



Rapport 3EE103

DIFFUSION DE LA CHALEUR

ZHU Conghui | GONG Weiyi | WAN Yufeng

Sommaire

1. Introduction
2. Description globale
3. Structuration des données
4. Déroulement du projet
 - I. Résolution du problème mono-dimensionnel
 - II. Résolution du problème bi-dimensionnel
5. Conclusion

Introduction

L'objectif de ce projet est d'analyser et puis modéliser la diffusion de la chaleur dans les composants électroniques. Pour ce faire, nous devons programmer en langage C en utilisant les méthodes numériques que nous avons appris dans le cours 3EE103 et à la fin, on présentera graphiquement les résultats à l'aide de Matlab.

Description globale

On donne d'abord la forme générale de l'équation de la chaleur. Soit Ω un domaine de \mathbb{R}^3 de frontière $\partial\Omega$ et $T(\mathbf{r}, t)$ la température en \mathbf{r} à l'instant t .

L'équation de la chaleur s'écrit :

$$\begin{cases} \frac{\partial T}{\partial t}(\mathbf{r}, t) - \alpha(\mathbf{r})\Delta T(\mathbf{r}, t) = f(\mathbf{r}, t) & \mathbf{r} \in \Omega, t > t_0 \\ T(\mathbf{r}, t) = f_c(\mathbf{r}) & \mathbf{r} \in \partial\Omega, t > t_0 \\ T(\mathbf{r}, t_0) = T_0(\mathbf{r}) & \mathbf{r} \in \Omega \end{cases}$$

Où $\alpha(\mathbf{r}) = \frac{K(\mathbf{r})}{\rho(\mathbf{r}) \cdot c(\mathbf{r})}$, $K(\mathbf{r})$ est la conductivité thermique (en $[W(K.m)^{-1}]$), $c(\mathbf{r})$ est la capacité calorifique (en $[J(kg.K)^{-1}]$) et $\rho(\mathbf{r})$ est la masse volumique (en $[kg.m^{-3}]$).

Ici, $f(\mathbf{r}, t)$ est une fonction pour décrire la source de la chaleur interne. Elle s'écrit en $f = \frac{P}{\rho c}$ où P est la production volumique interne de puissance calorifique (en $[W.m^{-3}]$). Par exemple, dans le cas d'un résistor, P est la puissance volumique dissipé par effet joule.

Donc le cœur de ce projet est de résoudre l'équation aux dérivées partielles qui est donné ci-dessous. Pour ce faire, il y a deux méthodes principales : Différences Finies et Runge-Kutta 2.

La méthode Différences Finies est une technique courante de recherche de solutions approchées d'équations aux dérivées partielles qui consiste à résoudre un système de relations (schéma numérique) liant les valeurs des fonctions inconnues en certains points suffisamment proches les uns des autres.

Les méthodes de Runge-Kutta sont des méthodes d'analyse numérique d'approximation de solutions d'équations différentielles. Ces méthodes

reposent sur le principe de l'itération, c'est-à-dire qu'une première estimation de la solution est utilisée pour calculer une seconde estimation, plus précise.

Dans la suite, on analyse premièrement le cas mono-dimensionnel et après on passera à la partie bi-dimensionnel.

Déroulement du projet

Résolution du problème mono-dimensionnel

Dans un premier temps, on considère le problème monodimensionnel (fil homogène + source ponctuelle) et on résout l'équation aux dérivées partielles (EDP) en x et en t par la méthode Différences Finies.

Dans ce cas, on considère que la température au bord ne varie pas en fonction du temps. La température initiale est donnée par la fonction $T_0(x)$.

$$\begin{cases} \frac{\partial T}{\partial t}(x, t) - \alpha \frac{\partial^2 T}{\partial x^2}(x, t) = f(x, t) & x \in]a; b[, t > t_0 \\ T(a, t) = T_a, \quad T(b, t) = T_b & t > t_0 \\ T(x, t_0) = T_0(x) & x \in [a; b] \end{cases}$$

Pour programmer la méthode différences finies en C, on doit d'abord discrétiser l'espace et le temps.

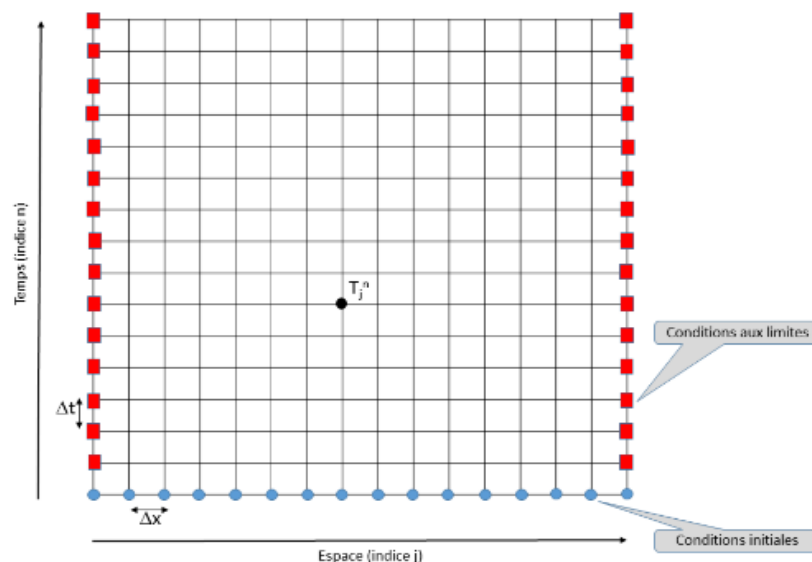


Figure 1 Matrice modélisé pour le mono-dimensionnel

Sur cette grille, chaque nœud est repéré par ses coordonnées d'espace et de temps. Ainsi chaque nœud peut être représenté de la forme $T(x + i\Delta x, t + j\Delta t)$. L'intérêt de la méthode différences finies est de ne plus calculer la solution de l'EDP en tous points dans le domaine mais seulement à chaque nœud de la grille.

À partir de la méthode différences finies, on en déduit les deux équations suivantes :

$$\begin{cases} \frac{\partial T}{\partial t}(x, t) = \frac{T(x, t + \Delta t) - T(x, t)}{\Delta t} \\ \frac{\partial^2 T}{\partial x^2}(x, t) = \frac{T(x + \Delta x, t) - 2T(x, t) + T(x - \Delta x, t)}{(\Delta x)^2} \end{cases}$$

En remplaçant la terme $\frac{\partial T}{\partial t}(x, t)$ et la terme $\frac{\partial^2 T}{\partial x^2}(x, t)$ dans l'équation de la chaleur, on obtient $T(x, t + \Delta t) = \left(\alpha \frac{T(x + \Delta x, t) - 2T(x, t) + T(x - \Delta x, t)}{(\Delta x)^2} + f(x, t) \right) \Delta t + T(x, t)$.

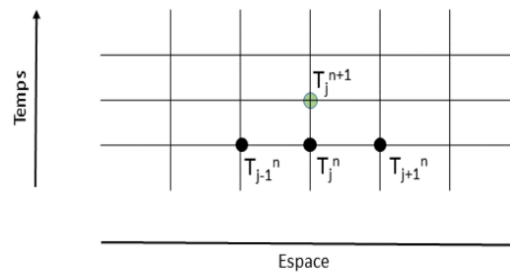


Figure 2

À l'aide du schéma ci-dessus, on peut trouver qu'on est capable de calculer le nœud vert (l'instant prochain) à partir de ces trois nœuds noirs.

Les paramètres choisis sont $\Delta x = 0.05m$, $\Delta t = 1s$, $f = 1$.

Nous avons choisi de placer la source ponctuelle au centre du fil.

REMARQUE

- f ne représente pas la température de la source interne, il est la puissance volumique, donc il faut bien choisir sa valeur pour que elle ne soit pas très grande.
- Sachant que les températures aux bords sont fixées pour simplifier la modèle, le système établi ne va jamais atteindre la régime permanent et en plus la température au centre (la position de la source interne) va tends vers l'infini. Il est logique parce que dans notre modèle, il n'existe pas l'échange de chaleur entre le système et l'atmosphère. Donc l'énergie

(chaleur) produite par les composants électroniques reste toujours dans le système. Pour améliorer, il faut changer les conditions aux limites.

Ci-dessous le résultat obtenu :

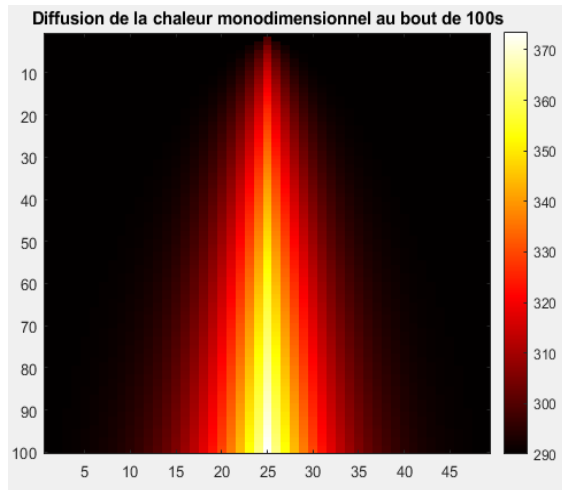


Figure 3 Diffusion de la chaleur monodimensionnel au bout de 100s (Source ponctuel)

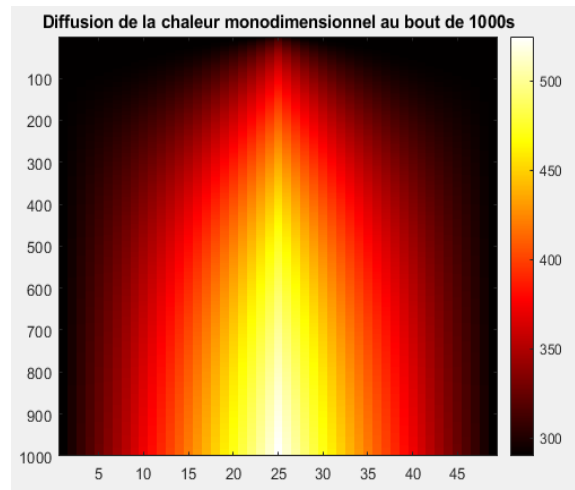


Figure 4 Diffusion de la chaleur monodimensionnel au bout de 1000s (Source ponctuel)

Nous avons observé que au bout de 100s, la température au centre a augmenté de 70°C et au bout de 1000s, la température a augmenté jusqu'à 500°C (différence de 210°C) et ce qui n'est quasiment pas possible pour le composant électronique. Ça montre bien que la proposition que j'ai donnée précédemment est logique et correcte. Mais au niveau théorique, nous avons bien modéliser et simuler le processus de la diffusion de la chaleur.

Résolution du problème bidimensionnel

Méthode Différences Finies

On considère ensuite des composants bidimensionnels (PCB et chips) et les sources correspondantes (ponctuelle ou linéaire).

L'équation de la chaleur devient donc

$$\left\{ \begin{array}{l} \frac{\partial T}{\partial t}(x, y, t) - \alpha \left(\frac{\partial^2 T}{\partial x^2}(x, y, t) + \frac{\partial^2 T}{\partial y^2}(x, y, t) \right) = f(x, y, t) \quad x \in]0; a[, y \in]0; b[, t > t_0 \\ T(0, y, t) = T_{x0}, \quad T(a, y, t) = T_a \quad y \in]0; b[, t > t_0 \\ T(x, 0, t) = T_{y0}, \quad T(x, b, t) = T_b \quad x \in]0; a[, t > t_0 \\ T(x, t_0) = T_0(x) \quad x \in]0; a[, y \in]0; b[\end{array} \right.$$

Dans ce cas, la température au bord ne varie pas en fonction du temps. La température initiale est donnée par la fonction $T_0(x, y)$.

Les étapes sont presque pareilles que le précédent. On discrétise en l'espace et en temps et on applique la méthode différences finies :

$$\begin{cases} \frac{\partial T}{\partial t}(x, y, t) = \frac{T(x, y, t + \Delta t) - T(x, y, t)}{\Delta t} \\ \frac{\partial^2 T}{\partial x^2}(x, y, t) = \frac{T(x + \Delta x, y, t) - 2T(x, y, t) + T(x - \Delta x, y, t)}{(\Delta x)^2} = A \\ \frac{\partial^2 T}{\partial y^2}(x, y, t) = \frac{T(x, y + \Delta y, t) - 2T(x, y, t) + T(x, y - \Delta y, t)}{(\Delta y)^2} = B \end{cases}$$

En remplaçant les terme $\frac{\partial T}{\partial t}(x, y, t)$, $\frac{\partial^2 T}{\partial x^2}(x, y, t)$ et $\frac{\partial^2 T}{\partial y^2}(x, y, t)$ dans l'équation de la chaleur, on obtient $T(x, y, t + \Delta t) = (\alpha(A + B) + f(x, y, t))\Delta t + T(x, y, t)$.

Les paramètres choisis sont $\Delta x = 0.05m$, $\Delta y = 0.05m$, $\Delta t = 1s$, $f = 5$.

La position de la source : le nœud au centre pour la source ponctuelle et la colonne au milieu pour la source linéaire.

REMARQUE

La résolution de l'équation en 2D est relativement lente voire le programme n'arrive même pas de calculer un volume de $50 \times 50 \times 1000$. Nous n'avons pas trouvé la bonne explication (la qualité du code est une raison possible) mais c'est sûr qu'on doit trouver une méthode numérique plus rapide et plus efficace.

Ci-dessous les résultats obtenus :

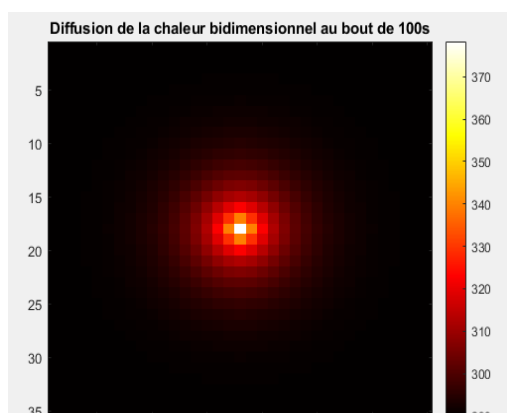


Figure 3 Diffusion de la chaleur bidimensionnel au bout de 100s (Source ponctuel)

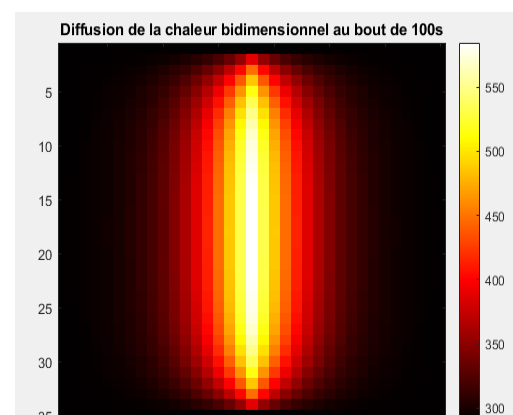


Figure 4 Diffusion de la chaleur bidimensionnel au bout de 100s (Source linéaire)

Nous avons observé que la chaleur se propage dans le bon sens. Dans le cas source ponctuelle, la température au centre a augmenté de 80°C. Et dans le cas source linéaire, la température de la colonne milieu a augmenté de 200°C (l'effet de la superposition). Le résultat obtenu est cohérent.

Méthode Runge-Kutta 2

En considérant le problème rencontré précédemment, nous sommes obligés de changer la méthode numérique. Donc pour diminuer le temps de calcul, on applique la méthode de Runge-Kutta 2 pour l'évolution temporelle.

En utilisant la méthode Runge-Kutta 2, on obtient les équations suivantes :

$$\left\{ \begin{array}{l} T_{\frac{1}{2}}(x, y) = T(x, y, t) + \frac{\Delta t}{2} (\alpha \Delta T(x, y, t) + f(x, y, t)) \\ \Delta T_{\frac{1}{2}}(x, y) = \frac{T_{\frac{1}{2}}(x-1, y) - 2T_{\frac{1}{2}}(x, y) + T_{\frac{1}{2}}(x+1, y)}{(\Delta x)^2} + \frac{T_{\frac{1}{2}}(x, y-1) - 2T_{\frac{1}{2}}(x, y) + T_{\frac{1}{2}}(x, y+1)}{(\Delta y)^2} \\ T(x, y, t+1) = T_{\frac{1}{2}}(x, y) + \frac{\Delta t}{2} \left(\alpha \Delta T_{\frac{1}{2}}(x, y) + f(x, y, t) \right) \end{array} \right.$$

On garde les mêmes paramètres $\Delta x = 0.05m, \Delta y = 0.05m, \Delta t = 1s, f = 5$ et ci-dessous le résultat obtenu :

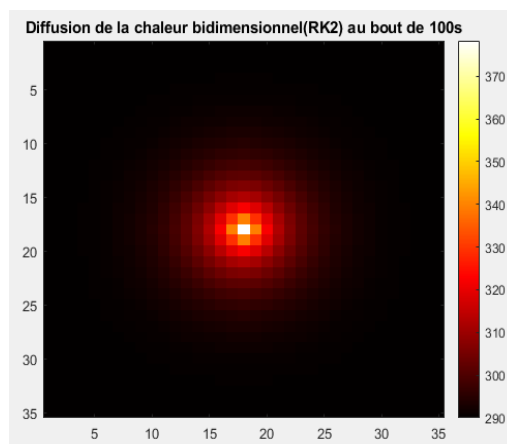


Figure 5 Diffusion de la chaleur bidimensionnel au bout de 100s (Source ponctuel) (RK2)

Nous avons observé que le résultat est quasiment la même par rapport le résultat qu'on a obtenu en utilisant la méthode différences finies. Et en même temps, la rapidité a beaucoup plus augmenté. Donc le choix d'utiliser la méthode Runge-Kutta 2 est tout à fait raisonnable.

Pour explorer plus de possibilité, nous avons appliqué les sources de chaleur variable en fonction du temps. Chaque Δt , la valeur de f augmente 0.1. Et on obtient le résultat suivant :

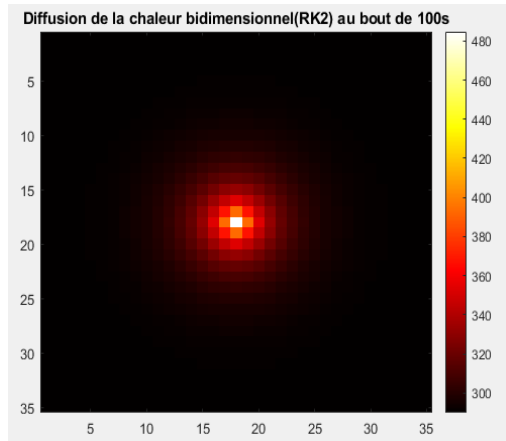


Figure 6 Diffusion de la chaleur bidimensionnel au bout de 100s (Source ponctuel variable) (RK2)

Structuration des données

Pour faciliter le codage et également pour rendre le code plus compréhensible, nous avons utilisé trois structures et 2 bibliothèques.

<pre>typedef struct{ float K; float C; float rou; }materiel;</pre>	La première structure est spécifiée au matériel. Il contient la conductivité thermique K, la capacité calorifique C et la masse volumique ρ .
<pre>typedef struct{ int nb_point; float* data; }source_1D;</pre>	La deuxième structure regroupe les informations nécessaires pour construire la source monodimensionnelle. Il contient le nombre de point et un tableau de valeur de la source.
<pre>typedef struct{ int nb_point_X; int nb_point_Y; float** data; }source_2D;</pre>	La troisième structure définit la source bidimensionnelle. Il contient le nombre de point pour chaque ligne et chaque colonne et un volume qui stock la valeur de la source.

Bibliothèque source.h permet de gérer les sources.

//Générer les différents types de source

```
source_1D creer_source1D_ponctuelle(int nb_point);
source_2D creer_source2D_ponctuelle(int nb_point_X,int nb_point_Y);
source_2D creer_source2D_lineaire(int nb_point_X,int nb_point_Y);
```

//Libérer l'espace utilisé par l'allocation dynamique.

```
void freeSource1D(source_1D source);
void freeSource2D(source_2D source);
```

Bibliothèque EDP.h contient principalement les algorithmes.

//L'ensemble de fonction pour calculer l'évolution temporelle.

```

float** creerMatrice(int dimX,int dimY);
float** Init_condition_1D(int nb_point,int nb_temps);
float calculDeriveeSeconde_X (float valeur_avant,float valeur_centre,float valeur_arriere);
float calculDeriveeSeconde_Y (float valeur_avant,float valeur_centre,float valeur_arriere);
float** Diffusion_1D(source_1D source,int nb_temps);
void Affichage_Diffusion_1D(float** diffusion,int nb_temps,source_1D source);
|
float*** creerVolume(int dimX,int dimY,int dimT);
float*** Init_condition_2D(int nb_point_X,int nb_point_Y,int nb_temps);
float*** Diffusion_2D_DF(source_2D source,int nb_temps);
float calcul_laplacien(float DeriveeSecondeX, float DeriveeSecondeY);
float*** Diffusion_2D_RK2 (source_2D source,int nb_temps);
void Affichage_Diffusion_2D(float*** diffusion,int nb_temps,source_2D source);

```

Pour plus d'information, veuillez consulter le programme original où il y a toutes les fonctions utilisées et les commentaires précises.

Conclusion

Pendant ce projet, nous avons utilisé la méthode des différences finies en 1D et 2D et aussi la méthode de Runge-Kutta 2 en 2D pour résoudre l'équation aux dérivées partielles (EDP) numériquement.

Le résultat est comme prévu et raisonnable en considérant la condition qu'on donne. La difficulté principale pendant ce projet est de bien choisir la valeur de la source. Il faut bien comprendre quelle grandeur il représente avant de choisir sa valeur.

Les améliorations qu'on peut faire est de changer la condition de limite. Ainsi, le résultat au bout d'un temps important sera plus proche de la loi naturelle. Le système sera stable quand t tends vers l'infini.

Bibliographie

[https://fr.wikipedia.org/wiki/M%C3%A9thodes_de_Runge-Kutta#La_m%C3%A9thode_de_Runge-Kutta_d'ordre_2_\(RK2\)](https://fr.wikipedia.org/wiki/M%C3%A9thodes_de_Runge-Kutta#La_m%C3%A9thode_de_Runge-Kutta_d'ordre_2_(RK2))

https://fr.wikiversity.org/wiki/Conduction_thermique/%C3%89quation_de_la_chaleur

https://fr.wikipedia.org/wiki/M%C3%A9thode_des_diff%C3%A9rences_finies