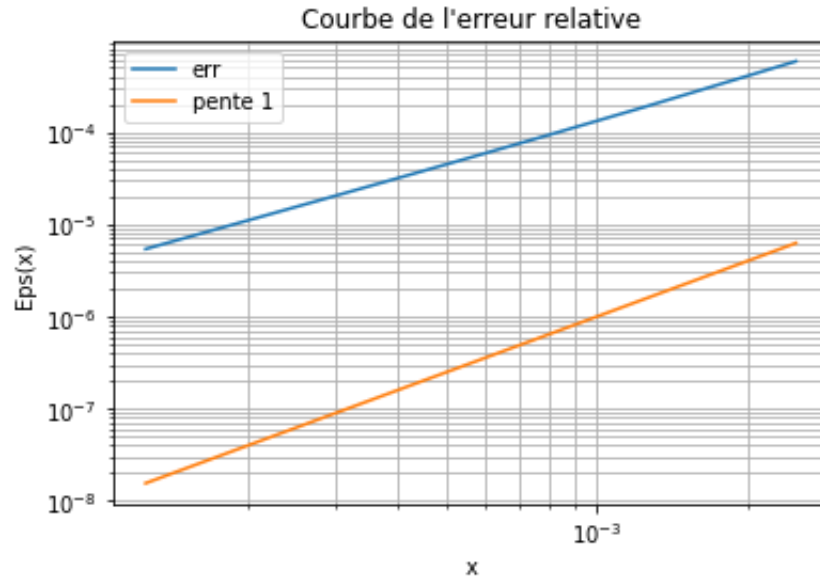


FIGURE 11 – Représentation de l'erreur relative en échelle logarithmique du problème non adiabatique avec des éléments P1

La pente de la droite est également de $\alpha = 1$, l'erreur relative est donc d'ordre 1 comme pour la condition adiabatique. C'est à dire que si l'on réduit le pas d'intégration par 2, alors l'erreur commise est réduite par 10.

L'erreur relative est donc bien cohérente avec les éléments finis qui sont dit linéaire ou encore P1.

4.2.2 Éléments finis Quadratiques

4.2.2.1 Graphe

On trace la solution obtenue à l'aide du programme python de la méthode des éléments finis unidimensionnels linéaires sur $N_e = 5$ éléments que l'on compare avec la solution exacte que l'on a calculé précédemment :

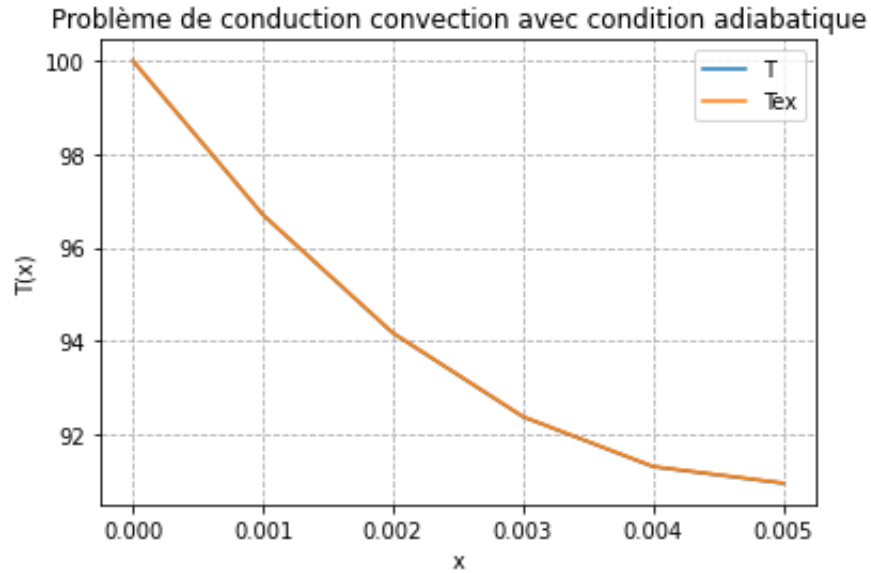


FIGURE 12 – Graphe de la solution éléments finis linéaire et la solution exacte du problème

On remarque que les solutions exacte et approchée sont tellement proche qu'elles se superposent. La méthode des éléments finis a donc bien approché la solution exacte. Cependant, on verra par la suite l'erreur commise grâce à la courbe de l'erreur relative.

L'évolution de la température dans le barreau subit encore une décroissance exponentielle ce qui est cohérent avec la forme de la solution exacte qui est une combinaison linéaire de cosinus et sinus hyperbolique.

L'évolution de la température est similaire à celle avec une condition adiabatique bien que la température au bout du barreau est plus basse pour la condition non adiabatique.

4.2.2.2 Erreur relative

On étudie l'ordre de précision de la méthode des éléments finis P2 en traçant la courbe de l'erreur relative ϵ en fonction de dx (longueur d'un élément) dans un graphe en échelle logarithmique.

On trace l'erreur relative pour la solution que l'on a calculé à l'aide des éléments finis :

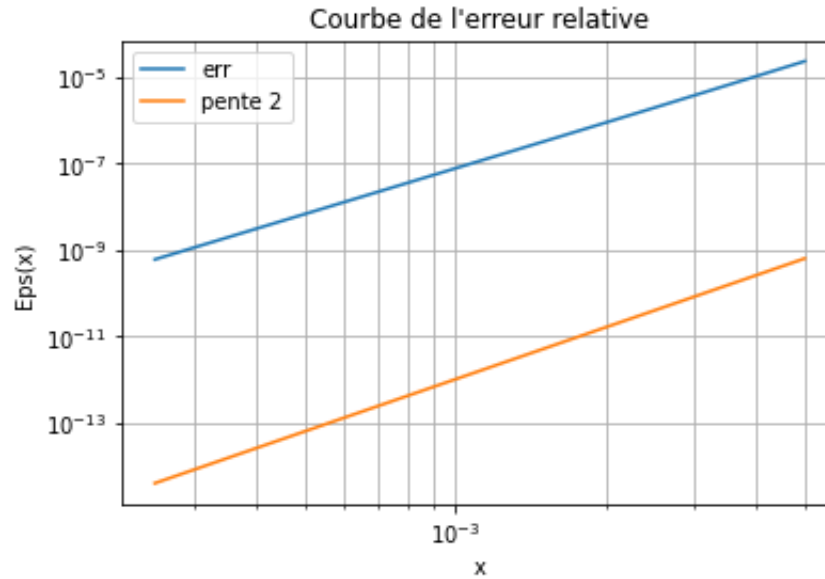


FIGURE 13 – Représentation de l'erreur relative en échelle logarithmique du problème adiabatique avec des éléments P2

La pente de la droite est de $\alpha = 2$, l'erreur relative est donc d'ordre 2. C'est à dire que si l'on réduit le pas d'intégration par 2, alors l'erreur commise est réduite par 100.

L'erreur relative est donc bien cohérente avec les éléments finis qui sont dit linéaire ou encore P2.

5 Conclusion

Pour conclure, on voit bien que la méthode des éléments finis en P1 et P2 permettent de bien approché la solution exacte même pour peu d'éléments. Avec des éléments P2 la précision est l'erreur relative est grandement améliorée.

Cependant cette méthode des éléments finis nécessite un certain formalisme mathématiques et la mise en équation nécessite un peu de calcul et d'autant plus pour des éléments quadratiques.

Deuxième partie

Éléments finis bidimensionnels

1 Présentation du problème

On cherche à obtenir la répartition de la température en 2D dans une ailette de longueur L grande par rapport à sa hauteur H et largeur $2r$. Pour des raisons de symétrie, on limitera le domaine d'étude à $[0, r] \cup [0, H]$. La géométrie du problème est présentée sur la figure suivante :

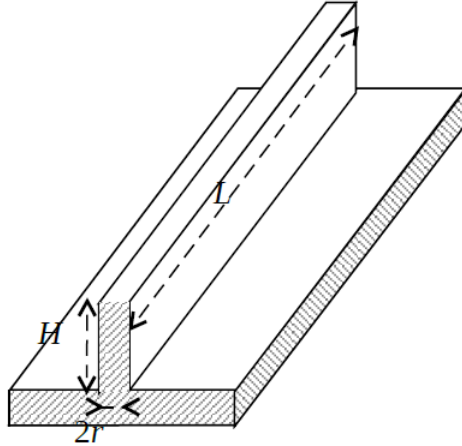


FIGURE 14 – Schéma du barreau métallique

La température sur l'extrémité du barreau en $y = 0$ est imposée et égale à $T_0 = 100^\circ \text{C}$. Le flux thermique est nul pour $x = 0$ et l'aillette est refroidie par convection avec le milieu ambiant ($h = 100 \text{ W.m}^{-2}.\text{C}^{-1}$, $T_\infty = 20^\circ \text{C}$).

La conductivité thermique du barreau est $\lambda = 40 \text{ W.m}^{-1}.\text{C}^{-1}$

L'équation d'équilibre du système en régime stationnaire 2D s'écrit :

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \quad (52)$$

En tenant compte de ses conditions aux limites suivantes :

$$\left\{ \begin{array}{l} T(x, 0) = T_0 \\ \frac{\partial T}{\partial y}(0, y) = 0 \\ -\lambda \frac{\partial T}{\partial x}(x, H) = h(T(x, H) - T_\infty) \\ -\lambda \frac{\partial T}{\partial y}(r, y) = h(T(r, y) - T_\infty) \end{array} \right. \quad (53)$$

2 Résolution numérique

2.1 Éléments finis bidimensionnels linéaires

2.1.1 Fonctions de base

On considère la fonction d'essai linéaire sous la forme :

$$\tilde{T}(x) = a_1 + a_2x + a_3y \quad (54)$$

On cherche à remplacer les valeurs de a_1 , a_2 et a_3 par les valeurs de T aux noeuds t_1, t_2 et t_3 :

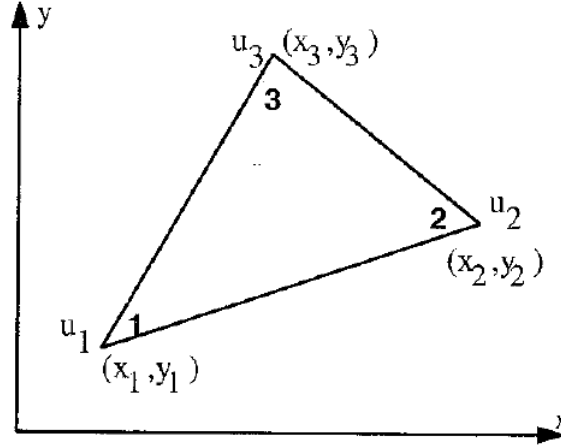


FIGURE 15 – Schéma de discrétisation

La fonction d'interpolation T représente les variables nodales aux trois noeuds. Par conséquent, la substitution des valeurs x et y à chaque noeuds donne :

$$\begin{Bmatrix} T_1 \\ T_2 \\ T_3 \end{Bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} \quad (55)$$

L'inversion de la matrice du système donne :

$$\begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} x_2y_3 - x_3y_2 & x_3y_1 - x_1y_2 & x_1y_2 - x_2y_1 \\ y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \end{Bmatrix} \quad (56)$$

où

$$A = \frac{1}{2} \det \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \quad (57)$$

On peut donc réécrire (54) :

$$\tilde{T}(x) = H_1(x, y)T_1 + H_2(x, y)T_2 + H_3(x, y)T_3 \quad (58)$$

où les $H_i(x, y)$ sont les fonctions de formes de l'élément triangulaire linéaire et dont les expressions sont :

$$\begin{aligned} H_1(x, y) &= \frac{1}{2A} [(x_2y_3 - x_3y_2) + (y_2 - y_3)x + (x_3 - x_2)y] \\ H_2(x, y) &= \frac{1}{2A} [(x_3y_1 - x_1y_3) + (y_3 - y_1)x + (x_1 - x_3)y] \\ H_3(x, y) &= \frac{1}{2A} [(x_1y_2 - x_2y_1) + (y_1 - y_2)x + (x_2 - x_1)y] \end{aligned} \quad (59)$$

Le domaine d'intégration est discrétisé en un nombre donné d'éléments triangulaires linéaires. On utilise le maillage triangulaire suivant :

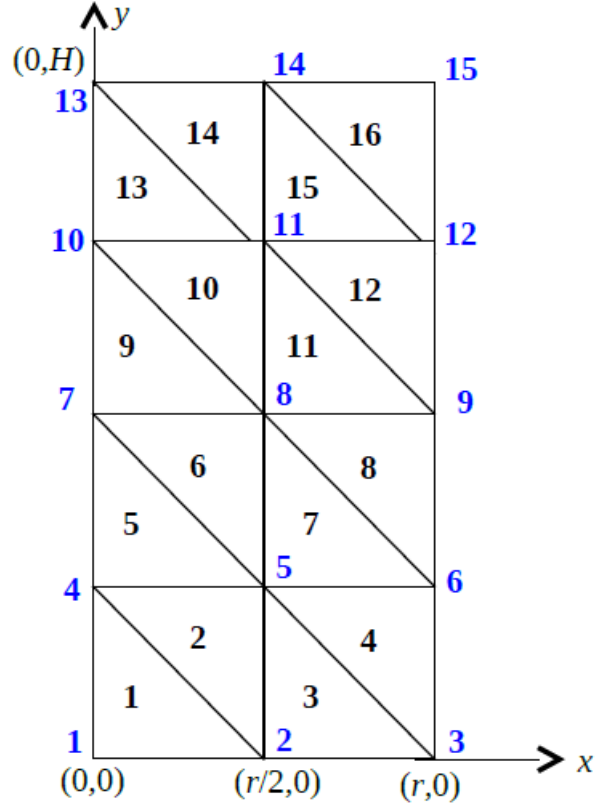


FIGURE 16 – Schéma de la discrétisation du barreau sur le domaine

2.1.2 Formulation faible discrète

Le résidu s'écrit en considérant la fonction d'essai \tilde{T} comme :

$$R(\tilde{T}) = \frac{\partial^2 \tilde{T}}{\partial x^2} + \frac{\partial^2 \tilde{T}}{\partial y^2} \quad (60)$$

On cherche à le minimiser :

$$||R(\tilde{T})|| = 0 \quad (61)$$

En introduisant une fonction de test (ou poids) ω telle que :

$$\omega(x, y) = \begin{bmatrix} \frac{\partial \tilde{T}_i}{\partial T_1} \\ \frac{\partial \tilde{T}_i}{\partial T_2} \\ \frac{\partial \tilde{T}_i}{\partial T_3} \end{bmatrix}$$

$$\omega(x, y) = \begin{bmatrix} H_1(x, y) \\ H_2(x, y) \\ H_3(x, y) \end{bmatrix}$$

On obtient la formulation forte :

$$\int_0^H \int_0^r \left(\frac{\partial^2 \tilde{T}}{\partial x^2} + \frac{\partial^2 \tilde{T}}{\partial y^2} \right) \omega(x, y) dx dy = 0 \quad (62)$$

La formule de Green permet d'écrire :

$$\int_0^H \left(\left[\omega(x, y) \frac{\partial \tilde{T}(x, y)}{\partial x} \right]_0^r - \int_0^r \frac{\partial \omega}{\partial x} \frac{\partial \tilde{T}}{\partial x} dx \right) dy + \int_0^r \left(\left[\omega(x, y) \frac{\partial \tilde{T}(x, y)}{\partial y} \right]_0^H - \int_0^H \frac{\partial \omega}{\partial y} \frac{\partial \tilde{T}}{\partial y} dy \right) dx = 0 \quad (63)$$

Les conditions aux limites s'écrivent pour les conditions de Neumann :

$$\begin{cases} \left[\omega(x, y) \frac{\partial \tilde{T}(x, y)}{\partial x} \right]_0^r = \frac{h}{\lambda} (-T(r, y) + T_\infty) \\ \left[\omega(x, y) \frac{\partial \tilde{T}(x, y)}{\partial y} \right]_0^H = \frac{h}{\lambda} (-T(x, H) + T_\infty) \end{cases} \quad (64)$$

La formulation faible devient :

$$- \int_0^H \int_0^r \left(\frac{\partial \omega}{\partial x} \frac{\partial \tilde{T}}{\partial x} + \frac{\partial \omega}{\partial y} \frac{\partial \tilde{T}}{\partial y} \right) dx dy + \int_0^r \frac{h}{\lambda} (-T(x, H) + T_\infty) \omega dx + \int_0^H \frac{h}{\lambda} (-T(r, y) + T_\infty) \omega dy = 0 \quad (65)$$

On écrit la formulation faible discrète pour N_e éléments :

$$\begin{aligned} \sum_{k=1}^{N_e} \int_{\Omega} \left(\left\{ \begin{array}{c} \frac{\partial H_1}{\partial x} \\ \frac{\partial H_2}{\partial x} \\ \frac{\partial H_3}{\partial x} \end{array} \right\} \left\{ \begin{array}{ccc} \frac{\partial H_1}{\partial x} & \frac{\partial H_2}{\partial x} & \frac{\partial H_3}{\partial x} \end{array} \right\} \left\{ \begin{array}{c} T_1 \\ T_2 \\ T_3 \end{array} \right\}^k + \left\{ \begin{array}{c} \frac{\partial H_1}{\partial y} \\ \frac{\partial H_2}{\partial y} \\ \frac{\partial H_3}{\partial y} \end{array} \right\} \left\{ \begin{array}{ccc} \frac{\partial H_1}{\partial y} & \frac{\partial H_2}{\partial y} & \frac{\partial H_3}{\partial y} \end{array} \right\} \left\{ \begin{array}{c} T_1 \\ T_2 \\ T_3 \end{array} \right\}^k \right) d\Omega \\ + \sum_{m=1}^{M=6} \int_{\Omega_m} \frac{h}{\lambda} \left\{ \begin{array}{c} H_1 \\ H_2 \\ H_3 \end{array} \right\} \left\{ \begin{array}{ccc} H_1 & H_2 & H_3 \end{array} \right\} \left\{ \begin{array}{c} T_1 \\ T_2 \\ T_3 \end{array} \right\}^m d\Omega = \sum_{m=1}^{M=6} \int_{\Omega_m} \frac{h}{\lambda} T_\infty \left\{ \begin{array}{c} H_1 \\ H_2 \\ H_3 \end{array} \right\} d\Omega \end{aligned} \quad (66)$$

2.1.3 Écriture du système matriciel

La formulation faible peut se réécrire :

$$\sum_{k=1}^{N_e} (K_x^{(k)} + K_y^{(k)}) \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}^k + \sum_{m=1}^{M=6} \frac{h}{\lambda} M^m \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}^m = \sum_{m=1}^{M=6} \frac{h}{\lambda} T_\infty B^m \quad (67)$$

où

$$\begin{aligned} K_x^{(k)} &= \frac{1}{2 \det(A)} \begin{bmatrix} (y_2 - y_3)^2 & (y_2 - y_3)(y_3 - y_1) & (y_2 - y_3)(y_1 - y_2) \\ (y_2 - y_3)(y_3 - y_1) & (y_3 - y_1)^2 & (y_3 - y_1)(y_1 - y_2) \\ (y_2 - y_3)(y_1 - y_2) & (y_3 - y_1)(y_1 - y_2) & (y_1 - y_2)^2 \end{bmatrix} \\ K_y^{(k)} &= \frac{1}{2 \det(A)} \begin{bmatrix} (x_2 - x_3)^2 & (x_2 - x_3)(x_3 - x_1) & (x_2 - x_3)(x_1 - x_2) \\ (x_2 - x_3)(x_3 - x_1) & (x_3 - x_1)^2 & (x_3 - x_1)(x_1 - x_2) \\ (x_2 - x_3)(x_1 - x_2) & (x_3 - x_1)(x_1 - x_2) & (x_1 - x_2)^2 \end{bmatrix} \\ B^k &= \int_{\Omega} \begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix} d\Omega \end{aligned} \quad (68)$$

$$M^k = \begin{bmatrix} \int_{\Omega_k} H_1^2 d\Omega & \int_{\Omega_k} H_1 H_2 d\Omega & \int_{\Omega_k} H_1 H_3 d\Omega \\ \int_{\Omega_k} H_1 H_2 d\Omega & \int_{\Omega_k} H_2^2 d\Omega & \int_{\Omega_k} H_2 H_3 d\Omega \\ \int_{\Omega_k} H_1 H_3 d\Omega & \int_{\Omega_k} H_2 H_3 d\Omega & \int_{\Omega_k} H_3^2 d\Omega \end{bmatrix}$$

2.1.4 Assemblage

Nous n'allons pas expliciter l'assemblage du système cela prendrait trop de place. Cependant, nous allons expliquer la démarche qui sera fait dans le programme python. Nous aurons un système de la forme $\mathcal{A}\mathcal{T} = \mathcal{B}$ avec \mathcal{A} est de la taille 16x16, \mathcal{T} et \mathcal{B} de taille 16x1.

La matrice \mathcal{A} est formée à partir des matrices élémentaires $M^k, K_x^{(k)}, K_y^{(k)}$ et de la table de connections. Cette dernière permet de déterminer l'emplacement des matrices élémentaires dans \mathcal{A} .

Le vecteur \mathcal{B} est construit de la même manière que \mathcal{A} mais en utilisant B^k . Enfin le vecteur \mathcal{T} contient la température de chaque éléments (de 1 à 16).

Nous avons utilisé la table de connexion suivante :

Élément	n ₁	n ₂	n ₃
1	1	2	4
2	2	5	4
3	2	3	5
4	3	6	5
5	4	5	7
6	5	8	7
7	5	6	8
8	6	9	8
9	5	8	10
10	8	11	10
11	8	9	11
12	9	12	11
13	10	11	13
14	11	14	13
15	11	12	14
16	12	15	14

TABLE 1 – Table de connexion

3 Présentation des résultats

3.1 Paramètres

Pour tracer le profil de température 2D de l'ailette, nous avons pris les paramètres de discrétisation suivant :

le nombre de noeuds $n = 15$
le nombre point de discretiation selon x $n_x = 3$
le nombre point de discretisation selon y $n_y = 5$

3.2 Graphe

Tout d'abord nous allons définir ce qu'est un contour. Un tracé de contour est une méthode graphique permettant de visualiser la surface 3D en traçant des tranches Z constantes appelées contours dans un format 2D. Le tracé de contour est une alternative à un tracé de surface 3D.

Nous allons tracer le profil de température en 2D dans l'ailette grâce à notre programme python. Pour le tracé du profil de température, on utilise la fonction contour de la bibliothèque matplotlib.pyplot.

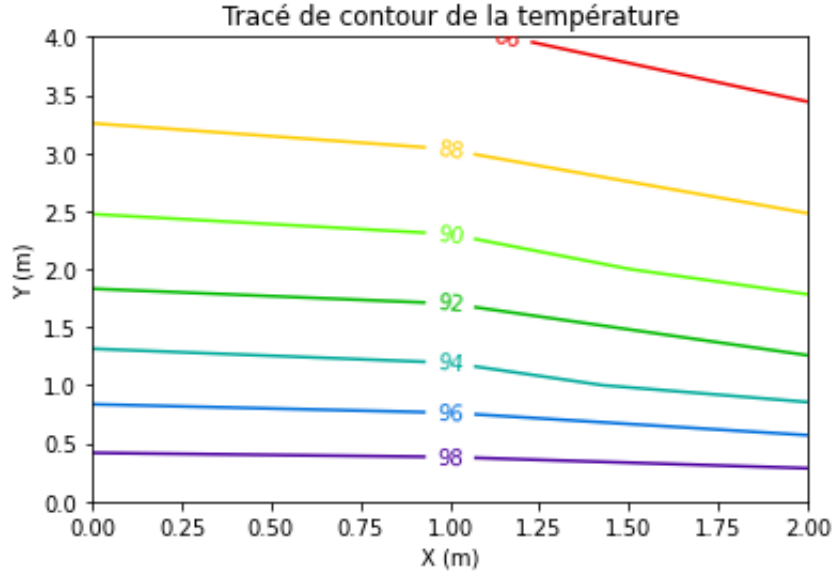


FIGURE 17 – Schéma du profil de température dans la demi-ailette

On peut remarquer que la température décroît selon y , ce qui est en accord avec la convection en $y = H$ mais aussi avec la diffusion qui s'opèrent comme l'ailette à une température fixée en son bord à $y = 0$.

On voit également que à y fixé, la température décroît si l'on augmente la valeur de x . On voit une nouvelle fois les effets de la diffusion et de la convection à la frontière où $x = r$.

Enfin, le flux nul pour $x = 0$ se retrouve par les isothermes qui sont quasi des droites sur l'intervalle $[0, r/2]$ qui traduisent la non variation de la température par rapport à la variable y .

4 Conclusion

Pour conclure, on voit bien que la méthode des éléments finis bidimensionnels permet d'avoir des résultats probants pour un problème de conduction/convection d'ailette pour un problème qu'il serait difficile de résoudre à la main.

Cependant, la mise en place est bien plus longue quand 1D et le programme pour faire l'assemblage est bien plus conséquent et l'assemblage plus fastidieux avec la table de connexion.

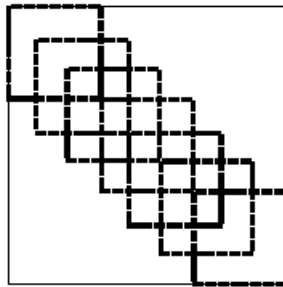
A Annexe

Nous allons présenter plusieurs fonctions Python qui nous ont été utile pour programmer ces méthodes des éléments finis.

A.1 Fonction sousblock

```
def sousblock(n,A):  
    """A=np.array([[2,1,1],[1,2,1],[1,1,2]])"""  
    a=np.zeros((9,9))  
    for k in range(0,7,n):  
        for i in range(3):  
            for j in range(3):  
                a[i+k,j+k]=a[i+k,j+k]+A[i,j]  
    return(a)
```

La fonction "sousblock" permet de faire l'assemblage de la matrice globale en créant une matrice de taille $n \times n$ à partir du matrice A de taille cohérente. La matrice résultante sera diagonale par blocs tel que les blocs soient décalés d'une ligne et d'une colonne à chaque itération comme suit :



A.2 Fonction pentadiagonale

```
def pentadiagonale(n):  
    M=np.zeros((n**2,n**2))  
    for i in range (n**2):  
        for j in range (n**2):  
            if i==j:  
                M[i,j]=4  
            if i==j+1:  
                M[i,j]=-1  
            if i==j-1:  
                M[i,j]=-1  
            if i==j-n:  
                M[i,j]=2  
            if i==j+n:  
                M[i,j]=2  
    return(M)
```

La fonction "pentadiagonale" permet de créer en fonction de la taille n souhaité, la matrice pentadiagonale A telle que :

$$A_{ij} = \begin{cases} 4, & \text{si } i=j \\ -1, & \text{si } i=j+1 \\ -1, & \text{si } i=j-1 \\ 2, & \text{si } i=j-n \\ 2, & \text{si } i=j+n \end{cases} \quad \text{avec } i=1..n^2, j=1..n^2.$$

A.3 La fonction triangle

```
def triangle(x1,y1,x2,y2,x3,y3):
    plt.plot([x1, x2], [y1, y2], 'r-', lw=2) # Red straight line
    plt.plot([x2, x3], [y2,y3], 'r-', lw=2)
    plt.plot([x3, x1], [y3,y1], 'r-', lw=2)
    plt.show()
    return()
```

La fonction "triangle" permet de tracer un triangle à partir des coordonnées de ses trois sommets. Cette dernière est bien utile pour les éléments finis 2D lorsqu'il s'agit d'établir le maillage du domaine d'étude.

A.4 La fonction tables

```
def tables(Lx,Ly,Nx,Ny,e):
    """ Donnees : * Lx - Float : Longueur selon la direction x
                  * Ly - Float : Longueur selon la direction y
                  * Nx - Int   : Nbre point de discretisation selon x
                  * Ny - Int   : Nbre point de discretisation selon y

    Resultats : * Noeud : Table coord. gloable Noeuds
                * Tbc : Table de connexion

    """
    nx = Nx - 1 # Nbre element sur x
    ny = Ny - 1 # Nbre element sur y

    lx = np.linspace(0,Lx,Nx)
    ly = np.linspace(0,Ly,Ny)
    Noeud = np.zeros((Nx*Ny,2))
    if e=='triangle':
        Tbc = np.zeros((2*nx*ny,3),dtype='int')
    elif e == 'carre':
        Tbc = np.zeros((nx*ny,4),dtype='int')

    Ne = 0
    Nn = 0
    i=0
    j=0
    compteur = 0

    while j < Ny-1:      # On se deplace sur les points sur y
```

```

i = 0
while i < Nx: # On se deplace sur les points sur x
    if e == 'triangle':
        if 0 < i and (Ne+1)%2 == 0:
            A1=j*Nx+i
            xA1 = lx[i]
            yA1 = ly[j]
            A2=(j+1)*Nx+i
            xA2 = lx[i]
            yA2 = ly[j+1]
            A3=(j+1)*Nx+i-1
            xA3 = lx[i-1]
            yA3 = ly[j+1]

        elif 0 <= i <= Nx-2:
            A1=j*Nx+i
            xA1 = lx[i]
            yA1 = ly[j]
            A2=j*Nx+i+1
            xA2 = lx[i+1]
            yA2 = ly[j]
            A3=(j+1)*Nx+i
            xA3 = lx[i]
            yA3 = ly[j+1]
            i=i+1

        elif (i+1)%Nx == 0:
            break;

    elif e == 'carre':
        if (i+1)%Nx == 0:
            break;
        else:
            A1=j*Nx+i
            xA1 = lx[i]
            yA1 = ly[j]
            A2=j*Nx+i+1
            xA2 = lx[i+1]
            yA2 = ly[j]
            A3=(j+1)*Nx+i+1
            xA3 = lx[i+1]
            yA3 = ly[j+1]
            A4=(j+1)*Nx+i
            xA4 = lx[i]
            yA4 = ly[j+1]
            i=i+1

    if e == 'triangle':
        Tbc[Ne,0]=int(A1)
        Tbc[Ne,1]=int(A2)
        Tbc[Ne,2]=int(A3)
    elif e == 'carre':
        Tbc[Ne,0]=int(A1)
        Tbc[Ne,1]=int(A2)

```

```

        Tbc[Ne,2]=int(A3)
        Tbc[Ne,3]=int(A4)

    Noeud[A1,0] = xA1
    Noeud[A1,1] = yA1
    Noeud[A2,0] = xA2
    Noeud[A2,1] = yA2
    Noeud[A3,0] = xA3
    Noeud[A3,1] = yA3

    if e == 'carre':
        Noeud[A4,0] = xA4
        Noeud[A4,1] = yA4
    Ne = Ne + 1 # Numero de element
    j=j+1
    return (Tbc,Noeud)

```

La fonction "tables" permet la création a partir du nombre de point et de la longueur du domaine la table de coordonnées globale des noeuds et la table de connexion pour un maillage triangulaire.