UNIVERSITÉ DE STRASBOURG

Simulation numérique d’un dispositif de refroidissement

Projet: C++

Effectué par:

Youness El Houssaini

Université de Strasbourg

Faculté: UFR mathématiques et informatique

Option: Calcul Scientifique et Mathématiques de l'Information

Examineur: Dr. Vincent Chabannes

Strasbourg

Décembre 2019

Résumé

Dans ce projet, on a étudié le comportement thermique d’un dispositif de refroidissement d’un micro-processeur. Ce comportement est simulé à travers un programme C++ en se basant sur la méthodes des différences finis et optant pour le schéma implicite. Le rapport présente en première partie une introduction au problème puis on traite le cas statique et on fini par le cas dynamique. La méthode du travail est inspiré du modèle de programmation vue en cours : Analyser (chapitre 1), conception globale (digramme de classe chaptire 2), conception détaillé (déscription : attributs et méthodes) et puis le code fourni en pièce jointe. Par suite on a commenté dans le code une partie de tests unitaire et de vérification et on a décommenté la partie de validation.

Remerciments

Je tiens à remercier toutes les personnes qui ont contribué à ce projet et qui m'ont aidé lors de la rédaction de ce rapport.

Je remercie également mon professeur, **Mr Vincent Chabannes** qui m'a beaucoup aidé dans ma formation et m'a permis d'avoir des bases solides en C++. Son écoute et ses conseils m'ont permis d'améliorer mon niveau de programmation, et surtout de comprendre les subtilités de concepts abordés pour la première fois.

Table de matière

I. Nomenclature

Symboles	Signification	Unité
$L_x, L_y et L_z$	Dimensions de l'ailette du dissipateur	m
p	Périmètre transversale $\frac{1}{2}(L_y + L_z)$	m
S	Aire transversale $(L_y \times L_z)$	m^2
Φ_p	Flux de la chaleur dus là a différence de température	m
T	Température le long de l'ailette	p
x	Position le long de l'axe x	m
T_e	Température extérieure	K
m	The vertical plan of the component	S
$(L_y \times L_z)$	Masse volumique	m^2
Φ_p	Chaleur spécifique à pression constante	$J.kg^{-1}.K^{-1}$
$W.m^{-2}$	Coefficient de transfert thermique	$W.m^{-2}.K^{-1}$

II. Abbréviation

Abbreviation	Signification
EDP	**E**quation aux **D**érivées **P**artielles
LU	Méthode de Gauss
CSV	**C**omma d **S**eparated **V**alue

1. Introduction

Ce projet a pour but de réaliser un programme C++ permettant d'étudier le comportement thermique d'un dispositif de refroidissement d'un micro-processeur. Un des moyens généralement utilisé pour réguler la température d'un processeur est l'utilisation d'un ventilateur.

1.1 Context

Le soufflage d'air va permettre d'évacuer la chaleur par convection sur la surface du processeur. Pour améliorer ce processus, le processeur est généralement attaché à un dissipateur (composé de plusieurs ailettes) qui est un élement très conducteur de chaleur. Ce disposif supplémentaire, représenté avec la figure 1, permet d'augmenter la surface d'échange avec le flux d'air et d'ainsi refroidir plus efficacement le composant électronique.

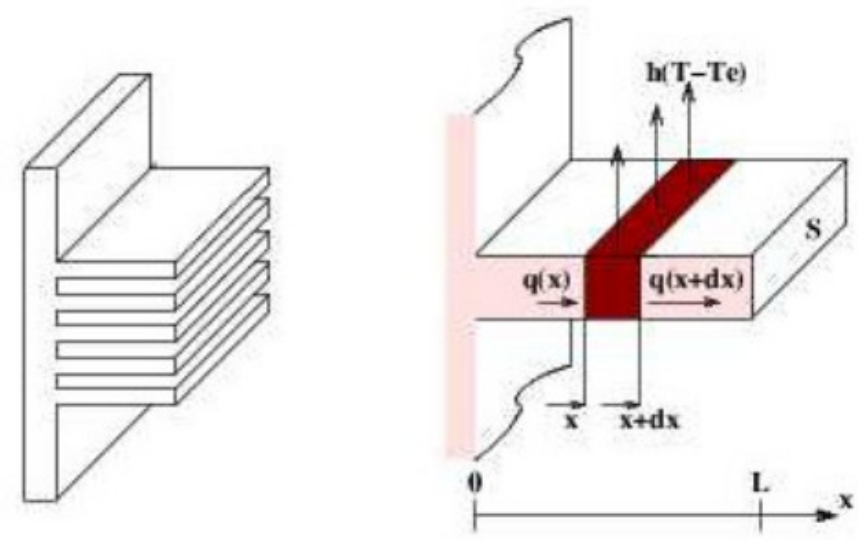


Figure 1 Modélisation d'une ailette du dissipateur.

1.2 Le cas d'étude

Pour ce projet, nous allons nous intéresser uniquememt à la simulation thermique d'une seule ailette du dissipateur. Les données du problème sont : la géométrie de l'ailette, figure 2, qui est décrite par les longueurs Lx , Ly et Lz ,le flux de chaleur Φp généré par le processeur et la température ambiante Te . De plus, nous supposons que l'ailette est suffisamment mince pour

considérer le problème unidimensionnel. La température T en un point donné dépendra uniquement de sa position selon x et de l'instant t.

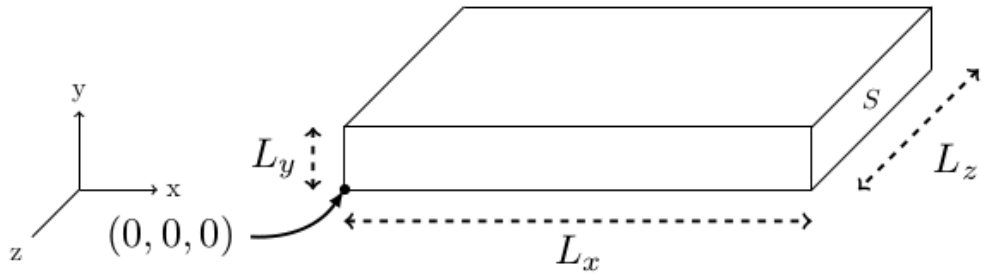


Figure 2 Géométrie de l'ailette.

1.3 EDP

En prenant en compte les pertes latérales par convection avec l'air, l'équation de la chaleur dans une ailette définie sur le domaine x = [0, Lx] est donnée par :

$$\rho C_p \frac{\partial T}{\partial t} - \kappa \frac{\partial^2 T}{\partial^2 x} + \frac{h_c p}{S}\left(T - T_e\right) = 0 \qquad (1)$$

Au niveau des conditions aux limites, nous supposons connaître le flux de chaleur Φp dégagé par le processeur en x = 0 et nous faisons l'hyphothèse que le transfert de chaleur en x = Lx est négligeable par rapport au flux apporté sur la section longitudinale. Cette modélisation nous conduit finalement à la recherche de la température T définie sur le domaine x = [0, Lx] qui vérifie :

$$-\kappa \frac{\partial T}{\partial x}_{x=0} = \Phi_p \qquad (2)$$

$$-\kappa \frac{\partial T}{\partial x}_{x=L_x} = 0 \qquad (3)$$

Les valeurs des paramètres utilisés pour les simulations sont données dans la table 1. On a choisi les propriétés physiques en considérant que l'ailette est constituée d'un alliage d'aluminium.

Cependant, les valeurs des paramètres pourront être modifiées dans le but d'analyser l'impact du paramètre sur le résultat. De plus, la valeur du coefficent de transfert de chaleur surfacique hc est donnée dans ce tableau pour le cas où le ventilateur est en marche. On pourra également traiter le cas où le ventilateur est éteint en utilisant hc = 10 W/(m 2 · K).

nom	valeur	unité	nom	valeur	unité
ρ	2700	$kg/(m^3)$	h_c	200	$W/(m^2 \cdot K)$
C_p	940	$J/(kg \cdot K)$	L_x	0.04	m
κ	164	$W/(m \cdot K)$	L_y	0.004	m
T_e	20	°C	L_z	0.05	m
Φ_p	$1.25 \cdot 10^5$	W/m^2			

Table 1 Données initiales.

2. Modèle stationnaire

Nous commençons par traiter le cas stationnaire, c'est-à-dire par calculer la distribution de température lorsque le temps t tend vers l'infini. On enlevera alors la dérivée en temps présente dans l'équation 1.

2.1 Schémas et conditions aux limites

Pour résoudre cette EDP, nous allons utiliser la métode des différences finies. Pour cela nous décomposons le segment [0, Lx] en M intervalles de longueur h = Lx /M . Nous obtenons ainsi un maillage composé de M+1 points d'abscisses $x_i = i * h$, i = 0, ..., M . La solution numérique sera alors décrite par ces M+1 points et on notera Ti la température calculée au point x_i .

Au niveau de la discrétisation de l'EDP, on utilise un développement de Taylor, ce qui permet de discrétiser la dérivée seconde à l'aide du schéma centré suivant dit central:

$$\frac{\partial^2 T}{\partial^2 x}(x_i) \approx \frac{T_{i-1} - 2T_i + T_{i+1}}{h^2}$$

Nous obtenons ainsi les équations discrètes du problème définies sur tous les noeuds internes :

$$-\kappa \frac{T_{i-1} - 2T_i + T_{i+1}}{h^2} + \frac{h_c p}{S}(T_i - T_e) = 0, \forall i \in [1, M-1] \quad (4)$$

Reste à appliquer les conditions aux limites :

$$-\kappa \frac{\partial T}{\partial x}_{x=0} \approx -\kappa \frac{T_1 - T_0}{h} = \Phi_p \quad (5)$$

$$-\kappa \frac{\partial T}{\partial x_{L_x=0}} \approx -\kappa \frac{T_{M-1} - T_M}{h} = 0 \quad (6)$$

2.2 Système algébrique

L'équation (4) peut être réecrite en réarrangeant les termes $T_i, T_{i-1} et\, T_{i+1}$ comme suivant:

$$\underbrace{\frac{-\kappa}{h^2}}_{a_i} T_{i-1} + \underbrace{\frac{-\kappa}{h^2}}_{c_i} T_{i+1} + \underbrace{\left(\frac{-\kappa}{h^2} + \frac{h_c p}{S}\right)}_{b_i} T_i = \underbrace{\frac{h_c p}{S} T_e}_{F_i}, \forall i \in [1, M-1] \qquad (4')$$

Les termes correspoandant à i = {0,M} sont déduite des conditions aux limites (5) et (6).

$$\underbrace{-\frac{\kappa}{h}}_{c_0} T_1 + \underbrace{\frac{\kappa}{h}}_{b_0} T_0 = \underbrace{\Phi_p}_{F_0} \qquad (5')$$

$$-\kappa \frac{\partial T}{\partial x_{L_x=0}} \approx -\kappa \frac{T_{M-1} - T_M}{h} = 0 \quad (6')$$

Le problème discret est décrit par la résolution d'un système AX = F avec A une matrice tridiagonale (taille (M+1) × (M +1)), F le vecteur second membre (taille M+1) et X le vecteur solution représentant la température en chaque point de discrétisation xi , i ∈ [0, M].

$$\underbrace{\begin{pmatrix} b_0 & c_0 & 0 & 0 & 0 & \cdots & 0 \\ a_1 & b_1 & c_1 & 0 & 0 & \cdots & 0 \\ 0 & a_2 & b_2 & c_2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & a_{M-1} & b_{M-1} & c_{M-1} \\ 0 & \cdots & \cdots & 0 & 0 & a_M & b_M \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} T_0 \\ T_1 \\ T_2 \\ \vdots \\ \vdots \\ T_{M-1} \\ T_M \end{pmatrix}}_{X} = \underbrace{\begin{pmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ \vdots \\ F_{M-1} \\ F_M \end{pmatrix}}_{F}$$

2.3 Résolution du système algébrique

La résolution du système algébrique $AX = F \Leftrightarrow LUX = F$ se fait en trois étape:

- Décomposition LU,
- Résolution LY = F,
- Résolution UX = Y

Pour ce, on a à disposition une classe “matrice” et une classe “Source” qui sont hérité par la classe “Resolution” comme c'est illustré par le diagramme des classes en dessous.

2.3.1 Classe matrice et classe Source

La classe matrice A étant tridiagonale contient une méthode 'LU' dédiée à sa décomposition en LU. Cette classe permet en effet le calcul des vecteurs $(a_i)_{i\in[1,M-1]}, (b_i)_{i\in[0,M-1]} et (c_i)_{i\in[1,M-1]}$ requis pour la résolution.Cette classe donc a besoin de tous les paramétres du système étudié, ce qui justifie la forme du diagramme des classes. La classe “ Source” regroupe tous les attributs liés à la source de chaleur et permet par suite le calcule du terme second membre.

2.3.2 Classe résolution

Cette classe permet le calcul du terme second membre $(F_i)_{i\in[0,M]}$ qui est déduit de la classe “Source”. Elle a entre autre une méthode pour exporter le résultat en format csv et puis nous permet de vérifier le résultat à travers une méthode pour le calcule de la solution exacte.

2.3.3 Classes propriétés

Pour faciliter l'accé aux différents paramètres de l'ailette on a construit plusieurs objets chacun représente une propriété physique indépendante. Ceci nous permet aussi de faire une analyse cause à effet de l'ailette. Le diagramme des classes suivant décrit les différentes classes élémentaires et leurs dépendances par héritage.

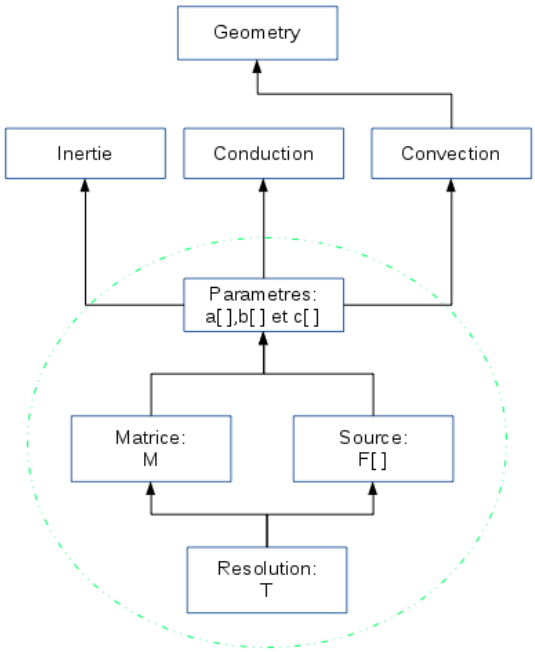


Figure 4 Diagrammes des classes.

Exportation

Dans le cas stationnaire, afin de visualiser la solution numérique 1d on exporte un fichier au format csv. On pourra afficher quelques solutions en fonction de la position x le long de l'ailette,pour voir l'évolution de sa température. Ce fichier csv sera alors composé de 3 colonnes, une première pour la position x_i , et les 2 autres pour la mesure de la température qui y correspond donnée par la solution exacte et la solution différence fini.

	Standard	Standard	Standard
1	xi	Tsolution	Texacte
2	0.000000e+00	4.121302e+01	4.121091e+01
3	4.000000e-06	4.120997e+01	4.120786e+01
4	8.000000e-06	4.120692e+01	4.120482e+01
5	1.200000e-05	4.120387e+01	4.120177e+01
6	1.600000e-05	4.120082e+01	4.119872e+01
7	2.000000e-05	4.119778e+01	4.119568e+01
8	2.400000e-05	4.119473e+01	4.119263e+01
9	2.800000e-05	4.119169e+01	4.118958e+01
10	3.200000e-05	4.118864e+01	4.118654e+01

Figure 5 Fichiers en csv

Résulat sur la console

La capture d'écran suivante est prise en éxecutant le programme en lisant un fichier “.cfg”.

```
Lx 0.04 Ly 0.0004 Lz 0.05
M 10000
Phi 125000
hc 200
Te 273
stationary 1
TFinal 300
N 600
Mx 0.05 My 0.01 Mz 0.03
```

En précisant “Stationnary” = 1, on a un régime statique et le programme exécute le cas stationnaire pour la résolution de AX =F. Le résultat affiché sur la console est le suivant.

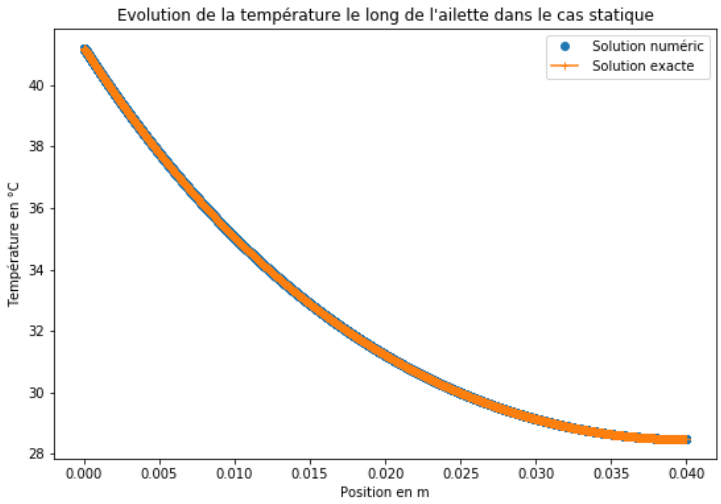
```
--------------Resolution statique----------

Voila les parametres de la simulation:

Geometry:  Lx 0.04 Ly 0.0004 Lz 0.05
Aire: 2e-05 Perimetres: 0.0252
Inertie : rho 2700 Cp 940
Conduction est : K= 164
Convection : Te 293 hc 200
Flux source :125000
le nombre de point de discritisation : 10000
-----------------------------------------------
La solution T(0):41.213 °C
La solution T(M/2):31.2173 °C
La solution T(M):28.4767 °C
-----------------------------------------------
-------------------Solution exacte-----------
La solution exacte T(0):41.2109 °C
La solution exacte T(M/2):31.216 °C
La solution exacte T(M):28.4752 °C
-------------------Exportation---------------
-----------------------------------------------
----------Résultat stationnaire exporté------
-----------------------------------------------
```

Dans ce cas statique la température au début, au milieu et au bout de l'ailette corréspond bien à la valeur donné par la solution exacte. Comme c'est illustré clairement par le graphe en bas développé en Pyhton.

```
data_stat = pd.read_csv('Statique_output.csv')
data_stat.head()
```

	xi	Tsolution	Texacte
0	0.000000	41.21302	41.21091
1	0.000004	41.20997	41.20786
2	0.000008	41.20692	41.20482
3	0.000012	41.20387	41.20177
4	0.000016	41.20082	41.19872

Comparaison



3. Modèle dynamique

Dans cette partie nous allons tenir en compte l'évolution en temps de la température modélisé dans l'équation (1) par $\frac{\partial T}{\partial t}(x_i, t_i)$.

3.1 Schéma implicite

Pour résoudre cette EDP, nous allons utiliser la métode des différences finies avec un shéma implicite car il est inconditionnelement stable. Nous commençons par échantillonnner l'intervalle de temps $[0, t_{final}]$ avec une discrétisation égulière composée de N+1 temps discret. Le pas de temps (i.e. la durée entre 2 temps discrets), sera donc constant et noté Δt. Ainsi, l'échantillonnage en temps est décrit par l'ensemble { t_n = nΔt, n = 0, ..., N }. On utilisera pour cette étude t final = 300s et N = 600. On note la Ti température discrète calculée au points x_i et à l'instant t_n . La dérivée en temps est discrétisée à l'aide d'un schéma d'Euler :

$$\frac{\partial T}{\partial t}(x_i, t_i) = \frac{T_i^{n+1} - T_i^n}{\Delta t} \qquad (7)$$

En adoptant un shéma implicite nous obtenons ainsi les équations discrètes du problème instationnaire définies sur tous les noeuds internes à chaque temps x , n = 1, ..., N :

$$\frac{\rho C_p}{\Delta t} X^{n+1} + A X^{n+1} = F + \frac{\rho C_p}{\Delta t} X^n \qquad (8)$$

Et les conditionsaux limites :

$$-\kappa \frac{\partial T}{\partial x}_{x=0} \approx -\kappa \frac{T_0^{n+1} - T_1^{n+1}}{h} = -\Phi_p \qquad (9)$$

$$-\kappa \frac{\partial T}{\partial x}_{x=0} \approx -\kappa \frac{T_M^{n+1} - T_{M-1}^{n+1}}{h} = -\Phi_p \qquad (10)$$

3.2 Système algébrique

En introduisant le terme d'inertie et en complétant le système algébrique du modèle stationnaire on obtient le système algébrique suivant:

$$MX^{n+1} = F + \frac{\rho C_p}{\Delta t} X^n \qquad (11)$$

Les conditions aux limites restent identiques à celles du modèle stationnaire donc on ne change pas les valeurs des coefficients $(a_i)_{i\in[1,M-1]}, et\,(c_i)_{i\in[1,M-1]}$. En revange, les coéfficients b doit inclure l'inertie et les valeurs de la solutionà l'instant précédent. De plus, il est néecessaire de dééfinir une solution initiale à t_0 . Dans ce projet on considéere que cette solution initiale est égale à la température ambiante, c'est-à-dire que $T^0{}_i = T_e$ pour $i \in [1, M-1]$.

Le problème discret est décrit par la résolution d'un système (11) avec M une matrice tridiagonale (taille (M+1) × (M +1)), F' le vecteur second membre (taille M+1) et le $Tn+1_i$ vecteurs temperature à l'instant n+1 et K de l'instant précédent. Le système algébrique se formule comme suite:

$$\underbrace{\begin{pmatrix} b_0 & c_0 & 0 & 0 & 0 & \cdots & 0 \\ a_1 & b_1+\frac{\rho C_p}{\Delta t} & c_1 & 0 & 0 & \cdots & 0 \\ 0 & a_2 & b_2+\frac{\rho C_p}{\Delta t} & c_2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & a_{M-1} & b_{M-1}+\frac{\rho C_p}{\Delta t} & c_{M-1} \\ 0 & \cdots & \cdots & 0 & 0 & a_M & b_M \end{pmatrix}}_{M} \underbrace{\begin{pmatrix} T^{n+1}{}_0 \\ T^{n+1}{}_1 \\ T^{n+1}{}_2 \\ \vdots \\ \vdots \\ T^{n+1}{}_{M-1} \\ T^{n+1}_M \end{pmatrix}}_{X^{n+1}} = \underbrace{\begin{pmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ \vdots \\ F_{M-1} \\ F_M \end{pmatrix}}_{F} + \frac{\rho C_p}{\Delta t} \underbrace{\begin{pmatrix} 0 \\ T^n{}_1 \\ T^n{}_2 \\ \vdots \\ \vdots \\ T^n{}_{M-1} \\ 0 \end{pmatrix}}_{X^n_{i\in[1,M-1]}}}_{F'}$$

3.3 Résolution du système algébrique

La résolution du système algébrique s $\frac{\rho C_p}{\Delta t} X^{n+1} + AX^{n+1} = F + \frac{\rho C_p}{\Delta t} X^n$ se fait à travers une boucle sur le pas de temps présenté ci-dessous.

- Initialiser T_0

For(n=0 à N) do

- Calculer T_{n+1} de T_n
    - Décomposition LU de M, Résolution LY = F' et Résolution UX = Y
- $T_n \leftarrow T_{n+1}$

End for

On résoud pour chaque pas de temps l'équation linéaire MX = F' de la même manière expliqué dans le chapitre précédent.

3.3.1 Classe matrice

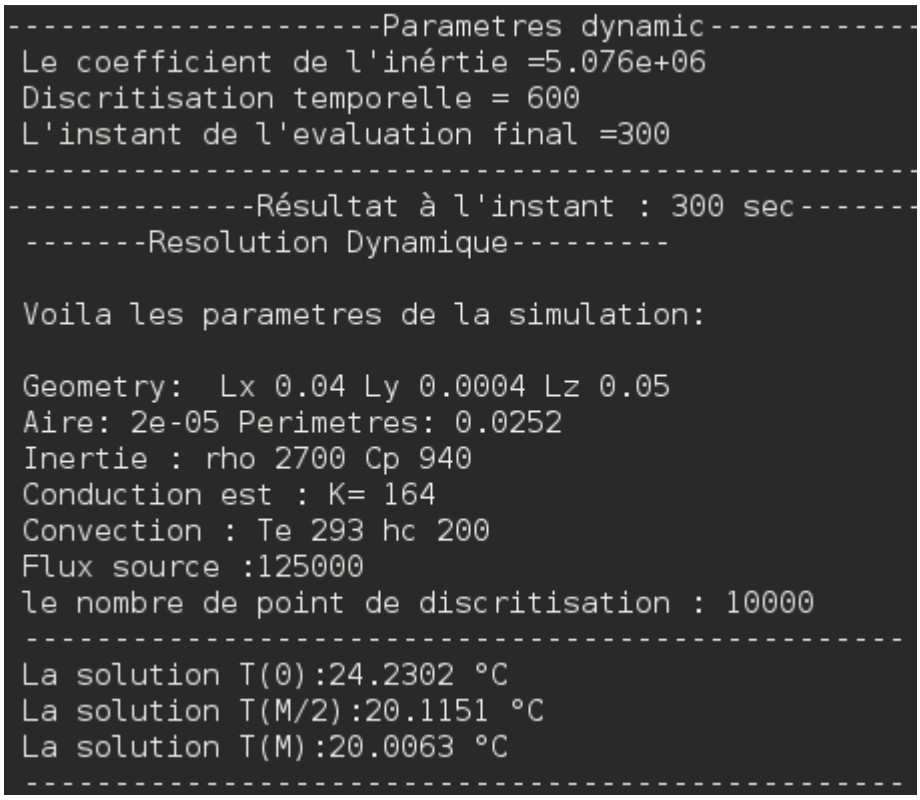
Cette classe comporte une option pour le cas dynamique qui modifie la diagonale de la matrice A pour devenir une matrice M adéquate au cas dynamique.

3.3.2 Classe Résolution

Cette classe comporte un cas traité a part dans la méthode “Solve_T()”pour le cas dynamique qui éxécute la boucle décrite en dessus et utilise la méme technique de résolution du modèle statique en l'occurence “la décomposition LU” suivie de l'algorithme de remonté et de déscente.

Résulat sur la console

En modifiant le fichier .cfg en méttant “Stationnary” = 0, on a le mode dynamique qui s'active et on obtient le résultat suivant à l'instant 300s.



Le résultat est issue du cas ou le flux source est constant, le cas du flux variable n'est pas traité dans ce rapport.

Cas non traités:

- Flux variable.
- Visualisation 3D.

Difficultés:

- Visualisation 3D.