

*Projet de Physique P6-3*  
*STPI/P6-3/2012 – 01*

**Diffusion de la chaleur dans une barre**



Étudiants :

Manuel BOURANE

Guillaume BREJUN

Guillaume FANCHON

Marie ROBCIS

Enseignant-responsable du projet :

Bernard GLEYSE



Date de remise du rapport : 18/06/2012

Référence du projet : STPI/P6-3/2012 – 01

Intitulé du projet : *diffusion de la chaleur dans une barre.*

Type de projet : *modélisation mathématique et implémentation informatique.*

Objectifs du projet :

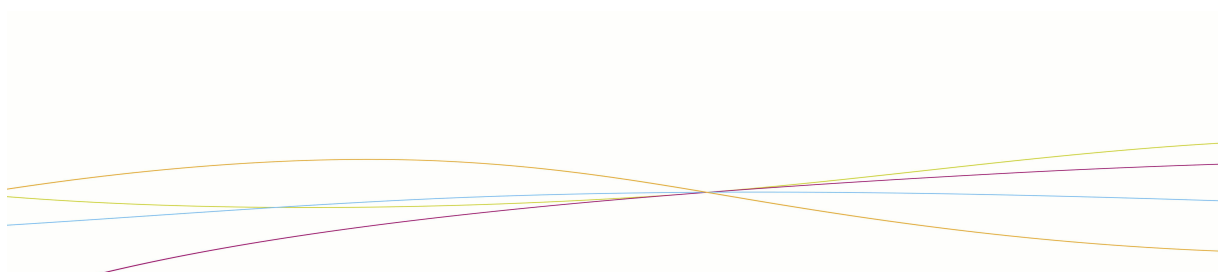
*Tout phénomène physique peut se mettre en équation. Les objectifs de ce projet ont été triples et mêlaient les mathématiques, la physique et l'informatique. Il a ainsi fallu tout d'abord établir une équation de la chaleur grâce à des principes fondamentaux de la physique. Ensuite, la résolution mathématique par deux méthodes différentes (implicite et explicite) a fait appel à des recherches approfondies concernant la discrétisation. Cette méthode nous amena ainsi à résoudre finalement un système linéaire. Enfin une fois la résolution de ce système effectuée, le dernier objectif était d'informatiser notre résolution et de permettre ainsi une comparaison des deux méthodes mathématiques étudiées. Le principal objectif était donc de modéliser l'échauffement d'une barre de métal par une source de chaleur.*

Mots-clefs du projet :

- **Modélisation**
- **Discrétisation**
- **Résolution de systèmes linéaires**
- **Implémentation informatique**

# TABLE DES MATIÈRES

- 1 Introduction .....6**
  
- 2 Méthodologie / Organisation du travail .....7**
  
- 3 Travail réalisé et résultat .....8**
  - 3.1 Établissement de l'équation de la chaleur .....8
  
  - 3.2 Résolution mathématique : méthode implicite 1D avec source de chaleur .....8
    - 3.2.1 Discrétisation de l'équation de la chaleur .....8
    - 3.2.2 Résolution avec la méthode de Cholesky .....10
  
  - 3.3 Résolution mathématique : méthode explicite 2D avec source de chaleur .....12
    - 3.3.1 Discrétisation de l'équation de la chaleur : cas bidimensionnel .....12
    - 3.3.2 Résolution du système linéaire .....15
  
  - 3.4 Implémentation informatique .....16
    - 3.4.1 Construction du programme.....16
    - 3.4.2 Comparaison des méthodes implicite et explicite .....16
  
- 4 Conclusion et perspectives .....19**
  
- 5 Bibliographie .....20**
  
- 6 Annexes .....21**



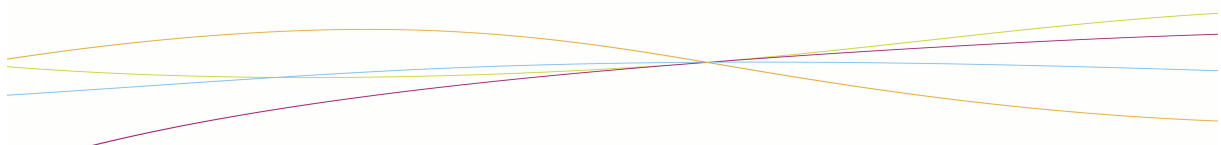
## 1. INTRODUCTION

Une part conséquente de la physique consiste à étudier les échanges thermiques entre des systèmes. On parle alors de diffusion thermique. Le but de ce projet est d'étudier cette diffusion thermique dans une barre. Ce mécanisme physique s'applique si l'on observe une différence de température dans la barre, c'est pourquoi il va falloir la chauffer et on observera un transfert d'énergie. L'étude de ce phénomène passera par une modélisation mathématique puis par sa résolution.

Les applications de l'étude de la diffusion thermique sont nombreux comme la métallurgie, l'électronique, la tribologie (étude des frottements)... C'est un des premiers problèmes physiques résolus de la physique moderne. En effet, nous devons l'étude de la diffusion thermique à un mathématicien et physicien français : Joseph Fourier. Né en 1768 et mort en 1830, il est un des plus renommé mathématicien français pour son œuvre sur les décompositions de fonctions périodiques en séries trigonométriques qui portent d'ailleurs son nom : les séries de Fourier. Mais il a également étudié la physique et plus particulièrement la propagation de la chaleur. Il s'est intéressé à ce principe en voulant justement modéliser la propagation de chaleur, donc la diffusion thermique, dans une barre de métal tout comme notre projet. En utilisant des principes simples de la thermodynamique, qui pourtant ne faisaient pas l'unanimité à l'époque, put trouver en 1811 l'équation de diffusion de la chaleur qui porte également son nom : l'équation de Fourier qui sera démontrer plus loin dans ce dossier.

La modélisation d'un échauffement dans une barre fait appel à différentes notions physiques. L'échauffement fait appel à la notion de chaleur alors que l'équation mathématique mesure une différentielle de température. Il ne faut donc pas confondre ces deux termes pourtant synonyme dans la vie de tous les jours. La chaleur est une énergie due, dans notre cas, aux mouvements des particules de la barre : on parle alors d'énergie interne. En chauffant notre barre, nous allons apporter de l'énergie à la barre qui va modifier l'énergie interne. Le but mathématique est donc de calculer cette modification. Quant à la température, utile dans l'équation elle n'a pas le même sens physique. En effet, la température n'est pas une énergie c'est une fonction qui détermine le degré d'agitation des particules au sein d'un système, ici dans notre barre. Même si chaleur et température sont intimement liés, elle n'ont d'ailleurs pas la même unité : le Kelvin (K) pour la température et le Joule (J) pour la chaleur. Enfin, nous pouvons relier ceux aspects physiques grâce à la capacité thermique, présent dans l'équation de Fourier, c'est une grandeur permettant de quantifier la possibilité qu'a un corps d'absorber ou restituer de l'énergie par échange thermique au cours d'une transformation pendant laquelle sa température varie, son unité est le J/K. Cette dernière grandeur fait donc le lien entre chaleur et température. Pour terminer, il ne faut absolument pas confondre température et chaleur au risque lors de la résolution mathématique d'avoir une interprétation erronée.

Enfin, ce projet physique est une opportunité de mêler deux matières, trop souvent cantonnées dans leur domaine dans l'enseignement, les mathématiques et la physique. Le choix sur ce projet a été une occasion de mettre en application physique des outils mathématiques étudiés en cours ou en projet mathématique par exemple. Les membres de ce projet, l'ont également choisi pour mener une véritable démarche scientifique : étude du phénomène, modélisation mathématique, résolution et implémentation informatique. Finalement, choisir ce projet nous a permis de construire une complète démarche scientifique qui sera utile dans notre futur carrière d'ingénieurs.



## 2. MÉTHODOLOGIE / ORGANISATION DU TRAVAIL

Comme tout commencement de projet, notre première étape fut de cerner le sujet et de comptabiliser les différents aspects que nous allions étudier. Après une première explication de M Bernard GLEYSE, nous nous sommes donc vite intéressés à l'équation de la chaleur. Le plus gros travail consista à comprendre les différentes techniques mathématiques pour résoudre cette équation et de décider lesquelles étaient les plus intéressantes. Il fut décidé de faire une méthode implicite et une autre explicite. Nous allons également résoudre les équations avec et sans production de chaleur. Ensuite chacun travailla sur sa partie et résolut l'équation avec la méthode de son choix. Enfin, l'aspect informatique commença avec l'implémentation informatique et la comparaison des deux méthodes utilisées.

Tableau récapitulatif des tâches de chacun :

Tâches à effectuer	Membre en charge	Semaine	
Familiarisation du sujet	TOUS	1 à 3	
Recherche des différentes équations	Bourane Manuel – Brejuin Guillaume – Robcis Marie	3 à 6	<u>Partie Physique</u>
Établissement de l'équation de la chaleur.	Fanchon Guillaume	4	
Choix et recherche des méthodes	TOUS	6	<u>Partie mathématique</u>
Méthode implicite en 1D	Fanchon Guillaume – Robcis Marie	6 à 8	
Méthode explicite en 2D	Bourane Manuel – Brejuin Guillaume	6 à 8	
Résolution mathématique	TOUS	8 à 10	
Implémentation et informatisation du problème	TOUS	10 et 11	<u>Partie informatique</u>
Comparaison des résultats	Fanchon Guillaume – Robcis Marie	11	
Écriture du rapport écrit	TOUS	12 et 13	
Rédaction de l'orale	TOUS	14	

Nous ne rencontrâmes que peu de problème le long de ce projet. Le plus difficile fut de cerner les méthodes mathématiques et de les comprendre. Le travail en groupe se déroula bien même s'il était difficile de faire une méthode mathématique tout en comprenant ce que faisaient les autres avec une méthode différente. Nous n'eûmes peu de problème de programmation car hélas, nous ne savions que coder en Pascal et nos talents de programmation limités mais avec l'aide des programmes de M Bernard GLEYSE que nous devons modifier, l'implémentation se passa bien. En conclusion, ce projet physique se déroula bien et nous pûmes entreprendre notre démarche scientifique sereinement.

### 3. TRAVAIL RÉALISÉ ET RÉSULTATS

#### 3.1 Établissement de l'équation de la chaleur.



Joseph Fourier,  
mathématicien à l'origine de  
l'équation de la chaleur

Pour étudier la diffusion de la chaleur dans une barre métallique, il faut d'abord ramener ce phénomène physique à une équation. C'est en 1811 que pour la première fois Joseph Fourier mis en place cette équation aux dérivées partielles et la résolut. Voici une méthode moderne d'établissement de l'équation de la chaleur avec des principes simples de physique.

Utilisons le premier principe de la thermodynamique :  $U(t+dt) = \delta W + \delta Q$

Vue que nous étudions un métal qui est un système isochore :  $\int -P dV = 0 = \delta W$

De plus on sait que :  $U(t) = \iiint \rho c T(t) dV$  avec

- $\rho$  = masse volumique en  $\text{kg/m}^3$
- $c$  = chaleur spécifique en  $\text{J/kg.K}$

On en déduit : 
$$U(t+dt) - U(t) = \iiint \rho c T(t+dt) - T(t) dV = \iiint \rho c \frac{dT}{dt} dV \quad (1)$$

On pose ensuite :

- $\varphi_q$  = le vecteur densité de flux de chaleur
- $P$  = densité volumique de source de chaleur par unité de temps en  $\text{W/m}^3$  (production de chaleur).

Ainsi on a : 
$$\delta Q = \pm \left( \iint \varphi_q \cdot d^2 S \right) dt + \iiint_v P dV dt$$

Or ici  $\varphi_q \cdot d^2 S > 0$  quand le flux est vers l'extérieur donc la variation de chaleur est négative dans le volume.

Ainsi 
$$\delta Q = - \iint \varphi_q \cdot d^2 S dt + \iiint_v P dV dt$$

Appliquons ensuite le théorème d'Ostrogradski :

$$\delta Q = - \iiint_v \text{divergence}(\varphi_q) + \iiint_v P dV dt$$

Ainsi en rassemblant (1) et (2) on obtient :

$$\iiint_v \rho c \frac{\partial T}{\partial t} dt dV = - \iiint_v \text{divergence}(\varphi_q) + \iiint_v P dV dt$$

Or nous avons l'expression  $\text{divergence}(\varphi_q) = -\rho c \frac{\partial T}{\partial t} + P$  existe car elle doit être valable pour tout volume  $V$ .

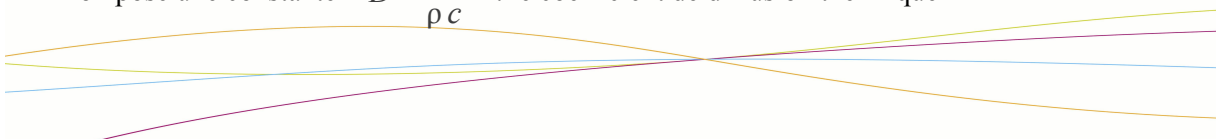
Utilisons maintenant la loi de Fourier :  $-\overrightarrow{k_{tr} \text{ grad}(T)}$  avec  $k_{tr}$  = coefficient de conductivité thermique.

On a également la propriété mathématique :  $\text{divergence}(\overrightarrow{\text{grad}}) = \nabla^2 = \Delta$

Ensuite on suppose que  $k_{tr}$  ne dépend pas des propriétés spatiales :

$$-k_{tr} \Delta T = -\rho c \frac{\partial T}{\partial t} + P \Leftrightarrow \frac{\rho c}{k_{tr}} \Delta T = \frac{P}{k_{tr}}$$

Enfin on pose une constante  $D = \frac{k_{tr}}{\rho c}$  : le coefficient de diffusion thermique



On obtient alors l'équation de Fourier, l'équation de la chaleur :  $\frac{\partial T}{\partial t} = D \Delta t + \frac{P}{\rho c}$

## 3.2 Résolution mathématique : méthode implicite 1D avec source de chaleur.

### 3.2.1 Discrétisation de l'équation de la chaleur.

Nous allons maintenant étudier le cas d'une barre 1D (suivant  $x$  et selon le temps seulement) avec une source de chaleur qui chauffe la barre.

L'équation de diffusion de la chaleur unidimensionnelle avec un terme source est donnée par :

$$\frac{\partial U}{\partial t} - a \frac{\partial^2 U}{\partial x^2} = f(x, t) \quad \text{avec : } U(x_n, t_n) = U_n^p \text{ et } f(x_n, t_n) = f_n^p \text{ fonction connue}$$

**Condition initiale :**  $U(x, 0) = U_0(x)$  , c'est-à-dire qu'à  $t = 0$ , on connaît la température dans toute la barre.

**Conditions limites :**  $U(0, t) = 0 \rightarrow$  condition au bord de la barre  
 $\frac{\partial U}{\partial x}(0, t) = 0$

On va ici utiliser la méthode du  $\theta$ -schéma dont la formule générale est :

$$\frac{U_{(n,p+1)} - U_{(n,p)}}{\partial t} - \theta a \frac{U_{(n-1,p+1)} - 2U_{(n,p+1)} + U_{(n+1,p+1)}}{\partial x^2} + (1-\theta)a \frac{U_{(n-1,p)} - 2U_{(n,p)} + U_{(n+1,p)}}{\partial x^2}$$

On sait que :

- $\theta = 0 \rightarrow$  Schéma explicite
- $\theta = 1 \rightarrow$  Schéma implicite
- $\theta = 0.5 \rightarrow$  Crank-Nicholson

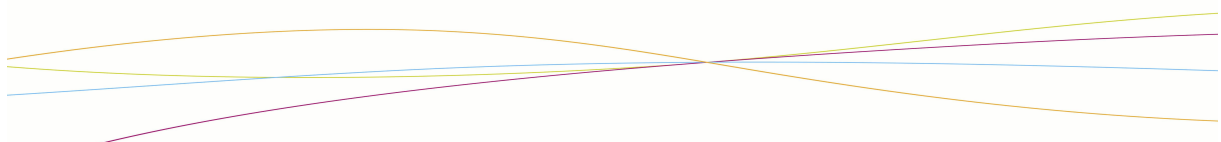
Nous nous intéressons ici à la méthode de **résolution mathématique implicite**, c'est-à-dire avec la valeur  $\theta = 1$ .

La formule du  $\theta$ -schéma pour la méthode implicite devient donc :

$$\frac{U_{(n,p+1)} - U_{(n,p)}}{\partial t} = a \frac{U_{(n-1,p+1)} - 2U_{(n,p+1)} + U_{(n+1,p+1)}}{\partial x^2}$$

On approxime  $U(x_n, t_n)$  grâce aux développements de Taylor :

- on a subdivisé l'axe de l'espace des  $x$  en  $N$  pas :  $h = \frac{l}{N}$  avec  $l$  la longueur totale de l'espace des  $x$  et  $h$  le pas.





- On a subdivisé l'espace du temps en P pas :  $\tau = \frac{t_p}{P}$  avec  $t_p$  la longueur de l'axe du temps et  $\tau$  le pas.

Les approximations de Taylor donnent :

$$U(x_n + h, t_p + \tau) = U(x_n, t_p) + h \frac{\partial U}{\partial x}(x_n, t_p) + \tau \frac{\partial U}{\partial t}(x_n, t_p) + \frac{h^2}{2} \frac{\partial^2 U}{\partial x^2}(x_n, t_p) + h^2 \varepsilon(h, \tau)$$

$$U(x_n - h, t_p + \tau) = U(x_n, t_p) - h \frac{\partial U}{\partial x}(x_n, t_p) + \tau \frac{\partial U}{\partial t}(x_n, t_p) + \frac{h^2}{2} \frac{\partial^2 U}{\partial x^2}(x_n, t_p) + h^2 \varepsilon'(h, \tau)$$

On ajoute les 2 approximations :

$$U(x_n + h, t_p + \tau) + U(x_n - h, t_p + \tau) = 2U(x_n, t_p) + 2\tau \frac{\partial U}{\partial t}(x_n, t_p) + \underbrace{h^2 \frac{\partial^2 U}{\partial x^2}(x_n, t_p) + h^2 \varepsilon''(h, \tau)}_{\text{négligeable}}$$

On a donc :

$$\frac{1}{h^2} (U(x_n + h, t_p + \tau) + U(x_n - h, t_p + \tau) - 2U(x_n, t_p)) = \frac{2\tau}{h^2} \frac{\partial U}{\partial t}(x_n, t_p) + \frac{\partial^2 U}{\partial x^2}(x_n, t_p) + \varepsilon''(h, \tau)$$

Donc, on peut approximer les termes de l'équation :

$$U(x_n, t_p + \tau) = U(x_n, t_p) + \tau \frac{\partial U}{\partial t}(x_n, t_p) \Leftrightarrow \frac{\partial U}{\partial t}(x_n, t_p) = \frac{1}{\tau} [U(x_n, t_p + \tau) - U(x_n, t_p)]$$

De même, on a :

$$U(x_{n+1}, t_p + \tau) + U(x_{n-1}, t_p + \tau) = 2U(x_n, t_p) + h^2 \frac{\partial^2 U}{\partial x^2}(x_n, t_p) \Leftrightarrow$$

$$\frac{\partial^2 U}{\partial x^2}(x_n, t_p) = \frac{1}{h^2} [U(x_{n+1}, t_p + \tau) - 2U(x_n, t_p) + U(x_{n-1}, t_p + \tau)]$$

Dans l'équation de la chaleur, on a :

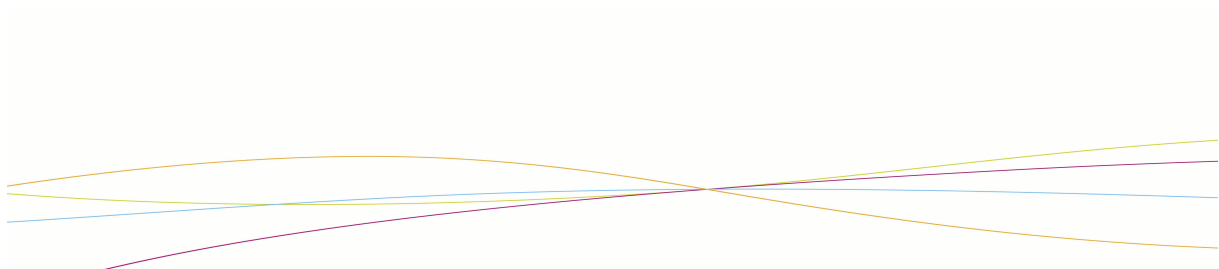
$$\frac{1}{\tau} [U(x_n, t_p + \tau) - U(x_n, t_p)] - \frac{a}{h^2} [U(x_{n+1}, t_p + \tau) - 2U(x_n, t_p) + U(x_{n-1}, t_p + \tau)] = f(x_n, t_p + \tau)$$

$$\Leftrightarrow \frac{1}{\tau} (U_n^{(p+1)} - U_n^p) - \frac{a}{h^2} (U_{n+1}^{(p+1)} - 2U_n^p + U_{n-1}^{(p+1)}) = f_n^{(p+1)}$$

$$\Leftrightarrow U_n^{(p+1)} = \frac{a\tau}{h^2} (U_{n+1}^{(p+1)} - 2U_n^p + U_{n-1}^{(p+1)}) + \tau f_n^{(p+1)} + U_n^p \Rightarrow \text{formule de récurrence d'une suite}$$

On peut mettre cette formule sous forme de système matricielle. On a :

- $n = 1, \dots, n-1$
  - $U_0^p = 0$
  - $U_n^0 = U_0(x_n)$
  - $U_n^p - U_{n-1}^p = 0$
- } Conditions initiales



Avec :  $U^{(p+1)} = \begin{pmatrix} U_1^{(p+1)} \\ U_2^{(p+1)} \\ \vdots \\ U_{(n-1)}^{(p+1)} \end{pmatrix}$  et  $AU^{(p+1)} = b$  avec  $U_0$  donné,

$$U_0(x) = x(1-x)^2$$

Pour les valeurs, on prend :

- $t_p = \frac{1}{2}$  ,  $\tau = \frac{1}{8}$  ,  $h = \frac{1}{3}$

**Donc, on a finalement le système matricielle suivant :**

$$\begin{pmatrix} \frac{1}{\tau} + \frac{2a}{h^2} & \frac{-a}{h^2} & 0 \dots \\ (\frac{-a}{h^2}) \dots & (\frac{1}{\tau} + \frac{2a}{h^2}) \dots & (\frac{-a}{h^2}) \dots \\ 0 & \frac{-a}{h^2} & \frac{1}{\tau} + \frac{a}{h^2} \end{pmatrix} \begin{pmatrix} U_1^{(p+1)} \\ U_2^{(p+1)} \\ \vdots \\ U_{(n-1)}^{(p+1)} \end{pmatrix} = \underbrace{\begin{pmatrix} f_1^{(p+1)} + \frac{U_1^p}{\tau} \\ f_2^{(p+1)} + \frac{U_2^p}{\tau} \\ \vdots \\ f_{(n-1)}^{(p+1)} + \frac{U_{(n-1)}^p}{\tau} \end{pmatrix}}_b$$

### 3.2.2. Résolution grâce à la méthode de Cholesky.

Une fois notre système obtenu grâce à une méthode implicite, il faut maintenant la résoudre. Pour cela nous allons utiliser la méthode de Cholesky : une méthode fiable et pratique pour peu qu'elle soit applicable. En effet notre matrice A doit être définie positive et symétrique mais c'est le cas ici. Il suffit d'observer la symétrie et de calculer le det(A) pour s'apercevoir que A est bien positive. Résolvons donc notre système linéaire avec cette méthode.

Utilisons le théorème de Cholesky : toute matrice définie positive et symétrique peut se décomposer en deux matrices triangulaires transposées l'une de l'autre tel que  $A=LL^t$ . Maintenant nous devons trouver ces deux matrices.



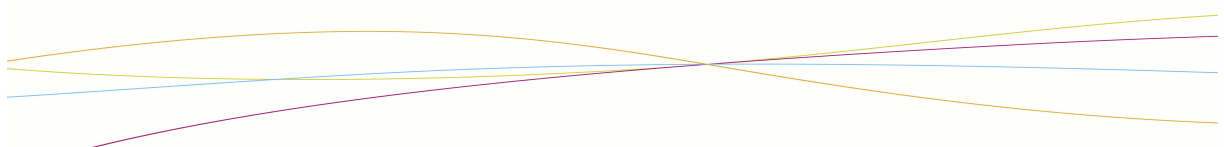
André-Louis Cholesky, inventeur de la factorisation de Cholesky

Notations des lignes i et colonnes j des matrices :

- $A = (a_{i,j})$
- $L = (l_{i,j})$

Comme  $A=LL^t$ , on en déduit que :  $a_{i,j} = \sum_{k=1}^{n-1} l_{i,k} l_{k,j}$  , on peut ainsi résoudre pour trouver L et  $L^t$ :

- $j=1 \quad a_{1,1} = l_{1,1} l_{1,1} \leftrightarrow l_{1,1} = \sqrt{a_{1,1}} = \sqrt{\frac{1}{\tau} + \frac{2a}{h^2}}$



- $j=2 \quad a_{1,2}=l_{1,1}l_{2,1} \leftrightarrow l_{2,1}=0$
- ...
- $j=i \quad a_{i,i}=l_{i,1}l_{i,1}+l_{i,2}l_{i,2}+\dots+l_{i,n}l_{i,n} \leftrightarrow l_{i,i}=\sqrt{a_{i,i}-l_{i,1}l_{i,1}-\dots-l_{i,j-1}l_{i,j-1}} \leftrightarrow$   
 $l_{i,i}=\sqrt{\frac{1}{\tau}+\frac{2a}{h^2}-\left(\sqrt{\frac{1}{\tau}+\frac{2a}{h^2}}\right)^2-\dots}=\sqrt{\frac{a}{h^2}}$
- $j=i+1 \quad a_{i,i+1}=l_{i,1}l_{1,i+1}+\dots+l_{i,i}l_{i+1,i} \leftrightarrow l_{i,i+1}=\frac{a_{i,i+1}-l_{i,1}l_{1,i+1}-\dots-l_{i,i-1}l_{i-1,i+1}}{l_{i,i}}=\sqrt{\frac{a}{h^2}}$
- etc.

Donc après tous les calculs intermédiaires, on obtient :

$$L = \begin{pmatrix} \sqrt{\frac{1}{\tau} + \frac{2a}{h^2}} & 0 \dots & 0 \\ 0 & \ddots & 0 \\ \vdots & & \vdots \\ 0 \dots & 0 & \sqrt{\frac{a}{h^2}} \end{pmatrix} \quad \text{et on peut ainsi en déduire facilement } L^t.$$

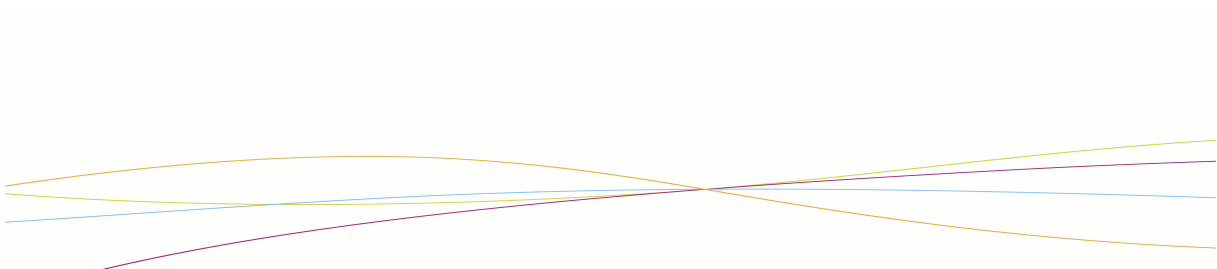
Ensuite, il faut définir le vecteur auxiliaire Y, tel que  $L^t Y = B$  :

$$L \times \begin{pmatrix} y_1 \\ \vdots \\ y_2 \end{pmatrix} = \begin{pmatrix} f_1^{(p+1)} + \frac{U_1^p}{\tau} \\ f_2^{(p+1)} + \frac{U_2^p}{\tau} \\ \vdots \\ f_{(n-1)}^{(p+1)} + \frac{U_{(n-1)}^p}{\tau} \end{pmatrix}$$

On calcule ensuite les termes de ce vecteur intermédiaire :

- $\sqrt{\frac{1}{\tau} + \frac{2a}{h^2}} \times y_1 = f_1^{p+1} + \frac{U_1^p}{\tau} \Leftrightarrow y_1 = \frac{f_1^{p+1} + \frac{U_1^p}{\tau}}{\sqrt{\frac{1}{\tau} + \frac{2a}{h^2}}}$
- ...
- $\sqrt{\frac{a}{h^2}} \times y_{n-1} = f_{n-1}^{p+1} + \frac{U_{n-1}^p}{\tau} \Leftrightarrow y_n = \frac{f_{n-1}^{p+1} + \frac{U_{n-1}^p}{\tau}}{\sqrt{\frac{a}{h^2}}}$

Finalement on peut trouver le vecteur x car on sait que  $Lx=Y$  :



$$\begin{pmatrix} \sqrt{\frac{1}{\tau} + \frac{2a}{h^2}} & 0 \dots & 0 \\ 0 & \ddots & 0 \\ \vdots & & \vdots \\ 0 \dots & 0 & \sqrt{\frac{a}{h^2}} \end{pmatrix} \times \begin{pmatrix} U_1^{(p+1)} \\ U_2^{(p+1)} \\ \vdots \\ U_{(n-1)}^{(p+1)} \end{pmatrix} = \begin{pmatrix} \frac{f_1^{p+1} + \frac{U_1^p}{\tau}}{\sqrt{\frac{1}{\tau} + \frac{2a}{h^2}}} \\ \vdots \\ \frac{f_{n-1}^{p+1} + \frac{U_{n-1}^p}{\tau}}{\sqrt{\frac{a}{h^2}}} \end{pmatrix}$$

Donc :

- $\sqrt{\frac{1}{\tau} + \frac{2a}{h^2}} U_1^{p+1} = \frac{f_1^{p+1} + \frac{U_1^p}{\tau}}{\sqrt{\frac{1}{\tau} + \frac{2a}{h^2}}} \Leftrightarrow U_1^{p+1} = \frac{f_1^{p+1} + \frac{U_1^p}{\tau}}{\frac{1}{\tau} + \frac{2a}{h^2}}$
- ...
- $\sqrt{\frac{a}{h^2}} U_{n-1}^{p+1} = \frac{f_{n-1}^{p+1} + \frac{U_{n-1}^p}{\tau}}{\sqrt{\frac{a}{h^2}}} \Leftrightarrow U_{n-1}^{p+1} = \frac{f_{n-1}^{p+1} + \frac{U_{n-1}^p}{\tau}}{\frac{a}{h^2}}$

Enfin, nous trouvons notre vecteur solution :

$$x = U^{p+1} = \begin{pmatrix} \frac{f_1^{p+1} + \frac{U_1^p}{\tau}}{\frac{1}{\tau} + \frac{2a}{h^2}} \\ \vdots \\ \frac{f_{n-1}^{p+1} + \frac{U_{n-1}^p}{\tau}}{\frac{a}{h^2}} \end{pmatrix}$$

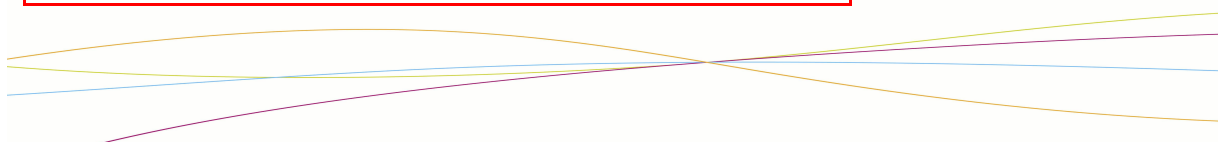
Nous avons ainsi trouver ce vecteur colonne (cas 1D) qui correspond ainsi à la variation de température.

### 3.3 Résolution mathématique : méthode explicite 2D avec source de chaleur.

#### 3.3.1 Discrétisation de l'équation de la chaleur : cas bidimensionnel

$\forall (x, y) \in ]0, L[ \times ]0, M[$  on considère l'équation (1) :

$$\frac{\partial (T(x, y, t))}{\partial t} - a \left( \frac{\partial^2 (T(x, y, t))}{\partial^2 x} + \frac{\partial^2 (T(x, y, t))}{\partial^2 y} \right) = f(x, y, t)$$



associée aux conditions suivantes :  $T(0,y,t) = 0$       Conditions limites  
 $T(x,0,t) = 0$

$u^p_{N,M} - u^p_{N,M-1} = 0$       Conditions initiales  
 $u^p_{N,M} - u^p_{N-1,M} = 0$

On pose  $\forall p, t_{p+1} - t_p = \tau \quad \forall n, x_{n+1} - x_n = h \quad \forall m, y_{m+1} - y_m = k$

Pour la suite, pour des questions de simplification on va considérer que  $k = h$  au moment de la résolution de la matrice et ainsi se placer dans le cas d'une barre carrée et non plus rectangulaire.

Avec l'approximation de  $T(x, y, t)$  par une suite :  $T(x_n, y_m, t_p) \approx u^p_{n,m}$

On résoudra l'équation (1) par méthode implicite, ce qui nous amène à nous placer au rang  $p+1$ , et par l'approximation des dérivées de la fonction, on obtient :

$$T(x_n+h, y_n+k, t_{p+2}) = T(x_n, y_n, t_{p+1}) + h \frac{\partial T}{\partial x}(x_n, y_n, t_{p+1}) + \tau \frac{\partial T}{\partial t}(x_n, y_n, t_{p+1}) + k \frac{\partial T}{\partial y}(x_n, y_n, t_{p+1}) \\ + \frac{h^2}{2} \frac{\partial^2 T}{\partial^2 x}(x_n, y_n, t_{p+1}) + \frac{k^2}{2} \frac{\partial^2 T}{\partial^2 y}(x_n, y_n, t_{p+1}) + k^2 \varepsilon(h, k, \tau) + h^2 \varepsilon(h, k, \tau)$$

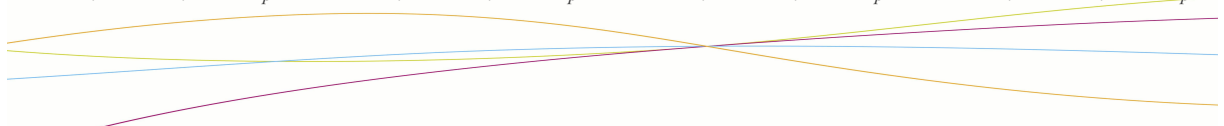
$$T(x_n+h, y_m-k, t_{p+2}) = T(x_n, y_m, t_{p+1}) + h \frac{\partial T}{\partial x}(x_n, y_m, t_{p+1}) + \tau \frac{\partial T}{\partial t}(x_n, y_m, t_{p+1}) - k \frac{\partial T}{\partial y}(x_n, y_m, t_{p+1}) \\ + \frac{h^2}{2} \frac{\partial^2 T}{\partial^2 x}(x_n, y_m, t_{p+1}) + \frac{k^2}{2} \frac{\partial^2 T}{\partial^2 y}(x_n, y_m, t_{p+1}) + k^2 \varepsilon(h, k, \tau) + h^2 \varepsilon(h, k, \tau)$$

$$T(x_n-h, y_m+k, t_{p+2}) = T(x_n, y_m, t_{p+1}) - h \frac{\partial T}{\partial x}(x_n, y_m, t_{p+1}) + \tau \frac{\partial T}{\partial t}(x_n, y_m, t_{p+1}) + k \frac{\partial T}{\partial y}(x_n, y_m, t_{p+1}) \\ + \frac{h^2}{2} \frac{\partial^2 T}{\partial^2 x}(x_n, y_m, t_{p+1}) + \frac{k^2}{2} \frac{\partial^2 T}{\partial^2 y}(x_n, y_m, t_{p+1}) + k^2 \varepsilon(h, k, \tau) + h^2 \varepsilon(h, k, \tau)$$

$$T(x_n-h, y_m-k, t_{p+2}) = T(x_n, y_m, t_{p+1}) - h \frac{\partial T}{\partial x}(x_n, y_m, t_{p+1}) + \tau \frac{\partial T}{\partial t}(x_n, y_m, t_{p+1}) - k \frac{\partial T}{\partial y}(x_n, y_m, t_{p+1}) \\ + \frac{h^2}{2} \frac{\partial^2 T}{\partial^2 x}(x_n, y_m, t_{p+1}) + \frac{k^2}{2} \frac{\partial^2 T}{\partial^2 y}(x_n, y_m, t_{p+1}) + k^2 \varepsilon(h, k, \tau) + h^2 \varepsilon(h, k, \tau)$$

Ainsi, en sommant le tout on obtient :

$$T(x_n+h, y_n+k, t_{p+2}) + T(x_n-h, y_n-k, t_{p+2}) + T(x_n+h, y_n-k, t_{p+2}) + T(x_n-h, y_n+k, t_{p+2})$$



$$= 4T(x_n, y_m, t_{p+1}) + 4\tau \frac{\partial T}{\partial t}(x_n, y_m, t_{p+1}) + 2h^2 \frac{\partial^2 T}{\partial^2 x}(x_n, y_m, t_{p+1}) + 2k^2 \frac{\partial^2 T}{\partial^2 y}(x_n, y_m, t_{p+1})$$

$$+ k^2 \varepsilon(h, k, \tau) + h^2 \varepsilon(h, k, \tau)$$

On en déduit alors :

$$\frac{\partial T}{\partial t}(x_n; t_p) \approx \frac{1}{\tau} [T(x_n; t_{p+1}) - T(x_n; t_p)]$$

$$\frac{\partial^2 T}{\partial x^2}(x_n; t_p) = \frac{1}{h^2} [2T(x_n; t_p + \tau) - 2T(x_n; t_p) - 2\tau \frac{\partial T}{\partial t}(x_n; t_p)]$$

En remplaçant, puis par approximation, on obtient alors :

$$\frac{\partial^2 T}{\partial x^2}(x_n, t_p) \approx \frac{1}{h^2} [T(x_{n+1}, y_m, t_p) - 2T(x_n, y_m, t_p) + T(x_{n-1}, y_m, t_p)]$$

De même pour la dérivée en y :

$$\frac{\partial^2 T}{\partial y^2}(x_n, t_p) \approx \frac{1}{h^2} [T(x_n, y_{m+1}, t_p) - 2T(x_n, y_m, t_p) + T(x_n, y_{m-1}, t_p)]$$

On remplace alors ces termes dans l'équation de la chaleur :

$$\frac{\partial T}{\partial t}(x, y, t) - a \left[ \frac{\partial^2 T}{\partial x^2}(x, y, t) - \frac{\partial^2 T}{\partial y^2}(x, y, t) \right] = f(x, y, t)$$

Et on associe alors les termes comme suit :

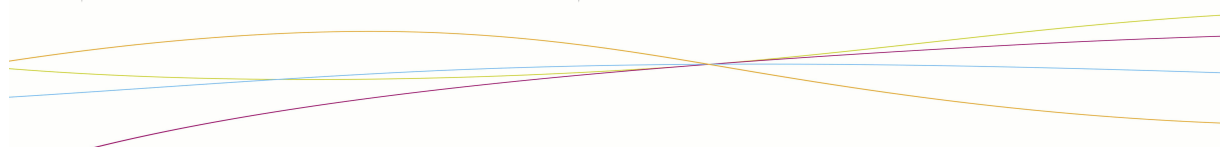
$$T(x_n, y_m, t_p) = u_{n,m}^p$$

On obtient donc l'équation suivante :

$$\frac{1}{\tau} u_{n,m}^{p+1} - \frac{a}{h^2} [u_{n+1,m}^{p+1} + u_{n-1,m}^{p+1} + u_{n,m+1}^{p+1} + u_{n,m-1}^{p+1} - 4u_{n,m}^{p+1}] = f_{n,m}^{p+1} + \frac{1}{\tau} u_{n,m}^p$$

Il s'agit d'une équation de la forme AU=B, on a A et U de la forme :

$$A = \begin{pmatrix} A_1 & C & 0 & \dots & \dots & \dots & 0 & 0 & 0 \\ C & A_2 & C & 0 & \dots & \dots & \dots & 0 & 0 \\ 0 & C & A_2 & C & 0 & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & 0 & C & A_2 & C & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 & C & A_2 & C \\ 0 & 0 & 0 & \dots & \dots & \dots & 0 & C & A_2 \end{pmatrix} \quad \text{et } U = \begin{pmatrix} u_{1,1} \\ u_{1,2} \\ \dots \\ u_{1,N-1} \\ u_{2,1} \\ \dots \\ u_{N-1,M-1} \end{pmatrix}$$



où A est une matrice par bloc, avec  $A_1$ ,  $A_2$  et C de la forme :

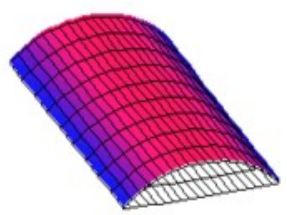
$$A_1 = \begin{pmatrix} a_1 & c & 0 & \dots & \dots & \dots & 0 & 0 & 0 \\ c & a_2 & c & 0 & \dots & \dots & \dots & 0 & 0 \\ 0 & c & a_2 & c & 0 & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & 0 & c & a_2 & c & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 & c & a_2 & c \\ 0 & 0 & 0 & \dots & \dots & \dots & 0 & c & a_1 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} a_2 & c & 0 & \dots & \dots & \dots & 0 & 0 & 0 \\ c & a_3 & c & 0 & \dots & \dots & \dots & 0 & 0 \\ 0 & c & a_3 & c & 0 & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & 0 & c & a_3 & c & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 & c & a_3 & c \\ 0 & 0 & 0 & \dots & \dots & \dots & 0 & c & a_2 \end{pmatrix}$$

$$C = \begin{pmatrix} c & 0 & 0 & \dots & \dots & \dots & 0 & 0 & 0 \\ 0 & c & 0 & 0 & \dots & \dots & \dots & 0 & 0 \\ 0 & 0 & c & 0 & 0 & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & 0 & 0 & c & 0 & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 & 0 & c & 0 \\ 0 & 0 & 0 & \dots & \dots & \dots & 0 & 0 & c \end{pmatrix}$$

Par identification, on obtient  $a_i = \frac{1}{\tau} + (1+i)\frac{a}{h^2}$  et  $c = -\frac{a}{h^2}$

### 3.3.2 Résolution du système linéaire

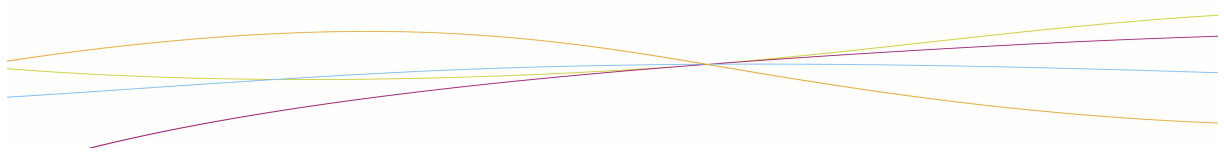


Modélisation de la propagation de la chaleur dans une barre

Nous obtenons ainsi comme dans l'exemple en 1D avec une méthode implicite, un système linéaire à résoudre. On remarque également que la matrice peut être décomposé en deux matrices symétriques. On peut donc également résoudre avec la méthode de Cholesky comme expliquer précédemment.

Une différence notable concerne la structure des matrices  $A_i$ . Ces matrices contiennent elles-mêmes une autre matrice C. Il faudra donc utiliser la méthode de Cholesky plusieurs fois. C'est grâce également à cet exemple que l'on voit le principal avantage de la méthode de Cholesky sur d'autres méthodes de résolution de système linéaire comme le pivot de Gauss par exemple. En effet, la méthode de Cholesky est plus pratique, pour peu qu'elle soit applicable, car elle limite les calculs. Le nombre de calcul pour Cholesky correspond à la dimension de la matrice : n. Quant au pivot de Gauss, le nombre de calculs correspond à  $n^3$ . Il est donc tout à fait raisonnable d'appliquer la méthode de Cholesky pour cet exemple.

Il est donc tout aussi important en physique de choisir sa méthode mathématique pour rendre les calculs simples et surtout limités leurs nombres car un nombre de calcul trop grand entraîne des erreurs d'arrondis.



### 3.4 Implémentation informatique

#### 3.4.1 Construction du programme

La physique moderne s'est d'abord construit grâce à des expériences et des observations. Mais de nos jours, la physique nécessite des expériences complexes et souvent très coûteuses. Il est ainsi primordiale d'abord de modéliser l'expérience et les résultats correspondants puis ensuite de faire l'expérience. Dans ce projet physique nous devons effectivement suivre cette démarche.

Le programme consiste à calculer les solutions des systèmes linéaires pour un système en une dimension, nous avons ainsi deux variables  $x$  et  $t$ . Avant de construire le programme, il est utile de décider dans quel langage le coder. Le problème est que nous ne connaissons que le Pascal ainsi notre choix se porta vite sur celui-ci.

Après avoir tenter quelques combinaisons de code, M Bernard Gleyse nous donna le code brut auquel il fallut modifier quelques lignes afin de correspondre avec nos données expérimentales et nos conditions initiales. Par exemple, il fallait qu'à  $t=0$ , la solution soit nulle ou que par exemple correspondre avec notre coefficient thermique :  $a$ .

```

geany_run_script.sh
u( 3, 4)= 7,68549185408241E-002
u( 0, 5)= 0,00000000000000E+000
u( 1, 5)= 9,57838910555823E-002
u( 2, 5)= 6,33315169572372E-002
u( 3, 5)= 6,33315169572372E-002
u( 0, 6)= 0,00000000000000E+000
u( 1, 6)= 7,89839025393757E-002
u( 2, 6)= 4,73101663298937E-002
u( 3, 6)= 4,73101663298937E-002
u( 0, 7)= 0,00000000000000E+000
u( 1, 7)= 6,22261986591393E-002
u( 2, 7)= 3,00547705731401E-002
u( 3, 7)= 3,00547705731401E-002
u( 0, 8)= 0,00000000000000E+000
u( 1, 8)= 4,61319217757219E-002
u( 2, 8)= 1,29846660757058E-002
u( 3, 8)= 1,29846660757058E-002
u( 0, 9)= 0,00000000000000E+000
u( 1, 9)= 3,13040651028549E-002
u( 2, 9)=-3,13294666088941E-003
u( 3, 9)=-3,13294666088941E-003
la difference maximum entre la solution approcheet la solution exacte est 8,706
66588328034E-002 en U 3 2
    
```

Capture d'écran lors de l'exécution de notre programme

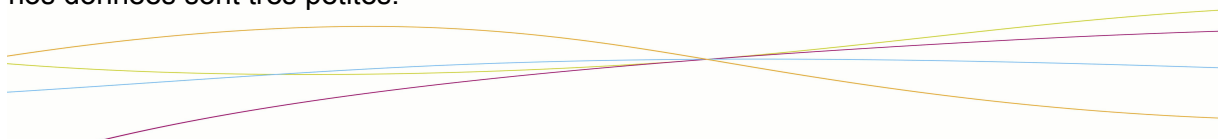
Une fois le programme construit, pour des valeurs de  $t$  et  $x$  de 0 à 9, il calcule les solutions  $U(x,t)$ . Mais ce qui est également intéressant est que le programme calcule également la solution exacte et estime ainsi la différence entre nos solutions approchées par les méthodes implicite et explicite et la solution exacte. Ainsi on remarque que la différence n'est que de :  $8.70666588328034E-002$ . Soit un écart assez faible, le programme nous donne également en quel point ce maximum d'écart est établi, en  $U(3,2)$ .

Finalement, notre programme nous a permis de vérifier que notre modèle mathématique correspond effectivement aux résultats exactes. Il nous reste maintenant à déterminer quel écart y-a-t-il entre les méthodes explicites et implicites et de conclure quelle méthode semble la plus appropriée.

#### 3.4.2 Comparaison des méthodes implicites et explicites

Nous allons ainsi comparer dans cette partie les méthodes implicites et explicites grâce au programme. En effet, le code pour ces deux méthodes est le même sauf une ligne (voir annexe) car la discrétisation mathématique n'est pas la même. D'or et déjà on s'aperçoit qu'à première vue les résultats semblent très proches mais pour s'en assurer, nous allons fixer des grandeurs et tracer des graphiques pour visualiser les différences.

On va tracer deux graphiques. Dans l'un nous allons fixer le temps et faire varier les  $x$  et dans l'autre faire varier les  $t$  et fixer  $x$ . Le logiciel utilisé a été Matlab car étudié en M8 et permet de visualiser parfaitement des courbes. Ce logiciel n'est pas à la portée de n'importe quel utilisateur toutefois notre choix s'est vite arrêté sur lui car le problème majeur est que nos données sont très petites.





Ainsi, par exemple avec un tableur ou avec Geogebra, les chiffres significatifs ne vont que jusqu'à 3 chiffres après la virgule ce qui n'est pas assez. Mais avec Matlab, nous pouvons garder tous nos chiffres significatifs soient 14 chiffres. Nous aurions pu également utiliser Maple mais un peu moins pratique pour tracer des courbes et les visualiser en plein écran.

Voici notre tableau de données et les lignes de code fournis sous Matlab :

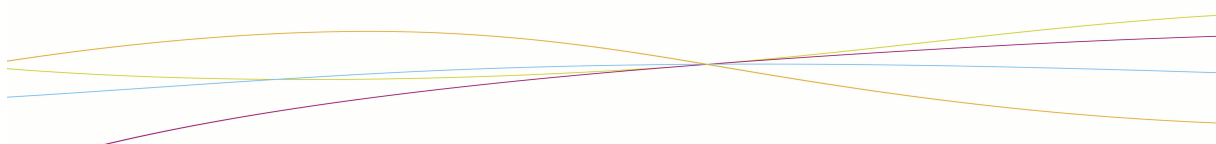
<b>x</b>	<b>Uimplicite (x,8)</b>	<b>Uexplicite (x,8)</b>	<b>F (solution exacte)</b>
0	0	0	0
1	2.48117320650020E-002	4.61319217757219E-002	3.58114785144129E-002
2	-2.02853745541023E-002	1.29846660757058E-002	-0.4785155362147423E-002
3	-2.02853745541023E-002	1.29846660757058E-002	1.05845698452124E-002

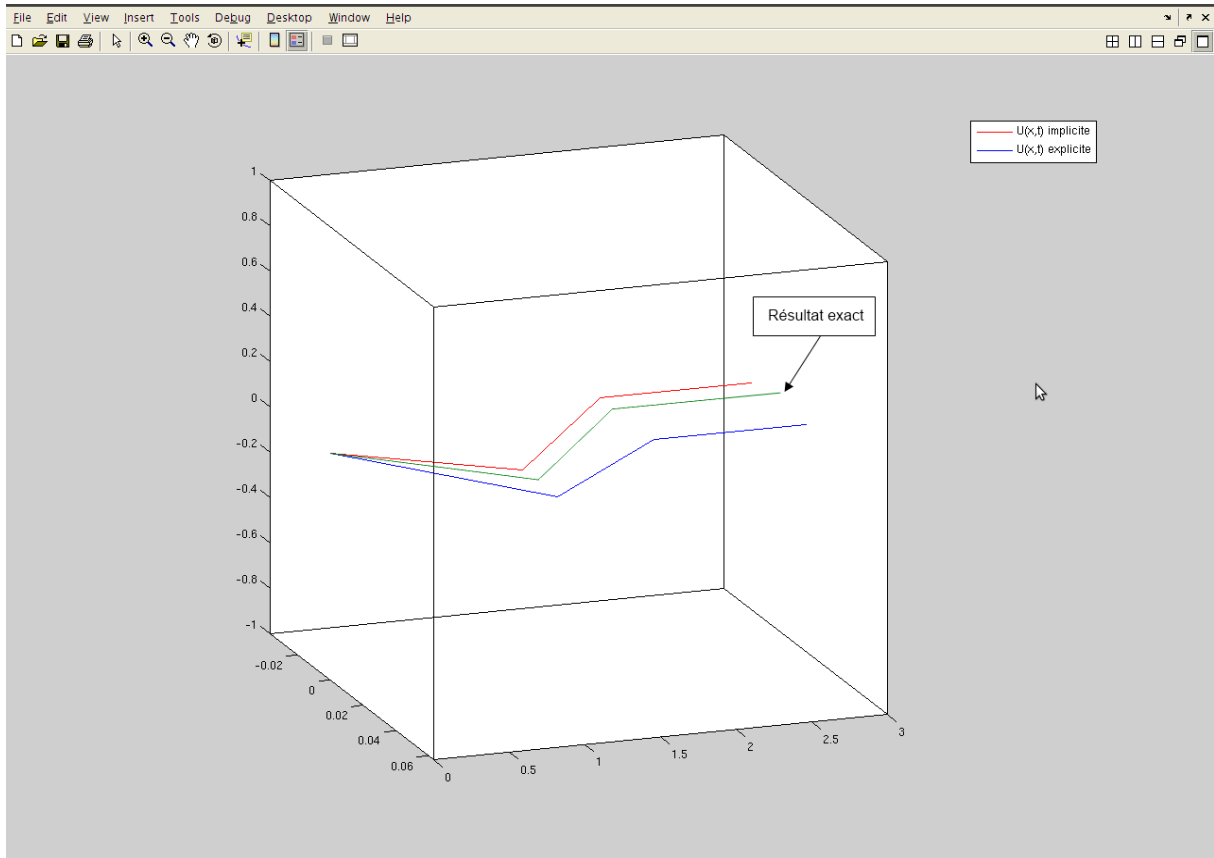
et

<b>t</b>	<b>Uimplicite (1,t)</b>	<b>Uexplicite (1,t)</b>	<b>F (solution exacte)</b>
0	1.48148148148148E-001	1.48148148148148E-001	1.48148148148148E-001
1	9.21574459647036E-002	1.39917695473251E-001	6.5547868951124E-002
2	8.03690888119954E-002	1.25079256664292E-001	5.03548124569453E-002
3	6.85185185185185E-002	1.23902418228397E-001	3.78213665478421E-002

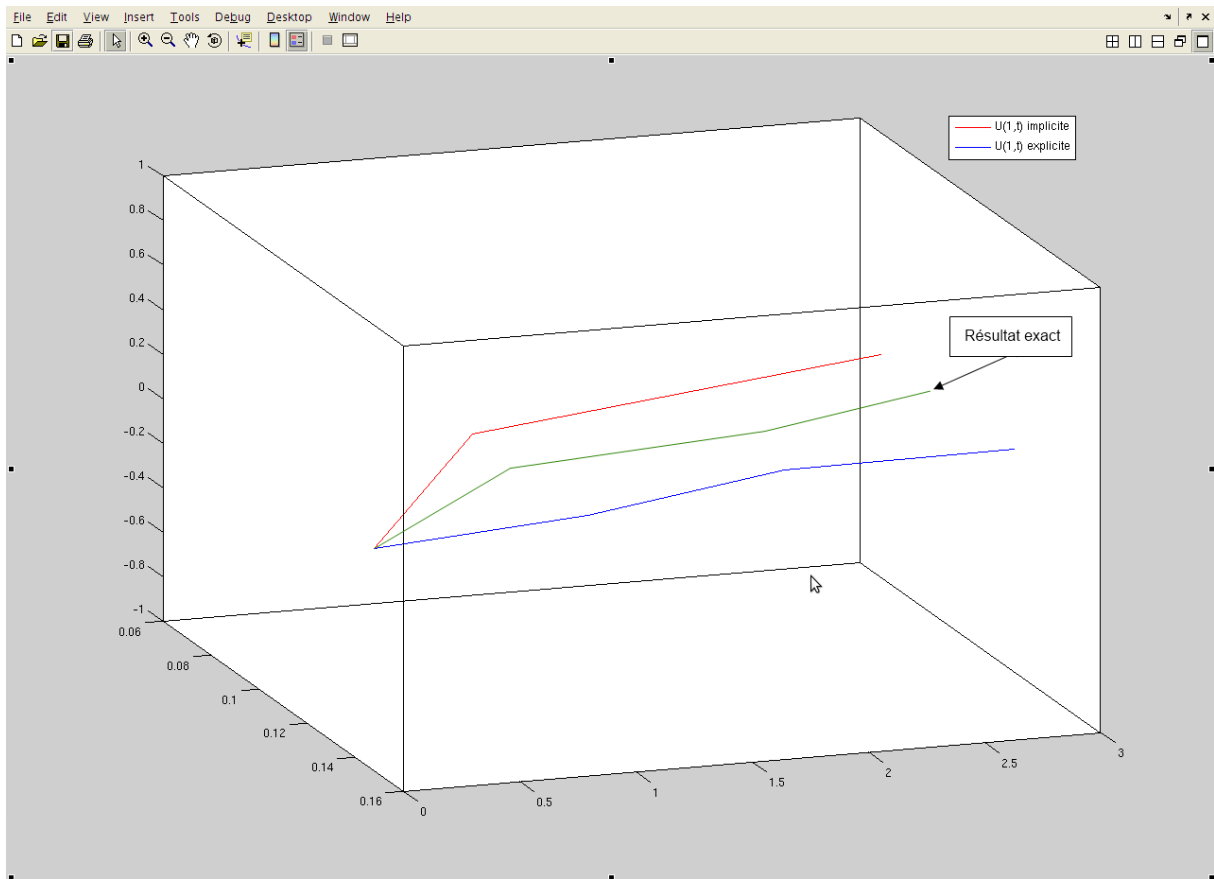
Exploitions maintenant nos graphiques. Sur le premier où t est fixé, on s'aperçoit que les deux courbes suivent bien les mêmes « pentes ». De plus notre solution exacte se situe bien au milieu de nos deux approximations. On peut donc valider nos résultats car nos deux courbes suivent les mêmes évolutions, seules les valeurs diffèrent quelque peu. De plus, notre solution exacte se situant entre les deux courbes indiquent bien que les valeurs trouvées ci-dessus sont de bonnes approximations. Ensuite sur le deuxième graphique, on peut tirer les mêmes conclusions : nos résultats semblent bons. Toutefois, on peut noter une différence plus nette entre nos résultats et les « pentes » se correspondent moins. Mais au vu des échelles, ces différences sont minimales car de l'ordre de  $10^{-2}$ . En effet, n'oublions pas que nous mesurons des différences de températures et de toute façon la plupart des thermomètres n'ont pas cette précision. Donc la conclusion reste la même : nos approximations sont tout à fait valable pour cet expérience.

On peut donc dire que la modélisation mathématique est une bonne méthode pour résoudre un problème mathématique tel que l'échauffement dans une barre.

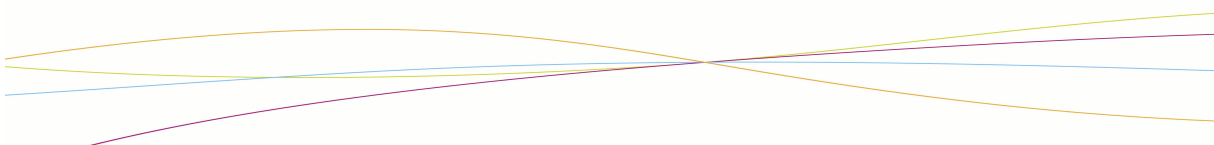




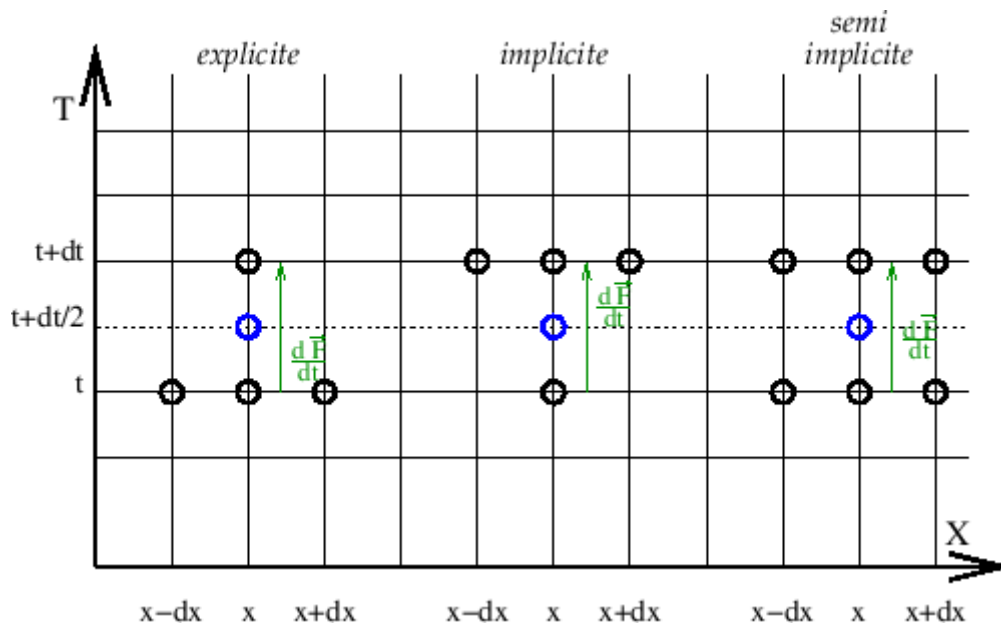
Graphique 1 :  $t$  est fixé,  $t=8 - U(x,8)$



Graphique 2 :  $x$  est fixé  $x=1 - U(1,t)$



Enfin, pour conclure, on peut même noter une meilleure approximation dans le cas de la méthode implicite car l'écart de courbe implicite  $\leftrightarrow$  résultat exact est moins grand que explicite  $\leftrightarrow$  résultat exact, pourquoi une telle différence ? Car tout cela provient de la discrétisation mathématique, en effet la méthode implicite est dite plus stable que l'explicite même si celle-ci est plus rapide. En effet la différence notable entre ces deux méthodes est la variation :  $\frac{d\vec{F}}{dt}$  mesurée. Une solution envisageable serait de faire une « moyenne » de ces deux méthodes où  $\theta=0.5$  et non pas 1 ou 0 : c'est la méthode Crank-Nicholson. Dans ce cas nos résultats auraient été encore plus précis. Voici d'ailleurs un schéma résumant les différences de mesures observées :



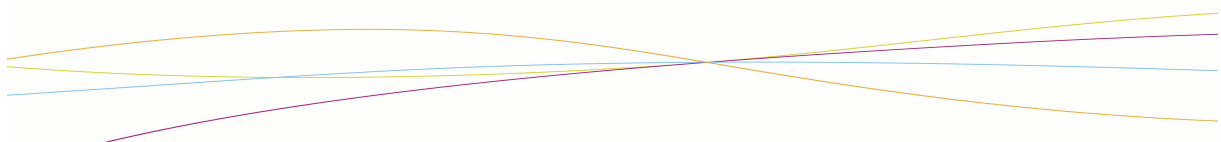
En conclusion, notre discrétisation mathématique est bel et bien validée par notre modélisation informatique.

#### 4. CONCLUSION ET PERSPECTIVES

Ce projet physique aura été pour nous une véritable découverte du monde des sciences. En effet, nous nous doutions pas que les mathématiques et ses modèles comme les méthodes implicites et explicites soient aussi présents dans n'importe quel phénomène physique où une équation aux dérivées partielles est exigée.

Des connaissances mathématiques ont donc dues être approfondies. Mais le côté informatique n'est également pas en reste puisque coder une discrétisation mathématique exige des compétences en Pascal assez élevées. Il a été aussi intéressant d'utiliser de grands principes de la physique pour établir une équation mathématique relativement complexe comparée au phénomène physique simple : l'échauffement dans une barre.

De plus, ce projet sera un vrai plus pour notre futur carrière d'ingénieur puisque l'étude d'un phénomène physique nous aura permis de mettre en place une véritable démarche scientifique : étude du phénomène, mise en équation, discrétisation du problème, résolution et modélisation informatique pour enfin comparaison de nos méthodes. Toutefois, il est à regretter que nous ne pûmes faire un TP afin de véritablement expérimenter nos modèles mais pour des raisons externes à M Bernard Gleyse, ce TP ne put se faire.



Pour conclure, ce projet aura été également une aventure humaine puisque travailler avec des personnes ne se connaissant pas au départ mais qui doit mener un projet commun est toujours difficile. Au début chacun travaillait de son côté mais très vite lorsque nous comprîmes que nos méthodes étaient intimement liées, nous avons alors travaillé ensemble pour une plus grande efficacité et compréhension de notre sujet.

Ce projet physique nous aura donc permis de mettre en place une démarche scientifique et de la mener en travaillant en groupe.

## 5. BIBLIOGRAPHIE

### Livres :

[1] Théorie analytique de la chaleur - Édition F. Diderot – 2007

[2] Éléments de thermocinétique : conduction de la chaleur – Martinet – Technique et Documentation – 1999

[3] Calcul scientifique, de la théorie à la pratique : équations différentielles et équations aux dérivées partielles - Florence Hubert et John Hubbard – Vuibert - 2006

### Sites internet :

[1] [http://fr.wikipedia.org/wiki/%C3%89quation\\_aux\\_d%C3%A9riv%C3%A9es\\_partielles](http://fr.wikipedia.org/wiki/%C3%89quation_aux_d%C3%A9riv%C3%A9es_partielles)

Consulté le 10/02/2012

[2] [http://fr.wikipedia.org/wiki/%C3%89quation\\_de\\_la\\_chaleur](http://fr.wikipedia.org/wiki/%C3%89quation_de_la_chaleur)

Consulté le 10/02/2012

[3] [http://fr.wikipedia.org/wiki/Conduction\\_thermique](http://fr.wikipedia.org/wiki/Conduction_thermique)

Consulté le 17/02/2012

[4] [http://fr.wikipedia.org/wiki/Conductivit%C3%A9\\_thermique](http://fr.wikipedia.org/wiki/Conductivit%C3%A9_thermique)

Consulté le 17/02/2012

[5] [http://laurent.baillet.voila.net/thermo\\_m1\\_barre\\_chauffee.pdf](http://laurent.baillet.voila.net/thermo_m1_barre_chauffee.pdf)

Consulté le 24/02/2012

[6] <http://promenadesmaths.free.fr/Equations%20de%20la%20physique.htm>

Consulté le 16/03/2012

[7] [http://robert.mellet.pagesperso-orange.fr/diff/diff\\_01.htm](http://robert.mellet.pagesperso-orange.fr/diff/diff_01.htm)

Consulté le 30/03/2012

[8] <http://physique-eea.ujf-grenoble.fr/intra/Organisation/CESIRE/TTE/DocsTTE/L3P/Diffusion.pdf>

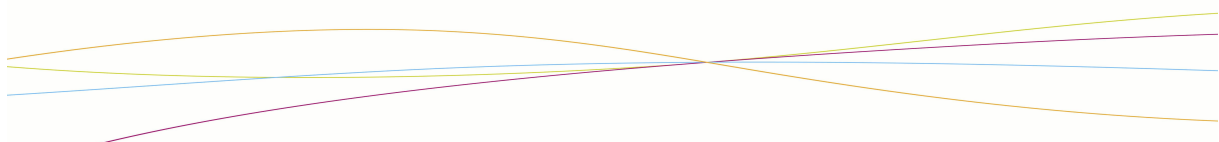
Consulté le 6/04/2012

[9] [http://www.lmm.jussieu.fr/~lagree/COURS/MECAVENIR/cours4\\_eqchal\\_loc.pdf](http://www.lmm.jussieu.fr/~lagree/COURS/MECAVENIR/cours4_eqchal_loc.pdf)

Consulté le 25/05/2012

[10] <https://docs.google.com/viewer?a=v&q=cache:B6SG0mleuNIJ:www.bonneel.com/TPEDP2%282%29.pdf>

Consulté le 20/04/2012



### 5.1. Programme de calcul des méthodes implicite et explicite

```

program edp;
const m=0.5;
      l=1;
      n=3;
      p=9;
type
      valu = array [0..n,0..p] of real;
var
      i,j,io,jo:integer;
      sup,t,h,r:real;
      u:valu;

function f(i,j:integer):real;
var x,y:real;
begin
      x := i*h;
      y := j*t;
      f := (x*sqr(1-x)*(3*x*(4*sqr(y)-1)-8*y)/(1+4*sqr(y))-2*(3*x-2-3*y*(1+x*(6*x-6))))/
(1+4*sqr(y))
end;

function limit(i:integer):real;
begin
      limit := i*h*sqr(1-i*h)
end;

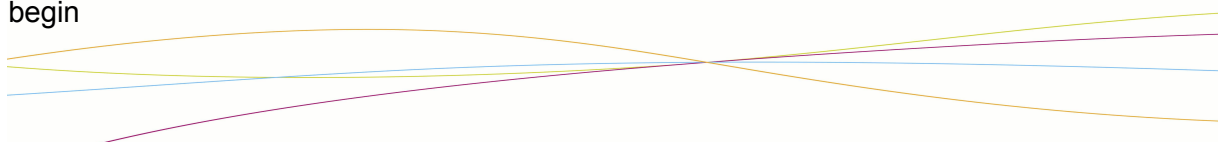
function exact(i,j:integer):real;
begin
      exact:=(sqr(1-i*h)*i*h*(1-3*j*t*i*h))/(1+4*sqr(j*t))
end;

```

```

procedure ecrire;
var i,j:integer;
begin

```



```

    for j:=0 to (p) do
      begin
        for i:=0 to (n) do
          writeln('u(',i:4,',',j:4,')=',u[i,j])
        end
      end
    end;

end;

procedure supdif(i,j:integer);
var a:real;
begin
  a:=abs(u[i,j]-exact(i,j));
  if a >= sup then
    begin
      sup := a;
      io :=i;
      jo:=j
    end
  end;
end;

begin
  io:=0;
  jo:=0;
  sup:=0;
  t:=m/p;
  h:=l/n;
  r:=t/sqr(h);
  if r>0.5 then writeln('le probleme ainsi discretise n`est pas stable,', 'changer les constantes
  pour que la condition r<=0.5 soit verifiee')
  else
    begin
      for i:=0 to (n) do
        u[i,0]:=limit(i);
      for j:=0 to(p-1) do
        begin
          u[0,j+1]:=0;
          for i:=1 to (n-1) do
            begin
              u[i,j+1]:=u[i,j]+r*(u[i-1,j]-2*u[i,j]+u[i+1,j])+t*f(i,j);
              // Pour la méthode implicite, on a u[i,j+1]:=u[i,j+1]+r*(u[i-
              1,j+1]-2*u[i,j+1]+u[i+1,j+1])+t*f(i,j+1)
            end
          end
        end
      end
    end
  end;
end;

```

```

supdif(i,j+1)
end;
u[n,j+1]:=u[n-1,j+1];
supdif(n,j+1)
end;
ecrire;
writeln('la difference maximum entre la solution approchee','et la solution
exacte est',sup,' en U',io:4,' ',jo:4)
end
end.

```

## 5.2. Programme de résolution de système linéaire par la méthode de Cholesky

Program resolutionSystemeLineaireMethodeCholesky;

Const

MAX=20;

Type

matrice=array[1..MAX,1..MAX] of real;

vecteur=array[1..MAX] of real;

Var

A,L,Lt: matrice;

B,x,y : vecteur;

N,i,j: Integer;

S : Real;

{Fonction permettant la factorisation de Cholesky de A, on obtient ainsi L, une matrice triangulaire}

function FactorisationCholesky(A:matrice):matrice;

Var

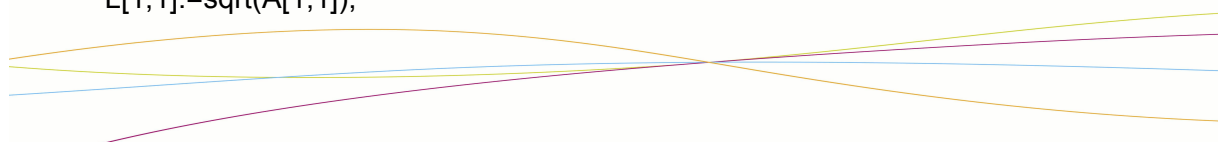
L : matrice;

puis, somme1, somme2, somme3 : Real;

i,j,k : Integer;

Begin

L[1,1]:=sqrt(A[1,1]);



```

for i:=2 to N do
  L[i,1]:=(A[1,i])/(L[1,1]);

somme1:=0;
for k:=2 to (N-1) do
  begin

    for i:=1 to (k-1) do
      begin
        puis:=1;
        puis:= puis*L[k,i]*L[k,i];
        somme1:= somme1 + puis;
      end;

    L[k,k]:=sqrt(A[k,k]-(somme1));
    end;

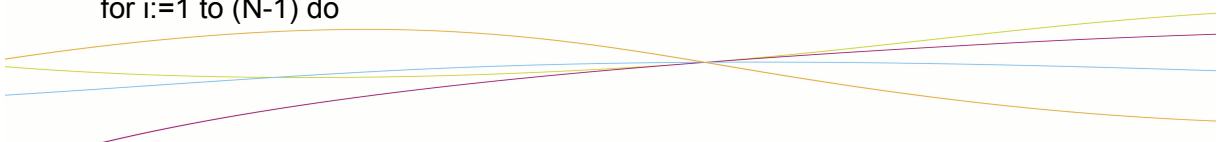
somme2:=0;
for k:=2 to (N-1) do
  begin
    for i:=(k+1) to N do
      begin
        somme2:=0;
        for j:=1 to (k-1) do

          begin
            puis:=1;
            puis:=puis*L[i,j]*L[k,j];
            somme2:= somme2 + puis;
          end;
          L[i,k]:=(A[i,k]-somme2)/(L[k,k]);
        end;
      end;
    end;
  end;

```

```
somme3:=0;
```

```
for i:=1 to (N-1) do
```





```
begin
    puis:=1;
    puis:=puis*L[N,i]*L[N,i];
    somme3:= somme3 + puis;
```

```
end;
```

```
L[N,N]:= sqrt(A[N,N]-somme3);
```

```
writeln("Voici la matrice L :");
```

```
for i:=1 to N do
    begin
        for j:=1 to N do
            write(L[i,j]:2);
            writeln;
```

```
FactorisationCholesky:=L;
```

```
End;
```

{Maintenant que nous avons la matrice L, on sait que si A est symétrique positive alors  $A=LL^t$ , cherchons  $L^t$

en effectuant une transposition dont voici le programme}

```
function Transposition(L:matrice):matrice;
```

```
Var
```

```
i,j : Integer;
```

```
Begin
```

```
for i:=1 to N do
    begin
        for j:=1 to N do
            Lt[j,i] := L[i,j];
```

```
end;
```

```
writeln ("Voici la transposée de la matrice L, Lt : ");
```



```

for j:= 1 to N do
  begin
    for i:= 1 to N do
      write(Lt[j,i]:5);
      writeln;
    end;
  end;

```

Transposition:=Lt;

End;

{Une fois nos deux fonctions créées, nous pouvons les utiliser pour résoudre un système linéaire du type  $Ax=b$ }

BEGIN

{Lecture de la dimension de A}

writeln('Quels est la dimension de votre matrice A ?');

Readln(N);

{Lecture des coefficients de la matrice A}

Writeln('Le système à résoudre est du type  $Ax=B$ , veuillez entrer la matrice A :');

for i:=1 to N do

begin

for j:=1 to N do

begin

writeln('Coefficient de la ligne ',i,' , et de la colonne ',j);

read(A[i,j]);

end;

end;

{On lit ensuite le vecteur b}

Writeln('Veuillez entrer le vecteur B :');

for i:=1 to N do

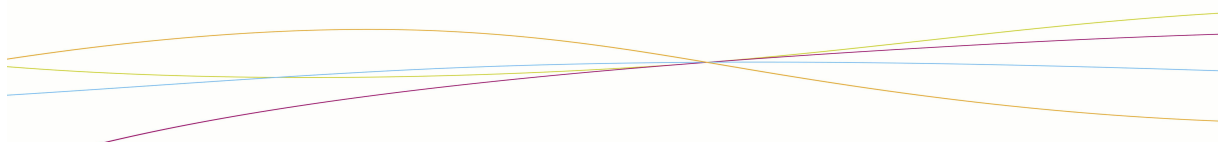
begin

writeln('Coefficients de la ligne ',i);

read(B[i]);

end;

{Enfin on commence la résolution par la méthode de Cholesky}



```

L:=FactorisationCholesky(A);
Lt:=Transposition(L);
writeln('Une fois triangularisée A, soit y un vecteur tel que  $y=Lt*x$  : ');

```

```
{Création du vecteur intermédiaire y}
```

```

y[1]:=B[1]/L[1,1];
for i:=1 to N do
  begin
    S:=0;
    for j:=1 to i-1 do
      S:=S+L[i,j]*y[j];
    y[i]:=(B[i]-S)/L[i,i];
    write(y[i]:5);
    writeln;
  end;

```

```
{Résolution du système avec la résolution de X}
```

```

x[N]:=y[N]/Lt[N,N];
writeln('Voici la solution du système linéaire, le vecteur x est : ');
for i:=N downto 1 do
  begin
    S:=0;
    for j:=i+1 to N do
      S:=S+Lt[i,j]*x[j];
    x[i]:=(y[i]-S)/Lt[i,i];
  end;

```

```
{Affichage de la solution x}
```

```

For i:=1 to N do
  begin
    write(x[i]:5);
    writeln;
  end;

```

```
END.
```

