

**Titre:** Méthodes de calcul numérique pour la simulation thermique des circuits intégrés  
Title:

**Auteur:** Alain Fourmigue  
Author:

**Date:** 2014

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** Fourmigue, A. (2014). Méthodes de calcul numérique pour la simulation thermique des circuits intégrés [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/1580/>  
Citation:

 **Document en libre accès dans PolyPublie**  
Open Access document in PolyPublie

**URL de PolyPublie:** <https://publications.polymtl.ca/1580/>  
PolyPublie URL:

**Directeurs de recherche:** Gabriela Nicolescu, & Giovanni Beltrame  
Advisors:

**Programme:** Génie informatique  
Program:

UNIVERSITÉ DE MONTRÉAL

MÉTHODES DE CALCUL NUMÉRIQUE POUR LA SIMULATION THERMIQUE DES  
CIRCUITS INTÉGRÉS

ALAIN FOURMIGUE  
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION  
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR  
(GÉNIE INFORMATIQUE)  
DÉCEMBRE 2014

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

MÉTHODES DE CALCUL NUMÉRIQUE POUR LA SIMULATION THERMIQUE DES  
CIRCUITS INTÉGRÉS

présentée par : FOURMIGUE Alain

en vue de l'obtention du diplôme de : Philosophiæ Doctor

a été dûment acceptée par le jury d'examen constitué de :

M. LANGLOIS J.M. Pierre, Ph. D., président

Mme NICOLESCU Gabriela, Doctorat, membre et directrice de recherche

M. BELTRAME Giovanni, Ph. D., membre et codirecteur de recherche

M. BOIS Guy, Ph. D., membre

Mme LIBOIRON-LADOUCEUR Odile, Ph. D., membre externe

## RÉSUMÉ

La simulation thermique des circuits intégrés est aujourd’hui incontournable afin de prévenir les problèmes de dissipation thermique. Les circuits intégrés modernes exhibent une densité de puissance très élevée, pouvant conduire à une température excessive, dégradant les performances du circuit et réduisant sa durée de vie.

Depuis leur invention en 1958, les circuits intégrés ont subi une évolution spectaculaire. La miniaturisation des composants, l’émergence des technologies tridimensionnelles et l’introduction de systèmes de refroidissement à base de liquide nécessitent de résoudre des problèmes thermiques très complexes. Les méthodes numériques fournissent un outil de simulation puissant, mais aussi très coûteux en calcul. Les simulateurs thermiques existants basés sur les méthodes numériques requièrent un temps d’exécution très long et une consommation de mémoire très importante. Ces inconvénients majeurs ralentissent considérablement la conception des circuits intégrés ou contraint à choisir des solutions thermiques non optimales.

Cette thèse propose deux méthodes de calcul numérique originales pour accélérer la simulation thermique des circuits intégrés. La première méthode est destinée à la simulation du régime transitoire, tandis que la deuxième méthode cible le régime permanent. Les méthodes de calcul proposées combinent des algorithmes puissants tels que la décomposition des opérateurs, les méthodes de projection et la décomposition de domaine. Les méthodes de calcul proposées sont évaluées sur des problèmes thermiques réalistes et variés, en étant comparés à des méthodes de calcul classiques bien éprouvées. La méthodologie utilisée pour modéliser le comportement thermique des circuits intégrés est validée en comparant les résultats obtenus par simulation à des mesures thermiques effectuées sur un circuit physique réel.

Dans le cadre de cette thèse, un simulateur thermique pour circuits intégrés, appelé « IC-Therm » a été développé. Ce logiciel implémente les méthodes de calcul proposées dans cette thèse pour simuler le régime transitoire et le régime permanent. Le logiciel ICTherm est actuellement utilisé en milieu universitaire pour des projets de recherche.

## ABSTRACT

Since their introduction in 1958, integrated circuits have gone through a spectacular evolution. The transistor miniaturization, the emergence of tridimensional technologies have caused an explosion in complexity. In addition, modern integrated circuits generate very high heat fluxes that can lead to a high temperature, degrading the performance and reducing the lifetime of the device. The thermal simulation of integrated circuits is extensively used to prevent this kind of heat dissipation issues, but its application is becoming increasingly difficult with the growing complexity of chip designs.

Numerical methods offer a powerful tool for the thermal simulation of integrated circuits. However, their high computation cost leads to long simulation times and to a vast memory usage, slowing down the design process or constraining the designer to unoptimized solutions. This thesis presents two highly-efficient methods for the thermal simulation of integrated circuits, for both transient and steady-state thermal problems. The proposed methods combine powerful algorithms, such as operator splitting, sub-space projection, and domain decomposition.

We evaluated the proposed techniques on a variety of representative thermal problems. Our results show up to an order of magnitude improvement in simulation speed when compared against well established methods for the same level of accuracy. We have also validated our approach by comparing simulation results with the temperature measured on a physical device via infrared thermography, achieving an error smaller than 6%.

In addition, we developed a thermal simulator for integrated circuits called ICTherm, implementing the methods proposed in this thesis. ICTherm can be used to simulate the transient and steady-state behaviour of arbitrary integrated circuits, and it is currently used for research by several universities.

## TABLE DES MATIÈRES

|  |      |
|--|------|
| RÉSUMÉ . . . . .   | iii  |
| ABSTRACT . . . . .   | iv   |
| TABLE DES MATIÈRES . . . . .                                     | v    |
| LISTE DES TABLEAUX . . . . .                                     | viii |
| LISTE DES FIGURES . . . . .                                      | ix   |
| LISTE DES SIGLES ET ABRÉVIATIONS . . . . .                       | xi   |
| <br>   |      |
| CHAPITRE 1 INTRODUCTION . . . . .                                | 1    |
| 1.1 Définitions et contexte . . . . .                            | 1    |
| 1.1.1 La dissipation thermique des circuits intégrés . . . . .   | 1    |
| 1.1.2 La simulation thermique des circuits intégrés . . . . .    | 4    |
| 1.1.3 Les défis actuels de la simulation thermique . . . . .     | 5    |
| 1.2 Éléments de la problématique . . . . .                       | 8    |
| 1.2.1 Analyse des requis . . . . .                               | 8    |
| 1.2.2 Critères d'évaluation . . . . .                            | 10   |
| 1.3 Hypothèse de recherche et contributions . . . . .            | 11   |
| 1.3.1 Hypothèse de recherche . . . . .                           | 11   |
| 1.3.2 Objectifs de recherche et contributions . . . . .          | 11   |
| <br>   |      |
| CHAPITRE 2 CONCEPTS DE BASE . . . . .                            | 14   |
| 2.1 Description structurelle des circuits intégrés . . . . .     | 14   |
| 2.1.1 Composition typique d'un circuit intégré . . . . .         | 14   |
| 2.1.2 Puces, transistors et interconnexions . . . . .            | 15   |
| 2.1.3 Évolution des boîtiers d'encapsulation . . . . .           | 17   |
| 2.1.4 Risques liés à une température élevée . . . . .            | 19   |
| 2.1.5 Systèmes de refroidissement . . . . .                      | 19   |
| 2.2 Théorie des transferts de chaleur . . . . .                  | 21   |
| 2.2.1 Définition de la température . . . . .                     | 21   |
| 2.2.2 Transferts de chaleur dans les circuits intégrés . . . . . | 22   |
| 2.2.3 Équation de bilan thermique . . . . .                      | 24   |

|  |   |    |
|--|---|----|
| 2.2.4  | Modèle thermique basé sur l'équation de la chaleur . . . . .        | 25 |
| 2.2.5  | Bilan . . . . .   | 27 |
| CHAPITRE 3 REVUE DE LITTERATURE . . . . .    |   | 28 |
| 3.1  | Modèles thermiques analytiques . . . . .                            | 28 |
| 3.1.1  | Principe . . . . .  | 28 |
| 3.1.2  | Modèles thermiques basés sur les séries de Fourier . . . . .        | 28 |
| 3.1.3  | Modèles thermiques basés sur les fonctions de Green . . . . .       | 29 |
| 3.1.4  | Bilan . . . . .   | 33 |
| 3.2  | Modèles thermiques compacts . . . . .                               | 33 |
| 3.2.1  | Principe . . . . .  | 33 |
| 3.2.2  | Modèles DELPHI . . . . .  | 34 |
| 3.2.3  | Modèle HotSpot . . . . .  | 35 |
| 3.2.4  | Bilan . . . . .   | 37 |
| 3.3  | Modèles thermiques numériques . . . . .                             | 37 |
| 3.3.1  | Principe . . . . .  | 37 |
| 3.3.2  | Modèles numériques du régime transitoire . . . . .                  | 46 |
| 3.3.3  | Modèles numériques du régime permanent . . . . .                    | 49 |
| 3.3.4  | Bilan . . . . .   | 52 |
| CHAPITRE 4 MÉTHODES PROPOSÉES . . . . .      |   | 54 |
| 4.1  | Méthode proposée pour la simulation du régime transitoire . . . . . | 54 |
| 4.1.1  | Décomposition des opérateurs . . . . .                              | 55 |
| 4.1.2  | Schéma itératif pour corriger l'erreur de décomposition . . . . .   | 57 |
| 4.1.3  | Algorithme de la méthode proposée . . . . .                         | 59 |
| 4.1.4  | Bilan . . . . .   | 60 |
| 4.2  | Méthode proposée pour la simulation du régime permanent . . . . .   | 60 |
| 4.2.1  | Méthodes de projection . . . . .                                    | 61 |
| 4.2.2  | Préconditionnement incomplet LU . . . . .                           | 66 |
| 4.2.3  | Décomposition de domaine . . . . .                                  | 68 |
| 4.2.4  | Algorithme de la méthode proposée . . . . .                         | 72 |
| 4.2.5  | Bilan . . . . .   | 72 |
| CHAPITRE 5 RÉSULTATS EXPÉRIMENTAUX . . . . . |   | 74 |
| 5.1  | Description des problèmes modèles . . . . .                         | 74 |
| 5.1.1  | Circuit intégré 3D refroidi par un dissipateur de chaleur . . . . . | 74 |
| 5.1.2  | Circuit intégré 3D refroidi par un liquide . . . . .                | 75 |

|            |   |     |
|------------|---|-----|
| 5.1.3      | Téléphone intelligent refroidi par convection naturelle . . . . . | 77  |
| 5.2        | Simulation du régime transitoire . . . . .                        | 78  |
| 5.2.1      | Évaluation de la précision . . . . .                              | 78  |
| 5.2.2      | Évaluation des performances . . . . .                             | 80  |
| 5.2.3      | Bilan . . . . .   | 83  |
| 5.3        | Simulation du régime permanent . . . . .                          | 83  |
| 5.3.1      | Évaluation de la précision . . . . .                              | 84  |
| 5.3.2      | Évaluation des performances . . . . .                             | 84  |
| 5.3.3      | Bilan . . . . .   | 86  |
| CHAPITRE 6 | VALIDATION PHYSIQUE . . . . .                                     | 87  |
| 6.1        | Description du système de validation . . . . .                    | 87  |
| 6.2        | Modélisation thermique . . . . .                                  | 91  |
| 6.2.1      | Éléments représentés dans le modèle . . . . .                     | 91  |
| 6.2.2      | Conditions aux limites . . . . .                                  | 92  |
| 6.2.3      | Implémentation du modèle . . . . .                                | 94  |
| 6.3        | Validation du modèle thermique . . . . .                          | 94  |
| 6.4        | Bilan . . . . .   | 96  |
| CHAPITRE 7 | CONCLUSION . . . . .  | 97  |
| 7.1        | Synthèse des travaux . . . . .                                    | 97  |
| 7.2        | Limitations des méthodes proposées . . . . .                      | 99  |
| 7.3        | Travaux futurs . . . . .  | 100 |
| RÉFÉRENCES | . . . . .   | 101 |



## LISTE DES TABLEAUX

|             |  |    |
|-------------|--|----|
| Tableau 1.1 | Niveaux de modélisation thermique . . . . .  | 6  |
| Tableau 5.1 | Erreur relative maximale (%) <b>sans correction</b> de l'erreur de décomposition . . . . .     | 79 |
| Tableau 5.2 | Erreur relative maximale (%) <b>avec correction</b> de l'erreur de décomposition . . . . .     | 80 |
| Tableau 5.3 | Temps de calcul par pas de temps <b>sans correction</b> de l'erreur de décomposition . . . . . | 81 |
| Tableau 5.4 | Temps de calcul par pas de temps <b>avec correction</b> de l'erreur de décomposition . . . . . | 81 |
| Tableau 5.5 | Temps mis pour simuler le régime transitoire (200 pas de temps) . . .                          | 82 |
| Tableau 5.6 | Consommation de mémoire pour simuler le régime transitoire . . . .                             | 82 |
| Tableau 5.7 | Temps mis pour simuler le régime permanent . . . . .   | 85 |
| Tableau 5.8 | Consommation de mémoire pour simuler le régime permanent . . . .                               | 85 |
| Tableau 6.1 | Éléments du modèle thermique . . . . .   | 91 |
| Tableau 6.2 | Scénarios de puissance considérés pour la validation du modèle thermique                       | 95 |

## LISTE DES FIGURES

|             |   |    |
|-------------|---|----|
| Figure 1.1  | Évolution de la densité de puissance dans les processeurs Intel <sup>†</sup> . . . . .                        | 2  |
| Figure 1.2  | Chemins suivis par la chaleur pour sortir du boîtier d'encapsulation . . . . .                                | 3  |
| Figure 1.3  | Simulation thermique de circuit intégré . . . . .   | 4  |
| Figure 1.4  | Technologies de circuits intégrés 3D . . . . .  | 7  |
| Figure 1.5  | Refroidissement d'un circuit intégré à l'aide de liquide . . . . .  | 8  |
| Figure 2.1  | Structure type d'un circuit intégré . . . . .   | 14 |
| Figure 2.2  | Composition d'une puce de circuit intégré . . . . .   | 15 |
| Figure 2.3  | Évolution du nombre de transistors dans les processeurs Intel . . . . .                                       | 16 |
| Figure 2.4  | Boîtier d'encapsulation avec puce retournée <sup>†</sup> . . . . .  | 18 |
| Figure 2.5  | Technologies de boîtiers d'encapsulation 3D . . . . .   | 18 |
| Figure 2.6  | Spectre électromagnétique . . . . .   | 24 |
| Figure 3.1  | Bloc homogène contenant une source de chaleur surfacique . . . . .  | 28 |
| Figure 3.2  | Géométrie et conditions aux limites considérées par Zhan <i>et al.</i> (Zhan and Sapatnekar, 2007) . . . . .  | 30 |
| Figure 3.3  | Décomposition du circuit proposée par Oh <i>et al.</i> (Oh et al., 2012) . . . . .                            | 31 |
| Figure 3.4  | Discrétisation du circuit en volumes de contrôle effectuée par Lee <i>et al.</i> (Lee et al., 2013) . . . . . | 32 |
| Figure 3.5  | Modèle compact d'un circuit intégré avec son boîtier d'encapsulation . . . . .                                | 33 |
| Figure 3.6  | Topologie typique d'un modèle compact DELPHI (Rosten and Lasance, 1995) . . . . .                             | 34 |
| Figure 3.7  | Modélisation compacte d'un circuit avec Hotspot (Huang et al., 2006) . . . . .                                | 36 |
| Figure 3.8  | Modélisation du boîtier d'encapsulation dans Hotspot (Huang et al., 2006) . . . . .                           | 36 |
| Figure 3.9  | Cellules de maillages tridimensionnels . . . . .  | 38 |
| Figure 3.10 | Discrétisation d'un domaine 2D à l'aide de la méthode FDM . . . . .   | 39 |
| Figure 3.11 | Volumes de contrôle sur un domaine polyédrique . . . . .  | 40 |
| Figure 3.12 | Problème thermique modèle . . . . .   | 40 |
| Figure 3.13 | Discrétisation du problème modèle en 2x2 cellules . . . . .   | 41 |
| Figure 3.14 | Méthodes classiques utilisées par les modèles numériques . . . . .  | 52 |
| Figure 4.1  | Problème de transfert de chaleur unidimensionnel . . . . .  | 55 |
| Figure 4.2  | Maillage orthogonal à $n_x \times n_y \times n_z$ cellules . . . . .  | 57 |
| Figure 4.3  | Méthode de projection . . . . .   | 61 |
| Figure 4.4  | Factorisation LU de la matrice A . . . . .  | 67 |

|             |   |    |
|-------------|---|----|
| Figure 4.5  | Décomposition d'un domaine $\Omega$ en quatre sous-domaines . . . . .   | 68 |
| Figure 4.6  | Problème original . . . . .   | 69 |
| Figure 4.7  | Décomposition de domaine . . . . .  | 70 |
| Figure 4.8  | Coloriage rouge-noir du problème . . . . .  | 71 |
| Figure 5.1  | Circuit Wioming (Santos et al., 2013) . . . . .   | 74 |
| Figure 5.2  | Plan de masse simplifié du circuit Wioming (Santos et al., 2013) . . .  | 75 |
| Figure 5.3  | Circuit Scc3D . . . . .   | 76 |
| Figure 5.4  | Plan de masse du circuit Scc3D . . . . .  | 76 |
| Figure 5.5  | Modélisation d'un téléphone intelligent . . . . .   | 77 |
| Figure 5.6  | Réponse transitoire du circuit Scc3D . . . . .  | 79 |
| Figure 5.7  | Simulation du régime transitoire avec SuperLU et la méthode proposée  | 83 |
| Figure 5.8  | Simulation du régime permanent pour le problème iPhone 5 par la<br>méthode du gradient conjugué avec un préconditionnement ILU(0) et<br>la méthode proposée . . . . . | 86 |
| Figure 6.1  | Éléments du système de validation . . . . .   | 87 |
| Figure 6.2  | Système de validation . . . . .   | 88 |
| Figure 6.3  | Capture d'écran du logiciel Flir Tool . . . . .   | 89 |
| Figure 6.4  | Plan de masse de l'atom D2700 . . . . .   | 89 |
| Figure 6.5  | Structure d'une cellule de Peltier <sup>†</sup> . . . . .   | 90 |
| Figure 6.6  | Distribution des billes du boîtier d'encapsulation de l'Atom D2700 ATOM   | 92 |
| Figure 6.7  | Modèle thermique proposé . . . . .  | 93 |
| Figure 6.8  | Discrétisation du système étudié . . . . .  | 94 |
| Figure 6.9  | Cartes thermiques en régime permanent pour le scénario 2 . . . . .  | 95 |
| Figure 6.10 | Cartes thermiques en régime permanent . . . . .   | 96 |

**LISTE DES SIGLES ET ABRÉVIATIONS**

|       |   |
|-------|---|
| 3D    | Three-dimensional                       |
| ADI   | Alternating Direct Implicit             |
| C4    | Controlled Collapse Chip Connection     |
| CMOS  | Complementary Metal Oxyde Semiconductor |
| CPU   | Central Processing Unit                 |
| DRAM  | Dynamic Random Access Memory            |
| FDM   | Finite Difference Method                |
| FEM   | Finite Element Method                   |
| FVM   | Finite Volume Method                    |
| GMRES | Generalised Minimal Residual Method     |
| GPU   | Graphics Processing Unit                |
| ILU   | Incomplete Lower Upper                  |
| LU    | Lower Upper                             |
| SCC   | Single Chip Cloud Computer              |
| TIM   | Thermal Interface Material              |

## CHAPITRE 1 INTRODUCTION

Ce chapitre présente le problème de la dissipation thermique dans les circuits intégrés modernes. Les enjeux et les défis de la simulation thermique appliquée à la conception des circuits intégrés sont analysés. Ce chapitre énonce également l'hypothèse de recherche et les contributions apportées par cette thèse.

### 1.1 Définitions et contexte

#### 1.1.1 La dissipation thermique des circuits intégrés

Dès la fabrication des premiers circuits intégrés dans les années 1960, les ingénieurs ont été confrontés au problème de la dissipation thermique. Durant leur fonctionnement, les circuits intégrés dégagent une chaleur importante au niveau des transistors et des connexions métalliques. Le passage du courant à travers la résistance d'un transistor ou d'une connexion consomme de la puissance électrique, dont la plus grande partie est dissipée sous forme de chaleur par effet Joule (Weste and Harris, 2010). Du point de vue physique, la chaleur correspond à un transfert d'énergie. Lorsqu'elle s'accumule dans le circuit, elle entraîne une élévation de la température néfaste pour le circuit. De manière générale, l'élévation de température dégrade les performances et diminue la fiabilité des circuits intégrés (Vassighi and Sachdev, 2006).

L'adoption de la technologie CMOS dans les années 1980 a pendant quelques années apporté une solution au problème de la dissipation thermique. Les transistors CMOS avaient initialement l'avantage de consommer très peu de puissance. Les premiers processeurs réalisés avec la technologie CMOS affichaient une consommation de puissance excellente, tel le i386 d'Intel dont la densité de puissance n'excédait pas  $2 W/cm^2$  (Weste and Harris, 2010). Cet état de grâce a cependant été de courte durée, car l'industrie microélectronique, en quête de performances, s'est lancée dans une course effrénée à l'intégration de composants. Durant plus d'une décennie, l'objectif a été de miniaturiser les transistors afin d'en intégrer davantage et de les faire fonctionner à des fréquences plus élevées. Pour alimenter tous ces transistors en énergie, les ingénieurs ont rapidement abandonné la tension d'alimentation historique de 5 V et adopté des tensions plus basses. L'utilisation de tensions d'alimentation plus basses a permis de réduire la consommation de puissance dynamique, mais a eu pour effet négatif d'augmenter la consommation de puissance statique. La miniaturisation des transistors dont l'épaisseur de grille est désormais de quelques atomes seulement, a rendu les phénomènes

quantiques non négligeables, augmentant davantage le courant de fuite et la consommation de puissance. Durant les trente dernières années, la densité de puissance des circuits intégrés n'a cessé d'augmenter. La figure 1.1 retrace l'évolution de la densité de puissance dans les processeurs Intel. Malgré l'adoption de la technologie CMOS, les problèmes de consommation de puissance et de dissipation thermique sont rapidement retournés au devant de la scène.

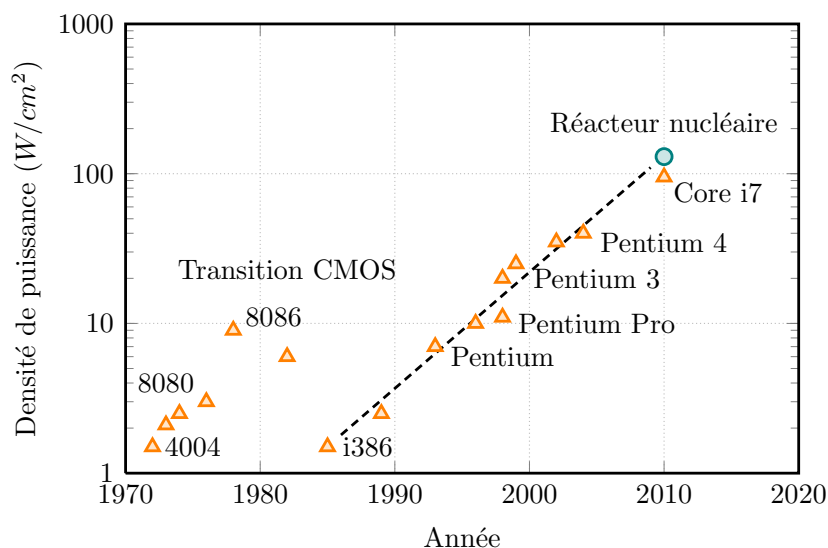


Figure 1.1 Évolution de la densité de puissance dans les processeurs Intel<sup>†</sup>

La figure 1.1 montre que si Intel avait continué à augmenter la fréquence de ses processeurs au même rythme, ceux-ci auraient atteint en 2010 une densité de puissance comparable à celle d'un réacteur nucléaire ! C'est une des raisons principales expliquant le virage brutal vers les processeurs multicœurs pris par Intel au milieu des années 2000. En 2009, Intel a lancé le Core i7, un de ses processeurs les plus puissants, puisqu'il dissipe une puissance maximum d'environ 95 W, mais il s'agit d'un processeur contenant 4 coeurs et non un seul comme le Pentium 4.

Les circuits intégrés modernes comportent des centaines de millions de transistors, qui dissipent une chaleur importante sur une surface réduite, d'où une densité de puissance très élevée. À titre de comparaison, une ampoule d'éclairage de 60 W, dont la surface est de  $120 \text{ cm}^2$ , est trop chaude pour pouvoir être touchée à la main. Un processeur moderne tel le Core i7 d'Intel dissipe la même puissance sur une surface d'à peine  $3 \text{ cm}^2$ , soit un dégagement de chaleur 40 fois plus intense. Le Core i7 est pourtant conçu pour équiper des ordinateurs portables, il est donc pourvu de toutes les améliorations technologiques possibles, pour réduire sa consommation de puissance et prolonger ainsi la durée de vie de la batterie. Il dégage

<sup>†</sup> Figure reproduite de (Weste and Harris, 2010)

cependant une énergie considérable dans un environnement confiné (le boîtier de l'ordinateur portable). Dissiper la chaleur de tels circuits est devenu un véritable défi technologique.

La chaleur générée en profondeur dans un circuit intégré doit souvent traverser plusieurs couches de matériaux avant de pouvoir rejoindre l'environnement ambiant. La figure 1.2 indique les principaux chemins empruntés par la chaleur pour sortir d'un circuit intégré type.

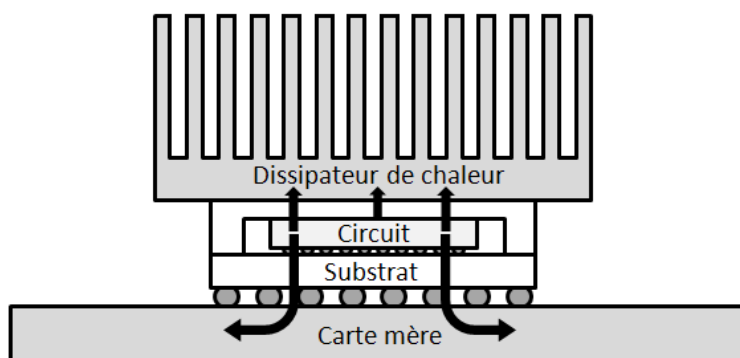


Figure 1.2 Chemins suivis par la chaleur pour sortir du boîtier d'encapsulation

La chaleur commence par se propager dans tout le circuit, ainsi que dans les couches adjacentes. Suivant le chemin emprunté, la chaleur atteint plus ou moins rapidement le substrat puis le boîtier d'encapsulation. Elle peut ensuite soit rejoindre le dissipateur de chaleur situé sur le haut du boîtier d'encapsulation pour être dissipée dans l'air par convection, soit se diffuser dans la carte mère en passant par les billes de contact entre le boîtier d'encapsulation et la carte mère. En théorie, la chaleur peut également passer dans l'air ambiant par convection naturelle sur les côtés du boîtier d'encapsulation, en pratique le boîtier d'encapsulation est isolant et la quantité de chaleur qui s'échappe par ce chemin est négligeable (Weste and Harris, 2010).

Les progrès réalisés dans les technologies de boîtier d'encapsulation et les dissipateurs de chaleur ont permis de repousser les limites de la dissipation thermique des boîtiers d'encapsulation. Les flux de chaleur évacués du circuit, sont passés de 8 W en 1985 à 130 W en 2008 (Weste and Harris, 2010). Cependant, les dissipateurs de chaleur, même les plus performants équipés de ventilateurs, ont atteint leur limite, contraignant à des efforts supplémentaires pour réduire la consommation de puissance.

De nombreuses techniques comme la coupure des signaux d'horloge, ont été développées pour optimiser la gestion de l'énergie et réduire la consommation de puissance dans les circuits intégrés. La tension d'alimentation a été abaissée jusqu'à une valeur très basse proche de 1V,

laissant peu de marge pour opérer une réduction draconienne de la dissipation de puissance dans les toutes prochaines années. L'industrie microélectronique s'efforce aujourd'hui de modéliser plus finement la dissipation thermique dans les circuits intégrés afin d'atténuer ses effets néfastes.

### 1.1.2 La simulation thermique des circuits intégrés

La simulation thermique des circuits intégrés s'est développée avec le besoin croissant de prédire et de contrôler la température dans les circuits intégrés. L'augmentation de la densité de puissance a contraint les ingénieurs à tenir compte du comportement thermique du circuit dès les premières phases de la conception. Étant donné le coût de développement d'un circuit et les pénalités en cas de surchauffe, la question de la dissipation thermique ne peut être laissée au hasard lors de la conception. L'époque où il était possible de trouver, après la conception du circuit, un dissipateur de chaleur assez puissant pour le refroidir, est désormais révolue. Aujourd'hui, la simulation thermique occupe une place centrale dans le processus de développement d'un circuit intégré (Weste and Harris, 2010; Loh et al., 2007).

La simulation thermique d'un circuit intégré vise à reproduire son comportement thermique, afin de prédire la température de ses composants. Pour cela, elle s'appuie sur un modèle du circuit. L'objectif de la simulation thermique est de s'assurer que le circuit en cours de conception sera capable de dissiper la chaleur produite, afin de maintenir une température supportable par les composants électroniques. Les simulations thermiques sont très souvent effectuées à un stade du flot de conception où le circuit physique n'est pas encore disponible. Il n'est donc pas possible de réaliser des mesures de température sur le circuit réel et la simulation thermique est le seul moyen de s'assurer que la température n'aura pas d'effet néfaste.

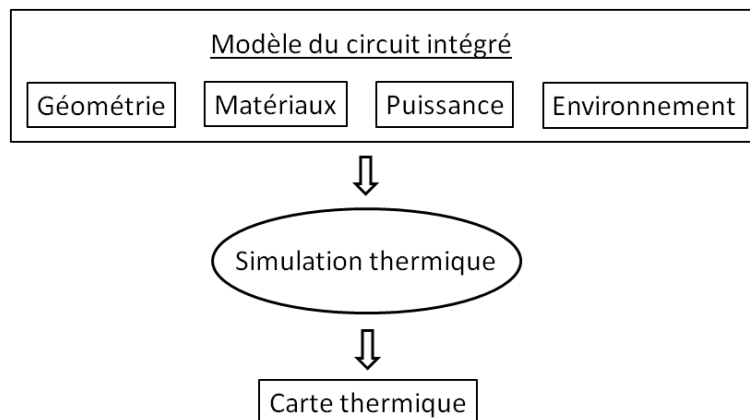


Figure 1.3 Simulation thermique de circuit intégré



La figure 1.3 présente le principe de la simulation thermique des circuits intégrés. En entrée, un modèle du circuit est nécessaire. Le modèle décrit la structure physique du circuit, c'est-à-dire la géométrie des composants, les matériaux de fabrication utilisés, ainsi que la position des sources de chaleur. Le modèle indique également les conditions environnementales, comme la température de la pièce. En sortie, la simulation thermique fournit une carte thermique du circuit. En interne, la simulation thermique résout des équations de physique pour calculer les transferts de chaleur dans le circuit et en déduire la distribution de température.

Beaucoup de modèles représentent un circuit intégré comme un empilement de couches homogènes, c'est-à-dire constituées d'un même matériau. Cette vision n'est valable que jusqu'à un certain niveau de détail. Une simulation thermique dont l'objectif est d'analyser finement la dissipation de chaleur au niveau des broches du circuit, ne peut se contenter de représenter ces connexions comme une couche homogène.

Suivant les objectifs poursuivis, quatre niveaux de modélisation sont possibles pour la simulation thermique des circuits intégrés. On distingue ainsi les simulations aux niveaux : système, boîtier d'encapsulation, puce et composant. Le tableau 1.1 résume les caractéristiques des différents niveaux de modélisation possibles pour la simulation thermique des circuits intégrés.

La simulation thermique est aujourd'hui une étape incontournable pour comprendre et analyser le comportement thermique des circuits intégrés. Cependant, de nombreux défis doivent être encore relevés pour répondre au besoin de simulation thermique des circuits intégrés modernes.

### 1.1.3 Les défis actuels de la simulation thermique

Les circuits intégrés ont beaucoup évolué depuis leur invention il y a plus de cinquante ans. Leur structure s'est considérablement complexifiée, aussi bien au niveau microscopique des transistors qu'à l'échelle macroscopique du boîtier d'encapsulation. Cette évolution spectaculaire suscite de nouveaux besoins concernant la simulation thermique.

Les défis actuels de la simulation thermique des circuits intégrés sont avant tout liés à l'apparition des technologies tridimensionnelles (3D), qui ont bouleversé la structure des circuits traditionnels. Les technologies 3D suscitent depuis quelques années un intérêt grandissant dans le monde de la microélectronique en raison de leurs capacités d'intégration importantes, dans un contexte où la miniaturisation des transistors connaît un ralentissement.

Les circuits intégrés 3D issus des technologies 3D, peuvent être définis au sens large comme un assemblage vertical de structures fabriquées séparément, voire dans des technologies dif-

Tableau 1.1 Niveaux de modélisation thermique

| Niveau    | Objectifs   | Échelle              | Transferts thermiques   |
|-----------|---|----------------------|---|
| Systeme   | <ul style="list-style-type: none"> <li>— Identification des points chauds sur la carte mère</li> <li>— Placement des circuits sur la carte mère</li> <li>— Dimensionnement du système de refroidissement</li> </ul> | $\geq 5cm$           | <ul style="list-style-type: none"> <li>— Conduction</li> <li>— Convection</li> <li>— Rayonnement</li> </ul> |
| Boîter    | <ul style="list-style-type: none"> <li>— Caractérisation de la résistance thermique</li> <li>— Dimensionnement du dissipateur de chaleur</li> </ul>   | $[3cm, 7cm]$         | <ul style="list-style-type: none"> <li>— Conduction</li> <li>— Convection</li> </ul>                        |
| Puce      | <ul style="list-style-type: none"> <li>— Identification des points chauds</li> <li>— Placement des unités fonctionnelles</li> </ul>   | $[1cm, 3cm]$         | <ul style="list-style-type: none"> <li>— Conduction</li> </ul>  |
| Composant | <ul style="list-style-type: none"> <li>— Etude fine du champ de température</li> </ul>  | $[1\mu m, 100\mu m]$ | <ul style="list-style-type: none"> <li>— Conduction</li> </ul>  |

férentes (Patti, 2006). Les circuits 3D se distinguent des circuits intégrés traditionnels, où l'intégration de nouveaux composants ne peut se faire que dans le plan horizontal. Les circuits intégrés 3D exploitent une troisième dimension pour créer des structures plus compactes, intégrant davantage de fonctionnalités, et combinant des technologies hétérogènes (Patti, 2006). Les circuits 3D peuvent ainsi intégrer des circuits analogiques ou numériques, des communications optiques et des éléments électromécaniques. Les circuits intégrés 3D offrent une solution technologique économiquement viable pour répondre au besoin d'intégrer toujours plus de fonctionnalités dans les systèmes numériques modernes.

La figure 1.4 illustre quelques-unes des nombreuses technologies de circuits intégrés 3D existantes.

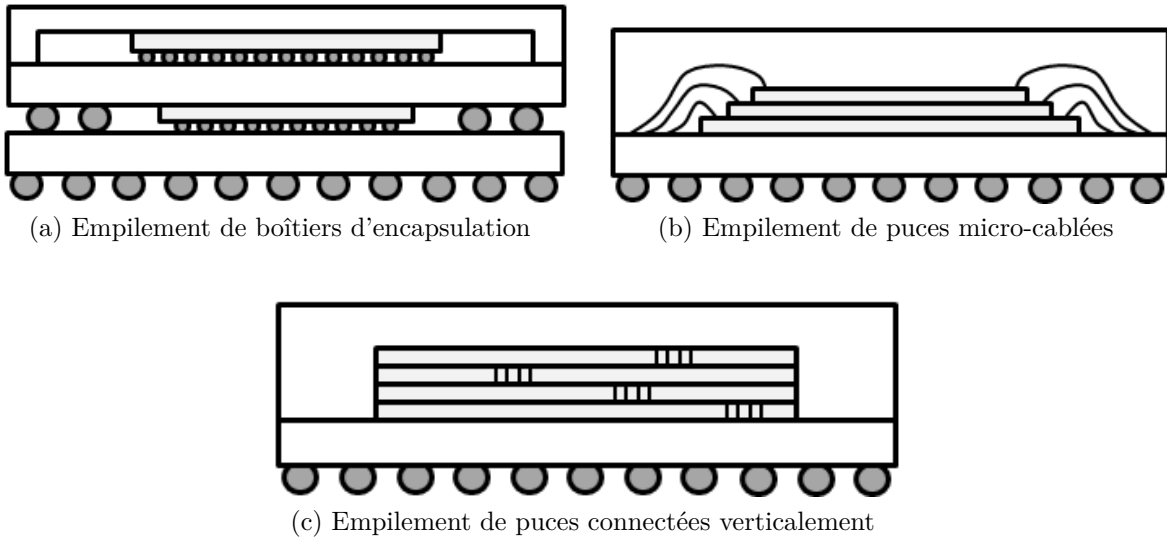


Figure 1.4 Technologies de circuits intégrés 3D

Les premières formes d'intégration 3D, datent de la fin des années 1990 et consistent à empiler plusieurs boîtiers d'encapsulation pour économiser de l'espace sur la carte mère. Ces structures ne sont cependant pas très compactes et ne permettent pas une densité d'interconnexion très élevée entre les différents circuits. Rapidement, sont apparues les technologies consistant à empiler plusieurs puces au sein d'un même boîtier d'encapsulation et à les relier par des micro-cablages à la périphérie des circuits. Plus récemment, de nouvelles technologies d'intégration 3D ont été développées, reposant sur l'utilisation d'interconnexions verticales, qui traversent le circuit de part en part, pour relier les différentes couches entre elles.

Les technologies d'intégration 3D à base de connexions verticales offrent des densités d'interconnexions très élevées et sont aujourd'hui au cœur des recherches sur l'intégration 3D (Lau, 2011). Les circuits 3D ont l'avantage d'être plus compacts et de permettre des densités d'intégration plus élevées que les circuits traditionnels. Ceci est particulièrement vrai pour les circuits 3D basés sur l'utilisation de connexions verticales.

Les circuits intégrés 3D ont l'inconvénient de concentrer les sources de chaleur des différentes couches sur la même surface. Les flux de chaleur générés par les différentes couches d'un circuit 3D s'additionnent. Ainsi, un circuit 3D contenant 4 à 8 couches est susceptible de produire un flux de chaleur de plusieurs centaines de  $W/cm^2$  (Pavlidis and Friedman, 2009). En raison de leur structure compacte, les circuits 3D sont particulièrement exposés aux problèmes de dissipation thermique. La simulation thermique est absolument indispensable à la

mise en place de ces nouvelles technologies. Cependant, l'introduction de boîtiers d'encapsulation 3D ou de connexions verticales complexifie la géométrie des modèles et rend inefficaces les approches utilisées classiquement pour la simulation thermique.

Pour compléter cette description, ajoutons que la nature des phénomènes physiques à simuler a également changé. Les dissipateurs de chaleur traditionnels à base de ventilateurs sont incapables d'évacuer des flux de chaleur aussi élevés que ceux rencontrés dans les circuits 3D. Des systèmes de refroidissement plus efficaces ont été développés spécifiquement pour les circuits 3D très puissants destinés au marché des super ordinateurs. De petites conduites, appelées micro-canaux, sont gravées dans le substrat de silicium afin d'y faire circuler un fluide pour évacuer la chaleur du circuit par convection forcée. La figure 1.5 illustre le principe du refroidissement d'un circuit intégré à l'aide d'un liquide.

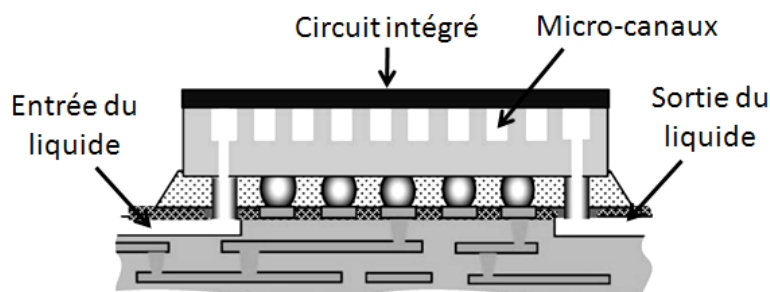


Figure 1.5 Refroidissement d'un circuit intégré à l'aide de liquide

L'introduction de micro-canaux dans le circuit nécessite de simuler également des phénomènes de mécanique des fluides. Certains micro-canaux permettent même un écoulement de fluide avec changement de phase. Le fluide liquide entre en ébullition et absorbe alors plus de chaleur lors du passage de l'état liquide à l'état vapeur. Ces progrès techniques s'accompagnent de nouveaux défis de simulation. Globalement, la simulation thermique des circuits intégrés modernes doit prendre en compte des structures plus complexes et des phénomènes physiques plus variés.

## 1.2 Éléments de la problématique

### 1.2.1 Analyse des requis

L'analyse des requis vise à déterminer les éléments essentiels qui doivent être pris en compte lors de la simulation thermique des circuits intégrés modernes.

On peut distinguer deux cas d'utilisation des simulations thermiques de circuits intégrés :

lors de la synthèse du circuit, notamment lors des opérations de placement et de routage, et lors de la phase de validation. Lors de la synthèse, l'architecture du circuit n'est pas encore connue et il s'agit d'obtenir une première estimation de la température. Les simulations thermiques sont plutôt utilisées pour déterminer la température en régime permanent, c'est-à-dire la température du circuit lorsqu'elle est stabilisée après un intervalle de temps plus ou moins long. Cette notion sera précisée dans la section 2.2.4. Lors de la phase de validation, l'architecture du circuit est bien définie. Il s'agit alors d'affiner l'analyse thermique et les simulations thermiques sont utilisées pour étudier le régime transitoire, avant que le régime permanent soit atteint. Il est donc essentiel de pouvoir simuler à la fois le régime permanent et le régime transitoire.

Les circuits intégrés modernes, en particulier les circuits 3D, se distinguent par des géométries compliquées de type empilement de boîtiers d'encapsulation ou de puces, avec des dimensions différentes. La simulation thermique de ces dispositifs doit s'appuyer sur des modèles réalistes, représentant fidèlement les différents composants du circuit et leurs matériaux. Il est également important de simuler les transferts thermiques dans les trois dimensions pour obtenir des résultats fiables. Par ailleurs, l'apparition de nouveaux systèmes de refroidissement basés sur l'écoulement d'un fluide dans le circuit implique d'élargir la nature des transferts de chaleurs simulés.

La simulation thermique de circuits intégrés modernes doit prendre en compte :

- le régime permanent et le régime transitoire afin de simuler l'intégralité du comportement thermique d'un circuit intégré
- les géométries complexes de type empilement de boîtiers d'encapsulation ou empilement de puces de dimensions différentes, afin de pouvoir modéliser correctement la géométrie des systèmes modernes
- les transferts de chaleur suivant les trois dimensions, afin de représenter fidèlement la nature tri-dimensionnelle des systèmes réels
- les sources de chaleur multiples, afin d'évaluer correctement la dissipation de chaleur dans les circuits intégrés
- les échelles de temps et de longueur différentes, afin de modéliser des composants de taille différente et des phénomènes de durée différente

La section suivante propose trois critères simples pour évaluer les méthodes de simulation thermique et aborde la problématique cruciale des performances.

### 1.2.2 Critères d'évaluation

Les méthodes de simulation thermique des circuits intégrés peuvent être évaluées à partir des trois critères suivants :

- le temps d'exécution
- la consommation de mémoire
- la précision des résultats

Le temps d'exécution d'une simulation thermique est le temps mis pour simuler le comportement du circuit et calculer la distribution de température. La plupart des méthodes de simulation thermique sont très coûteuses en terme de calcul et requièrent de longs temps de simulation. Cette complexité limite l'utilisation des modèles thermiques dans les outils de conception, aussi bien dans les environnements de simulation que dans les outils de synthèse.

La consommation de mémoire d'une simulation thermique est la quantité de mémoire vive utilisée pour calculer la température du circuit simulé. Le stockage des valeurs de température produites au cours de la simulation n'est pas pris en compte dans le calcul de la mémoire consommée, car ces résultats sont généralement écrits dans une mémoire externe. La consommation de mémoire conditionne la faisabilité de la simulation. À la différence du temps d'exécution, la consommation de mémoire est un critère binaire, qui ne pose pas de problème tant que la machine dispose d'assez de mémoire. Il est souvent possible de se donner plus de temps pour effectuer la simulation, mais il est généralement beaucoup plus difficile d'ajouter de la mémoire à la machine. La consommation de mémoire lors d'une simulation thermique est fortement liée à la complexité de la méthode de calcul utilisée. Très souvent, une simulation lente consomme beaucoup de mémoire et une simulation rapide en consomme peu. Cependant, ce n'est pas une règle absolue et il est nécessaire d'évaluer la consommation de mémoire en plus du temps d'exécution.

La précision d'une simulation thermique correspond à l'écart entre la température prédite par la simulation et la température que l'on pourrait mesurer sur le circuit réel. Une simulation est précise si elle commet une erreur faible sur les valeurs de température annoncées. Un circuit intégré étant une structure 3D, l'erreur varie dans l'espace. Lorsque le comportement thermique du circuit est simulé dans le temps, l'erreur varie également dans le temps. Une métrique commode est l'erreur maximale, dans tout le circuit ou tout au long de la simulation, afin de garantir un niveau minimum de précision. Généralement, la température exacte en tout point du circuit réel n'est pas connue, il est donc difficile de quantifier de manière certaine l'erreur de la simulation. Une approche courante est de se comparer avec un autre modèle désigné alors comme modèle de référence. L'analyse de l'erreur maximale est assez grossière. Pour mieux évaluer la précision de la simulation, il convient alors de calculer la distribution

de l'erreur pour mieux évaluer la précision de la simulation.

Les attentes par rapport à ces critères peuvent varier suivant les cas d'utilisation. Si la simulation thermique est effectuée lors de la synthèse pour décider du placement, le but est de faire une estimation rapide de la température. Il est alors important que la simulation thermique soit rapide, afin d'explorer de multiples configurations. Tant pis si le modèle est alors un peu moins précis, car de toute façon, les détails de l'architecture ne sont pas connus et il serait illusoire de penser faire un modèle très précis à ce stade. En revanche, pour valider l'architecture finale du circuit, une analyse fine du régime transitoire est souvent nécessaire. Il est alors préférable de privilégier la précision plutôt que la rapidité de la simulation.

### 1.3 Hypothèse de recherche et contributions

#### 1.3.1 Hypothèse de recherche

Cette thèse s'intéresse tout particulièrement aux méthodes de calcul utilisées pour faire la simulation thermique de circuits intégrés à l'aide de modèles numériques. La revue de littérature au chapitre 3 justifie le choix de ce sujet précis.

L'hypothèse de recherche considérée dans cette thèse est la suivante : *La combinaison de méthodes de calcul directes et itératives peut accélérer d'au moins un ordre de grandeur par rapport à l'état-de-l'art, la simulation thermique des circuits intégrés à l'aide de modèles numériques, tout en maintenant le même niveau de précision.*

La revue de littérature présentée au chapitre 3 montre que la plupart des modèles thermiques de circuits intégrés proposés se contentent d'utiliser des méthodes de calcul générales pour faire la simulation thermique de circuits intégrés. Malheureusement, les méthodes de calcul suffisamment générales pour être applicables à tout type de problème rencontrent souvent des performances médiocres et deviennent pénalisantes pour la simulation thermique de circuits modernes. La section suivante fixe plusieurs objectifs de recherche afin de contribuer à l'état de l'art dans le domaine de la simulation thermique des circuits intégrés.

#### 1.3.2 Objectifs de recherche et contributions

##### Objectifs de recherche :

Un premier objectif de recherche de cette thèse est de permettre la simulation thermique des circuits intégrés modernes. Dans un premier temps, le principe théorique sur lequel repose la simulation thermique doit être compris afin de déterminer la démarche générale à adopter ainsi que les éléments structurels des circuits qui doivent être modélisés. Dans un deuxième

temps, les méthodes existantes pour la simulation thermique (formules analytiques, modèles numériques, etc.) doivent être répertoriées et évaluées afin d'identifier leurs avantages et leurs inconvénients. Dans un troisième temps, de nouvelles méthodes doivent être mises en place afin d'apporter une solution aux limitations des méthodes existantes.

Un deuxième objectif est le développement d'un simulateur thermique pour circuits intégrés, capable de simuler le régime transitoire et le régime permanent. Le simulateur thermique proposé est un logiciel pourvu d'une interface (textuelle dans un premier temps) permettant de décrire le circuit à simuler et son environnement. Une syntaxe concise et flexible doit être définie afin de permettre aux utilisateurs du simulateur de décrire rapidement les différents composants du circuit. Le simulateur doit également fonctionner sous les environnements Windows, Linux et MacOS/X pour être accessible à une plus grande communauté d'utilisateurs.

Un troisième objectif est la validation du simulateur thermique développé. Dans un premier temps, le simulateur proposé peut être comparé à un autre logiciel de simulation. Plusieurs logiciels spécialisés dans la simulation thermique de circuits intégrés, tels Hotspot (Huang et al., 2006) et 3D-ICE (Sridhar et al., 2010) développés par des universités, peuvent être utilisés. Pour une validation plus approfondie, la confrontation à un simulateur commercial plus général, mais plus éprouvé, tels ANSYS (ANSYS) ou COMSOL (COMSOL) est sans doute préférable. Dans un deuxième temps, le simulateur thermique proposé peut être validé à l'aide de mesures de température effectuées sur un circuit réel. Pour cela, un système de validation doit être mis en place. Un circuit de test équipé de sondes de température peut fournir de précieuses informations sur la température interne. Une alternative est d'utiliser une caméra infrarouge pour réaliser des mesures de température à la surface du circuit.

### **Contributions :**

Une première contribution apportée par cette thèse est la proposition de deux méthodes de calcul performantes (au sens des critères énoncés en 1.2.2) pour simuler le comportement thermique des circuits intégrés modernes en régime transitoire et en régime permanent. Ces méthodes sont applicables aux modèles thermiques numériques de circuits intégrés.

Une deuxième contribution apportée par cette thèse est l'évaluation des méthodes de calcul proposées sur des problèmes thermiques réalistes et variés. Les méthodes proposées sont comparées à des méthodes de référence faisant actuellement autorité dans le domaine de la simulation thermique. Cette comparaison permet de cerner les points forts des méthodes proposées et leur champ d'application.



Une troisième contribution apportée par cette thèse est le simulateur thermique développé, appelé « ICTherm », qui implémente les méthodes de calcul proposées. Ce simulateur offre un outil pratique et performant pour analyser le comportement thermique de nouvelles architectures de circuits intégrés. Il est actuellement utilisé à l'école Polytechnique de Montreal (Canada) et à l'école Centrale de Lyon (France), à des fins de recherche, notamment sur les circuits intégrant des communications optiques, particulièrement sensibles aux effets de la température.

## CHAPITRE 2 CONCEPTS DE BASE

Depuis leur invention en 1958, les circuits intégrés ont subi une évolution spectaculaire. Ce chapitre présente les composants essentiels des circuits intégrés et leur structure, ainsi que le principe théorique de la simulation thermique.

### 2.1 Description structurelle des circuits intégrés

#### 2.1.1 Composition typique d'un circuit intégré

Un circuit intégré est formé d'un boîtier d'encapsulation contenant une ou plusieurs puces. La puce est l'élément de base, de forme rectangulaire, sur laquelle est implémentée la fonction du circuit. Il s'agit d'une plaque mince, réalisée dans un matériau semi-conducteur, généralement le silicium, sur laquelle sont gravés les transistors, nécessaires au fonctionnement du circuit. Ses dimensions latérales varient généralement entre 1 et 3 cm et son épaisseur est de l'ordre de quelques centaines de micromètres. Pour des raisons structurelles liées à sa fabrication, une puce n'excède généralement pas 3 cm de longueur ou de largeur. La figure 2.1 présente la structure type d'un circuit intégré.

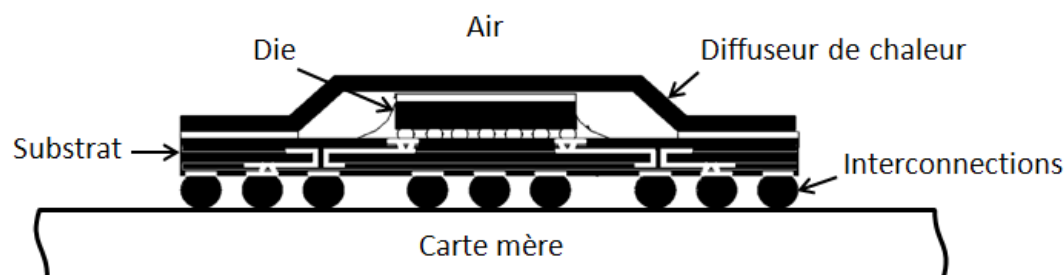


Figure 2.1 Structure type d'un circuit intégré

Le boîtier d'encapsulation remplit plusieurs fonctions. Premièrement, il protège la puce de l'environnement extérieur, en la maintenant à l'abri de la poussière et des chocs. Deuxièmement, il permet de connecter électriquement la puce à la carte mère. Historiquement, les premiers boîtiers d'encapsulation utilisaient des broches comme connecteurs, puis sont apparues les pattes et enfin les boules de soudure. Troisièmement, le boîtier d'encapsulation permet de dissiper la chaleur produite par le circuit dans l'environnement ambiant. Le boîtier d'encapsulation incorpore à cet effet un système de refroidissement comme un diffuseur de chaleur, éventuellement complété par un dissipateur de chaleur. Le diffuseur de chaleur est

généralement une simple plaque de métal, faite de cuivre ou d'aluminium, dont l'épaisseur varie entre 1 et 2 *mm*. Les dissipateurs de chaleur sont utilisés pour les circuits plus puissants, afin de maximiser le flux de chaleur évacué du boîtier d'encapsulation.

### 2.1.2 Puces, transistors et interconnexions

La puce est fabriquée à partir d'un mince substrat de silicium, sur lequel sont déposées plusieurs couches de matériaux, destinées à accueillir les transistors et les interconnexions. La puce est fabriquée couche par couche. Les transistors et les interconnexions sont réalisés au moyen d'un procédé appelé photolithogravure. Chaque nouvelle couche de la puce est d'abord recouverte d'un film de résine photosensible, sur lequel sont imprimés des motifs en faisant passer un rayonnement lumineux à travers un masque. Les matériaux dans les zones non protégées par la résine sont ensuite retirés par un procédé chimique. Le processus est répété pour chaque couche de la puce à l'aide d'un ensemble de masques. L'étape finale consiste à recouvrir le circuit d'une couche protectrice, formée d'un oxyde, appelée couche de passivation. La figure 2.2 représente la structure en couches d'une puce de circuit intégré.

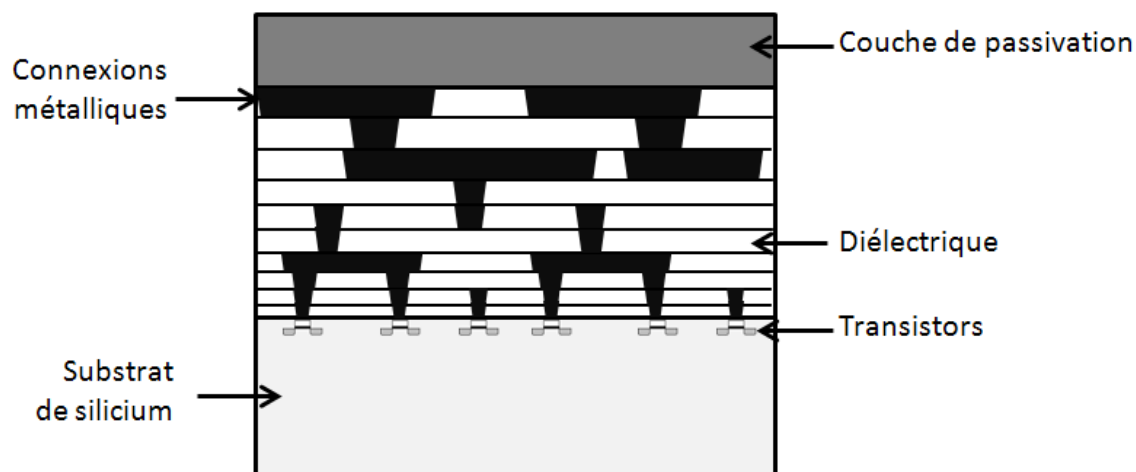


Figure 2.2 Composition d'une puce de circuit intégré

Les couches constituant la puce ont des épaisseurs très différentes. Les transistors atteignent seulement quelques dizaines de nanomètres d'épaisseur, tandis que l'ensemble des connexions métalliques a une épaisseur de l'ordre de 10  $\mu m$ . Le substrat de silicium qui joue également le rôle de support mécanique, a généralement une épaisseur de l'ordre de 100  $\mu m$ . Le nombre de couches et leurs dimensions exactes varient d'une technologie à l'autre et d'un fabricant à l'autre. La technologie d'Intel en 22 *nm* (Auth et al., 2012), mise sur le marché en 2012,

utilise par exemple des transistors d'à peine 2 nm d'épaisseurs et 9 niveaux de connexions métalliques, pour une épaisseur totale de 8  $\mu m$ . Malgré la constante évolution des technologies, la puce d'un circuit intégré reste essentiellement formée d'un bloc de silicium, sur lequel se trouve un enchevêtrement de matériaux semi-conducteurs, métalliques et diélectriques. L'ensemble forme une structure très complexe.

Les transistors peuvent être vus, de manière simplifiée, comme des interrupteurs très rapides commandés soit en tension, soit en courant. Les transistors sont agencés et regroupés en circuits élémentaires, appelés portes logiques, afin de réaliser des fonctions logiques simples. La combinaison de ces portes logiques permet de réaliser des circuits plus gros, capables d'effectuer des opérations arithmétiques ou de mémoriser des valeurs. Ces circuits sont eux-mêmes combinés entre eux pour former des systèmes plus complexes comme des processeurs, des mémoires, ou même des systèmes-sur-puce. Malgré la complexité élevée des circuits intégrés modernes, le transistor reste l'unité de base.

Depuis les années 1960, l'industrie s'est efforcée de miniaturiser les transistors, afin d'en intégrer davantage sur la même puce. Un transistor plus petit est également plus rapide, car les électrons doivent parcourir moins de distance pour le traverser. Un circuit fabriqué avec des transistors plus petits peut donc fonctionner à une fréquence plus élevée et effectuer un plus grand nombre d'opérations par seconde.

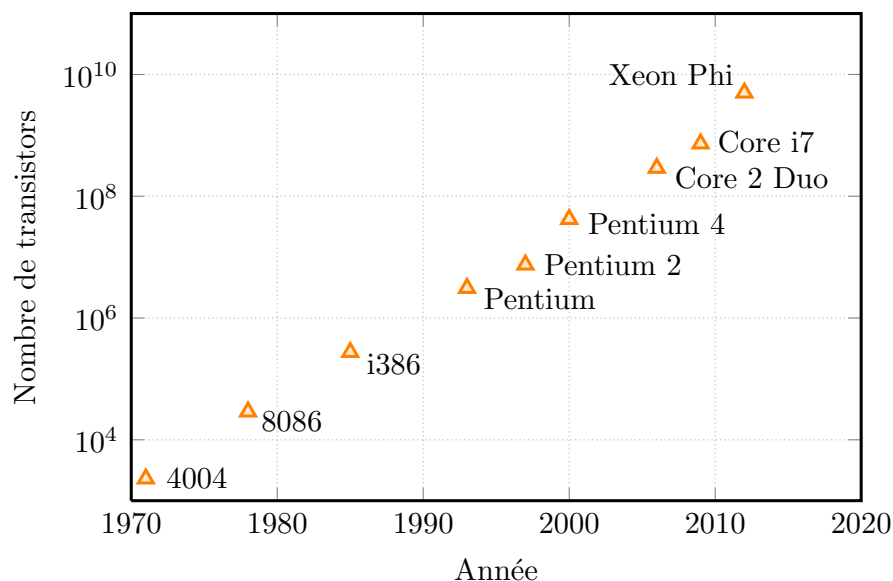


Figure 2.3 Évolution du nombre de transistors dans les processeurs Intel

La figure 2.3 montre l'évolution du nombre de transistors dans les processeurs Intel, depuis 1971 jusqu'à aujourd'hui. Ainsi que l'avait prédit Gordon Moore en 1965, le nombre de tran-

sistors intégrés dans une puce a suivi une progression exponentielle, en doublant quasiment tous les dix-huit mois. Ainsi, le processeur Xeon Phi d’Intel, mis sur le marché en 2012 et gravé en  $22\text{ nm}$ , contient quelques 5 milliards de transistors.

### 2.1.3 Évolution des boîtiers d’encapsulation

Parallèlement à la miniaturisation des transistors, les boîtiers d’encapsulation ont eux aussi subi une évolution importante. Pour concevoir un circuit performant, il est indispensable de considérer la conception du boîtier d’encapsulation lors de la conception du circuit. Idéalement, un boîtier d’encapsulation doit :

- connecter électriquement le circuit en minimisant le délai et la distorsion des signaux
- dissiper la chaleur produite par le circuit
- protéger le circuit des chocs
- avoir un faible coût de fabrication

Historiquement, les premiers circuits intégrés étaient reliés au boîtier d’encapsulation par des fils électriques soudés sur de petites zones rectangulaires, appelées « plages de connexions », situées à la périphérie du circuit. Afin de pouvoir souder un fil électrique sur une plage de connexion, la distance minimale entre deux plages était de l’ordre de  $100 - 200\ \mu\text{m}$ . Un circuit intégré de dimension  $1\text{ cm} \times 1\text{ cm}$  pouvait donc accueillir au plus une centaine d’entrée/sorties. Cette technologie convenait aux premiers circuits qui utilisaient peu d’entrées/sorties, comme le processeur 8086 d’Intel (Weste and Harris, 2010) qui ne comptait 18 entrées/sorties.

Dans les années 1990, beaucoup de fabricants ont opté pour la technologie de « puce retournée », afin de placer un nombre d’entrées/sorties plus élevé dans le boîtier d’encapsulation. Cette technologie, initialement développée par la société IBM, permet d’utiliser toute la surface du circuit pour connecter les entrées/sorties. Les plages de connexion sont placées sur toute la surface du circuit et en contact avec la dernière couche de métal. La puce est ensuite placée à l’envers dans son boîtier d’encapsulation, d’où le nom de la technologie. Sur les plages de connexions, sont soudées de petites billes, appelées « billes C4 ». La figure 2.4 présente une vue de côté d’un boîtier d’encapsulation de type « puce retournée ».

L’espacement minimal entre deux billes C4 est de l’ordre de  $100\ \mu\text{m}$ , ce qui permet de disposer de plusieurs milliers de connexions. Cette technologie a en outre l’avantage de rapprocher le circuit de la carte mère, éliminant ainsi les problèmes d’inductance associés aux longs fils électriques. Afin d’intégrer plus de composants et de raccourcir le délai des communications, les ingénieurs ont inventé les boîtiers d’encapsulation à plusieurs modules, qui contiennent plusieurs puces placées l’une à côté de l’autre sur un même substrat. Mise au point dans les années 1980, cette technologie était initialement réservée aux super ordinateurs.

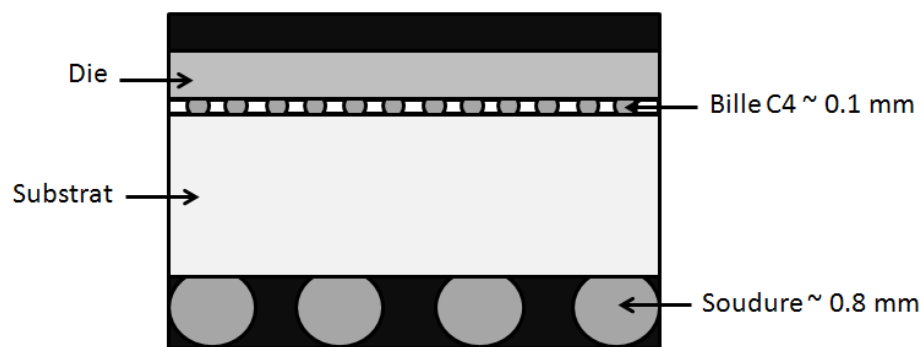


Figure 2.4 Boîtier d'encapsulation avec puce retournée<sup>†</sup>

Plus récemment, Intel a utilisé un boîtier d'encapsulation à plusieurs modules en 2010, pour associer son processeur Core i5 à un processeur graphique (COREi5).

Les boîtiers d'encapsulation à plusieurs modules présentent l'inconvénient d'occuper beaucoup d'espace sur la carte mère. Une solution efficace à ce problème est d'utiliser la dimension verticale. Ainsi les ingénieurs ont pensé à empiler les puces les unes sur les autres au sein d'un même boîtier d'encapsulation, ce qui a donné naissance aux boîtiers d'encapsulation 3D. À l'intérieur d'un boîtier d'encapsulation 3D, les puces sont reliées entre elles généralement à l'aide de micro-cablages. La puce située au bas de la pile peut éventuellement être retournée.

Une autre technologie de boîtier d'encapsulation 3D consiste à empiler directement les boîtiers plutôt que les puces. Cette technologie permet de tester chaque boîtier d'encapsulation séparément. Seuls les boîtiers d'encapsulation parfaitement fonctionnels sont ainsi assemblés, ce qui permet d'atteindre de très bons rendements lors de la production.

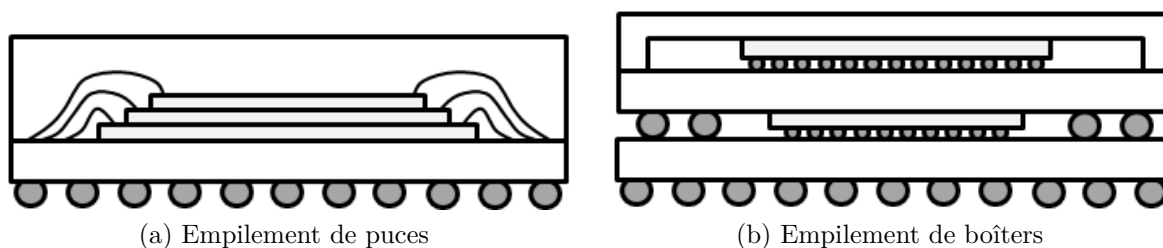


Figure 2.5 Technologies de boîtiers d'encapsulation 3D

Cette évolution des boîtiers d'encapsulation a permis d'intégrer plus de composants, augmentant davantage la densité de puissance et les risques liés à des températures trop élevées.

#### 2.1.4 Risques liés à une température élevée

Une température excessive dans un circuit intégré est néfaste à plusieurs points de vue. Les effets dommageables de la température sont par ordre croissant de dangerosité :

- la dégradation des performances en raison de la modification du comportement des transistors
- la réduction du temps de vie à cause du stress thermique et mécanique
- la destruction du composant par emballement thermique

L'augmentation de température conduit à une dégradation des performances du circuit. La température affecte les caractéristiques électriques des transistors, qui commutent moins rapidement et génèrent par un courant de fuite plus important. Le circuit consomme alors plus d'énergie tout en fonctionnant plus lentement. Une température non uniforme peut également entraîner un retard des signaux électriques en provoquant une variation du délai des portes logiques du circuit. Dans les cas extrêmes, l'augmentation de la température provoque une forte surconsommation électrique du circuit, faisant augmenter davantage la température. Le phénomène s'amplifie et conduit à un emballement thermique, pouvant provoquer la destruction du circuit.

Pour éviter les risques liés à l'élévation de température, les fabricants recommandent généralement de ne pas dépasser des températures de  $80 - 100^{\circ}\text{C}$  au sein des circuits intégrés. Pour maintenir une température acceptable, plusieurs systèmes de refroidissement ont été développés pour dissiper la chaleur des circuits intégrés.

#### 2.1.5 Systèmes de refroidissement

Les circuits intégrés modernes génèrent des flux de chaleur très élevés pouvant dépasser la centaine de watts. Cette chaleur, produite par effet joule dans les transistors et des connexions métalliques, doit être extraite du circuit sous peine d'une élévation de la température, qui serait néfaste aux composants électroniques. Extraire la chaleur des boîtiers d'encapsulation est devenu aujourd'hui un véritable défi technologique. Plusieurs systèmes de refroidissement ont été développés à cet effet, certains d'entre eux pouvant être combinés.

##### Diffuseurs de chaleur

Les diffuseurs de chaleur sont généralement de simples plaques de métal, faites de cuivre ou d'aluminium, placées directement sur le circuit ou faisant partie intégrante du boîtier d'encapsulation. Leur très bonne conductivité thermique et leur surface importante permettent de diffuser le flux de chaleur concentré dans le circuit. Pour être efficace, un diffuseur de

chaleur doit avoir une surface plus importante que celle du circuit. Si aucun autre système de refroidissement n'est utilisé, la chaleur s'échappe du diffuseur par convection naturelle avec l'air ambiant. Si le diffuseur de chaleur ne suffit pas, il est utilisé en combinaison avec un dissipateur de chaleur.

### **Dissipateurs de chaleur passifs et actifs**

Les dissipateurs de chaleurs existent en deux versions : passive et active. Les dissipateurs passifs utilisent la seule convection naturelle pour transmettre la chaleur du circuit à l'environnement. Leur conception est soigneusement optimisée afin de maximiser le transfert de chaleur avec l'air ambiant. Très souvent, ils ont une forme caractéristique avec des ailettes, augmentant la surface de contact avec l'air ambiant.

Les dissipateurs de chaleur actifs peuvent extraire davantage de chaleur en utilisant un ventilateur pour créer une circulation d'air dans les ailettes. La chaleur est alors évacuée par convection forcée. Plus l'air circule vite, plus le transfert thermique par convection forcée est important. La puissance du ventilateur peut être ajustée avec l'activité du circuit, pour éviter de surdimensionner inutilement le système de refroidissement. En plus d'être bruyant et encombrant, ce système de refroidissement a également l'inconvénient de consommer de l'électricité.

### **Refroidissement par un fluide**

Les technologies modernes de circuits intégrés, en particulier les technologies 3D, peuvent dégager des flux de chaleur de plusieurs centaines de watts. Très souvent, les dissipateurs de chaleur actifs équipés de puissants ventilateurs ne sont pas capables de dissiper des flux de chaleur aussi importants. Le refroidissement par un fluide représente une alternative efficace aux traditionnels dissipateurs de chaleur. Le principe du refroidissement par un fluide est d'extraire la chaleur directement des composants actifs du circuit en faisant circuler un fluide dans des micro-canaux gravés dans le substrat de silicium. L'inconvénient majeur de tels systèmes de refroidissement est bien sûr la complexité de mise en oeuvre. Un circuit d'alimentation avec un réservoir et une pompe sont nécessaires pour faire circuler le fluide. De tels dispositifs ne peuvent généralement pas être utilisés dans les systèmes embarqués. Le refroidissement par un fluide reste pour l'instant réservé aux super ordinateurs, comme l'Aquasar d'IBM (Zimmermann et al., 2012).



## 2.2 Théorie des transferts de chaleur

### 2.2.1 Définition de la température

Dans la vie de tous les jours, la température est une grandeur physique assez intuitive, liée à la sensation de chaud et de froid, dont on fait l'expérience à l'aide d'un simple thermomètre. En physique, la température traduit à l'échelle macroscopique l'agitation cinétique des particules microscopiques qui constituent la matière. Plusieurs approches sont possibles pour définir la température. La théorie cinétique des gaz parfaits définit la température  $T$  d'un gaz parfait comme une mesure de l'énergie cinétique moyenne  $\langle E_c \rangle$  de ses particules, en faisant intervenir la constante de Boltzmann  $k_B$ .

$$T = \frac{2 \langle E_c \rangle}{3k_B} \quad (2.1)$$

Une autre définition de la température est donnée par la thermodynamique, qui définit la température d'un système comme l'inverse du taux de variation de son entropie  $S$  par rapport à son énergie interne  $U$ .

$$T = \frac{1}{\left(\frac{\partial S}{\partial U}\right)_V} \quad (2.2)$$

Bien que d'autres définitions de la température existent, le but n'est pas ici d'en faire une liste exhaustive, mais plutôt de souligner la multiplicité des théories existantes. Heureusement, à l'échelle macroscopique, toutes ces définitions coïncident et il est possible de parler de LA température. Cependant, à mesure que l'on considère des systèmes de plus en plus petits et des instants de plus en plus brefs, définir la notion de température devient plus délicat.

### Équilibre thermodynamique local (ETL) :

Pour pouvoir définir localement la température en tout point d'un système, celui-ci doit se trouver à l'état d'équilibre thermodynamique local (ETL). Cette hypothèse signifie que le déséquilibre thermique reste suffisamment faible à l'échelle d'un volume infinitésimal  $dV$  pour que celui-ci puisse être étudié comme s'il était à l'équilibre thermique. Plus précisément, l'ETL signifie que les particules microscopiques qui constituent le volume  $dV$  sont réparties selon une distribution de Maxwell-Boltzmann. L'hypothèse d'ETL permet de définir la température en tout point du système considéré et à tout instant. L'hypothèse d'ETL n'est valide que si les dimensions caractéristiques du système considéré sont suffisamment grandes devant le libre parcours moyen des porteurs d'énergie, c'est-à-dire la distance moyenne parcourue par les particules microscopiques entre deux collisions. En pratique, pour les matériaux rencontrés

dans les circuits intégrés, l'hypothèse d'ETL est vérifiée pour des échelles spatiales supérieures à  $1 \mu m$  et des échelles temporelles supérieures à  $1 ns$  (Sobhan and Peterson, 2008). On veillera donc à ne pas calculer la température dans des composants trop petits ou sur des laps de temps trop brefs.

La question de savoir s'il est possible de parler de température dans des composants nanométriques, comme les transistors, ou sur des périodes de temps de l'ordre de la nanoseconde, sort du cadre de cette thèse.

### Unités :

La température peut être exprimée en Kelvin (K), en degré Celsius ( $^{\circ}C$ ) ou en degré Fahrenheit ( $^{\circ}F$ ). Un Kelvin correspond à un degré Celsius et une température de  $0^{\circ}C$  est égale à  $273.5 K$ . Exprimée en Kelvin, la température est toujours positive, car elle mesure la vitesse moyenne des particules élémentaires. La température de zéro Kelvin correspond à un état théorique où les particules constituantes de la matière seraient immobiles. L'échelle Fahrenheit est en revanche beaucoup moins commode à utiliser et de ce fait très peu employée pour effectuer la modélisation thermique d'un système.

### 2.2.2 Transferts de chaleur dans les circuits intégrés

Dans les circuits intégrés, on rencontre trois modes de transfert de chaleur : la conduction, la convection et le rayonnement.

#### Conduction thermique :

La conduction thermique dans un milieu homogène est décrite à l'échelle macroscopique par la loi de Fourier (Holman, 2002).

$$\vec{q}_{cd} = -k\vec{\nabla T}S \quad (2.3)$$

Le flux de chaleur conductif  $\vec{q}_{cd}$  qui passe à travers une surface  $S$  est proportionnel au gradient de température  $\vec{\nabla T}$  à travers  $S$ . La constante  $k$  est appelée *conductivité thermique*. Elle dépend du milieu considéré et s'exprime en  $W/m.K$ . La présence d'un signe (-) dans la loi de Fourier assure le respect du second principe de la thermodynamique, selon lequel la chaleur se propage toujours des régions les plus chaudes vers les régions les plus froides.

L'expérience montre que cette loi phénoménologique, découverte par le mathématicien français Joseph Fourier, rend très bien compte des phénomènes conductifs à l'échelle macroscopique. Il faut cependant garder à l'esprit que la loi de Fourier et la notion de conductivité

thermique ne sont valables que dans le cadre de l'ETL, c'est-à-dire pour des échelles spatiales  $dl > 1 \mu m$  et temporelles  $dt > 1 ns$ . En effet, cette loi relie de manière linéaire et instantanée la cause du phénomène de diffusion, le gradient de température, à son effet, l'existence d'un transfert de chaleur. La linéarité de la loi de Fourier peut être remise en cause si le gradient de température devient trop élevé, en considérant une échelle spatiale trop petite. Le caractère instantané de la réponse peut être remis en cause si l'échelle temporelle caractéristique devient trop faible. Dans la pratique, savoir si la loi de Fourier peut encore être utilisée avec des longueurs  $dl < 1 \mu m$  ou des temps  $dt < 1 ns$  reste un problème ouvert (Liu et al., 2012a). Aussi, nous éviterons de nous placer dans ce cas-là.

### Convection thermique :

Les transferts de chaleurs par convection sont caractérisés par un déplacement de matière. On les rencontre dans diverses situations en présence de fluides (liquides ou gaz). On parle typiquement de convection lorsqu'un fluide circule autour d'un solide, pour désigner l'échange de chaleur entre le solide et le fluide à travers la surface de contact. Selon le mécanisme qui génère le mouvement du fluide, on parle de convection naturelle ou de convection forcée. Lorsque le mouvement du fluide résulte du seul effet des différences de densités dues aux écarts de température à travers la paroi de contact, on parle de convection naturelle. Lorsque le mouvement du fluide résulte d'une cause externe (pompe, turbine, ventilateur, etc.), on parle de convection forcée.

Dans les circuits intégrés, les transferts de chaleur par convection ont lieu à travers les surfaces où le circuit est en contact avec un fluide. Il s'agit principalement des surfaces du boîtier d'encapsulation, où la chaleur qui s'échappe du circuit passe dans l'air ambiant par convection naturelle. Si le boîtier d'encapsulation est équipé d'un dissipateur de chaleur avec ventilateur, il s'agit de convection forcée. On rencontre également le phénomène de convection dans les circuits refroidis par la circulation d'un liquide. La chaleur du circuit passe par convection forcée dans le liquide au niveau des parois des micro-canaux.

Les transferts de chaleur par convection peuvent être modélisés par la loi de Newton (Holman, 2002).

$$\vec{q}_{cv} = h(T_{surface} - T_{fluide})S\vec{n}_{surface} \quad (2.4)$$

Le flux de chaleur convectif  $q_{cv}$  échangé entre le solide et le fluide à travers la surface de contact  $S$  est proportionnel à la différence de température entre la surface et le fluide. Le coefficient  $h$  est appelé *coefficient de transfert par convection*. Il s'exprime en  $W/m^2.K$  et dépend du fluide considéré, de la nature de l'écoulement et de la surface de contact.

## Rayonnement thermique :

Quelle que soit sa température, tout corps émet un rayonnement électromagnétique plus ou moins intense. Son énergie thermique est transformée en un rayonnement électromagnétique qui se propage dans le milieu ambiant. Le transfert de chaleur associé au rayonnement électromagnétique est appelé rayonnement thermique.

La figure 2.6 présente le spectre électromagnétique correspondant au rayonnement thermique, qui est compris entre les longueurs d'onde de  $0.1 \mu m$  et  $100 \mu m$ , il comprend donc le domaine du visible.

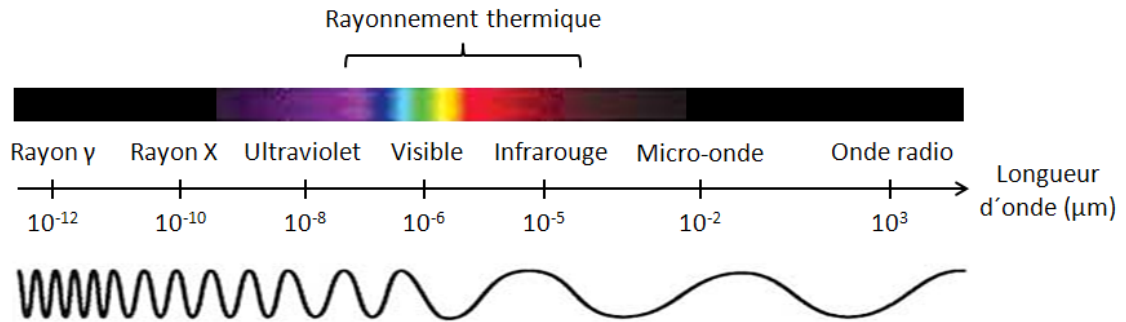


Figure 2.6 Spectre électromagnétique

Le rayonnement thermique est décrit par la loi de Stefan-Boltzman (Sobhan and Peterson, 2008).

$$\vec{q}_{rd} = \epsilon S \sigma (T_{surface}^4 - T_{air}^4) S \vec{n}_{surface} \quad (2.5)$$

Cette relation donne le flux de chaleur  $q_{rd}$  émit par le rayonnement d'un corps à travers sa surface  $S$ . La constante  $\epsilon$  est l'émissivité du corps, il s'agit d'un coefficient sans unité, compris entre 0 et 1. Pour un corps noir totalement absorbant,  $\epsilon = 1$ . La constante  $\sigma$  est la constante de Stefan-Boltzman, qui vaut  $\sigma \simeq 5.67 \times 10^{-8} W/m^2.K^{-4}$ . Pour un circuit intégré type, le rayonnement thermique joue cependant un faible rôle dans la dissipation de chaleur du circuit (Cheng et al., 2000).

### 2.2.3 Équation de bilan thermique

Le bilan thermique permet d'écrire une équation dont la résolution conduit à la distribution de température dans le système. Le bilan thermique est une application directe du principe de conservation de l'énergie au système. L'équation qui gouverne la température dans le système peut prendre plusieurs formes suivant la nature des transferts de chaleur (conduction,

convection, rayonnement). La plus célèbre de ces équations est l'équation de la chaleur, qui est obtenue dans le cas de transferts conductifs. Le bilan thermique sur un système de volume  $V$  s'écrit :

$$\int_V c_v \frac{\partial T}{\partial t} dv = \int_S \vec{q} \cdot \vec{ds} + \int_V P_v dv \quad (2.6)$$

Où  $c_v$  désigne la capacité volumique thermique du matériau considéré,  $T$  est la température au sein du volume  $V$ ,  $q$  est le flux de chaleur surfacique reçu par le volume  $V$  à travers la surface  $S$  et  $P_v$  est la puissance volumique dissipée dans le volume  $V$ .

L'équation 2.6 obtenue par un bilan d'énergie est assez générale. Aucune hypothèse n'est faite sur les matériaux constituant le système. Suivant la nature du système, le flux de chaleur  $q$  peut être conductif, convectif ou advectif. Il convient alors de remplacer  $q$  par la loi physique appropriée :

- loi de Fourier, pour un flux de chaleur conductif
- loi de Newton, pour un flux de chaleur convectif
- loi de Stefan-Boltzman, pour un flux de chaleur radiatif

Pour obtenir la température du système à tout instant  $t$ , il suffit de résoudre le problème défini par :

- l'équation 2.6
- les conditions aux limites sur la surface du volume
- la distribution de température à l'instant initial  $t = 0$

La résolution de ce problème dans le cas général est difficile. L'équation 2.6 est une équation aux dérivées partielles dépendant à la fois de variables d'espace et de temps. Dans le cas général, il n'existe pas de solution analytique simple à ce problème.

Dans le cas des circuits intégrés, plusieurs éléments viennent compliquer le problème :

- Le problème est tridimensionnel, ce qui rend fastidieux, voire impossible la résolution sous forme analytique.
- Le système est souvent très hétérogène, c'est-à-dire constitué de matériaux différents. La conductivité thermique varie donc dans l'espace.
- Certains circuits sont refroidis par un liquide et l'équation 2.6 devient plus complexe dans le cas d'un volume de fluide.

#### 2.2.4 Modèle thermique basé sur l'équation de la chaleur

Dans le cas des circuits intégrés, quelques hypothèses simplificatrices peuvent être formulées, afin de construire un modèle thermique. Sauf dans le cas très particulier du refroidissement par un fluide, les transferts thermiques **au sein** d'un circuit intégré se font par conduction.

En injectant la loi de Fourier dans l'équation 2.6, on obtient la célèbre équation de la chaleur :

$$c_v \frac{\partial T}{\partial t} = -\vec{\nabla} \cdot [k \vec{\nabla} T] + P_v \quad (2.7)$$

L'expérience montre que la conductivité thermique  $k$  des matériaux rencontrés dans les circuits intégrés, dépend peu de la température. Cette hypothèse rend l'équation de la chaleur linéaire :

$$c_v \frac{\partial T}{\partial t} + k \nabla^2 T = P_v \quad (2.8)$$

Cette dernière équation étant linéaire, sa solution peut se mettre sous la forme  $T(r, t) = T_1(r, t) + T_2(r, t)$  où :

- $T_1(r, t)$  est la solution de l'équation homogène associée :  $c_v \frac{\partial T}{\partial t} + k \nabla^2 T = 0$
- $T_2(r, t)$  est une solution particulière de l'équation complète.

La solution  $T_1(r, t)$  est appelée *régime propre* tandis que la solution  $T_2(r, t)$  est appelée *régime permanent*.

### Régime propre :

Le régime propre correspond à l'évolution du système lorsque celui-ci est mis hors équilibre en l'absence de source. À cause des phénomènes dissipatifs et de l'absence de source, le régime propre s'amortit au cours du temps.

### Régime permanent :

Le régime permanent correspond à la réponse du système quand le régime propre est amorti. En toute rigueur, le régime permanent ne signifie pas nécessairement que la température ne varie plus dans le temps. Par exemple, si les sources de puissance sont des fonctions sinusoïdales du temps, alors la température en régime permanent est aussi une fonction sinusoïdale du temps. On parle de régime sinusoïdal permanent.

### Régime stationnaire :

Lorsque la température est effectivement indépendante du temps, on parle du régime stationnaire. La plupart du temps, les notions de régime permanent et de régime stationnaire se recoupent, car les sources de chaleur ont des valeurs relativement constantes dans le temps. Pour cette raison, il est très fréquent de voir les termes « régime stationnaire » et « régime permanent » employés comme synonymes.

Le régime stationnaire correspond donc à une situation où la température n'évolue plus. La dérivée partielle de la température par rapport au temps s'annule et l'équation de la chaleur se simplifie en donnant l'équation de Poisson :

$$\nabla^2 T = \frac{P_v}{\lambda_{th}} \quad (2.9)$$

Les modèles thermiques qui s'intéressent seulement au régime stationnaire se contentent de résoudre l'équation de Poisson.

### Régime transitoire :

Le régime transitoire correspond à la réponse du système quand le régime propre n'est pas encore amorti. Dans les circuits intégrés classiques équipé d'un dissipateur de chaleur, le temps d'amortissement du régime propre varie entre quelques secondes et une minute, suivant la résistance et la capacitance thermiques du circuit et de son boîtier d'encapsulation (Cheng et al., 2000).

Si la distribution de puissance  $P_v$  change de manière significative avant que le régime propre ne soit amorti, alors le régime permanent n'est jamais atteint et il faut étudier le régime transitoire pour connaître la température dans le circuit. Le régime permanent peut également être calculé comme la limite dans le temps du régime transitoire.

### 2.2.5 Bilan

La simulation thermique des circuits intégrés consiste à résoudre un problème de transfert de chaleur. Les circuits intégrés contiennent des composants actifs, comme les transistors et les connexions électriques qui dissipent de la chaleur. Trois modes de transfert de la chaleur (conduction, convection et rayonnement) permettent au circuit de dissiper cette chaleur vers l'environnement ambiant. Le calcul de la température dans le circuit peut être effectué en résolvant une équation de bilan d'énergie appliquée au domaine spatial constitué par le circuit. Dans le cas de circuits où les transferts thermiques sont purement conductifs, l'équation de bilan d'énergie prend la forme d'une équation aux dérivées partielles, appelée équation de la chaleur. Cette équation peut être résolue en régime transitoire ou permanent.

## CHAPITRE 3 REVUE DE LITTERATURE

L'intérêt croissant pour la simulation thermique des circuits intégrés a donné lieu à une littérature abondante. De nombreux modèles ont été proposés pour simuler le comportement thermique des circuits intégrés. On peut distinguer trois classes de modèles thermiques, suivant l'approche utilisée pour résoudre l'équation de la chaleur : les modèles analytiques, les modèles compacts et les modèles numériques.

### 3.1 Modèles thermiques analytiques

#### 3.1.1 Principe

Les modèles thermiques analytiques s'appuient sur des outils mathématiques pour résoudre de manière exacte l'équation de la chaleur associée au circuit. Ces modèles permettent d'obtenir une formulation analytique du champ de température dans le circuit. Les modèles thermiques analytiques de circuits intégrés sont principalement basés sur les séries de Fourier et les fonctions de Green.

#### 3.1.2 Modèles thermiques basés sur les séries de Fourier

Les séries de Fourier jouent un rôle important en mathématique, notamment en analyse harmonique et en théorie du signal. Les séries de Fourier ont été introduites en 1822 par le mathématicien français Joseph Fourier, qui cherchait alors à résoudre l'équation de la chaleur. Les séries de Fourier peuvent être utilisées pour résoudre l'équation de la chaleur dans des domaines à géométrie rectangulaire. Considérons par exemple le bloc homogène de la figure 3.1, qui pourrait servir de modèle très simplifié de puce de circuit intégré.

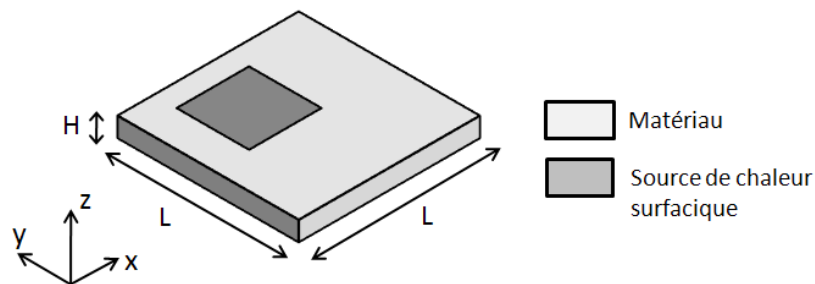


Figure 3.1 Bloc homogène contenant une source de chaleur surfacique



La méthode des séries de Fourier permet d'écrire la température dans le bloc sous la forme :

$$T = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \cos(m\pi \frac{x}{L}) \cos(n\pi \frac{y}{L}) (A_{mn} e^{\gamma z} + B_{mn} e^{-\gamma z}) e^{j\omega t} \quad (3.1)$$

où les coefficients  $A_{mn}$  et  $B_{mn}$  sont calculés à partir des conditions aux limites et  $\gamma$  est un nombre complexe vérifiant la relation :

$$\gamma^2 = (\frac{m\pi}{L})^2 + (\frac{n\pi}{L})^2 + j\omega \frac{\rho c_p}{k} \quad (3.2)$$

Les développements en séries de Fourier ont l'avantage d'être rapides à calculer, notamment si l'algorithme de transformation rapide de Fourier est appliqué. La méthode des séries de Fourier reste néanmoins complexe à appliquer aux problèmes multi-dimensionnels, aussi leur utilisation est souvent limitée à des géométries très simples.

Le logiciel THERMAN (Székely et al., 2000), basé sur les séries de Fourier, permet de calculer le régime permanent dans des circuits intégrés à géométries simples. Le logiciel THERMAN est capable de modéliser des structures contenant un nombre quelconque de couches homogènes, des sources de puissance et même des cavités.

Les séries de Fourier ne sont cependant pas le seul outil mathématique permettant de résoudre l'équation de la chaleur. Les fonctions de Green offrent également cette possibilité.

### 3.1.3 Modèles thermiques basés sur les fonctions de Green

Les fonctions de Green sont très utilisées en physique pour la résolution de problèmes aux dérivées partielles. Pour certains types d'équations et certaines conditions aux limites, la forme analytique des fonctions de Green est connue. La solution du problème s'obtient alors par un produit de convolution avec la fonction de Green.

Les fonctions de Green sont utilisées par Zhan *et al.* (Zhan and Sapatnekar, 2007) pour calculer efficacement la température en régime permanent dans un circuit intégré. Zhan *et al.* considèrent une géométrie et des conditions aux limites extrêmement simplifiées, illustrées sur la figure 3.2. Le circuit est modélisé comme un empilement de couches homogènes ayant les mêmes dimensions latérales. La chaleur s'échappe uniquement par convection à travers la surface supérieure du circuit.

Pour calculer la température à l'aide des fonctions de Green, Zhan *et al.* procèdent en trois étapes :

1. Calcul dans le domaine fréquentiel de la distribution de puissance en utilisant la transformée en cosinus discrète (DCT).

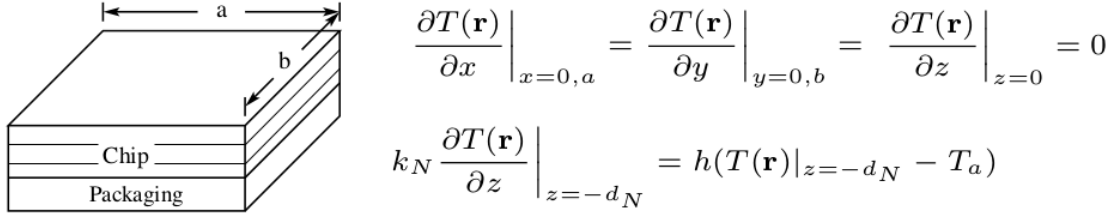


Figure 3.2 Géométrie et conditions aux limites considérées par Zhan *et al.* (Zhan and Sapatnekar, 2007)

2. Calcul dans le domaine fréquentiel de la température en multipliant chaque composante fréquentielle de la densité de puissance par la réponse fréquentielle correspondante du système, donnée par la fonction de Green.
3. Utilisation de la transformée inverse (IDCT) pour obtenir la température à partir de sa représentation fréquentielle.

La transformée en cosinus discrète (DCT) et sa transformée inverse (IDCT) sont calculées en utilisant la transformée de Fourier rapide, qui est un algorithme très efficace. Pour un circuit contenant  $L$  couches et discrétisé en  $L$  grilles de  $N$  cellules, la complexité du modèle proposé par Zhan *et al.* est en  $O(L^2 N \log N)$ . Un temps d'exécution de 6.4 *sec* est rapporté pour calculer  $1.05 \times 10^6$  valeurs de température. L'erreur relative du modèle est évaluée à 1%. Le modèle thermique proposé par Zhan *et al.* peut donc être considéré comme précis et efficace. Il n'est cependant applicable qu'au calcul du régime permanent pour des géométries très simplifiées.

Pour modéliser la présence de connexions verticales dans les circuits 3D, Oh *et al.* (Oh et al., 2012) proposent un modèle thermique également basé sur les fonctions de Green. Encore une fois, le circuit est modélisé de manière très simpliste, comme un empilement de couches homogènes contenant ponctuellement des matériaux différents, supposés représenter la présence des connexions verticales au sein des couches. Le modèle proposé par Oh *et al.* est en réalité à mi-chemin entre un modèle analytique et un modèle numérique, car le circuit est discrétisé à l'aide de la méthode des différences finies (FDM), qui est une méthode numérique. En revanche, la température est calculée par une méthode semi-analytique. Oh *et al.* décomposent le circuit en deux structures, la première formée uniquement des couches homogènes et la deuxième contenant les connexions verticales du circuit. Cette décomposition est illustrée sur la figure 3.3.

La matrice de conductance  $G$  du circuit s'écrit donc logiquement comme la somme de deux

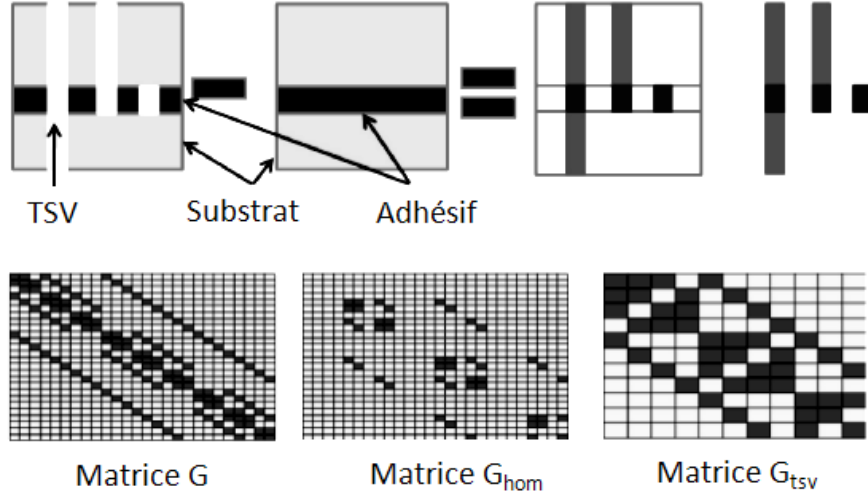


Figure 3.3 Décomposition du circuit proposée par Oh *et al.* (Oh et al., 2012)

matrices :

$$G = G_{hom} + G_{tsv} \quad (3.3)$$

La matrice  $G_{hom}$  correspond à la conductance thermique de la structure constituée des couches homogènes et la matrice  $G_{tsv}$  correspond à la conductance des connexions verticales.

Le calcul de la température en régime permanent nécessite d'inverser la matrice  $G$ . Pour réaliser cette opération efficacement, Oh *et al.* utilisent l'identité matricielle de Woodbury :

$$(\mathbf{A} + \mathbf{UCV}^T)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}^T\mathbf{A}^{-1} \quad (3.4)$$

en prenant  $\mathbf{A} = G_{hom}$ . L'intérêt de cette manipulation est que la matrice  $G_{hom}$  peut être inversée rapidement à l'aide des fonctions de Green. Quelques calculs supplémentaires permettent alors d'obtenir l'inverse de la matrice complète  $G$  sans trop d'effort. Il faut tout de même noter que ces opérations sont peu coûteuses en calcul seulement dans le cas de matrices très creuses, ce qui est le cas ici car les connexions verticales sont de petite taille et dispersées dans le circuit. La complexité du modèle proposé par Oh *et al.* reste donc globalement en  $O(L^2N \log N)$  pour un circuit contenant  $L$  couches formées chacune de  $N$  cellules. Oh *et al.* évaluent l'erreur de leur modèle à moins de  $1^\circ\text{C}$  en se comparant au logiciel commercial COMSOL(COMSOL). Le modèle proposé par Oh *et al.* représente une amélioration par rapport au modèle de Zhan *et al.* car il prend en compte la présence de connexions verticales dans les circuits 3D. Il reste cependant limité au régime permanent et à un petit nombre de connexions verticales dispersées dans le circuit.

Pour modéliser des circuits contenant une proportion de matériaux hétérogènes plus impor-

tante, Lee *et al.* proposent NUMANA (Lee et al., 2013), un modèle thermique du régime permanent, qui combine une approche hybride basée sur une discrétisation numérique et une résolution analytique. Le circuit est d’abord discrétisé en petites cellules, appelées volumes de contrôles, représentés sur la figure 3.4, à l’aide de la méthode des volumes finis, qui est une méthode numérique, puis la température est calculée à l’aide des fonctions de Green.

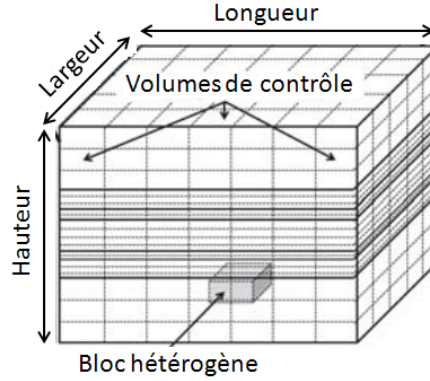


Figure 3.4 Discrétisation du circuit en volumes de contrôle effectuée par Lee *et al.* (Lee et al., 2013)

Lee *et al.* effectuent une décomposition du circuit similaire à celle proposée par Oh *et al.*, avec d’une part la structure composée de couches homogènes et d’autre part tous les matériaux hétérogènes. La matrice de conductance  $G$  du circuit est écrite comme la somme de deux matrices  $G = G_h + \Delta G$ , où  $G_h$  est la matrice de conductance correspondant à l’empilement de couches homogènes. En notant  $P$  la dissipation de puissance dans le circuit, la température en régime permanent est calculée à l’aide de la relation suivante :

$$T = \sum_{q=0}^{\infty} (-1)^q (G_h^{-1} \Delta G)^q G_h^{-1} P \quad (3.5)$$

Lee *et al.* retiennent seulement les termes les plus significatifs qui sont calculés de manière récursive. Lee *et al.* rapportent un temps d’exécution de 8.4 sec pour résoudre un problème contenant  $1.6 \times 10^6$  valeurs de températures. Il s’agit d’une très bonne performance de calcul, mais l’erreur du modèle dans la même expérimentation est évaluée à 1.8 °C, ce qui est une précision moyenne. Par ailleurs, le modèle NUMANA présente les mêmes limitations que les modèles précédents : il ne calcule que le régime permanent et la géométrie supportée reste très simplifiée, les couches du modèle doivent avoir les mêmes dimensions latérales ; et les conditions aux limites restent très simplifiées, la chaleur ne peut s’échapper du circuit que des faces supérieures.

### 3.1.4 Bilan

Lorsqu'ils sont applicables, les modèles thermiques analytiques de circuits intégrés ont l'avantage d'être précis, rapides et de consommer peu de mémoire. En outre, ils permettent de calculer la température à un endroit particulier du circuit sans avoir à calculer la distribution de température dans l'ensemble du circuit.

Les modèles analytiques sont cependant assez lourds à mettre en place. Ils sont basés sur des outils mathématiques complexes, applicables seulement à des géométries simplifiées. De manière générale, les modèles analytiques peinent à prendre en compte les matériaux hétérogènes, les sources de chaleur multiples et la grande variété de conditions aux limites, rencontrés dans les circuits intégrés.

Une alternative aux modèles analytiques, capable de prendre en compte des structures plus réalistes, tout en restant très rapide à simuler, est la modélisation compacte.

## 3.2 Modèles thermiques compacts

### 3.2.1 Principe

Les modèles thermiques compacts ont été développés dans les années 1980, afin de fournir une estimation rapide de la quantité de chaleur pouvant être dissipée par le boîtier d'encapsulation. Les modèles compacts sont des modèles comportementaux qui visent à prédire la température seulement en quelques points précis du circuit et du boîtier d'encapsulation (Shidore, 2007). Ils sont constitués d'un réseau de résistances thermiques reliant les points où l'on désire évaluer la température. La figure 3.5 représente le modèle compact associé à un boîtier d'encapsulation de type puce-retournée.

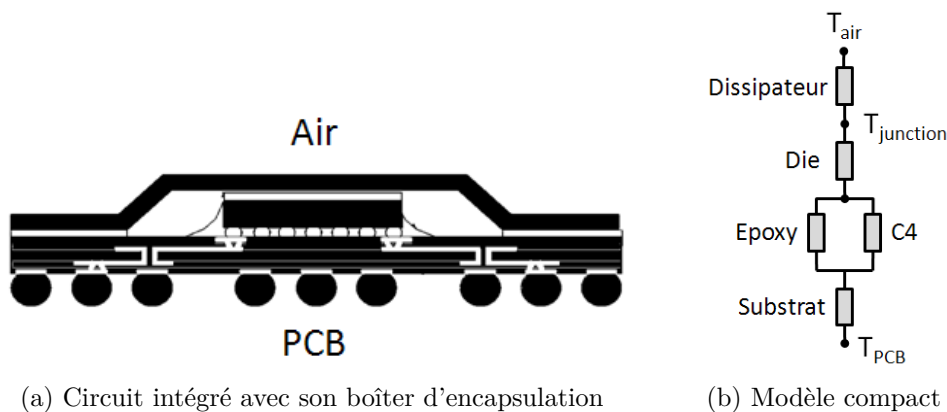


Figure 3.5 Modèle compact d'un circuit intégré avec son boîtier d'encapsulation

Un modèle compact n'est pas construit en essayant de reproduire fidèlement la structure physique du circuit, c'est-à-dire la géométrie, les matériaux et les sources de chaleur. Il s'agit plutôt d'une abstraction reproduisant le comportement thermique du circuit pour des conditions environnementales variées. La caractéristique principale d'un modèle compact est de contenir un petit nombre d'inconnues et d'être très rapide à simuler.

### 3.2.2 Modèles DELPHI

DELPHI (Rosten and Lasance, 1995) est le nom d'une méthodologie développée au début des années 1990 dans le cadre d'un projet européen, qui permet de générer des modèles compacts de circuits intégrés et de boîtiers d'encapsulation, indépendants des conditions aux limites.

Les modèles compacts DELPHI sont composés de résistances thermiques reliant un noeud de jonction, représentant la température de la puce, à plusieurs autres noeuds de surface. La figure 3.6 représente la topologie typique d'un modèle compact DELPHI.

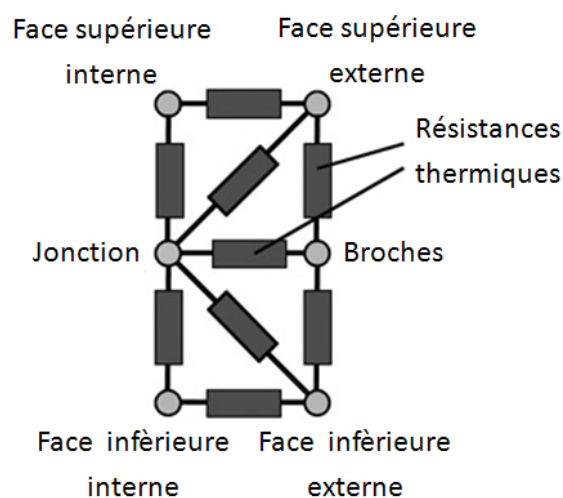


Figure 3.6 Topologie typique d'un modèle compact DELPHI (Rosten and Lasance, 1995)

Pour construire un modèle thermique compact indépendant des conditions aux limites, la méthodologie DELPHI procède par identification. Plusieurs simulations numériques détaillées sont réalisées au moyen du logiciel FLoTHERM (FLOTHERM), afin d'évaluer le comportement thermique du système placé dans un environnement particulier.

La méthodologie DELPHI propose un ensemble de conditions aux limites, censées couvrir toutes les conditions environnementales que peut rencontrer le circuit durant son fonctionnement. Pour chacune des conditions aux limites DELPHI, une simulation numérique détaillée est effectuée, pour finalement générer un modèle indépendant des conditions aux limites.

Le projet DELPHI avait pour ambition de devenir un standard de l'industrie. L'objectif était d'amener les fabricants de boîtiers d'encapsulation à accompagner systématiquement chacun de leurs produits de son modèle compact DELPHI. Malheureusement, la méthodologie DELPHI n'a pas été universellement adoptée. Seuls quelques vendeurs de boîtiers d'encapsulation fournissent aujourd'hui systématiquement un modèle compact DELPHI pour tous leurs produits.

Les modèles DELPHI possèdent en outre deux limitations importantes :

- Le régime transitoire n'est pas modélisé, car les modèles DELPHI sont constitués uniquement de composants passifs (résistances thermiques). Par la suite, une extension (Monier-Vinard et al., 2011) à la méthodologie DELPHI a été proposée afin de générer des modèles compacts dynamiques, capables de simuler le régime transitoire. Le principe de cette extension est de rajouter des éléments capacitifs au réseau de résistances thermiques. Néanmoins, la génération de modèles DELPHI dynamiques reste très lente, car elle nécessite plusieurs simulations du régime transitoire à l'aide d'un modèle numérique détaillé.
- La modélisation du circuit intégré reste grossière, car seules quelques valeurs de température sont calculées pour l'ensemble du circuit. Par conséquent, les modèles DELPHI ne permettent pas de détecter les éventuels points de surchauffe dans le circuit.

### 3.2.3 Modèle HotSpot

Hotspot (Huang et al., 2006) est un modèle thermique compact de circuit intégré, développé au début des années 2000 dans le cadre d'un projet de doctorat à l'université de Virginia. Hotspot modélise les régimes permanent et transitoire, et comprend un modèle de boîtier d'encapsulation. Hotspot est basé sur l'analogie qui existe entre les lois électriques et les lois thermiques. Le circuit et son boîtier d'encapsulation sont représentés à l'aide d'un modèle compact constitué de résistances et de capacités thermiques. Dans sa version de base, Hotspot modélise le circuit au niveau architectural, c'est-à-dire en se concentrant sur l'architecture du circuit. Chaque unité fonctionnelle du circuit (banque de registres, mémoire, etc.) est associée à un noeud du modèle compact. Hotspot calcule donc plusieurs valeurs de température au niveau du circuit, contrairement aux modèles DELPHI présentés à la section 3.2.2. Les éléments du boîtier d'encapsulation comme le substrat, le diffuseur ou le dissipateur de chaleur sont modélisés dans Hotspot par un nombre plus réduit de noeuds. La figure 3.7 représente le modèle compact construit par Hotspot pour un circuit contenant trois unités fonctionnelles.

La modélisation du boîtier d'encapsulation faite dans Hotspot est assez rudimentaire. Un

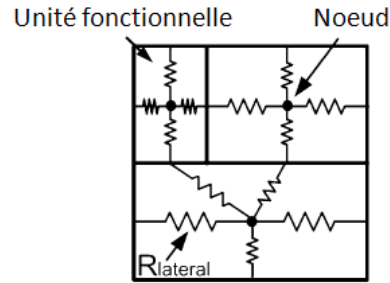


Figure 3.7 Modélisation compacte d'un circuit avec Hotspot (Huang et al., 2006)

nombre fixe d'éléments est représenté et très peu de noeuds sont utilisés pour calculer la température dans ces éléments. La figure 3.8 présente le modèle de boîtier d'encapsulation implémenté dans Hotspot.

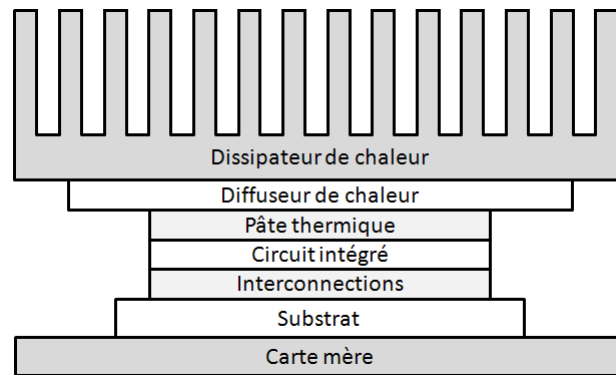


Figure 3.8 Modélisation du boîtier d'encapsulation dans Hotspot (Huang et al., 2006)

Ce modèle a cependant l'avantage de permettre à l'utilisateur de se concentrer sur la description du circuit en faisant abstraction des conditions environnementales. Une version plus récente d'Hotspot permet de modéliser la structure en couches du circuit. Le modèle compact qui permet de calculer les échanges thermiques dans le circuit, devient une grille dont l'utilisateur peut fixer les dimensions. Cette version d'Hotspot est donc à mi-chemin entre un modèle thermique compact et un modèle numérique détaillé.

Bien que rudimentaire, le modèle Hotspot est très populaire dans le milieu académique. Il se présente sous la forme d'un petit simulateur qui fonctionne en ligne de commande sous Linux. La description du circuit est lue depuis un fichier texte et les valeurs de température sont retournées directement à l'écran ou dans un fichier texte. Hotspot est implémenté dans le langage C. Le code source est ouvert et distribué librement.

Le modèle compact utilisé par Hotspot pouvant contenir un nombre arbitraire de noeuds,



le nombre d'équations à résoudre peut devenir important. Pour calculer la température en régime permanent, HotSpot 5.02 (la version la plus récente à ce jour) utilise la méthode multi-grille, dont le principe est présenté à la section 3.3.3. Pour le calcul de la température en régime transitoire, HotSpot 5.02 utilise la méthode Runge-Kutta. Contrairement aux premières versions, HotSpot 5.02 modélise les transferts thermiques dans les trois dimensions dans le circuit, grâce à la possibilité de représenter ses différentes couches. En revanche, la modélisation de couches hétérogènes, n'est pas possible directement dans Hotspot. Une extension (Meng et al., 2012) a cependant été apportée à Hotspot afin de pouvoir modéliser la présence de connexions verticales dans les couches des circuits 3D. HotSpot a été validé avec deux simulateurs commerciaux, SOLIDWORKS Flow Simulation (SOLIDWORKS) et ANSYS (ANSYS), ainsi que par des mesures sur un processeur réel et un FPGA. Une erreur inférieure à 10% est rapportée (Huang et al., 2006). Pour un problème contenant  $40.9 \times 10^3$  cellules, le calcul du régime permanent prend environ 5 sec.

Initialement construit sur le principe des modèles compacts, Hotspot, dans ses dernières versions, calcule la température en de nombreux points et se rapproche nettement des modèles numériques, dont il emprunte les méthodes de calcul.

### 3.2.4 Bilan

Les modèles thermiques compacts de circuits intégrés ont l'avantage d'être très rapides à simuler et de fournir des résultats directement exploitables, comme la température de jonction. La principale limitation des modèles thermiques compacts est de donner des résultats peu précis en raison du petit nombre de noeuds utilisés pour représenter le circuit. Les modèles compacts qui tentent de rajouter des noeuds pour évaluer plus précisément la température dans le circuit, tel HotSpot, tendent progressivement à devenir des modèles numériques.

## 3.3 Modèles thermiques numériques

### 3.3.1 Principe

Les modèles thermiques numériques discrétisent l'équation de la chaleur par rapport aux variables d'espace et de temps (Holman, 2002). Cette discrétisation permet de transformer le problème aux dérivées partielles, qui n'admet pas de solution analytique, en un problème algébrique, qui peut être résolu par une méthode de calcul matriciel.

## Discrétisation spatiale

La discrétisation de l'équation de la chaleur par rapport aux variables d'espace est généralement effectuée à l'aide de l'une des trois méthodes numériques suivantes :

- la méthode des différences finies (*Finite Difference Method* - FDM)
- la méthode des volumes finis (*Finite Volume Method* - FVM)
- la méthode des éléments finis (*Finite Element Method* - FEM)

Ces méthodes opèrent sur un maillage, c'est-à-dire une discrétisation spatiale du domaine considéré en petits éléments de forme régulière, appelés cellules. Les cellules du maillage peuvent prendre des formes diverses (hexaèdre, tétraèdre, prisme, pyramide, etc.). La figure 3.9 illustre quelques-unes des formes les plus classiques de maillages tridimensionnels.

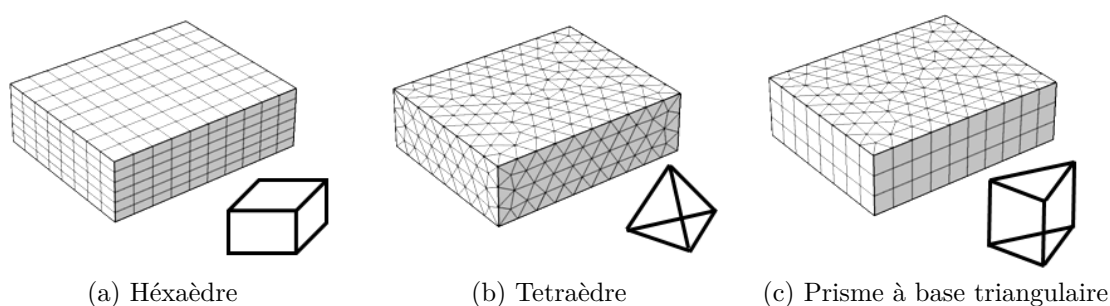


Figure 3.9 Cellules de maillages tridimensionnels

## La méthode des différences finies (FDM)

La méthode des différences finies (FDM) est historiquement la méthode numérique la plus ancienne. Les opérateurs différentiels de l'équation à résoudre sont discrétisés à l'aide des développements en séries de Taylor. Dans la plupart des cas, quelques termes des séries de Taylor suffisent à obtenir une précision suffisante. La méthode FDM est très simple à mettre en oeuvre sur des maillages orthogonaux, car les opérateurs différentiels discrétisés ont des expressions assez simples. Considérons par exemple un domaine 2D rectangulaire et une fonction  $\phi(x, y)$  définie et différentiable en tout point de ce domaine. Le domaine est discrétisé à l'aide d'un maillage orthogonal, comme illustré sur la figure 3.10.

La méthode FDM utilise trois approximations pour discrétiser les dérivées partielles de la fonction  $\phi$  aux noeuds du maillage :

- Le schéma « décentré en avant », qui est une approximation du 1er ordre :

$$\left. \frac{\partial \phi}{\partial x} \right|_{i,j} \simeq \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x} \qquad \left. \frac{\partial \phi}{\partial y} \right|_{i,j} \simeq \frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta y} \qquad (3.6)$$

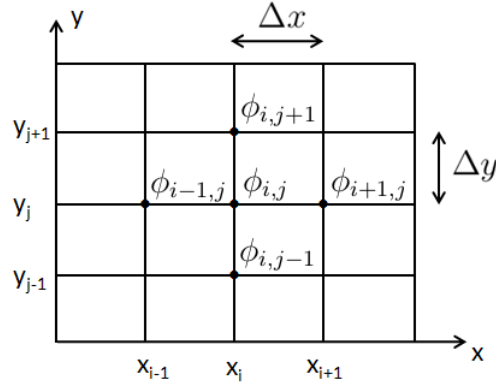


Figure 3.10 Discrétisation d'un domaine 2D à l'aide de la méthode FDM

— Le schéma « décentré en arrière », qui est une approximation du 1er ordre :

$$\left. \frac{\partial \phi}{\partial x} \right|_{i,j} \simeq \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x} \quad \left. \frac{\partial \phi}{\partial y} \right|_{i,j} \simeq \frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta y} \quad (3.7)$$

— Le schéma « centré », qui est une approximation du 2ème ordre :

$$\left. \frac{\partial \phi}{\partial x} \right|_{i,j} \simeq \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x} \quad \left. \frac{\partial \phi}{\partial y} \right|_{i,j} \simeq \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2\Delta y} \quad (3.8)$$

La méthode FDM peut également être utilisée avec des maillages non orthogonaux, notamment des maillages triangulaires, mais sa mise en oeuvre devient alors plus délicate et d'autres méthodes numériques lui sont préférées. Les circuits intégrés ayant une géométrie rectangulaire se prêtent bien à une discrétisation à l'aide d'un maillage orthogonal. De ce fait, la méthode FDM est très utilisée pour développer des modèles thermiques de circuits intégrés.

### La méthode des volumes finis (FVM)

La méthode des volumes finis (FVM) a été développée pour discrétiser des lois de conservation sur des géométries complexes. Une loi de conservation exprime la conservation d'une grandeur physique ; l'équation de la chaleur exprimant la conservation de l'énergie thermique en est un bon exemple.

La méthode FVM travaille avec des volumes de contrôle définis à partir du maillage et considère les valeurs moyennes des grandeurs physiques dans ces volumes. La figure 3.11 illustre le découpage d'un domaine polyédrique en volumes de contrôle définis à partir d'un maillage triangulaire. Les flux entrants et sortants des volumes de contrôle sont discrétisés en faisant intervenir les valeurs moyennes des grandeurs physiques dans les volumes de contrôles.

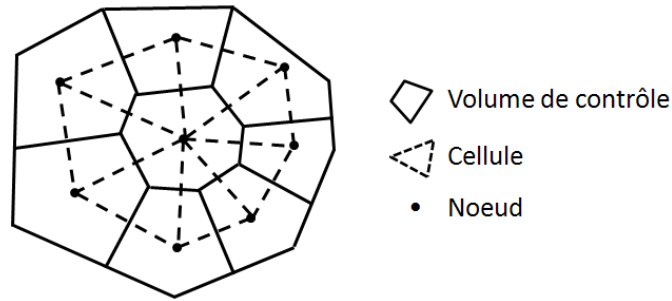


Figure 3.11 Volumes de contrôle sur un domaine polyédrique

La méthode FVM permet de traiter de façon élégante les problèmes thermiques rencontrés dans les structures hétérogènes, grâce à la possibilité de définir des volumes de contrôle respectant la distribution des matériaux. La méthode FVM est donc très utile pour développer des modèles thermiques numériques de circuits intégrés, composés d'un grand nombre de matériaux différents.

Illustrons le principe de la méthode FVM sur un problème thermique modèle. Considérons le bloc hétérogène de dimension  $L \times L \times H$ , représenté à la figure 3.12. Ce bloc est constitué de deux matériaux  $A$  et  $B$ , et contient une source de chaleur rectangulaire qui dégage une puissance  $P$ . La chaleur s'échappe du bloc par convection à travers la surface supérieure. Le coefficient de transfert par convection est noté  $h$  et la température de l'air ambiant est notée  $T_a$ .

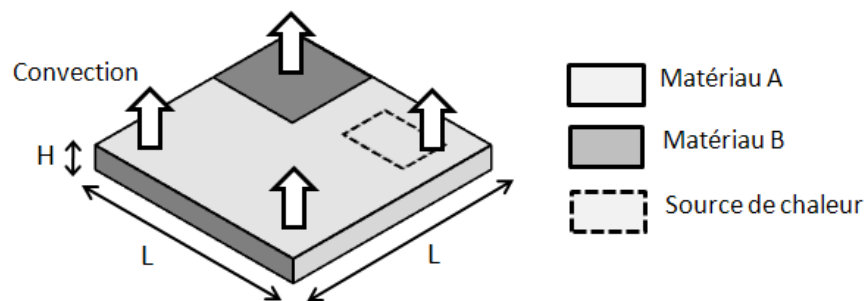


Figure 3.12 Problème thermique modèle

Le bloc est discrétisé à l'aide d'un maillage orthogonal constitué de  $2 \times 2$  cellules, représenté à la figure 3.13. Dans cet exemple, les cellules du maillage ont des dimensions identiques, mais ce n'est pas une contrainte de la méthode FVM. L'important est que le maillage respecte la distribution des matériaux, afin que les cellules soient constituées d'un matériau homogène.

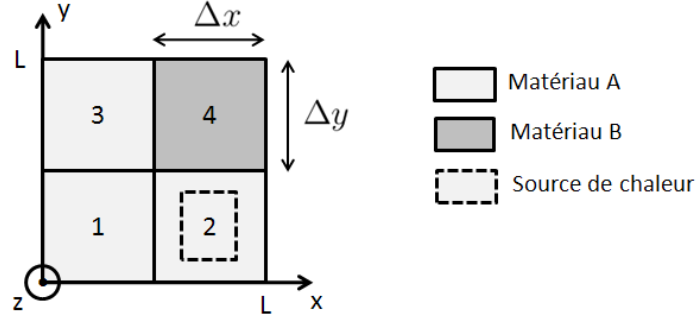


Figure 3.13 Discrétisation du problème modèle en 2x2 cellules

Les volumes de contrôle de la méthode FVM sont les cellules du maillage. Appliquons l'équation de la chaleur à une cellule quelconque du maillage :

$$\int_V c_v \frac{\partial T}{\partial t} dv + \int_S -k \overrightarrow{\nabla T} d\vec{s} = \int_V P_{vol} dv \quad (3.9)$$

où  $k$  et  $c_v$  sont respectivement la conductivité et la capacité thermique du matériau constituant la cellule. On note  $T$  la température moyenne dans cette cellule et  $V$  son volume. Puisque la cellule est constituée d'un seul matériau, on peut écrire :

$$\int_V c_v \frac{\partial T}{\partial t} dv = C \dot{T} \quad (3.10)$$

où  $C = c_v V$  est à la capacité thermique de la cellule. Le flux conductif  $q_{cd}$  reçu par la cellule sur une de ses faces  $S$  est donné par l'intégrale  $q_{cd} = \int_S -k \overrightarrow{\nabla T} d\vec{s}$ . En faisant intervenir la température moyenne  $T_S$  de la face  $S$ , le flux conductif  $q_{cd}$  est discrétisé par une approximation du 1er ordre :

$$q_{cd} \simeq kS \frac{T_S - T}{\frac{\Delta l}{2}} \quad (3.11)$$

où  $\Delta l$  est la longueur de la cellule dans la direction du transfert thermique. La quantité  $G = \frac{2kS}{\Delta l}$  est homogène à une conductance thermique. Le flux thermique reçu par la cellule sur sa face  $S$  s'écrit donc  $q_{cd} = G(T_S - T)$ .

Si la face  $S$  est au contact d'une autre cellule, de température  $T'$  et de conductance  $G'$ , alors  $q_{cd}$  est aussi égal au flux  $G'(T' - T_S)$  sortant de la cellule voisine. En écrivant l'égalité des flux thermiques à travers la face  $S$ , la température  $T_S$  de cette face, est donnée par la relation :

$$T_S = \frac{G}{G + G'} T + \frac{G'}{G + G'} T' \quad (3.12)$$

ce qui permet d'écrire le flux conductif  $q_{cd}$  échangé par les deux cellules, uniquement en fonction de leurs températures  $T$ ,  $T'$  et leurs conductances  $G$ ,  $G'$  :

$$q_{cd} = \frac{GG'}{G + G'}(T' - T) \quad (3.13)$$

Si la face  $S$  de la cellule est au contact de l'air ambiant, la cellule reçoit sur cette face le flux convectif  $q_{cv} = hS(T_{air} - T_S)$ , où  $T_{air}$  est la température de l'air ambiant et  $h$  le coefficient du transfert par convection. On pose  $G' = 1/hS$ . En écrivant l'égalité des flux conductif et convectif à travers la face  $S$ , on élimine la température  $T_S$  des relations et le flux convectif reçu par la cellule s'écrit finalement :

$$q_{cv} = \frac{GG'}{G + G'}(T_{air} - T) \quad (3.14)$$

À l'aide des équations 3.13 et 3.14, l'équation de la chaleur est discrétisée pour chaque cellule du maillage  $2 \times 2$ , et conduit à un système d'équations de la forme :

$$C_1\dot{T}_1 + G_{12}(T_2 - T_1) + G_{13}(T_3 - T_1) + G_{1a}(T_{air} - T_1) = 0 \quad (3.15)$$

$$C_2\dot{T}_2 + G_{12}(T_1 - T_2) + G_{24}(T_4 - T_2) + G_{2a}(T_{air} - T_2) = P \quad (3.16)$$

$$C_3\dot{T}_3 + G_{13}(T_1 - T_3) + G_{34}(T_4 - T_3) + G_{3a}(T_{air} - T_3) = 0 \quad (3.17)$$

$$C_4\dot{T}_4 + G_{24}(T_2 - T_4) + G_{34}(T_3 - T_4) + G_{4a}(T_{air} - T_4) = 0 \quad (3.18)$$

Mis sous forme matricielle, ce système s'écrit :

$$\mathbf{C}\dot{\mathbf{T}} + \mathbf{G}\mathbf{T} = \mathbf{P} \quad (3.19)$$

Les matrices  $\mathbf{C}$  et  $\mathbf{G}$  sont appelées respectivement matrices de capacitance et de conductance. Le vecteur  $\mathbf{T}$  est le vecteur température, contenant les valeurs moyennes de température de chaque cellule. Le vecteur  $\mathbf{P}$  est le vecteur puissance, contenant les valeurs de puissance générées dans chaque cellule, ainsi que la contribution de l'environnement ambiant. Cet exemple simple, mais complet montre comment la discrétisation spatiale de l'équation de la chaleur à l'aide de la méthode FVM, aboutit à une équation différentielle matricielle du 1er ordre.

La méthode FVM est un outil très puissant, bien que son champ d'application soit limité aux lois de conservation. Pour traiter des problèmes plus généraux, il faut recourir à la méthode des éléments finis.

## La méthode des éléments finis (FEM)

La méthode des éléments finis (FEM) construit une approximation continue de la solution du problème, à partir de fonctions d'interpolation définies sur les cellules du maillage. Les fonctions d'interpolation sont choisies avec des expressions simples, par exemple des fonctions affines, pour être faciles à discrétiser. Les coefficients des fonctions d'interpolation deviennent les inconnues du problème. Ils sont calculés de manière à ce que la solution ainsi construite satisfasse le problème considéré. Bien que plus flexible et plus générale que les méthodes FDM et FVM, la méthode FEM est plus complexe à mettre en oeuvre. La très grande majorité des modèles thermiques numériques de circuits intégrés se contentent d'utiliser les méthodes FDM ou FVM, dont l'application à des maillages orthogonaux est immédiate.

## Bilan

En résumé, quelle que soit la méthode employée (FDM, FVM ou FEM) et le type de maillage utilisé (orthogonal, tétraédrique, etc.), la discrétisation spatiale de l'équation de la chaleur conduit à l'équation matricielle :

$$\mathbf{C}\dot{\mathbf{T}} + \mathbf{G}\mathbf{T} = \mathbf{P} \quad (3.20)$$

où  $\mathbf{C}$  et  $\mathbf{G}$  sont respectivement les matrices de capacitance et de conductance,  $\mathbf{T}$  est le vecteur température et  $\mathbf{P}$  est le vecteur puissance.

Dans le cas où l'on s'intéresse seulement au régime permanent, l'équation 3.19 peut être simplifiée en remarquant que la température n'évolue plus lorsque le régime permanent est atteint et que sa dérivée par rapport au temps s'annule. La température en régime permanent est donc obtenue en résolvant l'équation 3.21.

$$\mathbf{G}\mathbf{T} = \mathbf{P} \quad (3.21)$$

Dans le cas où l'on s'intéresse au régime transitoire, l'équation 3.19 doit encore être discrétisée dans le temps avant de pouvoir être résolue.

## Discrétisation temporelle

La discrétisation de l'équation de la chaleur par rapport à la variable de temps est effectuée à l'aide de la méthode FDM. Le temps continu  $t$  est découpé en intervalles de temps  $\Delta t$  et la température est calculée aux instants  $t_n = n\Delta t$ , où  $n \in \mathbb{N}$ . On note  $\mathbf{T}^n$  et  $\mathbf{P}^n$  les vecteurs

température et puissance à l'instant  $t_n$ . Les trois schémas de discrétisation « décentré en avant », « décentré en arrière » et « centré », présentés précédemment, peuvent être utilisés pour discrétiser la dérivée temporelle du vecteur température. Ces trois schémas conduisent respectivement aux méthodes d'Euler, d'Euler-arrière et de Crank-Nicholson.

### La méthode d'Euler

La méthode d'Euler utilise le schéma « décentré en avant » pour discrétiser la dérivée temporelle du vecteur température :

$$\mathbf{C} \frac{\mathbf{T}^{n+1} - \mathbf{T}^n}{\Delta t} + \mathbf{G}\mathbf{T}^n = \mathbf{P}^n \quad (3.22)$$

et conduit à une formulation explicite du problème transitoire :

$$\mathbf{T}^{n+1} = (\mathbf{I} + \Delta t \mathbf{C}^{-1} \mathbf{G}) \mathbf{T}^n + \Delta t \mathbf{C}^{-1} \mathbf{P}^n \quad (3.23)$$

On parle de formulation explicite, car la température future  $\mathbf{T}^{n+1}$  peut être calculée directement à partir de la température courante  $\mathbf{T}^n$ , en effectuant seulement une multiplication d'une matrice par un vecteur et une addition entre deux vecteurs. L'avantage de la méthode d'Euler est sa simplicité et son coût de calcul relativement faible à chaque pas de temps. La méthode d'Euler s'implémente très facilement et possède une complexité à chaque pas de temps, qui augmente linéairement avec la taille du problème. L'inconvénient majeur de la méthode d'Euler est d'être conditionnellement stable. Si les pas de temps  $\Delta t$  sont choisis trop grands, les valeurs de température divergent. La méthode d'Euler est donc soumise à un critère de stabilité sur le pas de temps, qui peut être très restrictif pour certains problèmes. Ce critère de stabilité peut contraindre à utiliser des pas de temps très petits, ce qui augmente drastiquement le nombre d'itérations à effectuer pour simuler une durée de temps donnée. En outre, il n'existe pas de moyen simple de calculer le critère de stabilité, ce qui complique la mise en oeuvre de la méthode d'Euler.

### La méthode d'Euler-arrière

La méthode d'Euler-arrière utilise le schéma « décentré en arrière » pour discrétiser la dérivée temporelle du vecteur température :

$$\mathbf{C} \frac{\mathbf{T}^n - \mathbf{T}^{n-1}}{\Delta t} + \mathbf{G}\mathbf{T}^n = \mathbf{P}^{n-1} \quad (3.24)$$



et conduit à une formulation implicite du problème transitoire :

$$(\mathbf{I} + \Delta t \mathbf{C}^{-1} \mathbf{G}) \mathbf{T}^{n+1} = \mathbf{T}^n + \Delta t \mathbf{C}^{-1} \mathbf{P}^n \quad (3.25)$$

On parle de formulation implicite, car pour calculer la température future  $\mathbf{T}^{n+1}$ , il est nécessaire d'inverser au préalable la matrice  $(\mathbf{I} + \Delta t \mathbf{C}^{-1} \mathbf{G})$ , opération coûteuse dont la complexité est cubique dans le cas de matrices denses. L'avantage de la méthode d'Euler-arrière est d'être inconditionnellement stable. Quel que soit le pas de temps choisi  $\Delta t$ , la méthode d'Euler-arrière ne diverge pas et fournit des résultats remarquablement stables. Il est donc possible de choisir de grands pas de temps  $\Delta t$  pour réduire le nombre d'itérations à faire afin de simuler une durée de temps donnée et raccourcir ainsi globalement le temps de simulation. L'utilisation de grands pas de temps se fait bien sûr au détriment de la précision temporelle. Cependant en pratique, pour des pas de temps de l'ordre de 100 *ms*, on observe que la méthode d'Euler-arrière conserve une précision acceptable pour la modélisation thermique des circuits intégrés, ce qui lui confère une supériorité sur la méthode d'Euler.

### La méthode de Crank-Nicholson :

La méthode de Crank-Nicholson utilise le schéma « centré » pour discrétiser la dérivée temporelle du vecteur température :

$$\mathbf{C} \frac{\mathbf{T}^{n+1} - \mathbf{T}^n}{\Delta t} + \mathbf{G} \frac{\mathbf{T}^{n+1} + \mathbf{T}^n}{2} = \mathbf{P}^n \quad (3.26)$$

et conduit également à une formulation implicite du problème transitoire :

$$\left(\mathbf{I} + \frac{\Delta t}{2} \mathbf{C}^{-1} \mathbf{G}\right) \mathbf{T}^{n+1} = \left(\mathbf{I} - \frac{\Delta t}{2} \mathbf{C}^{-1} \mathbf{G}\right) \mathbf{T}^n + \Delta t \mathbf{C}^{-1} \mathbf{P}^n \quad (3.27)$$

L'inconvénient de la méthode de Crank-Nicholson est lié à sa formulation implicite, qui nécessite une inversion de matrice. L'avantage de cette méthode est d'être inconditionnellement stable, c'est-à-dire que les pas de temps peuvent être choisis librement. Cependant, en pratique, on observe des oscillations dans les valeurs température obtenues, lorsque la méthode de Crank-Nicholson est utilisée avec des pas de temps  $\Delta t > 10$  *ms* (Wang and Chen, 2002). Le véritable avantage de cette méthode réside dans sa précision en  $O(\Delta t^2)$  obtenue grâce à l'utilisation d'un schéma « centré » pour la discrétisation temporelle. Or, ce gain de précision ne peut être exploité qu'avec des pas de temps relativement petits  $\Delta t < 1$  *ms*. La méthode de Crank-Nicholson n'est donc pas la méthode à privilégier, pour accélérer la simulation des modèles transitoires de circuits intégrés en choisissant de grands pas de temps.

Les méthodes d'Euler-arrière et Crank-Nicholson ramènent la résolution du problème transitoire à la résolution d'une équation matricielle linéaire.

### Résolution du problème discrétisé

La résolution du problème thermique discrétisé, que ce soit pour simuler le régime transitoire ou permanent, se ramène à la résolution d'une équation matricielle linéaire. Pour résoudre ce type d'équation, on distingue deux classes de méthodes : les méthodes directes et les méthodes itératives.

Les méthodes directes permettent de résoudre une équation matricielle linéaire en un nombre fini d'opérations arithmétiques. Sans les erreurs d'arrondi dues au calcul en virgule flottante, les méthodes directes fourniraient la solution exacte du problème discrétisé. Le point faible de ces méthodes réside dans leur complexité élevée, complexité cubique par rapport au nombre d'inconnues dans le cas général. L'utilisation des méthodes directes est limitée à des problèmes de petite taille, comportant typiquement moins de  $10^5$  inconnues.

Pour résoudre des problèmes de plus grande taille, comportant typiquement plus de  $10^5$  inconnues, il faut recourir aux méthodes itératives. Ces méthodes calculent une solution approchée en construisant de manière itérative une suite de vecteurs qui converge vers la solution du problème. En pratique les itérations sont arrêtées dès la satisfaction d'un critère de convergence fixé a priori. Le point faible des méthodes itératives réside dans leur vitesse de convergence, qui s'avère parfois être trop lente pour résoudre un problème en un temps raisonnable.

En résumé, les modèles thermiques numériques comportent toujours deux aspects bien distincts : la discrétisation du problème à l'aide des méthodes FDM, FVM ou FEM ; et la résolution du problème discrétisé à l'aide d'une méthode directe ou itérative. Les sections suivantes présentent quelques-uns des modèles thermiques numériques les plus performants, développés au cours des quinze dernières années.

#### 3.3.2 Modèles numériques du régime transitoire

Les modèles numériques du régime transitoire sont très souvent basés sur la méthode d'Euler-arrière, en raison de ses propriétés de stabilité.

La méthode d'Euler-arrière conduit à résoudre à chaque pas de temps de la simulation transitoire, une équation de la forme  $Ax = b$  où  $x$  est la température à l'instant futur et  $b$  la température présente à laquelle est ajoutée la chaleur accumulée dans le circuit durant le pas de temps. Les modèles thermiques basés sur la méthode d'Euler-arrière, nécessitent donc

plusieurs résolutions successives du même système avec des seconds membres différents. Les méthodes de résolution directes basées sur une décomposition LU (Chapra and Canale, 2009) sont particulièrement appropriées pour cette tâche. La décomposition LU est un algorithme très classique, qui dérive de l'élimination gaussienne. La matrice  $A$  est décomposée comme un produit  $A = LU$  d'une matrice triangulaire inférieure  $L$  par une matrice triangulaire supérieure  $U$ . La résolution du système  $Ax = b$  se ramène alors à la résolution de deux systèmes triangulaires  $Ly = b$  et  $Ux = y$ . La décomposition LU est un calcul coûteux dont la complexité est cubique dans le cas de matrices denses (Chapra and Canale, 2009). En revanche, la résolution d'un système triangulaire possède une complexité quadratique (Chapra and Canale, 2009). Une fois la décomposition LU effectuée, la température peut être calculée à chaque pas assez rapidement.

3D-ICE (Sridhar et al., 2010) et 3D-ACME (Hanhua Qian, 2013) sont deux modèles thermiques utilisant une méthode de résolution directe de type décomposition LU. Tous deux modélisent le régime transitoire dans les circuits 3D refroidis par un liquide et discrétisent l'équation de la chaleur à l'aide de la méthode d'Euler-arrière. 3D-ICE utilise la librairie SuperLU (Demmel et al., 1999), tandis que 3D-ACME utilise la librairie KLU (Davis and Palamadai Natarajan, 2010). SuperLU est une librairie performante, dédiée à la décomposition LU de matrices creuses, symétriques ou non symétriques. SuperLU utilise des algorithmes avancés de factorisation symbolique pour réduire le nombre d'opérations lors de la décomposition LU. SuperLU effectue la décomposition LU d'une matrice creuse de taille  $N$  avec une complexité moyenne estimée à  $O(N^{1.6})$ , ce qui est un progrès notable par rapport à la complexité en  $O(N^3)$  de l'algorithme LU standard (Chapra and Canale, 2009). La décomposition LU d'un problème thermique type de taille  $N = 10^5$ , prend environ 43 sec avec SuperLU. KLU poursuit le même objectif que SuperLU, mais utilise des algorithmes différents, qui lui permettent d'atteindre une efficacité légèrement supérieure. KLU effectue la décomposition d'un problème thermique type de taille  $N = 10^5$  en 30 sec environ, ce qui confère à 3D-ACME de meilleures performances de simulation que 3D-ICE. Les performances de 3D-ACME sont également comparées à celles du modèle Hotspot 5.0. Pour des problèmes comportant moins de  $2 \times 10^4$  cellules, 3D-ACME est plus rapide qu'Hotspot. Au-delà, la méthode multi-grille utilisée par Hotspot s'impose. 3D-ICE est validé avec un prototype et le logiciel commercial ANSYS (ANSYS). Une erreur inférieure à 4% est rapportée. 3D-ACME est validé avec le logiciel commercial COMSOL (COMSOL), puis comparé à 3D-ICE. 3D-ACME s'avère plus précis et plus fiable que 3D-ICE (Hanhua Qian, 2013).

Le point faible de 3D-ICE et 3D-ACME est bien sûr la complexité élevée des librairies SuperLU et KLU utilisées, qui rendent ces modèles trop lents et trop gourmands en mémoire. Le blâme ne doit pas être jeté sur les librairies SuperLU et KLU, qui restent très performantes

dans leur catégorie, mais plutôt sur le choix de la méthode LU, qui est une méthode directe et donc limitée à des problèmes de petite taille, du moins aujourd’hui. En résumé, 3D-ICE et 3D-ACME sont incapables de traiter des problèmes de plus de  $10^5$  inconnues et donc limités à des modélisations assez grossières.

Pour accélérer 3D-ICE, Vincenzi *et al.* proposent une méthode de calcul basée sur les réseaux de neurones (Vincenzi et al., 2012). Un réseau de neurones est entraîné avec 3D-ICE pour imiter le comportement transitoire du circuit modélisé. Une fois entraîné, le réseau de neurones est capable de reproduire le comportement transitoire du circuit, simulé avec d’autres cartes de puissance, tant que la structure du circuit n’est pas modifiée. Vincenzi *et al.* implémentent leur modèle sur GPU, et obtiennent un facteur d’accélération  $35 - 100\times$  par rapport à 3D-ICE exécuté sur CPU. Hormis le fait que l’utilisation de GPUs est nécessaire pour obtenir des temps de calcul acceptables, cette méthode a deux limitations importantes. Premièrement, le réseau de neurones doit être entraîné avec 3D-ICE à chaque modification de sa structure, ce qui ne permet pas vraiment une exploration de l’espace de conception. Deuxièmement, la technique proposée par Vincenzi *et al.* est conditionnée à l’utilisation de 3D-ICE. Si la simulation avec 3D-ICE est impossible faute de ressources suffisantes, alors la technique proposée est inutile.

La méthode Euler-arrière est également utilisée par Liu *et al.* (Liu et al., 2012b) pour simuler le régime transitoire de circuits 3D refroidis par un liquide. Liu *et al.* proposent de résoudre le problème discrétisé à l’aide d’une méthode itérative de minimalisation des résidus, la méthode GMRES (Saad, 2003). Ce choix est a priori étonnant, car les méthodes itératives ne peuvent pas exploiter le fait que le même système est résolu plusieurs fois. La méthode GMRES calcule une suite de vecteurs  $x_k$  pour laquelle la norme du résidu  $\|Ax_k - b\|$  converge vers 0. Chaque vecteur  $x_k$  est cherché dans un espace vectoriel généré par les  $k - 1$  vecteurs précédents. De ce fait, tous les vecteurs  $x_k$  doivent être conservés en mémoire et la complexité de cette méthode croît à chaque itération. Pour réduire la consommation de mémoire, la méthode GMRES est relancée périodiquement en conservant seulement le dernier vecteur  $x_k$  calculé, ce qui a pour effet de ralentir sa convergence. Liu *et al.* utilisent un préconditionnement AINV pour accélérer la vitesse de convergence et implémentent leur modèle sur GPU. Malgré ces efforts, les performances de ce modèle restent assez médiocres. Un temps de 864 *sec* est nécessaire pour simuler 1600 pas de temps du régime transitoire pour un problème de taille  $N \simeq 1.7 \times 10^5$ . Globalement, le modèle de Liu *et al.* implémenté sur GPU n’est que  $2.3\times$  plus rapide qu’une résolution par la méthode directe LU sur un CPU multi-core.

Une alternative plus efficace est présentée par Wang *et al.* (Wang and Chen, 2002), qui utilisent la méthode ADI (Douglas and Gunn, 1964) pour simuler le régime transitoire. La

méthode ADI dérive de la méthode de Crank-Nicholson. Le calcul est décomposé suivant les trois dimensions spatiales  $x, y, z$  qui sont traitées séparément l'une après l'autre. Pour chaque direction, la contribution des cellules situées dans la même direction est calculée par une équation implicite, tandis que la contribution des cellules situées dans le plan orthogonal est calculée de manière explicite. Les contributions calculées implicitement conduisent à des systèmes tri-diagonaux, qui sont résolus par l'algorithme de Thomas (Chapra and Canale, 2009) avec une complexité linéaire. Le calcul des contributions explicites a naturellement une complexité linéaire. La méthode ADI possède donc une complexité globale linéaire. Wang *et al.* reportent de très bons résultats. La simulation de 1200 pas de temps pour un problème de taille  $N \simeq 2.8 \times 10^6$  prend 311 min. Le modèle de Wang *et al.* possède néanmoins deux limitations importantes. Premièrement, il n'est applicable qu'à des problèmes discrétisés à l'aide d'un maillage orthogonal, car les cellules doivent être alignées suivant les directions  $x, y, z$ . Deuxièmement, un comportement oscillatoire peut se produire, si le pas de temps  $\Delta t$  est choisi trop grand. Pour un circuit type, Wang *et al.* reportent des oscillations pour  $\Delta t > 10ms$ .

En cas d'oscillations, la réponse transitoire est perdue. Le pas de temps critique au-delà duquel apparaissent des oscillations est difficile à établir a priori. Il dépend de la structure du circuit et il n'existe pas de modèle théorique pour le déterminer à l'avance.

### 3.3.3 Modèles numériques du régime permanent

Les modèles numériques du régime permanent doivent résoudre une équation matricielle  $Ax = b$  où  $x$  est le vecteur température,  $A$  est la matrice de conductance du circuit et  $b$  le vecteur puissance. La plupart de ces modèles utilisent une méthode de résolution itérative, généralement plus efficace qu'une méthode directe pour un problème de grande taille (Chapra and Canale, 2009; Saad, 2003).

Parmi les nombreuses méthodes itératives existantes, la méthode de Gauss-Seidel est très populaire en raison de son extrême simplicité (Chapra and Canale, 2009). Mizunuma *et al.* l'utilisent pour calculer la température en régime permanent dans un circuit 3D refroidi par un liquide (Mizunuma et al., 2011). À partir d'une estimation initiale  $x_0$  de la température dans le circuit, la méthode de Gauss-Seidel construit une suite de vecteurs  $(x_k)_{k \geq 0}$  qui converge vers la solution du problème. À chaque itération, la solution approchée  $x_k$  est affinée en corrigeant systématiquement ses composantes. Le nombre d'opérations à chaque itération pour calculer un vecteur  $x_k$  dépend du nombre de coefficients non nuls de la matrice  $A$ , ce qui est avantageux dans le cas de matrices creuses. C'est une des raisons qui motivent l'utilisation de la méthode de Gauss-Seidel. En effet, la discrétisation du circuit à l'aide d'une méthode

numérique conduit à des matrices creuses, c'est-à-dire des matrices contenant relativement peu de coefficients non nuls. La méthode Gauss-Seidel (et les méthodes itératives en général) sont capables d'exploiter la structure creuse des matrices pour calculer efficacement la température dans le circuit. La méthode de Gauss-Seidel présente cependant l'inconvénient de converger très lentement lors de la résolution de problèmes thermiques types. Mizunuma *et al.* rapportent un temps de calcul de 11sec pour calculer la température dans un maillage contenant environ 200 cellules, ce qui est très long.

Toujours pour calculer le régime permanent d'un circuit 3D refroidi par un liquide, Kim *et al.* utilisent une technique de relaxation appelée « Successive Under Relaxation » (SUR) (Kim et al., 2009). Il s'agit d'une variante de la méthode Gauss-Seidel conçue pour converger plus rapidement. L'idée est de remplacer la solution calculée par la méthode de Gauss-Seidel par une moyenne pondérée avec la valeur calculée à l'itération précédente. Le vecteur température  $x_k$  calculé à l'itération  $k$  vérifie ainsi la relation  $x_{k+1} = \lambda x_{k+1} + (1-\lambda)x_k$  avec  $0 < \lambda < 1$ . Cette méthode est plutôt employée avec  $\lambda > 1$ , elle s'appelle alors « Successive Over Relaxation » (SOR). Kim *et al.* ne justifient pas l'emploi de la méthode SUR plutôt que la méthode SOR et ne donnent aucun résultat sur la performance de cette méthode de calcul, ni en terme de temps de calcul ni en terme de précision.

Kim *et al.* s'intéressent plutôt à la formulation du modèle thermique qu'à la résolution des équations qui en découlent. Koo *et al.* utilisent également la méthode SUR dans un modèle thermique (Koo et al., 2005) similaire à celui de Kim *et al.*, mais sans donner plus de précision sur les performances de cette méthode appliquée à ce problème précis.

Une autre méthode itérative très connue, la méthode du gradient conjugué, est utilisée par Goplen *et al.* pour résoudre le problème matriciel issu d'une discrétisation par la méthode FEM (Goplen and Sapatnekar, 2006). Goplen *et al.* s'intéressent au placement des vias thermiques dans un circuit 3D de manière à favoriser la remontée de chaleur. Goplen *et al.* proposent un algorithme de placement couplé à un modèle thermique. La méthode FEM est employée pour discrétiser l'équation de la chaleur sur une grille orthogonale. Pour résoudre le problème discrétisé, Goplen *et al.* utilisent la méthode du gradient conjugué qui a la propriété de converger en un nombre fini d'itérations au plus égal à la dimension du système. La méthode du gradient conjugué s'applique aux problèmes  $Ax = b$ , dont la matrice  $A$  est symétrique définie positive. Elle peut donc être employée comme méthode de résolution dans les modèles thermiques de circuits intégrés sans liquide calculant seulement le régime permanent. La méthode du gradient conjugué est une méthode de descente. La solution approchée

$x_k$  est construite itérativement selon la relation 3.28.

$$x_{k+1} = x_k + \alpha_k p_k \quad (3.28)$$

La méthode du gradient conjugué exploite le produit scalaire défini par la matrice  $A$  (symétrique définie positive) pour choisir optimalement  $\alpha_k$  et  $p_k$ . Pour améliorer la convergence, Goplen *et al.* emploient un préconditionnement ILU. L'utilisation d'un bon préconditionneur nécessite cependant beaucoup de calculs. Il n'existe pas de préconditionnement à la fois fiable et général pour tous les types de problèmes rencontrés en ingénierie. Le préconditionnement ILU employé par Goplen *et al.* est basé sur une factorisation LU incomplète. Goplen *et al.* ne reportent pas directement le temps mis par leur modèle pour calculer la température, mais plutôt le temps mis par leur algorithme de placement. Il est donc difficile de comparer l'efficacité de ce modèle thermique. La précision du modèle n'est pas non plus évaluée.

Une autre utilisation de la méthode du gradient conjugué est proposée par Feng *et al.* pour calculer la température en régime permanent dans un circuit refroidi par un liquide (Feng and Li, 2010). En théorie, il n'est pas possible d'utiliser la méthode du gradient conjugué pour cette application, car le problème matriciel n'est plus symétrique. Feng *et al.* ne détaillent pas les équations discrétisées de leur modèle, ce qui ne permet pas de contrôler que le modèle des flux advectifs soit correctement construit. Il est également possible que Feng *et al.* utilisent une extension de la méthode du gradient conjugué aux problèmes non symétriques. Afin d'accélérer l'exécution du modèle thermique, Feng *et al.* proposent une implémentation sur GPU. Le modèle thermique développé est relativement simplifié afin de pouvoir être implémenté sur GPU. La dissipation de chaleur latérale est négligée. La précision du modèle n'est pas évaluée.

Un autre modèle thermique de circuits intégrés efficace est proposé par Li *et al.* (Li et al., 2006), qui utilise la méthode multi-grille. Le modèle proposé par Li *et al.* permet de calculer la température en régime permanent et transitoire dans un circuit classique (constitué seulement de matériaux solides). La méthode multi-grille accélère la convergence des méthodes de relaxation en employant des grilles avec différentes granularités. La température est d'abord calculée approximativement sur une grille fine en faisant quelques itérations d'une méthode de relaxation. Le résidu est ensuite corrigé sur une grille grossière qui permet d'éliminer les composantes à haute fréquence de l'erreur. La correction est ensuite interpolée sur la grille fine. L'approche proposée par Li *et al.* utilise plusieurs grilles intermédiaires pour accélérer la vitesse de convergence. La température en régime permanent est calculée en 2.5 sec sur un circuit discrétisé en  $784 \times 10^3$  cellules. La combinaison de grilles qui permet une convergence optimale est cependant difficile à déterminer a priori.

### 3.3.4 Bilan

Les modèles numériques sont plus flexibles et plus robustes que les modèles analytiques ou les modèles compacts. Ils sont logiquement imposés comme l'outil indispensable pour simuler le comportement thermique des circuits intégrés modernes. Les modèles numériques sont capables de simuler des structures complexes et des phénomènes physiques variés. Ils permettent également de travailler avec des échelles de temps et de longueur très différentes. La figure 3.14 illustre les deux aspects, discrétisation et résolution, d'un modèle numérique et situe les différentes méthodes citées dans ce chapitre.

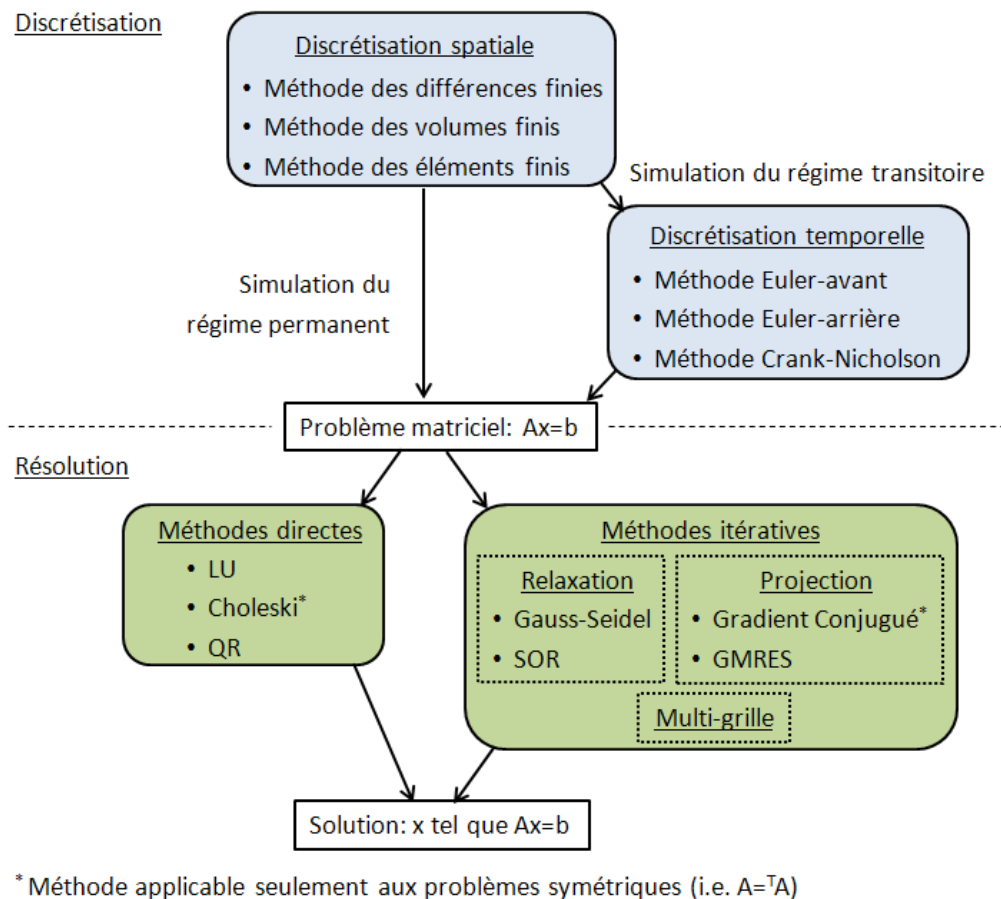


Figure 3.14 Méthodes classiques utilisées par les modèles numériques

La principale limitation des modèles numériques reste leur coût de calcul élevé, qui implique trop souvent un temps de simulation déraisonnable ou une consommation de mémoire excessive. C'est donc tout naturellement que cette thèse s'intéresse aux méthodes de calcul utilisées par les modèles numériques. La revue de littérature montre que les modèles du régime transitoire utilisent généralement des méthodes directes, tandis que les modèles du



régime permanent utilisent plutôt des méthodes itératives.

Cette thèse propose deux méthodes de calcul pour résoudre le problème matriciel qui apparaît dans les modèles numériques. La première méthode proposée cible la simulation du régime transitoire, tandis que la seconde méthode proposée concerne la simulation du régime permanent. Les deux méthodes proposées sont basées sur une combinaison des méthodes directes et itératives les plus performantes connues à ce jour, afin d'améliorer l'état de l'art.

## CHAPITRE 4 MÉTHODES PROPOSÉES

Ce chapitre propose deux méthodes de calcul pour la simulation thermique de circuits intégrés ; la première méthode permet de simuler le régime transitoire et la deuxième méthode le régime permanent. Les méthodes proposées sont applicables aux modèles numériques de circuits intégrés obtenus à l'aide d'un maillage orthogonal.

### 4.1 Méthode proposée pour la simulation du régime transitoire

La simulation du régime transitoire d'un circuit intégré à l'aide d'un modèle numérique se ramène à la résolution d'une équation matricielle de la forme :

$$C\dot{X} + GX = P \quad (4.1)$$

où  $C$  et  $G$  sont respectivement les matrices de capacitance et de conductance thermique du circuit,  $X$  est le vecteur température et  $P$  est le vecteur puissance. L'équation 4.1 étant obtenue en discrétisant un problème de transfert de chaleur, la matrice  $C$  est diagonale et inversible, tandis que la matrice  $G$  est creuse avec une structure en bande et inversible.

L'équation 4.1 est une équation différentielle linéaire du premier ordre par rapport à la variable de temps  $t$ . Pour la résoudre, le temps continu est divisé en petits intervalles de temps  $\Delta t$  et la température est calculée aux instants discrets  $t_m = m\Delta t$ , où  $m \in \mathbb{N}$ . Les vecteurs température et puissance à l'instant  $t_m$  sont notés respectivement  $X_m$  et  $P_m$ . Parmi les méthodes existantes permettant de discrétiser la dérivée temporelle  $\dot{X}$ , la méthode Euler-arrière (Holman, 2002) est sélectionnée pour ses propriétés de stabilité. À chaque pas de temps  $\Delta t$  de la simulation, la méthode Euler-arrière résout l'équation matricielle suivante :

$$(I + \Delta t C^{-1}G)X_{m+1} = X_m + \Delta t C^{-1}P_m \quad (4.2)$$

Le calcul de la température future  $X_{m+1}$  nécessite d'inverser la matrice  $(I + \Delta t C^{-1}G)$ . Cette opération est très coûteuse, avec une complexité plus que quadratique si l'algorithme LU standard est appliqué aux matrices creuses des problèmes thermiques.

La méthode proposée pour simuler le régime transitoire résout efficacement l'équation 4.2 en combinant les deux techniques suivantes : une décomposition des opérateurs pour fournir efficacement une bonne estimation de la température ; et un schéma itératif pour affiner l'estimation de température en corrigeant l'erreur introduite par la décomposition des opérateurs.

### 4.1.1 Décomposition des opérateurs

La décomposition des opérateurs permet d'appliquer aux problèmes à deux et trois dimensions, les algorithmes de calcul efficaces des problèmes unidimensionnels, en particulier l'algorithme de Thomas (Saad, 2003). Pour un problème de transfert de chaleur unidimensionnel, la matrice de conductance  $G$  et par conséquent la matrice  $(I + \Delta t C^{-1}G)$  sont tri-diagonales. La figure 4.1 illustre la discrétisation d'un problème de transfert de chaleur unidimensionnel et la matrice de conductance associée.

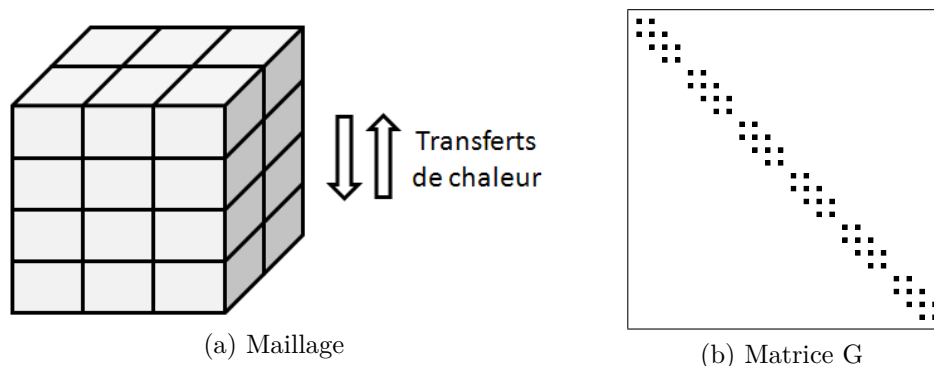


Figure 4.1 Problème de transfert de chaleur unidimensionnel

Dans cet exemple, les transferts de chaleur sont unidimensionnels, bien que la structure physique servant de support soit tridimensionnelle. Lorsque la matrice  $G$  est tridiagonale, l'équation 4.2 du régime transitoire peut être résolue efficacement par l'algorithme de Thomas.

---

#### Algorithme 1 : Algorithme de Thomas

---

**Input** :  $A \in \mathbb{R}^{n \times n}$  telle que  $a_{i,j} \neq 0$  si  $|i - j| \leq 1$ ,  $b \in \mathbb{R}^n$   
**Output** :  $x \in \mathbb{R}^n$  tel que  $Ax = b$

```

1  $a_{0,0} \leftarrow 1/a_{0,0}$  ;
2 for  $i = 2 \dots n$  do % Factorisation LU
3    $a_{i,i-1} \leftarrow a_{i,i-1} * a_{i-1,i-1}$  ;
4    $a_{i,i} \leftarrow 1/(a_{i,i} - a_{i,i} * a_{i-1,i})$  ;
5 end
6 for  $i = 2 \dots n$  do % Substitution avant
7    $x_i \leftarrow x_i - a_{i,i} * x_{i-1}$  ;
8 end
9  $x_n \leftarrow x_n * a_{n,n}$  ;
10 for  $i = n - 1 \dots 1$  do % Substitution arrière
11    $x_i \leftarrow (x_i - a_{i,i+1} * x_{i+1}) * a_{i,i}$  ;
12 end

```

---

L'algorithme de Thomas est basé sur la décomposition LU d'une matrice tridiagonale. Il permet de résoudre un système tridiagonal avec une complexité linéaire par rapport à la taille du système. Il constitue la primitive de calcul de la méthode de décomposition des opérateurs. Les étapes de calcul de l'algorithme de Thomas sont données dans l'algorithme 1.

Pour pouvoir appliquer l'algorithme de Thomas à des problèmes tridimensionnels, la matrice  $G$  est décomposée selon les trois directions  $\vec{x}, \vec{y}, \vec{z}$  du maillage orthogonal. On note  $G = G_x + G_y + G_z$  la décomposition de la matrice  $G$  selon les trois directions  $\vec{x}, \vec{y}, \vec{z}$ . En utilisant un maillage orthogonal pour la discrétisation, les opérateurs  $G_x, G_y, G_z$  sont tridiagonaux à une permutation près. Appliquée à l'équation 4.2, la technique de décomposition des opérateurs conduit à résoudre successivement les équations suivantes :

$$(I - \Delta t C^{-1} G_x) X_{m+\frac{1}{3}} = X_m + \Delta C^{-1} P_m \quad (4.3)$$

$$(I - \Delta t C^{-1} G_y) X_{m+\frac{2}{3}} = X_{m+\frac{1}{3}} \quad (4.4)$$

$$(I - \Delta t C^{-1} G_z) X_{m+1} = X_{m+\frac{2}{3}} \quad (4.5)$$

Puisque les matrices  $G_x, G_y, G_z$  sont semblables à des matrices tridiagonales, chacune de ces équations peut être résolue efficacement par l'algorithme de Thomas. Les vecteurs  $X_{m+\frac{1}{3}}$  et  $X_{m+\frac{2}{3}}$  sont des variables intermédiaires sans aucune signification physique particulière. Les équations 4.3, 4.4 et 4.5 sont mathématiquement équivalentes à l'équation suivante :

$$(I + \Delta t C^{-1} G_x)(I + \Delta t C^{-1} G_y)(I + \Delta t C^{-1} G_z) X_{m+1} = X_m + \Delta t C^{-1} P_m \quad (4.6)$$

En développant le produit  $(I + \Delta t C^{-1} G_x)(I + \Delta t C^{-1} G_y)(I + \Delta t C^{-1} G_z)$ , on montre que l'équation 4.6 est égale à l'équation 4.2 plus le terme suivant :

$$\Gamma = \Delta t^2 (C^{-1} G_x C^{-1} G_y + C^{-1} G_x C^{-1} G_z + C^{-1} G_y C^{-1} G_z) - \Delta t^3 C^{-1} G_x C^{-1} G_y C^{-1} G_z \quad (4.7)$$

Le terme  $\Gamma$  se comporte comme un terme perturbateur par rapport à l'équation initiale. Ce terme, introduit par la décomposition des opérateurs, est appelé « erreur de décomposition ». L'équation 4.7 montre que le terme perturbateur peut être majoré par  $O(\Delta t^2)$ . La décomposition des opérateurs introduit donc à chaque pas de temps une erreur en  $O(\Delta t^2)$ . L'erreur commise sur l'ensemble du régime transitoire peut être évaluée en sommant les erreurs commises à chaque pas de temps (Chapra and Canale, 2009). La simulation d'un intervalle de temps  $L$  comportant  $\frac{L}{\Delta t}$  pas de temps verra son erreur majorée par  $\frac{L}{\Delta t} \times O(\Delta t^2) = O(\Delta t)$ . L'erreur globale due à la décomposition des opérateurs reste du premier ordre.

La décomposition des opérateurs offre également un haut degré de parallélisme. Chacune des trois équations 4.3, 4.4, 4.5 résout un problème de transfert de chaleur unidimensionnel, qui peut être parallélisé très facilement sur un maillage orthogonal. Considérons le maillage orthogonal de la figure 4.2. L'équation 4.3 résout un système tridiagonal pour chaque ligne

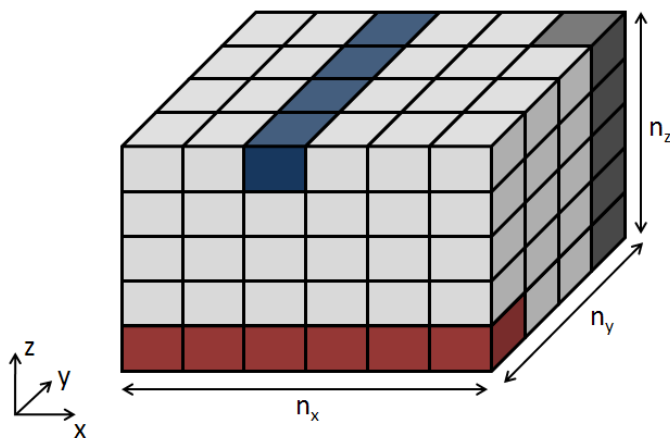


Figure 4.2 Maillage orthogonal à  $n_x \times n_y \times n_z$  cellules

de cellules dans la direction  $\vec{x}$ . La résolution de l'équation 4.3 peut être parallélisée en  $n_y \times n_z$  résolutions de systèmes tridiagonaux indépendants. Le même raisonnement s'applique aux deux autres directions  $\vec{y}$  et  $\vec{z}$ .

La décomposition des opérateurs appliquée à l'équation Euler-arrière conduit à un schéma de calcul ayant une complexité linéaire et un haut degré de parallélisme. La principale limitation de cette technique est l'erreur de décomposition qui s'ajoute à l'erreur inhérente à la discrétisation temporelle. Bien que l'erreur de décomposition reste du premier ordre, c'est-à-dire majorée par  $O(\Delta t)$ , elle peut conduire à des résultats imprécis si de grands pas de temps  $\Delta t$  sont employés. Pour remédier à ce problème, la méthode de calcul proposée utilise un schéma itératif pour corriger l'erreur de décomposition.

#### 4.1.2 Schéma itératif pour corriger l'erreur de décomposition

Le schéma itératif proposé pour corriger l'erreur de décomposition est basé sur le principe des itérations de Richardson (Saad, 2003). Une suite de vecteurs  $(X_k)_{k \in \mathbb{N}}$  est construite pour converger vers la solution d'une équation de la forme  $X = f(X)$  où  $f$  est une fonction continue, à l'aide du schéma itératif  $X_{k+1} = f(X_k)$  et d'un vecteur initial  $X_0$ .

Pour appliquer le principe des itérations de Richardson à l'équation 4.2, la matrice  $I + \Delta t C^{-1} G$  est « cassée » en deux matrices  $M$  et  $N$  telles que  $I + \Delta t C^{-1} G = M - N$  où  $M$  est une

matrice facilement inversible. L'équation 4.2 devient alors :

$$X_{m+1} = f(X_{m+1}) = M^{-1}NX_{m+1} + M^{-1}X_m + M^{-1}\Delta tC^{-1}P_m \quad (4.8)$$

L'efficacité du schéma itératif repose sur le choix des matrices  $M$  et  $N$ . La matrice  $M$  doit être facilement inversible, car l'opérateur  $M^{-1}$  est appliqué à chaque itération, tout en permettant une convergence rapide vers la solution exacte. Il est possible de montrer que la vitesse de convergence de ce schéma itératif dépend directement du rayon spectral  $\rho(M^{-1}N)$  de la matrice  $M^{-1}N$ , c'est-à-dire de sa plus grande valeur propre en module. Plus exactement, l'erreur  $e_k = \|X - X_k\|$  commise à la  $k$ -ème itération vérifie :

$$e_k \leq \rho(M^{-1}N)^k e_0 \quad (4.9)$$

où  $e_0 = \|X - X_0\|$  est l'erreur associée au vecteur initial  $X_0$ .

La nature du problème physique sous-jacent peut guider le choix des matrices  $M$  et  $N$ . Les circuits intégrés sont formés d'un empilement de couches minces et la chaleur tend à s'échapper principalement par les faces supérieure et inférieure du circuit. Par conséquent, les transferts thermiques dans la direction verticale  $\vec{z}$  prédominent par rapport aux transferts dans les directions  $\vec{x}$  et  $\vec{y}$ . Ce phénomène se traduit par un fort couplage thermique entre les cellules du maillage selon l'axe  $\vec{z}$ . Un choix naturel pour la matrice  $M$  est l'opérateur  $I + \Delta tC^{-1}G_z$  afin de résoudre le problème de transfert de chaleur selon l'axe  $\vec{z}$  à chaque itération. Étant donné que la matrice  $I + \Delta tC^{-1}G_z$  est tridiagonale, elle est facile à inverser et constitue un bon candidat pour développer un schéma itératif efficace. La matrice  $N$  correspondante est  $N = -\Delta tC^{-1}(G_y + G_x)$ . Le schéma itératif proposé conserve une complexité de calcul à chaque itération linéaire ainsi qu'un haut niveau de parallélisme.

Il reste maintenant à décider d'un critère de convergence pour déterminer quand arrêter le processus itératif. Étant donné un paramètre de précision  $\epsilon > 0$ , le critère de convergence idéal serait basé sur la distance entre la solution exacte  $X$  et la solution approchée  $X_k$  :

$$\|X - X_k\| < \epsilon \quad (4.10)$$

Ce test est bien sûr irréaliste, car il suppose de connaître la solution exacte du problème. Les deux tests les plus couramment utilisés sont basés soit sur la norme du vecteur résidu, soit sur la distance entre deux approximations successives. Ces deux tests sont mathématiquement équivalents. Par commodité de calcul, la méthode proposée utilise la distance entre deux

approximations successives comme test d'arrêt.

$$\|X_{k+1} - X_k\| < \epsilon \quad (4.11)$$

Ce test a l'avantage de stopper les itérations quand la méthode a convergé, mais également quand les erreurs d'arrondis (dues au calcul avec une précision finie) sont du même ordre de grandeur que le gain de précision obtenu à la dernière itération.

Employé avec un vecteur initial quelconque, il serait déraisonnable de penser que le schéma itératif proposé puisse à lui seul, donner une solution précise en un temps de simulation acceptable. Pour cette raison, la méthode proposée commence les itérations à partir de l'estimation fournie par l'algorithme de décomposition des opérateurs, dont l'erreur est déjà majorée par  $O(\Delta t^2)$ .

### 4.1.3 Algorithme de la méthode proposée

La méthode proposée pour simuler le régime transitoire résout efficacement l'équation 4.2 de la méthode Euler-arrière, en combinant une décomposition des opérateurs à un schéma itératif. Les étapes de la méthode proposée sont détaillées dans l'algorithme 2.

---

#### Algorithme 2 : Méthode proposée pour la simulation du régime transitoire

---

**Input** :  $C, G, (P_m)_{m \in \mathbb{N}}, X_0$  : température initiale,  $\Delta t, \epsilon$  : précision,  $s$  : nombre de pas à simuler

**Output** :  $(X_m)_{m \in \mathbb{N}}$  : vecteurs température aux instants  $t_m = m\Delta t$

```

1 for  $m = 1 \dots s$  do /* Pour chaque pas de temps */
2    $X_m \leftarrow X_{m-1} + \Delta t C^{-1} P_{m-1}$ ;
3    $X_m \leftarrow (I + \Delta t C^{-1} G_x)^{-1} X_m$ ;
4    $X_m \leftarrow (I + \Delta t C^{-1} G_y)^{-1} X_m$ ;
5    $X_m \leftarrow (I + \Delta t C^{-1} G_z)^{-1} X_m$ ;
6    $\tilde{X}_m \leftarrow 0$ ;
7   while  $\|X_m - \tilde{X}_m\| > \epsilon$  do /* Schéma itératif pour corriger l'erreur de
   décomposition */
8      $\tilde{X}_m \leftarrow X_m$ ;
9      $X_m \leftarrow -\Delta t C^{-1} (G_y + G_z) X_m$ ;
10     $X_m \leftarrow X_m + X_{m-1} + \Delta t C^{-1} P_{m-1}$ ;
11     $X_m \leftarrow (I + \Delta t C^{-1} G_z)^{-1} X_m$ ;
12  end
13 end

```

---

#### 4.1.4 Bilan

La méthode proposée permet de simuler le régime transitoire avec une bonne précision tout en ayant une complexité linéaire par rapport à la taille du problème. Elle possède en outre un haut niveau de parallélisme, très facilement exploitable. Pour cela, une technique de décomposition des opérateurs est utilisée pour calculer avec une complexité linéaire une bonne estimation de la température. Un schéma itératif est ensuite appliqué à cette estimation de température pour corriger l'erreur introduite par la décomposition des opérateurs et fournir une solution précise.

## 4.2 Méthode proposée pour la simulation du régime permanent

La simulation du régime permanent d'un circuit intégré à l'aide d'un modèle numérique se ramène à la résolution de l'équation matricelle :

$$GX = P \tag{4.12}$$

où  $G$  représente la matrice de conductance,  $X$  le vecteur température et  $P$  le vecteur puissance. L'équation 4.12 étant obtenue en discrétisant un problème de transfert de chaleur, la matrice  $G$  est creuse et inversible. La plupart du temps, la matrice  $G$  est symétrique, car les transferts de chaleur au sein des circuits intégrés se font par conduction thermique et la chaleur se propage aussi bien dans un sens que dans l'autre. Dans le cas très particulier des circuits refroidis par la circulation d'un fluide, le transport de chaleur par le fluide se fait uniquement dans le sens de l'écoulement du fluide, ce qui rend la matrice de conductance asymétrique. Pour être applicable à tout type de modèle thermique numérique du régime permanent, la méthode de calcul idéale devrait résoudre les problèmes symétriques comme asymétriques. La revue de littérature a montré que les méthodes préférées pour simuler le régime permanent sont les méthodes itératives par projection, en particulier la méthode du gradient conjugué (applicable aux problèmes symétriques) et la méthode GMRES (applicable aux problèmes symétriques et asymétriques).

Afin d'améliorer l'état de l'art, la méthode proposée pour simuler le régime permanent s'appuie sur les deux méthodes de projection du gradient conjugué et GMRES, puis les combine à un préconditionnement LU et une technique de décomposition de domaine. Pour comprendre comment combiner efficacement ces différentes méthodes, les sections suivantes détaillent les points essentiels de ces techniques et le rapport entre elles.



### 4.2.1 Méthodes de projection

Les méthodes de projection constituent une famille importante de méthodes itératives pour résoudre un problème de la forme  $Ax = b$  où  $A \in \mathbb{R}^{n \times n}$  est une matrice creuse,  $x$  et  $b$  sont deux vecteurs de taille  $n$ . Le principe général des méthodes de projection est de chercher une approximation de la solution par projection dans un sous-espace vectoriel. La figure 4.3 illustre le principe des méthodes de projection.

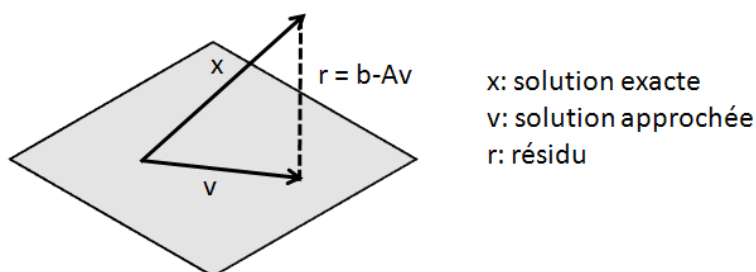


Figure 4.3 Méthode de projection

Les méthodes de projection utilisent deux suites  $(U_k)_{k \in \mathbb{N}}$  et  $(V_k)_{k \in \mathbb{N}}$  de sous-espaces vectoriels de  $\mathbb{R}^n$ . À la  $k$ -ème itération, la solution approchée  $x_k$  est cherchée dans le sous-espace  $U_k$  en imposant au résidu  $r_k$  d'être orthogonal à  $V_k$ . Les sous-espaces  $U_k$  et  $V_k$  sont emboîtés et de dimension croissante :

$$U_k \subset U_{k+1} \subset \dots \subset \mathbb{R}^n \quad (4.13)$$

$$V_k \subset V_{k+1} \subset \dots \subset \mathbb{R}^n \quad (4.14)$$

On a donc  $U_n = V_n = \mathbb{R}^n$ . Au bout de  $n$  itérations, la condition d'orthogonalité sur le résidu implique donc  $r_n \perp \mathbb{R}^n$ , c'est-à-dire  $r_n = 0$ . Une méthode de projection donne théoriquement le résultat exact en au plus  $n$  itérations. En pratique, les itérations sont arrêtées bien avant d'atteindre le nombre  $n$ , dès la satisfaction d'un critère de convergence  $\|r\| < \epsilon$  sur la norme du vecteur résidu.

La différence entre les nombreuses méthodes de projection existantes réside dans le choix des espaces vectoriels  $U_k$  et  $V_k$ . Ceux-ci sont généralement construits à partir des espaces de Krylov. L'espace de Krylov d'ordre  $k$  relatif à la matrice  $A$  et à un vecteur  $v$ , noté  $K_k(A, v)$ , est l'espace engendré par les vecteurs  $v, Av, A^2v, \dots, A^k v$ .

Pour calculer le régime permanent, la méthode de projection utilisée est la méthode de gradient conjugué quand la matrice  $A$  est symétrique, ou bien la méthode GMRES quand la matrice  $A$  est asymétrique.

## La méthode du gradient conjugué :

La méthode du gradient conjugué (Hestenes and Stiefel, 1952) est l'une des méthodes itératives les plus performantes connues à ce jour pour résoudre des systèmes de grande taille, dont la matrice est symétrique définie positive et creuse. Introduite en 1952, la méthode du gradient conjugué est devenue populaire surtout à partir des années 1970, en raison des applications industrielles nécessitant de résoudre des problèmes de grande taille et des progrès en informatique. Seuls les aspects les plus pertinents de cette méthode sont examinés ici, à savoir le principe théorique, l'algorithme, la complexité, la consommation mémoire, la vitesse de convergence et les limitations.

Principe théorique : La méthode du gradient conjugué est une méthode de projection sur un espace de Krylov. La solution approchée  $x_k$  est cherchée dans un sous-espace affine  $x_0 + K_k(A, r_0)$  en imposant la condition d'orthogonalité  $r_k \perp K_k(A, r_0)$ . Il est possible de montrer que la méthode du gradient conjugué sélectionne comme solution approchée le vecteur de l'espace de recherche qui minimise la distance à la solution exacte du problème.

Algorithme : La méthode du gradient conjugué prend comme données en entrée la matrice  $A$  du système, le vecteur second membre  $b$  et un paramètre de précision  $\epsilon > 0$ . Le test d'arrêt porte sur la norme du vecteur résidu. Dès que la norme du vecteur résidu est inférieure à la tolérance  $\epsilon$ , les itérations sont arrêtées. Le résultat retourné par la méthode est le vecteur solution calculé lors de la dernière itération. La méthode du gradient conjugué est très simple à mettre en oeuvre. Les étapes de calcul sont données dans l'algorithme 3.

Complexité : La complexité de la méthode du gradient conjugué à chaque itération est linéaire avec la taille  $n$  du problème (dans le cas de matrices creuses). À chaque itération, la méthode effectue une multiplication une matrice-vecteur, deux produits scalaires et trois additions de vecteurs. La complexité de la méthode est dominée par le coût de la multiplication matrice-vecteur, ce qui la rend particulièrement bien adaptée aux matrices creuses. Dans le cas d'un problème thermique discrétisé sur un maillage orthogonal, la matrice de conductance contient exactement 6 coefficients par ligne. Chaque itération de la méthode du gradient conjugué comporte donc environ  $10n$  opérations à virgule flottante.

Consommation de mémoire : La consommation de mémoire de la méthode du gradient conjugué augmente linéairement avec la taille du problème. En plus des données en entrée et en sortie, seulement trois vecteurs de taille  $n$  et trois scalaires doivent être alloués pour

---

**Algorithme 3** : Méthode du gradient conjugué
 

---

**Input** :  $A \in \mathbb{R}^{n \times n}$ ,  $b \in \mathbb{R}^n$ ,  $\epsilon$  (précision)

**Output** :  $x \in \mathbb{R}^n$  tel que  $\|b - Ax\| < \epsilon$ 
**1 Initialisation** :  $x \leftarrow 0$ ;  $r \leftarrow b$ ;  $p \leftarrow r$ ;

**2 while**  $\|r\| > \epsilon$  **do**
**3**      $q \leftarrow Ap$  ;

**4**      $\alpha \leftarrow \frac{\langle r|r \rangle}{\langle q|p \rangle}$  ;

**5**      $\beta \leftarrow \frac{1}{\|r\|}$  ;

**6**      $x \leftarrow x + \alpha p$  ;

**7**      $r \leftarrow r - \alpha q$  ;

**8**      $\beta \leftarrow \beta * \|r\|$  ;

**9**      $p \leftarrow r + \beta p$  ;

**10 end**


---

stocker les variables internes utilisées dans l'algorithme. La faible consommation de mémoire est un des principaux avantages de cette méthode.

Convergence : La convergence de la méthode du gradient conjugué s'effectue en au plus  $n$  itérations. Il s'agit donc théoriquement d'une méthode directe puisqu'elle donne le résultat exact en un nombre fini d'opérations. Il est possible de montrer (Hestenes and Stiefel, 1952) que la norme de l'erreur  $e_k = \|x - x_k\|$  commise à la  $k$ -ème itération vérifie :

$$e_k \leq 2 \left( \frac{\sqrt{\text{cond}(A) - 1}}{\sqrt{\text{cond}(A) + 1}} \right)^k e_0 \quad (4.15)$$

où  $\text{cond}(A) = \|A\|_2 \|A^{-1}\|_2$  est appelé nombre de conditionnement de la matrice  $A$ . La norme matricielle  $\|A\|_2$  est définie comme la racine carrée de la plus grande des valeurs propres de  $A$ . La vitesse de convergence de la méthode du gradient conjugué dépend directement du nombre de conditionnement  $\text{cond}(A)$  de la matrice  $A$ . Plus le système est bien conditionné et plus la méthode du gradient conjugué converge vite.

Limitations : La principale limitation de la méthode du gradient conjugué est d'être applicable seulement aux problèmes symétriques définis positifs. En effet, l'opérateur symétrique définie positif permet de définir un produit scalaire utilisé pour trouver la solution approchée par projection. De nombreuses méthodes alternatives existent pour résoudre des

systèmes plus généraux dont la matrice n'est pas nécessairement symétrique, notamment la méthode GMRES.

### La méthode GMRES :

La méthode GMRES (Saad, 2003) est considérée comme l'une des méthodes itératives les plus performantes pour la résolution de systèmes matriciels linéaires généraux. Développée au milieu des années 1980, la méthode GMRES fait encore l'objet de recherches mathématiques actives.

Principe théorique : La méthode GMRES fait également partie des méthodes de projection sur un espace de Krylov. La solution approchée  $x_k$  est cherchée dans un sous-espace affine  $x_0 + K_k(A, r_0)$  en imposant la condition d'orthogonalité  $r_k \perp AK_k(A, r_0)$ . Contrairement à la méthode du gradient conjugué qui minimise la norme de l'erreur  $\|x - x_k\|$  à chaque itération  $k$ , la méthode GMRES minimise la norme du résidu  $\|b - Ax_k\|$ .

---

#### Algorithme 4 : Méthode GMRES

---

**Input** :  $A \in \mathbb{R}^{n \times n}$ ,  $b \in \mathbb{R}^n$ ,  $\epsilon$  (précision),  $m$  (redémarrage)  
**Output** :  $x \in \mathbb{R}^n$  tel que  $\|b - Ax\| < \epsilon$

- 1 **Initialisation** :  $x \leftarrow 0$ ;  $r \leftarrow b$ ;
- 2 **while**  $\|r\| > \epsilon$  **do**
- 3      $v_1 \leftarrow r / \|r\|$ ;
- 4     **for**  $j=1, \dots, m$  **do**
- 5          $w \leftarrow Av_j$ ;
- 6         **for**  $i=1, \dots, j$  **do**
- 7              $h_{i,j} \leftarrow w^T v_i$ ;
- 8              $w \leftarrow w - h_{i,j} v_i$ ;
- 9         **end**
- 10          $h_{j+1,j} \leftarrow \|w\|_2$ ;
- 11          $v_{j+1} \leftarrow w / \|w\|_2$ ;
- 12     **end**
- 13      $V_m \leftarrow [v_1 \ v_2 \ \dots \ v_m]$ ;
- 14      $x \leftarrow x + V_m y_m$  où  $y_m$  minimise  $\|r - AV_k y\|_2$ ;
- 15      $r \leftarrow b - Ax$ ;
- 16 **end**

---

Algorithme : La méthode GMRES construit une base orthonormée  $\{v_1, \dots, v_k\}$  de l'espace de Krylov  $K_k(A, r_0)$  pour y rechercher plus facilement la solution approchée  $x_k$  qui minimise la norme du résidu. L'algorithme d'Arnoldi est généralement employé au coeur de la méthode

GMRES pour construire la base orthonormée de l'espace de Krylov. Le coût de stockage des vecteurs de la base orthonormée devient rapidement prohibitif à mesure que la dimension  $k$  de l'espace de Krylov croît. Ceci est particulièrement vrai pour les systèmes de grande taille. Pour remédier à ce problème, la méthode GMRES est périodiquement redémarrée après  $m$  itérations en utilisant la solution approchée  $x_m$  comme nouvelle donnée initiale. Les étapes de la méthode GMRES sont données dans l'algorithme 4.

Complexité : La complexité de la méthode GMRES est dominée par le procédé d'orthonormalisation permettant de construire une base orthonormée des espaces de Krylov. En supposant que la matrice  $A$  soit creuse, la complexité de la multiplication matrice-vecteur se situe quelque part entre  $O(n)$  et  $O(n^2)$ . En revanche, le reste des opérations d'orthonormalisation a une complexité en  $O(nk)$ , qui croît avec la dimension  $k$  des espaces de Krylov. La complexité de la méthode GMRES augmente donc à chaque itération. Si la méthode est redémarrée toutes les  $m$  itérations, sa complexité est majorée par  $O(nm)$ .

Consommation de mémoire : La consommation mémoire de la méthode GMRES peut devenir rapidement très importante. Elle dépend en fait du paramètre  $m$  choisi pour redémarrer la méthode. En effet, la méthode GMRES doit conserver tous les vecteurs de la base orthonormée des espaces de Krylov. Pour résoudre un problème de taille  $n$  en redémarrant la méthode toutes les  $m$  itérations, il faut être préparé à stocker  $m$  vecteurs de taille  $n$  en plus de la matrice  $A$  et des vecteurs  $x, b, w$ .

Convergence : En arithmétique exacte, la convergence de la méthode GMRES sans redémarrage est assurée en au plus  $n$  itérations, car il s'agit d'une méthode de projection. La convergence de la méthode GMRES avec redémarrage est prouvée lorsque la matrice  $A$  est définie positive. Concernant la vitesse de convergence de la méthode GMRES, il n'existe pas à ce jour de résultat théorique vraiment exploitable. Dans le cas où la matrice  $A$  est diagonalisable, il est possible de majorer la norme  $\|r_k\|_2$  du résidu obtenu après  $k$  itérations par

$$\|r_k\|_2 \leq \|r_0\| \text{cond}(P) \min_{q \in Q_k} \max_{\lambda \in S(A)} |q(\lambda)| \quad (4.16)$$

où  $P$  est la matrice formée des vecteurs propres de  $A$ ,  $Q_k$  est l'ensemble des polynômes  $q$  de degré  $\leq k$  tels que  $q(0) = 1$  et  $S(A)$  est le spectre de  $A$ . Puisque le conditionnement de la matrice  $P$  n'est pas connu et peut être relativement large, cette majoration a peu d'intérêt en pratique. Il faut donc procéder à des tests numériques pour évaluer la vitesse de convergence de la méthode GMRES sur une famille de problèmes donnés.

Limitations : La principale limitation de la méthode GMRES est sa consommation mémoire importante dans le cas où les redémarrages sont effectués tardivement. D'un autre côté, des redémarrages trop fréquents dégradent la vitesse de convergence de la méthode. Employée sur des problèmes mal conditionnés, la méthode GMRES converge très lentement, tout comme la méthode du gradient conjugué. L'utilisation d'un préconditionneur permet de remédier à ce problème.

#### 4.2.2 Préconditionnement incomplet LU

Le préconditionnement d'un système matriciel vise à augmenter la vitesse de convergence des méthodes itératives (Saad, 2003). Comme il a été mentionné à la section 4.2.1, la vitesse de convergence de la méthode du gradient conjugué dépend du nombre de conditionnement  $cond(A)$  de la matrice  $A$  et la vitesse de convergence de la méthode GMRES dépend du rayon spectral  $\rho(A)$  de  $A$ . Sans être équivalentes, ces deux notions sont liées. Globalement, plus  $cond(A)$  est proche de 1, plus le système est bien conditionné et plus vite converge la méthode itérative.

Le principe du préconditionnement consiste à remplacer le système  $Ax = b$  par un système équivalent  $C^{-1}Ax = C^{-1}b$  plus facile à résoudre par une méthode itérative. Pour cela, la matrice  $C^{-1}$  doit être choisie de façon à minimiser  $cond(C^{-1}A)$ . Pour obtenir un gain par rapport au système original, il faut au moins avoir  $cond(C^{-1}A) \leq cond(A)$ . La matrice  $C$  est appelée « matrice de préconditionnement » ou « préconditionneur ».

Trouver un bon préconditionneur pour résoudre un système matriciel relève parfois plus d'un art que d'une véritable science (Lascaux and Théodor, 1987). Les résultats théoriques sont rares, et contrairement à toute attente, certains préconditionneurs fonctionnent très bien tandis que d'autres non. En théorie, le meilleur préconditionneur que l'on puisse trouver vérifie  $C^{-1} = A^{-1}$ , car on a alors  $cond(C^{-1}A) = 1$  et la méthode itérative converge en une seule itération ! En pratique, on cherche à avoir  $C^{-1}$  le plus proche possible de  $A^{-1}$  sans que les calculs de  $C^{-1}$  soient trop coûteux. Très souvent, le calcul explicite de  $C$  ou même de  $C^{-1}A$  n'est pas possible, car trop coûteux. Le préconditionnement est appliqué lors de la résolution itérative en travaillant avec les opérateurs  $A$  et  $C^{-1}$ .

En résumé, un bon préconditionneur doit être peu coûteux en calcul, c'est-à-dire avec une complexité en  $O(N)$  ou  $O(N \log N)$ , car il est appliqué à chaque itération et il doit permettre de diminuer sensiblement le nombre de conditionnement du système à résoudre. De nombreux préconditionneurs simples et performants sont basés sur la factorisation LU de la matrice  $A$ .

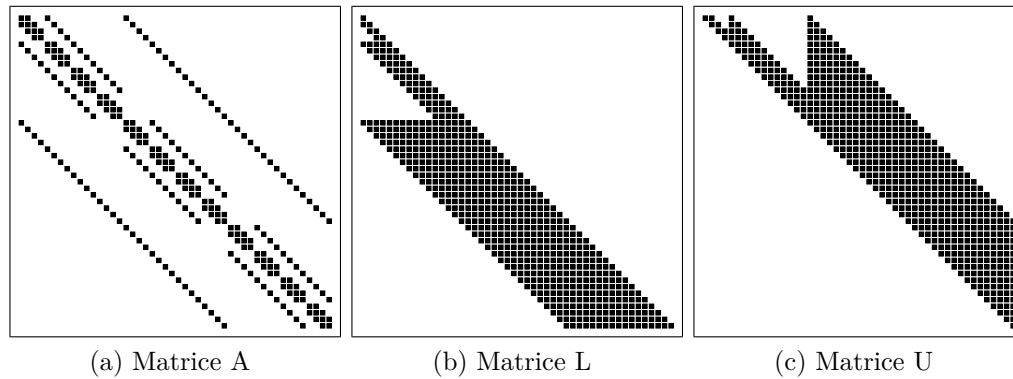


Figure 4.4 Factorisation LU de la matrice A

Lors de la factorisation LU illustrée sur la figure 4.4, on observe un remplissage des matrices  $L$  et  $U$  par des coefficients non nuls. Ce phénomène de remplissage est responsable du coût de calcul et de stockage en mémoire extrêmement lourd de la factorisation LU. La factorisation LU incomplète évite le remplissage des matrices  $L$  et  $U$  tout en tenant compte des coefficients prépondérants. Le principe est de trouver deux matrices creuses  $L$  et  $U$ , respectivement triangulaires inférieure et supérieure, telle que  $A \simeq LU$ . Pour cela, les coefficients trop petits qui apparaissent lors de la factorisation LU de  $A$  sont écartés, afin de conserver une structure creuse pour les matrices  $L$  et  $U$ . Une factorisation LU incomplète simple et performante est la factorisation ILU(0), qui impose aux matrices  $L$  et  $U$  d'avoir la même structure que la matrice  $A$ . Les étapes du calcul du préconditionnement ILU(0) (Saad, 2003) sont détaillées dans l'algorithme 5.

---

**Algorithme 5 : Algorithme ILU(0)**


---

```

Input :  $A \in \mathbb{R}^{n \times n}$ 
Output :  $L, U \in \mathbb{R}^{n \times n}$  (stockées dans A)
1 for  $k=1, \dots, n-1$  do
2   for  $i=k+1, \dots, n$  do
3     if  $(i, k) \in NZ$  then
4        $a_{ik} \leftarrow a_{ik}/a_{kk}$  ;
5       for  $j=k+1, \dots, n$  do
6         if  $(i, k) \in NZ$  then
7            $a_{ij} \leftarrow a_{ij} - a_{ik} * a_{kj}$  ;
8         end
9       end
10    end
11  end
12 end

```

---

Les coefficients de la matrice  $A$  sont notés  $(a_{i,j})$  avec  $i, j \in [1; n]$ . L'ensemble des couples (ligne, colonne) correspondant aux coefficients non nuls de la matrice  $A$  est noté  $NZ$ . Le coût de calcul pour appliquer le préconditionnement  $ILU(0)$  est en fait le même que celui de la matrice  $A$ , puisque ces matrices contiennent le même nombre de coefficients non nuls. L'approximation  $A \simeq LU$  est bien sûr très grossière, ce qui se traduit la plupart du temps par une norme  $\|I - U^{-1}L^{-1}A\| \gg 1$ . Cependant, le préconditionnement  $ILU(0)$  donne d'excellents résultats en pratique. Pour certains types de matrices, il est possible de montrer que le rayon spectral de la matrice  $U^{-1}L^{-1}A$ , c'est-à-dire la plus grande valeur propre en valeur absolue, est nettement plus petit que celui de  $A$  et donc  $cond(U^{-1}L^{-1}A) \leq cond(A)$ . La qualité d'un préconditionneur n'est donc pas forcément liée à la qualité de l'approximation de la matrice  $A$ . Par conséquent, la recherche intuitive d'un préconditionneur en se basant sur le sens physique du problème n'est pas toujours un gage de succès. Pour augmenter la qualité du préconditionneur  $ILU(0)$ , c'est-à-dire diminuer le nombre  $cond(U^{-1}L^{-1}A)$  il est possible d'autoriser un remplissage plus important des matrices  $L$  et  $U$ . Le préconditionnement  $ILU(n)$  remplit jusqu'à  $n$  coefficients en plus de ceux initialement présents dans la matrice  $A$ .

L'utilisation d'un préconditionnement LU incomplet pour augmenter la vitesse de convergence des méthodes du gradient conjugué et GMRES est une technique désormais assez bien acceptée en analyse numérique, déjà appliquée à la simulation thermique des circuits intégrés. Pour améliorer cet état de l'art de la simulation thermique des circuits intégrés, la méthode proposée combine en plus une méthode de décomposition de domaine.

### 4.2.3 Décomposition de domaine

Les méthodes de décomposition de domaine (Saad, 2003) sont basées sur le célèbre principe « diviser pour régner ». Ces méthodes tentent de résoudre un problème défini sur un domaine  $\Omega = \bigcup_i \Omega_i$  à partir des solutions calculées sur les sous-domaines  $\Omega_i$ .

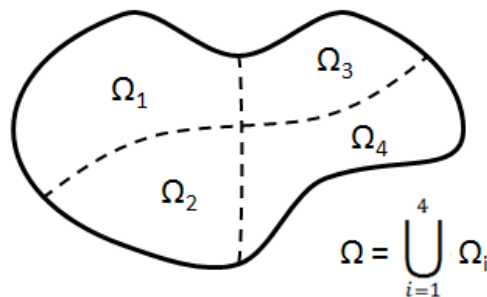


Figure 4.5 Décomposition d'un domaine  $\Omega$  en quatre sous-domaines



Plusieurs raisons motivent une décomposition de domaine. Premièrement, la résolution du problème sur un sous-domaine  $\Omega_i$  de dimension plus réduite est souvent plus rapide et moins coûteuse en mémoire. Deuxièmement, lorsque le problème implique des phénomènes physiques de nature différente, il est généralement plus simple de le décomposer sur des domaines séparés où les équations à résoudre sont de même nature. Troisièmement, le calcul dans les différents sous-domaines peut souvent être effectué en parallèle, ce qui permet d'exploiter le parallélisme des architectures informatiques modernes. Aujourd'hui, la popularité des méthodes de décomposition de domaine est principalement due à l'émergence des technologies de calcul parallèle.

On distingue deux familles de méthodes de décomposition de domaine : les méthodes basées sur la procédure alternative de Schwarz et les méthodes basées sur le complément de Schur. La méthode proposée est basée sur le complément de Schur, utilisé avec un coloriage « rouge-noir ».

### Complément de Schur :

Le complément de Schur (Saad, 2003) est une matrice qui apparaît naturellement lors d'une décomposition de domaine. Pour illustrer cette méthode, considérons le problème de conduction thermique bidimensionnel discrétisé à l'aide d'un maillage orthogonal, représenté à la figure 4.6. Le cas bidimensionnel est considéré seulement afin d'alléger la représentation de la méthode. L'extension au cas tridimensionnel est immédiate. La figure 4.6 présente la numé-

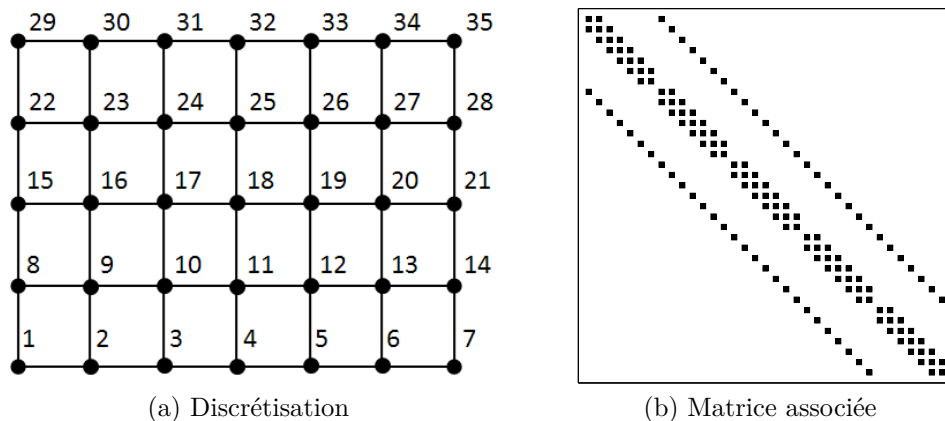


Figure 4.6 Problème original

rotation des cellules du maillage et la matrice de conductance associée au problème original. Une décomposition de domaine possible est illustrée sur la figure 4.7. Le maillage est divisé en quatre sous-domaines (repérés par des ellipses) séparés par des interfaces (colorées en rouge).

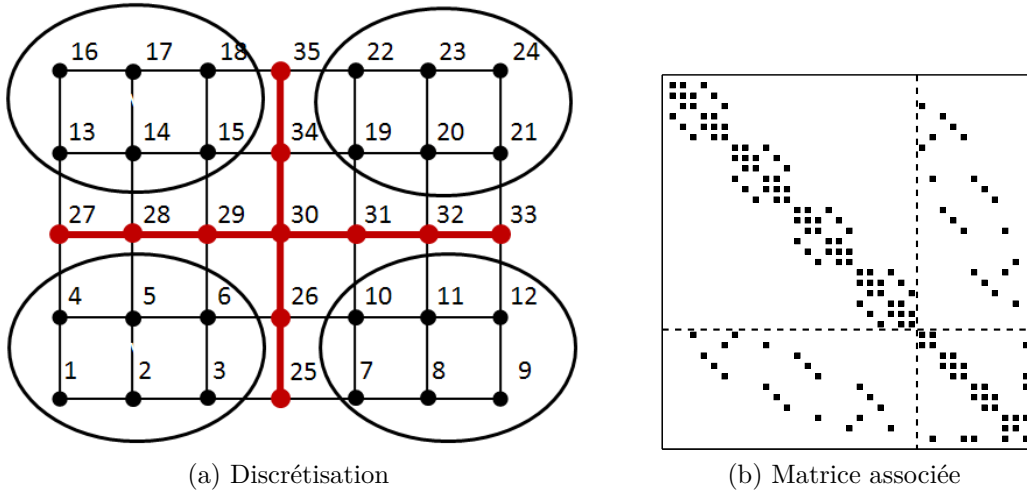


Figure 4.7 Décomposition de domaine

La décomposition de domaine se traduit par une numérotation différente des cellules du maillage. Les cellules situées dans les sous-domaines sont numérotées en premier, puis viennent les cellules situées sur les interfaces. Il est important de numérotter les cellules d'un même sous-domaine consécutivement afin de faire apparaître une structure par bloc dans la matrice associée au problème. On note  $B$  la matrice associée aux sous-domaines,  $C$  la matrice associée aux interfaces,  $E$  et  $F$  les matrices de couplage entre les sous-domaines et les interfaces. L'équation  $Ax = b$  initiale peut se mettre sous la forme du système :

$$\begin{pmatrix} B & E \\ F & C \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \quad (4.17)$$

Le vecteur  $u$  représente les variables associées aux sous-domaines et  $v$  les variables associées aux interfaces. Le vecteur  $u$  s'exprime en fonction des variables d'interface par :

$$u = B^{-1}(f - Ev) \quad (4.18)$$

En substituant 4.18, dans l'équation 4.17, on obtient :

$$(C - FB^{-1}E)v = g - FB^{-1}f \quad (4.19)$$

La résolution de l'équation 4.19 donne accès aux variables d'interfaces, qui permettent de calculer les variables restantes en les injectant dans l'équation 4.18. La matrice

$$S = C - FB^{-1}E \quad (4.20)$$

est appelée « complément de Schur ». La méthode de calcul basée sur le complément de Schur est résumée par l'algorithme 6.

---

**Algorithme 6** : Méthode du complément de Schur

---

**Input** :  $B \in \mathbb{R}^{p \times p}$ ,  $C \in \mathbb{R}^{q \times q}$ ,  $E \in \mathbb{R}^{q \times p}$ ,  $F \in \mathbb{R}^{p \times q}$ ,  $f \in \mathbb{R}^p$ ,  $g \in \mathbb{R}^q$

**Output** :  $u \in \mathbb{R}^p$ ,  $v \in \mathbb{R}^q$

- 1  $u \leftarrow B^{-1}f$  ;
  - 2  $v \leftarrow g - Fu$  ;
  - 3  $v \leftarrow S^{-1}v$  ;
  - 4  $u \leftarrow u - B^{-1}Ev$  ;
- 

Les étapes délicates de cet algorithme sont les étapes 1 et 3, impliquant la résolution des systèmes matriciels  $B$  et  $S$ . La méthode de calcul proposée exploite un coloriage rouge-noir afin que la matrice  $B$  issue de la décomposition de domaine, ait une structure permettant une résolution rapide.

**Coloriage rouge-noir :**

Le coloriage rouge-noir est la version à deux couleurs du célèbre problème de coloriage. Le principe est de colorier les cellules du maillage à l'aide de deux couleurs (rouge et noir) de sorte que deux cellules voisines n'aient jamais la même couleur. La figure 4.8 illustre la décomposition de domaine rouge-noir sur le problème bidimensionnel considéré à titre d'exemple.

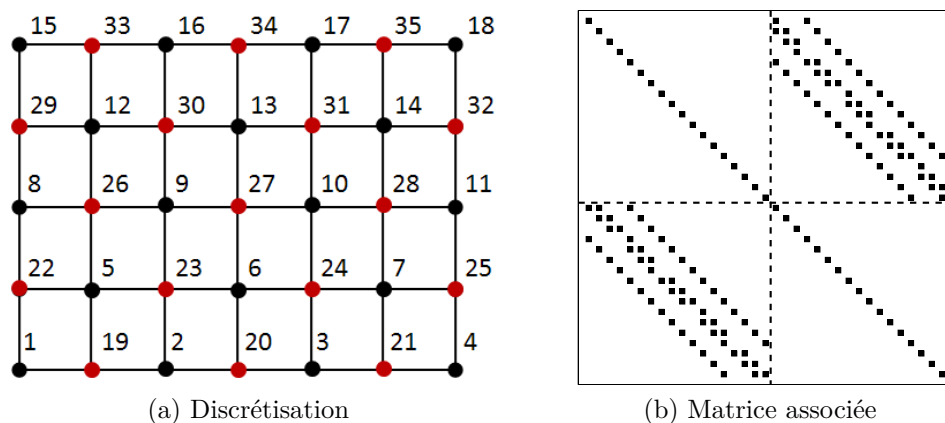


Figure 4.8 Coloriage rouge-noir du problème

Lorsque le maillage est orthogonal, il est toujours possible de trouver un coloriage rouge-noir, quelle que soit la dimension, un, deux ou trois, du problème. La méthode proposée exploite le

coloriage rouge-noir pour effectuer une décomposition de domaine. Les cellules noires constituent les domaines tandis que les cellules rouges jouent le rôle des interfaces. Il s'agit d'une décomposition de domaine très particulière avec des domaines de taille 1 et autant d'interfaces que de domaines à peu de chose près. Comme illustrée sur la figure 4.8, les matrices  $B$  et  $C$  associées respectivement aux domaines et aux interfaces sont devenues diagonales. Leur inversion est donc immédiate! Toutes les méthodes discutées jusqu'ici peuvent maintenant être réunies pour proposer une méthode de calcul performante pour simuler le régime permanent des circuits intégrés.

#### 4.2.4 Algorithme de la méthode proposée

La méthode proposée pour simuler le régime permanent résout efficacement l'équation 4.12 en combinant une décomposition de domaine rouge-noir, la méthode du complément de Schur et une méthode de projection avec un préconditionnement incomplet LU. Les étapes de la méthode proposée sont détaillées dans l'algorithme 7.

---

#### Algorithme 7 : Méthode proposée pour le calcul du régime permanent

---

**Input** :  $A \in \mathbb{R}^{n \times n}$ ,  $b \in \mathbb{R}^n$ ,  $\epsilon$  (précision)  
**Output** :  $x \in \mathbb{R}^n$  tel que  $\|b - Ax\| < \epsilon$

- 1 Trouver la matrice de permutation  $P$  correspondant au coloriage rouge-noir
- 2 Calculer la décomposition  $PAP^{-1} = \begin{pmatrix} B & E \\ F & C \end{pmatrix}$  et  $Pb = \begin{pmatrix} f \\ g \end{pmatrix}$ ;
- 3  $S \leftarrow C - FB^{-1}E$  ;
- 4 Calculer un préconditionnement  $S \simeq LU$  ;
- 5  $u \leftarrow B^{-1}f$  ;
- 6  $w \leftarrow g - Fu$  ;
- 7 **if**  $S = S^T$  **then**
- 8 | Résoudre  $Sv = w$  par la méthode du gradient conjugué préconditionnée avec  $L, U$  ;
- 9 **else**
- 10 | Résoudre  $Sv = w$  par la méthode GMRES préconditionnée avec  $L, U$  ;
- 11 **end**
- 12  $u \leftarrow u - B^{-1}Ev$  ;
- 13  $x \leftarrow P^{-1}[u \ v]^T$  ;

---

#### 4.2.5 Bilan

La méthode proposée permet de simuler efficacement le régime permanent avec une bonne précision et une complexité qui augmente linéairement avec la taille du problème. Pour cela, les méthodes du gradient conjugué et GMRES sont combinées à un préconditionnement LU

incomplet, afin d'améliorer leur vitesse de convergence, ainsi qu'à une technique de décomposition de domaine basée sur la méthode du complément de Schur. Un coloriage rouge-noir est également utilisé pour trouver une décomposition de domaine efficace.

## CHAPITRE 5 RÉSULTATS EXPÉRIMENTAUX

L'objectif de ce chapitre est d'évaluer la précision et la performance des méthodes de calcul proposées au chapitre 4 pour la simulation thermique des circuits intégrés. Trois problèmes modèles sont considérés pour cette évaluation. Après une brève description, ces problèmes sont simulés en régime transitoire et en régime permanent.

### 5.1 Description des problèmes modèles

#### 5.1.1 Circuit intégré 3D refroidi par un dissipateur de chaleur

Le premier problème modèle considéré est le circuit Wioming (Dutoit et al., 2013; Santos et al., 2013), dont la structure est représentée sur la figure 5.1. Il s'agit d'un circuit intégré 3D développé conjointement par les compagnies ST-Microelectronics, ST-Ericsson et Cadence. Ce circuit représente l'état de l'art en matière de circuit intégré 3D.

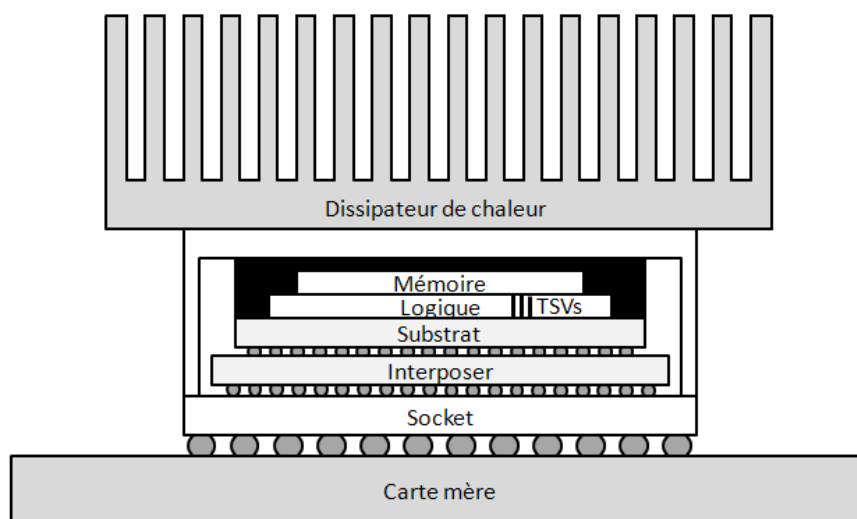


Figure 5.1 Circuit Wioming (Santos et al., 2013)

Le circuit Wioming est composé de deux puces reliées par une matrice de connexions verticales dans une configuration « face-arrière ». La première puce est une couche de logique intégrant un réseau sur puce asynchrone, tandis que la deuxième puce est une couche de mémoire. Le circuit Wioming est muni d'un dissipateur de chaleur classique à ailettes, dont les spécifications sont données dans (Santos et al., 2013).

La figure 5.2 présente le plan de masse simplifié de la couche de logique du circuit Wioming, indiquant la position des sources de chaleur.

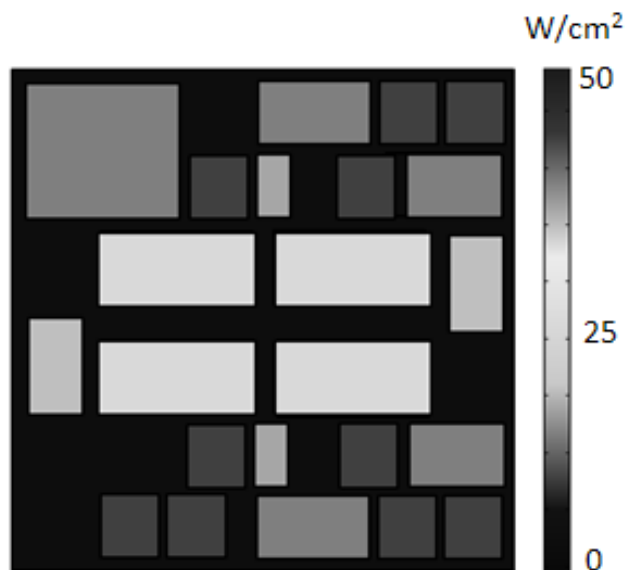


Figure 5.2 Plan de masse simplifié du circuit Wioming (Santos et al., 2013)

Le circuit Wioming est discrétisé à l'aide d'un maillage orthogonal avec différentes résolutions spatiales dans le plan horizontal  $400\mu m \times 400\mu m$ ,  $200\mu m \times 200\mu m$  et  $100\mu m \times 100\mu m$  conduisant respectivement à trois problèmes matriciels de taille  $0.6 \times 10^5$ ,  $1.3 \times 10^5$  et  $3.0 \times 10^5$  inconnues.

### 5.1.2 Circuit intégré 3D refroidi par un liquide

Le deuxième problème considéré est un circuit intégré 3D refroidi par la circulation d'un liquide, inspiré du circuit Single Chip Cloud Computer (Scc) développé par Intel (Howard and al., 2011) Ce circuit, appelé « Scc3D », est représenté sur la figure 5.3.

Le circuit Scc3D est composé de deux puces, reliées par des connexions verticales. La puce inférieure correspond au circuit Scc d'Intel, il s'agit d'une couche de logique. La puce supérieure est une couche de mémoire, constituée de banques de mémoire DRAM. Le Scc est un circuit à haute performance destiné aux tâches de calcul intensives. Étant donné la dissipation de puissance importante du Scc (jusqu'à 125 watts), ce circuit est un bon candidat pour le refroidissement par circulation de liquide.

La figure 5.4 présente le plan de masse des deux puces du Scc3D, c'est-à-dire la couche de logique et la couche de mémoire, indiquant la position des différentes sources de chaleur.

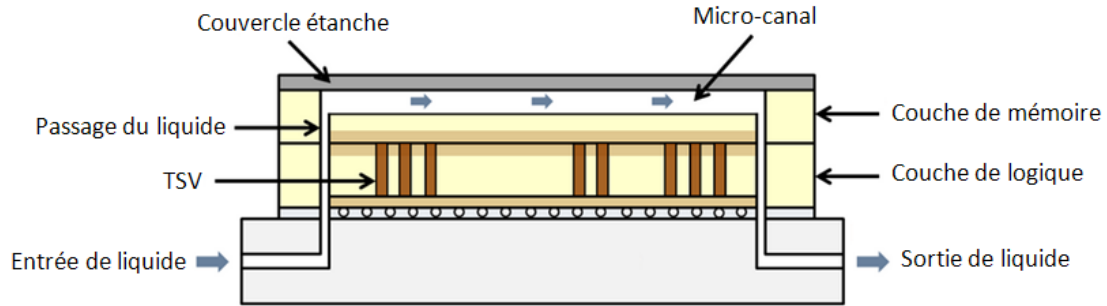
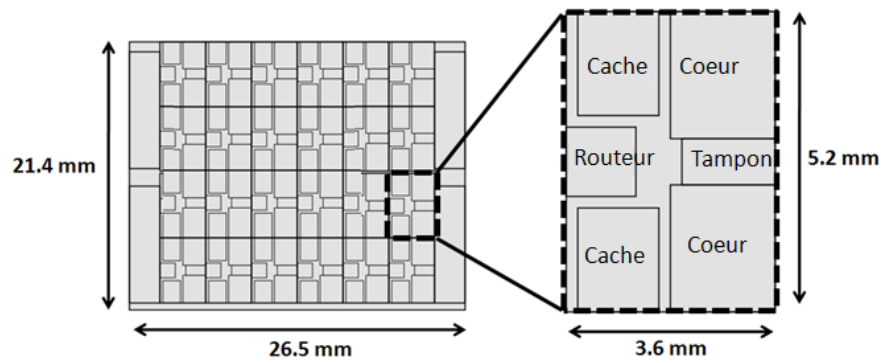
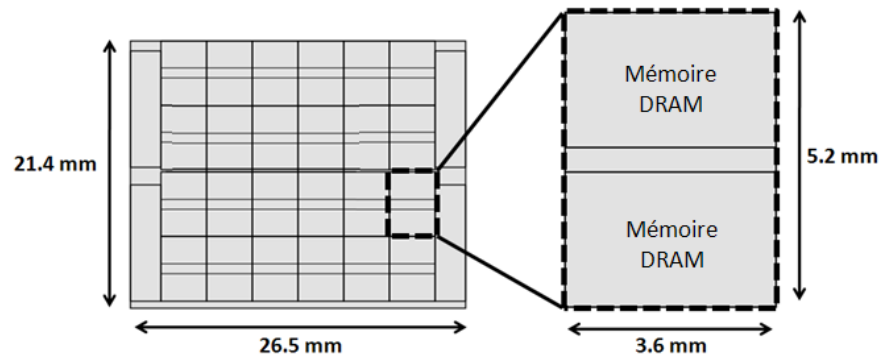


Figure 5.3 Circuit Scc3D



(a) Couche de logique



(b) Couche de mémoire

Figure 5.4 Plan de masse du circuit Scc3D

Le circuit Scc3D est discrétisé à l'aide d'un maillage orthogonal. La dimension des micro-canaux impose la taille des cellules du maillage. Trois largeurs typiques de micro-canaux,  $300\mu m$ ,  $200\mu m$  et  $100\mu m$  sont considérées, conduisant respectivement à trois problèmes matriciels de taille  $2.1 \times 10^5$ ,  $4.0 \times 10^5$  et  $6.1 \times 10^5$  inconnues.



### 5.1.3 Téléphone intelligent refroidi par convection naturelle

Le troisième problème considéré est un téléphone intelligent, dont le modèle simplifié est basé sur l'iPhone 5 d'Apple. Bien que le téléphone intelligent ne soit pas vraiment un circuit intégré, il s'agit d'un système électronique de géométrie rectangulaire, composé d'une carte mère sur laquelle sont fixés plusieurs composants, dont des circuits intégrés. La méthodologie employée pour la simulation thermique des circuits intégrés est donc applicable à ce système, qui constitue un cas d'étude très intéressant par sa complexité. La figure 5.5 présente les composants de l'iPhone 5 et le maillage du modèle simplifié de l'iPhone 5.

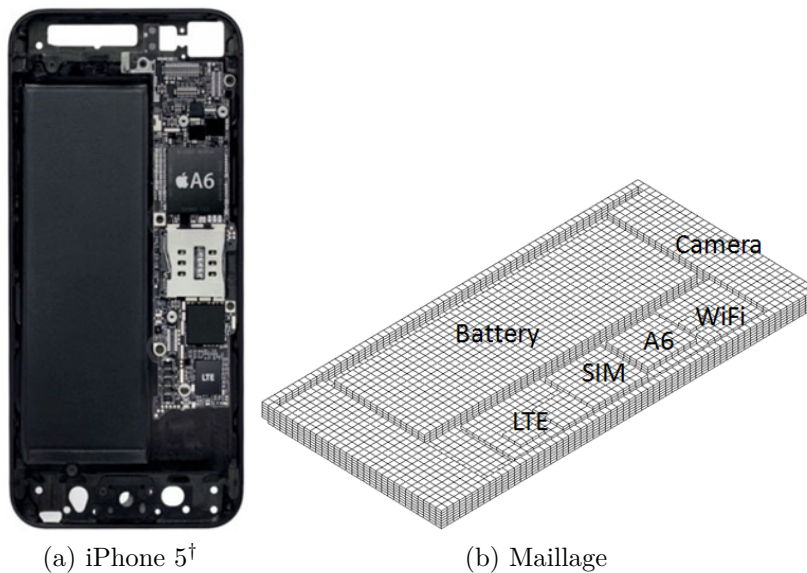


Figure 5.5 Modélisation d'un téléphone intelligent

La chaleur s'échappe du téléphone par convection naturelle avec l'air ambiant. Les coefficients de transfert par convection pour les six faces du téléphone ainsi que les propriétés thermiques des différents matériaux sont tirés des travaux de Luo *et al.* (Luo et al., 2008). Étant donné les dimensions et la complexité importantes du téléphone intelligent, un grand nombre de cellules est nécessaire pour le discrétiser convenablement. Le téléphone est discrétisé avec une résolution spatiale variant de  $100 \mu m \times 100 \mu m$  à  $400 \mu m \times 400 \mu m$  dans les circuits intégrés embarqués (Apple A6, module WiFi, module LTE, etc.) et de  $400 \mu m \times 400 \mu m$  à  $800 \mu m \times 800 \mu m$  dans les autres éléments (batterie, carte mère, etc.). Ces différentes discrétisations conduisent à trois problèmes matriciels volumineux de  $1.4 \times 10^6$ ,  $4.0 \times 10^6$  et  $7.2 \times 10^6$  inconnues.

† Image tirée de [www.ifixit.com](http://www.ifixit.com)

## 5.2 Simulation du régime transitoire

La méthode proposée pour la simulation du régime transitoire est comparée à la méthode LU pour matrices creuses implémentée dans la librairie SuperLU (Demmel et al., 1999). Le choix la méthode LU de SuperLU comme méthode de référence est justifié par ses très bonnes performances et son utilisation dans un grand nombre de modèles thermiques du régime transitoire (Sridhar et al., 2010; Liu et al., 2012b).

### 5.2.1 Évaluation de la précision

La précision de la méthode proposée est évaluée sur la base de l'erreur maximale ( $^{\circ}C$ ) commise au cours la simulation transitoire, rapportée à l'élévation de température maximale dans le circuit. Il s'agit donc d'une erreur relative, calculée de la manière suivante :

$$e = \max_{1 \leq m \leq s} \frac{\|X_m - Y_m\|_{\infty}}{\|Y_{\infty}\|_{\infty}} \quad (5.1)$$

où  $s$  est le nombre de pas de temps de la simulation,  $X_m$  est le vecteur température à l'instant  $t_m = m\Delta t$  calculé à l'aide la méthode proposée,  $Y_m$  est le vecteur température à l'instant  $t_m$  calculé avec la méthode de référence et  $Y_{\infty}$  est la température en régime permanent calculée avec la méthode de référence. La norme vectorielle  $\|\cdot\|_{\infty}$  est définie par  $\|X\|_{\infty} = \max_{1 \leq i \leq N} (x_i)$ , où  $(x_i)_{1 \leq i \leq N}$  sont les composantes du vecteur  $X$ .

Dans un premier temps, la méthode proposée est appliquée sans corriger l'erreur de décomposition. Le tableau 5.1 présente l'erreur relative maximale commise pour les différents problèmes, sans corriger l'erreur de décomposition. Les différents problèmes sont simulés avec des pas de temps  $\Delta t$  typiques allant de 0.1 *ms* à 100 *ms*. La taille importante de certains problèmes ne permet pas d'appliquer la méthode LU de la librairie SuperLU, car malgré les bonnes performances de cette librairie, la consommation de mémoire vive est trop élevée. C'est le cas des problèmes dont la taille excède  $5.4 \times 10^5$  inconnues, dont la résolution avec SuperLU nécessite plus de 15 *GB* de mémoire. Les résultats de la comparaison sont alors indiqués comme non disponibles (N/A).

Le tableau 5.1 montre la variation de l'erreur de décomposition en fonction du pas de temps utilisé et de la taille du problème. L'erreur de décomposition est à peu près proportionnelle au pas de temps  $\Delta t$  tel que le prédit la théorie. Dans le cas  $\Delta t = 100ms$ , les résultats sont particulièrement imprécis pour le circuit Scc3D, ce qui s'explique en analysant la réponse la réponse transitoire du circuit Scc3D à une dissipation de puissance variable, illustrée sur la figure 5.6.

Tableau 5.1 Erreur relative maximale (%) **sans correction** de l'erreur de décomposition

|         |                       | $\Delta t = 0.1$ ms | $\Delta t = 1$ ms | $\Delta t = 10$ ms | $\Delta t = 100$ ms |
|---------|-----------------------|---------------------|-------------------|--------------------|---------------------|
| Wyoming | $n = 0.6 \times 10^5$ | 0.03%               | 0.3%              | 1.4%               | 18.6%               |
|         | $n = 1.3 \times 10^5$ | 0.03%               | 0.3%              | 1.6%               | 18.9%               |
|         | $n = 3.0 \times 10^5$ | 0.03%               | 0.4%              | 1.7%               | 19.1%               |
| Scc3D   | $n = 2.1 \times 10^5$ | 2.4%                | 22.3%             | 105.7%             | 315.2%              |
|         | $n = 4.0 \times 10^5$ | 2.6%                | 22.5%             | 106.1%             | 316.9%              |
|         | $n = 6.1 \times 10^5$ | N/A                 | N/A               | N/A                | N/A                 |
| iPhone5 | $n = 1.4 \times 10^6$ | N/A                 | N/A               | N/A                | N/A                 |
|         | $n = 4.0 \times 10^6$ | N/A                 | N/A               | N/A                | N/A                 |
|         | $n = 7.2 \times 10^6$ | N/A                 | N/A               | N/A                | N/A                 |

N/A : Problème de taille trop importante pour y appliquer la méthode SuperLU

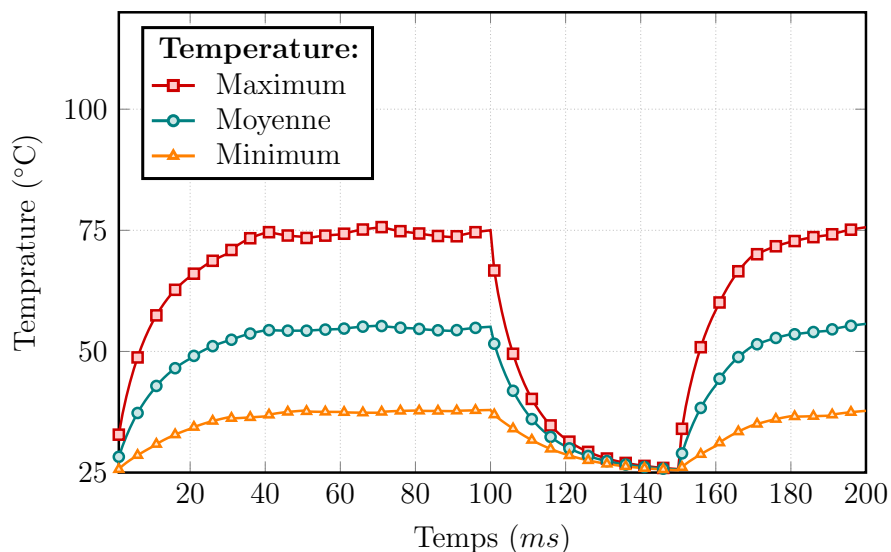


Figure 5.6 Réponse transitoire du circuit Scc3D

La figure 5.6 montre que la constante de temps du circuit Scc3D est de l'ordre de 30 ms. Tout se passe comme si le régime permanent était calculé comme limite du régime transitoire en une seule itération. Il n'est donc pas surprenant d'obtenir des résultats aberrants. Le tableau 5.1 montre également que l'erreur de décomposition augmente légèrement avec la taille du problème, donc avec la finesse du maillage. L'utilisation d'un maillage plus fin, c'est-à-dire avec des cellules de plus petite taille, entraîne une augmentation en valeur absolue des coefficients de la matrice  $C^{-1}G$ , et donc une augmentation de l'erreur de décomposition. L'enseignement principal du tableau 5.1 est que l'erreur de décomposition doit absolument être corrigée pour obtenir des résultats fiables.

Dans un deuxième temps, la méthode proposée est appliquée avec la correction de l'erreur de décomposition. Le paramètre de précision  $\epsilon = 10^{-4}$  est utilisé pour déterminer l'arrêt du processus itératif servant à corriger l'erreur de décomposition. Le tableau 5.2 montre l'erreur relative maximale commise sur l'ensemble de la simulation transitoire.

Tableau 5.2 Erreur relative maximale (%) **avec correction** de l'erreur de décomposition

|         |                       | $\Delta t = 0.1$ ms | $\Delta t = 1$ ms | $\Delta t = 10$ ms | $\Delta t = 100$ ms |
|---------|-----------------------|---------------------|-------------------|--------------------|---------------------|
| Wyoming | $n = 0.6 \times 10^5$ | 0.01%               | 0.01%             | 0.01%              | 0.01%               |
|         | $n = 1.3 \times 10^5$ | 0.01%               | 0.01%             | 0.01%              | 0.01%               |
|         | $n = 3.0 \times 10^5$ | 0.01%               | 0.01%             | 0.01%              | 0.01%               |
| Scc3D   | $n = 2.1 \times 10^5$ | 0.01%               | 0.01%             | 0.01%              | 0.01%               |
|         | $n = 4.0 \times 10^5$ | 0.01%               | 0.01%             | 0.01%              | 0.01%               |
|         | $n = 6.1 \times 10^5$ | N/A                 | N/A               | N/A                | N/A                 |
| iPhone5 | $n = 1.4 \times 10^6$ | N/A                 | N/A               | N/A                | N/A                 |
|         | $n = 4.0 \times 10^6$ | N/A                 | N/A               | N/A                | N/A                 |
|         | $n = 7.2 \times 10^6$ | N/A                 | N/A               | N/A                | N/A                 |

N/A : Problème de taille trop importante pour y appliquer la méthode SuperLU

Sans surprise, le processus itératif corrige très bien l'erreur de décomposition. Les résultats obtenus diffèrent des résultats donnés par la méthode de référence (SuperLU) de moins de 0.01% pour tous les problèmes considérés. Il convient maintenant d'analyser l'impact sur les performances de la correction de l'erreur de décomposition.

### 5.2.2 Évaluation des performances

Toutes les expérimentations sont réalisées sur un ordinateur de bureau équipé d'un processeur Intel(R) Core(TM) i7-3770 CPU@3.40GHz avec 16 GB de mémoire principale. Les résultats reportés correspondent à des versions non parallélisées de la méthode proposée et de la méthode de référence SuperLU.

Afin d'évaluer le coût de calcul associé à la correction de l'erreur de décomposition, le tableau 5.3 reporte le temps de calcul moyen de la méthode proposée à chaque pas de temps, sans corriger l'erreur de décomposition. Comme attendu, le temps de calcul de l'algorithme de décomposition ne dépend pas du pas de temps utilisé, mais seulement de la taille du problème. Le tableau 5.4 reporte le temps de calcul moyen de la méthode proposée à chaque pas de temps, avec la correction de l'erreur de décomposition.

On remarque que le temps de calcul pour corriger l'erreur de décomposition augmente avec le pas de temps utilisé, ce qui est logique car l'erreur de décomposition augmente elle-même

Tableau 5.3 Temps de calcul par pas de temps **sans correction** de l'erreur de décomposition

|         |                       | $\Delta t = 0.1$ ms | $\Delta t = 1$ ms | $\Delta t = 10$ ms | $\Delta t = 100$ ms |
|---------|-----------------------|---------------------|-------------------|--------------------|---------------------|
| Wyoming | $n = 0.6 \times 10^5$ | 2.3 ms              | 2.3 ms            | 2.4 ms             | 2.3 ms              |
|         | $n = 1.3 \times 10^5$ | 4.0 ms              | 4.1 ms            | 4.0 ms             | 4.0 ms              |
|         | $n = 3.0 \times 10^5$ | 8.5 ms              | 8.5 ms            | 8.5 ms             | 8.4 ms              |
| Scc3D   | $n = 2.1 \times 10^5$ | 5.8 ms              | 5.8 ms            | 5.9 ms             | 5.8 ms              |
|         | $n = 4.0 \times 10^5$ | 11.1 ms             | 11.0 ms           | 11.1 ms            | 11.2 ms             |
|         | $n = 6.1 \times 10^5$ | 16.7 ms             | 16.7 ms           | 16.7 ms            | 16.7 ms             |
| iPhone5 | $n = 1.4 \times 10^6$ | 42.1 ms             | 42.1 ms           | 42.1 ms            | 42.1 ms             |
|         | $n = 4.0 \times 10^6$ | 121 ms              | 121 ms            | 121 ms             | 121 ms              |
|         | $n = 7.2 \times 10^6$ | 212 ms              | 212 ms            | 212 ms             | 212 ms              |

Tableau 5.4 Temps de calcul par pas de temps **avec correction** de l'erreur de décomposition

|         |                       | $\Delta t = 0.1$ ms | $\Delta t = 1$ ms | $\Delta t = 10$ ms | $\Delta t = 100$ ms |
|---------|-----------------------|---------------------|-------------------|--------------------|---------------------|
| Wyoming | $n = 0.6 \times 10^5$ | 2.5 ms              | 3.9 ms            | 8.9 ms             | 12.6 ms             |
|         | $n = 1.3 \times 10^5$ | 5.3 ms              | 8.4 ms            | 19.6 ms            | 27.9 ms             |
|         | $n = 3.0 \times 10^5$ | 12.2 ms             | 19.4 ms           | 43.4 ms            | 61.7 ms             |
| Scc3D   | $n = 2.1 \times 10^5$ | 56.1 ms             | 263 ms            | 1.02 sec           | 1.51 sec            |
|         | $n = 4.0 \times 10^5$ | 108 ms              | 574 ms            | 2.1 sec            | 3.1 sec             |
|         | $n = 6.1 \times 10^5$ | 167 ms              | 792 ms            | 3.1 sec            | 4.6 sec             |
| iPhone5 | $n = 1.4 \times 10^6$ | 44.8 ms             | 70.1 ms           | 378 ms             | 3.1 sec             |
|         | $n = 4.0 \times 10^6$ | 128 ms              | 200 ms            | 1.1 sec            | 9.0 sec             |
|         | $n = 7.2 \times 10^6$ | 230 ms              | 360 ms            | 1.9 sec            | 16.7 sec            |

avec le pas de temps. Lorsque l'erreur de décomposition est inférieure à 10%, le temps supplémentaire pour appliquer la correction itérative n'est pas excessivement pénalisant. Ce temps reste en première approximation inférieur à 10 fois le temps de calcul de l'algorithme de décomposition. En revanche, lorsque l'erreur de décomposition est très importante, comme pour le circuit Scc3D simulé avec  $\Delta t = 100$  ms, le schéma itératif peut ralentir la simulation d'un facteur supérieur à  $\times 100$ . Ce résultat confirme que le schéma itératif ne converge pas assez rapidement pour résoudre à lui seul le problème en un temps acceptable. Il doit être employé à partir d'une bonne estimation de la solution, comme celle fournie par l'algorithme de décomposition utilisé avec un pas de temps modéré.

Le tableau 5.5 reporte le temps mis pour simuler le régime transitoire au complet avec SuperLU et la méthode proposée. Le régime transitoire est simulé sur 200 pas de temps, afin de capturer précisément la réponse du système. En raison de leur structure différente, les circuits Wyoming, Scc3D et le téléphone iPhone5 ont des régimes transitoires de durées très

différentes. L'expérience montre que le régime transitoire dure environ 60 *sec* pour le circuit Wyoming, 100 *ms* pour le circuit Scc3D et 100 *sec* pour le téléphone iphone5. La simulation du régime transitoire de ces différents systèmes sur 200 pas de temps, se traduit donc par un pas de temps  $\Delta t = 300ms$  pour le circuit Wyoming,  $\Delta t = 0.5ms$  pour le circuit Scc3D et  $\Delta t = 500ms$  pour l'iphone5.

Tableau 5.5 Temps mis pour simuler le régime transitoire (200 pas de temps)

|         |                       | SuperLU | Méthode proposée | Accélération |
|---------|-----------------------|---------|------------------|--------------|
| Wyoming | $n = 0.6 \times 10^5$ | 38 sec  | 3.1 sec          | $\times 12$  |
|         | $n = 1.3 \times 10^5$ | 127 sec | 6.0 sec          | $\times 21$  |
|         | $n = 3.0 \times 10^5$ | 463 sec | 13.1 sec         | $\times 35$  |
| Scc3D   | $n = 2.1 \times 10^5$ | 232 sec | 23 sec           | $\times 10$  |
|         | $n = 4.0 \times 10^5$ | 867 sec | 46 sec           | $\times 18$  |
|         | $n = 6.1 \times 10^5$ | N/A     | 73 sec           | N/A          |
| iPhone5 | $n = 1.4 \times 10^6$ | N/A     | 972 sec          | N/A          |
|         | $n = 4.0 \times 10^6$ | N/A     | 2887 sec         | N/A          |
|         | $n = 7.2 \times 10^6$ | N/A     | 5160 sec         | N/A          |

N/A : Problème de taille trop importante pour y appliquer la méthode SuperLU

Le tableau 5.6 reporte la consommation de mémoire lors de la simulation du régime transitoire. La consommation de mémoire de SuperLU est très importante même pour des problèmes de taille modeste. La méthode proposée permet de réduire significativement la consommation de mémoire durant la simulation.

Tableau 5.6 Consommation de mémoire pour simuler le régime transitoire

|         |                       | SuperLU | Méthode proposée | Réduction    |
|---------|-----------------------|---------|------------------|--------------|
| Wyoming | $n = 0.6 \times 10^5$ | 93 MB   | 13 MB            | $\times 7$   |
|         | $n = 1.3 \times 10^5$ | 2.2 GB  | 27 MB            | $\times 82$  |
|         | $n = 3.0 \times 10^5$ | 6.1 GB  | 62 MB            | $\times 98$  |
| Scc3D   | $n = 2.1 \times 10^5$ | 3.7 GB  | 42 MB            | $\times 88$  |
|         | $n = 4.0 \times 10^5$ | 9.4 GB  | 80 MB            | $\times 117$ |
|         | $n = 6.1 \times 10^5$ | N/A     | 123 MB           | N/A          |
| iPhone5 | $n = 1.4 \times 10^6$ | N/A     | 281 MB           | N/A          |
|         | $n = 4.0 \times 10^6$ | N/A     | 802 MB           | N/A          |
|         | $n = 7.2 \times 10^6$ | N/A     | 1.4 GB           | N/A          |

N/A : Problème de taille trop importante pour y appliquer la méthode SuperLU

La figure 5.7 illustre la comparaison de performance entre SuperLU et la méthode proposée.

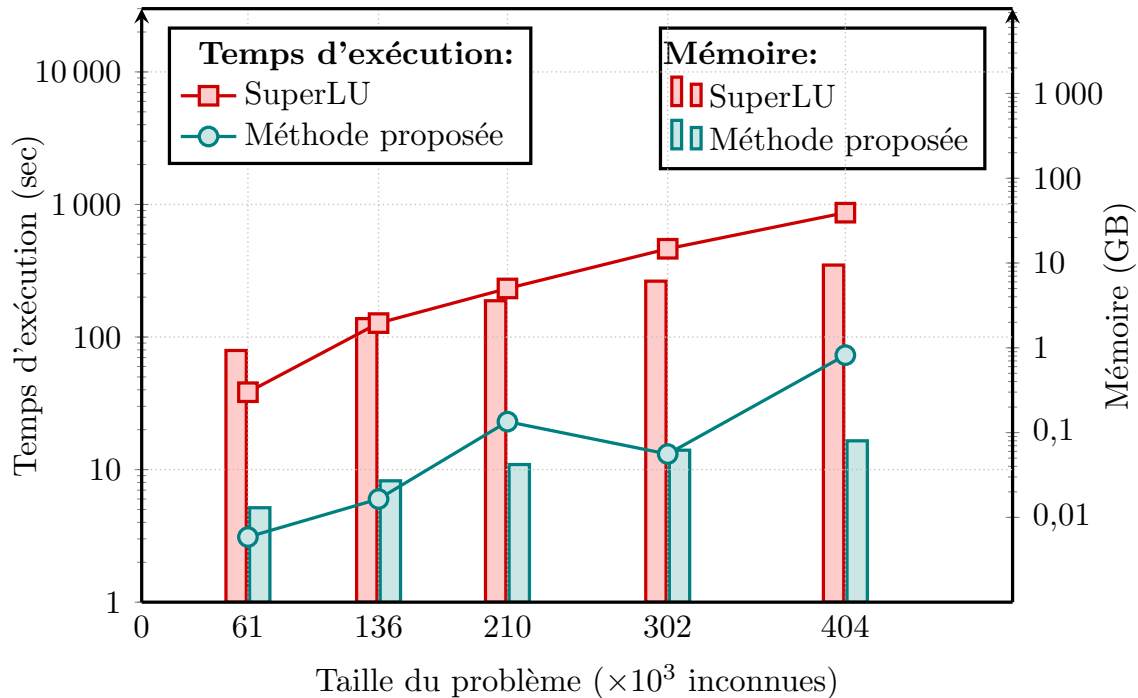


Figure 5.7 Simulation du régime transitoire avec SuperLU et la méthode proposée

### 5.2.3 Bilan

La méthode proposée pour la simulation du régime transitoire est comparée sur des problèmes thermiques réalistes et variés, à la méthode éprouvée SuperLU, utilisée dans les modèles thermiques les plus performants (Sridhar et al., 2010; Liu et al., 2012b). Les expérimentations montrent que la méthode proposée accélère la simulation d'un facteur  $\times 10$  à  $\times 35$ , réduit la consommation de mémoire d'un facteur  $\times 7$  à  $\times 117$  tout en offrant une précision équivalente.

### 5.3 Simulation du régime permanent

La méthode proposée pour simuler le régime permanent est comparée aux méthodes du gradient conjugué (Hestenes and Stiefel, 1952) et GMRES (Saad, 2003) préconditionnées avec une factorisation ILU(0). La méthode du gradient conjugué est la méthode de référence pour résoudre les problèmes symétriques (circuit Wyoming, iPhone 5), tandis que la méthode GMRES est la méthode de référence pour résoudre les problèmes asymétriques (circuit Scc3D). Le choix des méthodes de référence est justifié par leurs très bonnes performances et

leur utilisation dans un grand nombre de modèle thermique du régime permanent (Lee et al., 2013; Goplen and Sapatnekar, 2006; Feng and Li, 2010).

### 5.3.1 Évaluation de la précision

La précision de la méthode proposée est évaluée sur la base de l'erreur maximale ( $^{\circ}C$ ) commise sur le calcul de la température en régime permanent, rapportée à l'élévation de température maximale dans le circuit. Il s'agit donc d'une erreur relative, calculée de la manière suivante :

$$e = \frac{\|X_{\infty} - Y_{\infty}\|_{\infty}}{\|Y_{\infty}\|_{\infty}} \quad (5.2)$$

où  $X_{\infty}$  est la température en régime permanent calculée à l'aide la méthode proposée et  $Y_{\infty}$  est la température en régime permanent calculée avec la méthode de référence. La norme vectorielle  $\|\cdot\|_{\infty}$  est définie par  $\|X\|_{\infty} = \max_{1 \leq i \leq N}(x_i)$ , où  $(x_i)_{1 \leq i \leq N}$  sont les composantes du vecteur  $X$ .

Les différents problèmes thermiques considérés sont simulés à l'aide des méthodes de référence et de la méthode proposée. Le critère de convergence utilisé est un résidu  $\|r\|_1 < 10^{-9}$  qui est la précision maximale pouvant être obtenue pour les problèmes considérés résolus avec les méthodes de référence. En effet, l'expérience montre qu'en raison des erreurs d'arrondi, il n'est pas possible d'obtenir par exemple un résidu  $\|r\|_1 < 10^{-10}$ . Lorsque la méthode GMRES est utilisée, elle est redémarrée toutes les 100 itérations pour réduire sa consommation de mémoire. L'expérience montre que ce paramètre de redémarrage offre un bon compromis entre la vitesse de convergence et la complexité de la méthode.

Les expérimentations montrent que dans tous les cas, l'erreur relative maximale commise par la méthode proposée est inférieure à  $2 \times 10^{-5}$ . La différence entre les résultats obtenus avec la méthode proposée et les méthodes de référence apparaît seulement à partir du 4ème chiffre après la virgule dans le pire cas. Ces résultats prouvent que la méthode proposée est tout aussi précise que les méthodes de référence (gradient conjugué et GMRES).

### 5.3.2 Évaluation des performances

Toutes les expérimentations sont réalisées sur un ordinateur de bureau équipé d'un processeur Intel(R) Core(TM) i7-3770 CPU@3.40GHz avec 16 GB de mémoire principale.

Le tableau 5.7 reporte le temps mis pour simuler le régime permanent à l'aide des méthodes de référence et de la méthode proposée. La méthode proposée permet d'accélérer la simulation d'un facteur  $\times 3$  au minimum. Le temps de calcul rapporté à la taille du problème est beaucoup



Tableau 5.7 Temps mis pour simuler le régime permanent

|          |                       | Gradient conjugué/GMRES | Méthode proposée | Accélération |
|----------|-----------------------|-------------------------|------------------|--------------|
| Wyoming  | $n = 0.6 \times 10^5$ | 287 ms <sup>†</sup>     | 79 ms            | ×3.6         |
|          | $n = 1.3 \times 10^5$ | 900 ms <sup>†</sup>     | 214 ms           | ×4.2         |
|          | $n = 3.0 \times 10^5$ | 3.01 sec <sup>†</sup>   | 751 ms           | ×4.0         |
| Scc3D    | $n = 2.1 \times 10^5$ | 19 sec <sup>‡</sup>     | 3.9 sec          | ×4.8         |
|          | $n = 4.0 \times 10^5$ | 47 sec <sup>‡</sup>     | 9.6 sec          | ×4.9         |
|          | $n = 6.1 \times 10^5$ | 76 sec <sup>‡</sup>     | 14.9 sec         | ×5.1         |
| iPhone 5 | $n = 1.4 \times 10^6$ | 43.3 sec <sup>†</sup>   | 11.7 sec         | ×3.7         |
|          | $n = 4.0 \times 10^6$ | 139 sec <sup>†</sup>    | 36.6 sec         | ×3.8         |
|          | $n = 7.2 \times 10^6$ | 335 sec <sup>†</sup>    | 81.7 sec         | ×4.1         |

<sup>†</sup> Résolu par la méthode du gradient conjugué avec un préconditionnement ILU(0)

<sup>‡</sup> Résolu par la méthode du GMRES avec un préconditionnement ILU(0)

plus long pour le circuit Scc3D. Ce problème asymétrique est résolu à l'aide de la méthode GMRES, qui est moins efficace que la méthode du gradient conjugué.

Le tableau 5.8 indique la consommation de mémoire des méthodes de référence et de la méthode proposée. La consommation de mémoire de la méthode proposée est supérieure d'un

Tableau 5.8 Consommation de mémoire pour simuler le régime permanent

|          |                       | Gradient conjugué/GMRES | Méthode proposée | Réduction |
|----------|-----------------------|-------------------------|------------------|-----------|
| Wyoming  | $n = 0.6 \times 10^5$ | 8.6 MB <sup>†</sup>     | 12 MB            | ×0.7      |
|          | $n = 1.3 \times 10^5$ | 18.5 MB <sup>†</sup>    | 26 MB            | ×0.7      |
|          | $n = 3.0 \times 10^5$ | 42 MB <sup>†</sup>      | 59 MB            | ×0.7      |
| Scc3D    | $n = 2.1 \times 10^5$ | 170 MB <sup>‡</sup>     | 92 MB            | ×1.8      |
|          | $n = 4.0 \times 10^5$ | 354 MB <sup>‡</sup>     | 183 MB           | ×1.9      |
|          | $n = 6.1 \times 10^5$ | 521 MB <sup>‡</sup>     | 266 MB           | ×1.9      |
| iPhone 5 | $n = 1.4 \times 10^6$ | 140 MB <sup>†</sup>     | 196 MB           | ×0.7      |
|          | $n = 4.0 \times 10^6$ | 402 MB <sup>†</sup>     | 521 MB           | ×0.8      |
|          | $n = 7.2 \times 10^6$ | 723 MB <sup>†</sup>     | 955 MB           | ×0.7      |

<sup>†</sup> Résolu par la méthode du gradient conjugué avec un préconditionnement ILU(0)

<sup>‡</sup> Résolu par la méthode du GMRES avec un préconditionnement ILU(0)

facteur ×1.4 à la consommation de mémoire du gradient conjugué et inférieure d'un facteur ×1.9 à la consommation de mémoire de GMRES. Ce comportement s'explique en réalité assez bien. La consommation de mémoire de la méthode GMRES est dominée par le stockage des vecteurs servant à orthonormaliser les espaces de Krylov successifs. La méthode proposée applique la méthode GMRES au seul calcul des variables d'interface dans la décomposition de

domaine. En raison du coloriage rouge-noir utilisé, la méthode GMRES se retrouve appliquée à un problème réduit de moitié, ce qui explique que la consommation de mémoire soit réduite environ de moitié.

La figure 5.8 illustre le temps d'exécution et la consommation de mémoire pour simuler le régime permanent du problème iPhone 5 avec la méthode de référence et la méthode proposée.

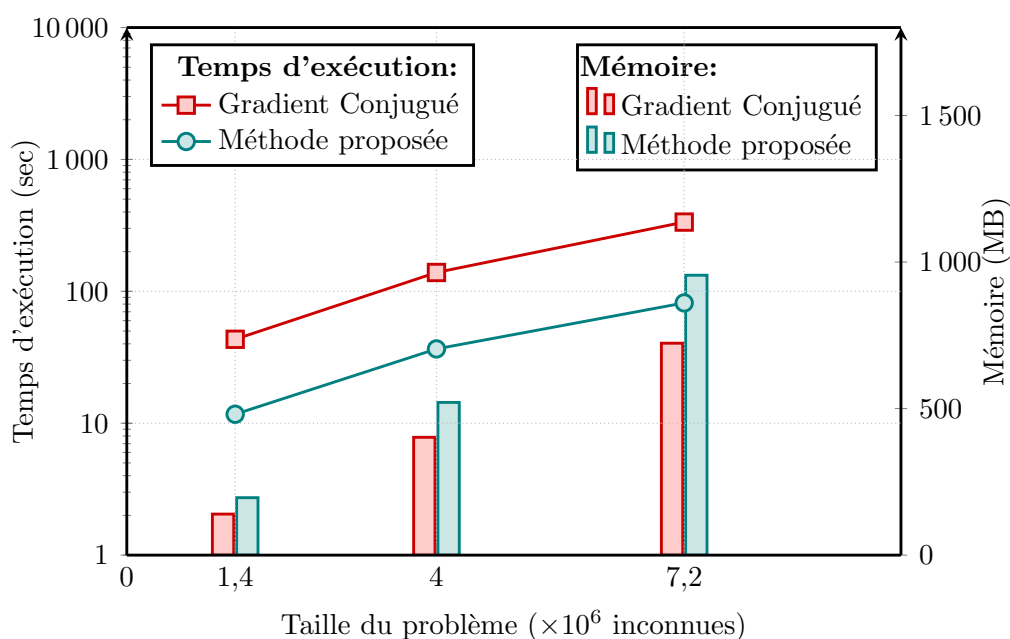


Figure 5.8 Simulation du régime permanent pour le problème iPhone 5 par la méthode du gradient conjugué avec un préconditionnement ILU(0) et la méthode proposée

### 5.3.3 Bilan

La méthode proposée pour la simulation du régime permanent est comparée aux deux méthodes très classiques, du gradient conjugué et GMRES, préconditionnées par une factorisation incomplète ILU(0). La méthode proposée accélère la simulation d'un facteur au moins  $\times 3.6$  dans toutes les expérimentations. Sa consommation de mémoire légèrement supérieure à celle de la méthode du gradient conjugué ne saurait être un inconvénient majeur.

## CHAPITRE 6 VALIDATION PHYSIQUE

L'objectif de ce chapitre est de valider la méthodologie utilisée pour faire la simulation thermique de circuits intégrés, en comparant les résultats obtenus par simulation à des mesures thermiques sur un processeur réel.

### 6.1 Description du système de validation

Le système de validation utilisé pour réaliser les mesures thermiques sur un processeur réel est composé des éléments suivants :

- une carte mère Intel BOXD2700DC, contenant le système étudié, le processeur Intel Atom D2700 ATOM, ainsi que divers composants électroniques (mémoires, périphériques) nécessaires à son fonctionnement
- une cellule de Peltier pour refroidir la carte mère
- une caméra thermique FLIR A40 FLIR pour faire les mesures de température et un ordinateur pour faire l'acquisition des données

La figure 6.1 présente les principaux éléments qui composent le système de validation.

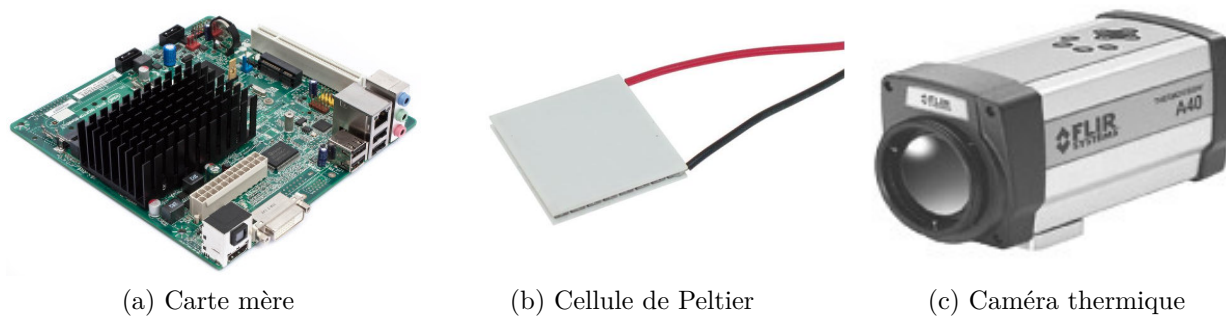


Figure 6.1 Éléments du système de validation

La carte mère Intel BOXD2700DC est un véritable petit ordinateur, contenant tous les composants indispensables pour une utilisation multimédia conviviale : processeur, carte réseau, ports USB, port HDMI, etc. Lors des expérimentations, le système d'opération Ubuntu 12.04, une distribution Linux légère est embarqué sur la carte mère. Le benchmark Impact Parboil Stratton et al. (2012) est utilisé pour générer de l'activité et provoquer une élévation de température au sein des composants électroniques.

La figure 6.2 présente une vue d'ensemble du système de validation, une fois ses divers composants assemblés<sup>†</sup>.

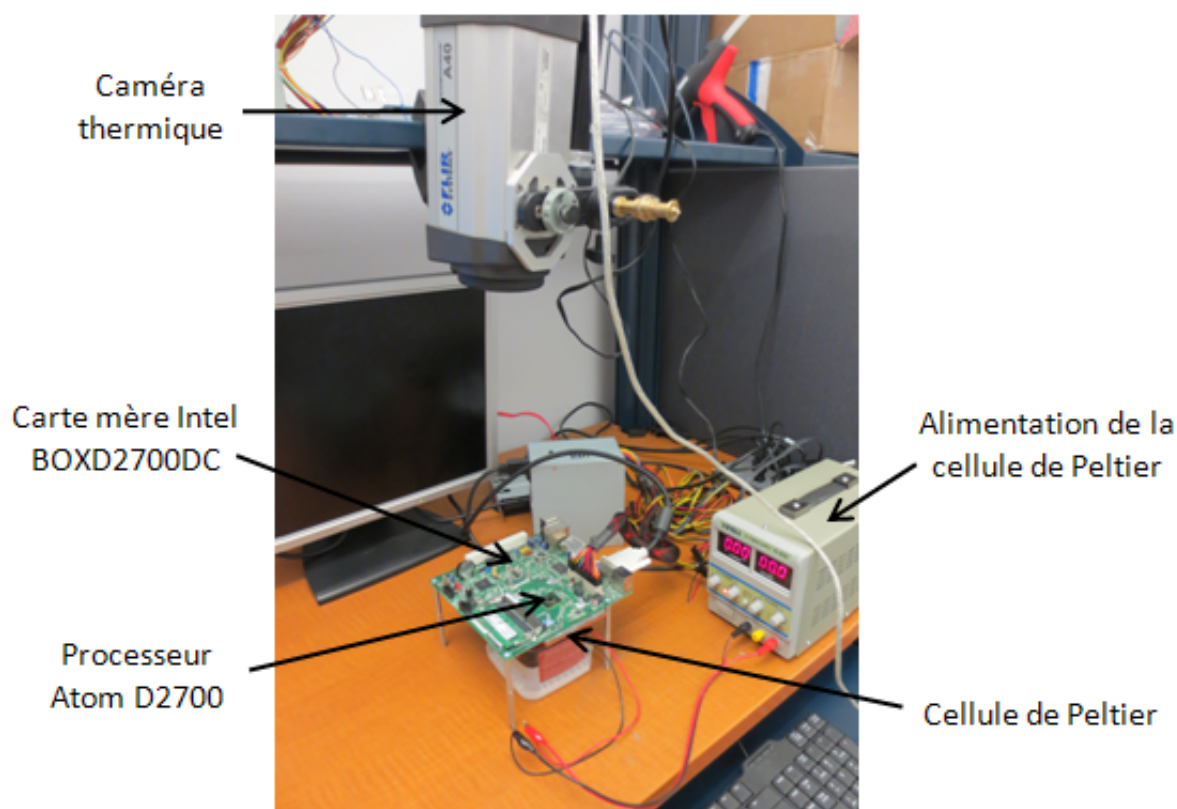


Figure 6.2 Système de validation

La caméra thermique utilisée pour effectuer les mesures de température est une caméra FLIR A40 FLIR. En mesurant l'intensité du rayonnement infrarouge, cette caméra est capable de relever la température à la surface d'un objet avec une précision de  $\pm 2^{\circ}C$ . Le rayonnement infrarouge est un rayonnement électromagnétique indétectable à l'oeil nu. Il s'agit d'une partie du spectre électromagnétique que nous percevons comme chaleur. La caméra utilisée produit des images de  $320 \times 240$  pixels, avec une fréquence maximale de 60 Hz. Un logiciel fourni par la société FLIR est utilisé pour faire l'acquisition des données et traiter les images thermiques. La figure 6.3 présente une capture d'écran de ce logiciel.

Le système étudié, sur lequel sont réalisées les mesures thermiques, est le processeur Intel Atom D2700 ATOM. Il s'agit d'un processeur gravé en 32 nm et composé de deux coeurs cadencés à 2.13 GHz. Commercialisé en 2011, ce processeur d'entrée de gamme est avant

<sup>†</sup> Le système de validation a été assemblé par Sami Riahi, étudiant du laboratoire MIST, sous la direction du professeur Giovanni Beltrame, Ecole Polytechnique Montreal.

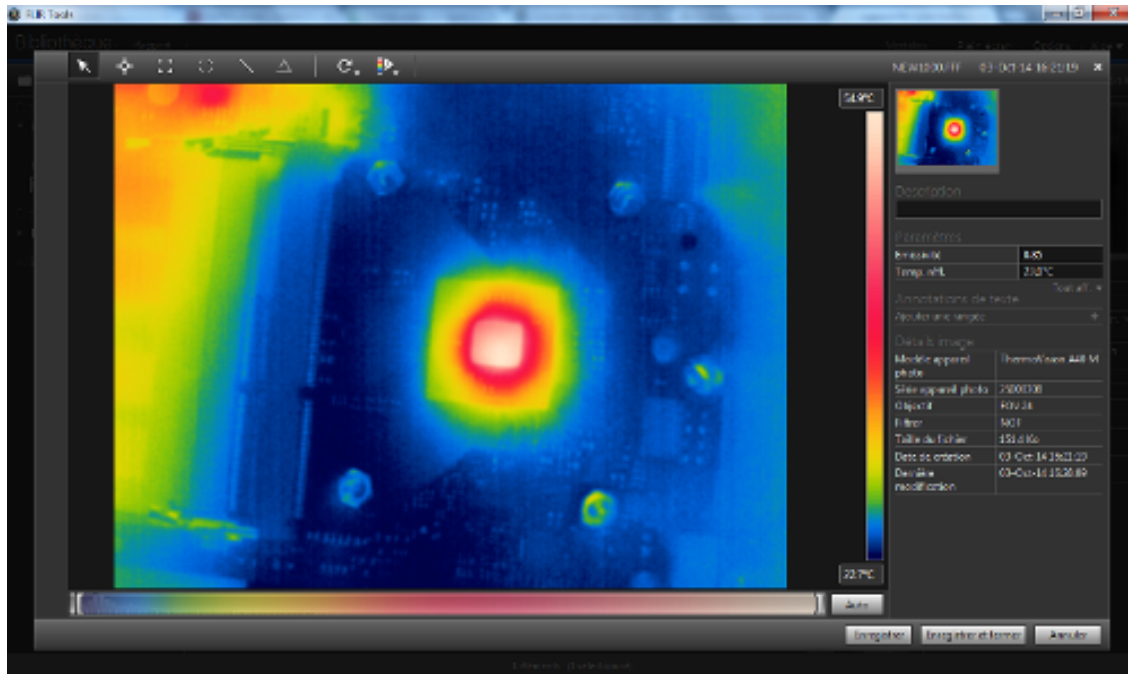


Figure 6.3 Capture d'écran du logiciel Flir Tool

tout destiné aux stations netbook. Les coeurs de l'Atom D2700 ne sont pas très puissants et conviennent seulement aux tâches de calcul légères. La figure 6.4 présente le plan de masse du processeur Atom D2700.

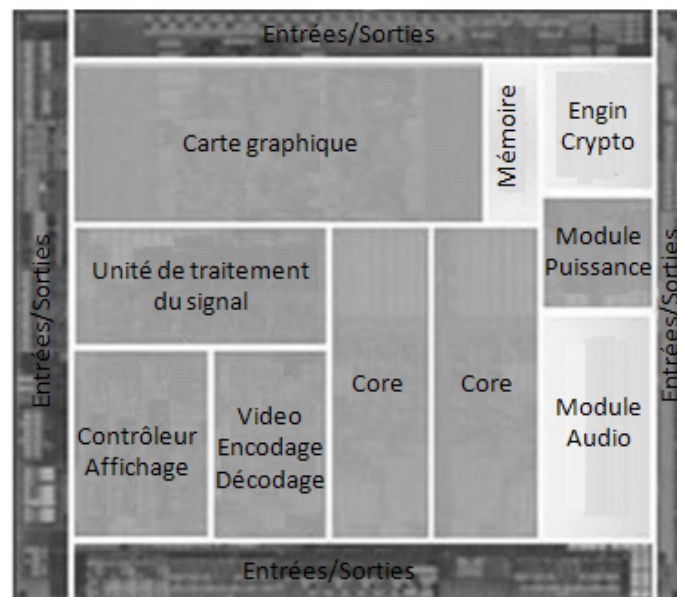


Figure 6.4 Plan de masse de l'atom D2700

Comme beaucoup de systèmes modernes, ce processeur est en réalité un système sur puce, qui comprend en plus des deux coeurs, une carte graphique, un module pour décoder les flux vidéo haute définition, un contrôleur HDMI et un engin cryptographique.

La caméra thermique utilisée peut relever la température de surface, mais elle n'est pas capable de voir la température à travers les couches du circuit. Afin de mesurer le plus précisément possible la température de jonction, c'est-à-dire la température au niveau des transistors et des connexions métalliques, le dissipateur de chaleur présent sur le boîtier d'encapsulation de l'Atom D2700 est retiré. La température mesurée est donc celle de la face supérieure de la puce, qui reste néanmoins assez proche de la température de jonction. Bien que l'Atom D2700 ne soit pas un processeur très puissant (son enveloppe thermique se situe autour de 10 W), un système de refroidissement est tout de même nécessaire. Une cellule de Peltier est placée sous la carte mère afin d'extraire la chaleur du processeur. Peu encombrante et peu onéreuse, la cellule de Peltier permet un contrôle efficace de la température. La structure d'une cellule de Peltier est représentée sur la figure 6.5.

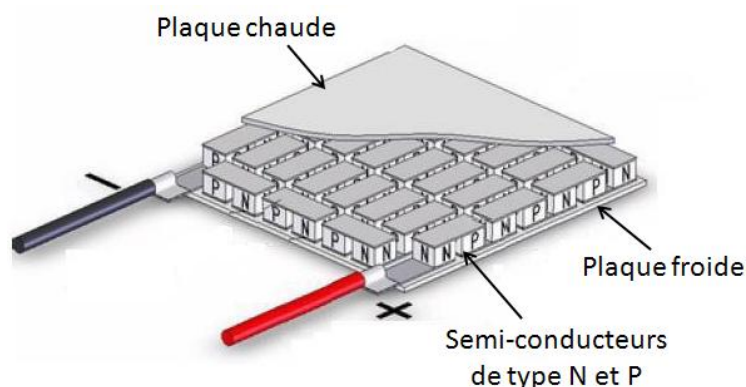


Figure 6.5 Structure d'une cellule de Peltier<sup>†</sup>

La cellule de Peltier est constituée d'éléments semi-conducteurs, placés entre deux couches minces conductrices de la chaleur. En faisant passer un courant électrique dans la cellule, une de ses faces absorbe de la chaleur, tandis que l'autre face la libère. Le phénomène thermo-électrique à la base d'une cellule de Peltier est un transport d'entropie dans deux matériaux conducteurs de nature différente mis en contact. En passant d'un matériau à l'autre, les porteurs de charge, dans un sens, gagnent de l'entropie et absorbent de la chaleur au niveau de la jonction entre les matériaux, et dans l'autre sens, ils perdent de l'entropie qu'ils libèrent sous forme de chaleur. Ce phénomène, baptisé « effet Peltier », a été découvert en 1834 par le physicien français Jean-Charles Peltier. La quantité de chaleur absorbée ou libérée est

<sup>†</sup> Image adaptée de [www.kryothermec.com](http://www.kryothermec.com)

proportionnelle à l'intensité du courant qui traverse la cellule de Peltier. La cellule de Peltier peut être vue comme une pompe à chaleur, capable de prendre l'énergie thermique d'une source froide pour la transférer à une source chaude. Il convient néanmoins de rester vigilant lors des expérimentations, car la différence de température entre les deux faces de la cellule de Peltier peut atteindre plus de  $60^{\circ}\text{C}$ .

## 6.2 Modélisation thermique

Le modèle thermique doit permettre de simuler avec précision le comportement du système étudié, c'est-à-dire le processeur Atom D2700. Prédire la température dans les autres composants de la carte mère n'est pas pertinent en soi. Pour autant, le système étudié échange de la chaleur avec son environnement, c'est-à-dire l'air ambiant et le reste de la carte mère. Afin de limiter la complexité du modèle sans perdre de précision, il convient de choisir convenablement les éléments à représenter dans le modèle et les conditions aux limites pour tenir compte des échanges de chaleur entre le modèle et son environnement.

### 6.2.1 Éléments représentés dans le modèle

Pour construire un modèle thermique précis, toutes les couches de matériaux présentes dans le système étudié (puce, couche active, billes c4, substrat) doivent être représentées. Étant donné le fort couplage thermique dans la direction verticale, il convient de représenter également la carte mère, la pâte thermique et le diffuseur de chaleur, qui se trouvent sous le système étudié. Le tableau 6.1 présente les éléments représentés dans le modèle, leur matériau et leur épaisseur.

Tableau 6.1 Eléments du modèle thermique

| Élément              | Matériau                   | Épaisseur (mm) |
|----------------------|----------------------------|----------------|
| Die                  | Silicium                   | 0.7            |
| Couche active        | Mélange métal-diélectrique | 0.01           |
| Billes C4            | Alliage Étain-Plomb        | 0.1            |
| Substrat             | FR4                        | 1              |
| Billes BGA           | Alliage Étain-Plomb        | 0.3            |
| Carte mère           | FR4                        | 1.5            |
| Pâte thermique       | TIM                        | 2              |
| Diffuseur de chaleur | Cuivre                     | 4              |

Certains éléments du modèle sont constitués de matériaux ayant des propriétés thermiques très différentes. C'est notamment le cas de la couche assurant la connexion du boîtier d'en-

capsulation à la carte mère. Cette couche, constituée d'époxy, un matériau isolant, contient de nombreuses billes métalliques fortement conductrices de la chaleur, dont la distribution est représentée par la figure 6.6,

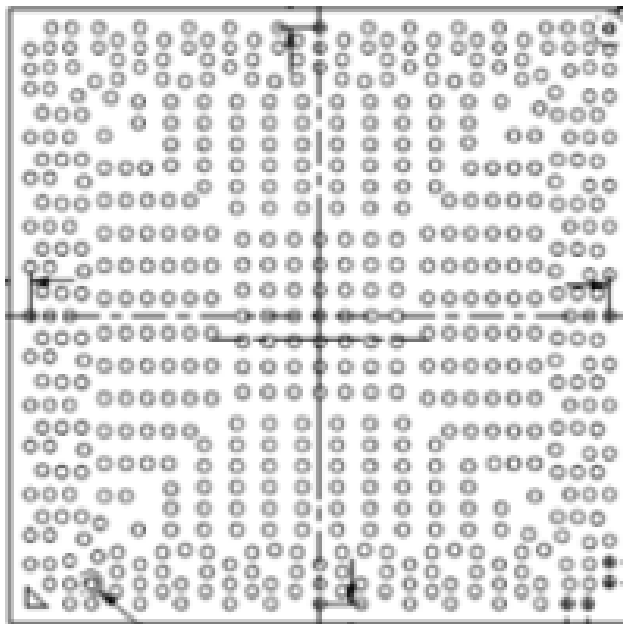


Figure 6.6 Distribution des billes du boîtier d'encapsulation de l'Atom D2700 ATOM

La modélisation individuelle de chaque bille n'est pas nécessaire pour tenir compte de l'hétérogénéité de cette couche. En effet, il est possible de calculer les propriétés thermiques d'un matériau isotrope équivalent à l'aide des formules de mélange suivantes :

$$k_{eq} = \frac{k_1 V_1 + k_2 V_2}{V_1 + V_2} \quad (6.1)$$

$$c_{veq} = \frac{c_{v1} V_1 + c_{v2} V_2}{V_1 + V_2} \quad (6.2)$$

où  $k_i$  représente la conductivité thermique,  $c_{vi}$  la capacité thermique volumique et  $V_i$  le volume du matériau  $i$ .

### 6.2.2 Conditions aux limites

Le modèle thermique ayant une géométrie rectangulaire, six conditions aux limites, correspondant aux six faces du modèle, sont à déterminer. Les images prises par la caméra thermique montrent que la température chute très vite à l'extérieur de la puce pour devenir relativement constante sur le bord du boîtier d'encapsulation. L'expérience justifie donc de limiter le



modèle dans le plan horizontal aux frontières du boîtier d'encapsulation et d'imposer comme condition aux limites sur les faces latérales une température fixe  $T_{lateral}$ .

Au niveau de la face supérieure, la chaleur s'échappe dans l'air ambiant par convection naturelle, car le dissipateur de chaleur a été retiré afin de pouvoir faire les mesures thermiques. La condition aux limites appliquée sur la face supérieure est un flux de chaleur modélisé par la loi de Newton (Equation 2.4). Le coefficient de transfert  $h$  est déduit à partir du nombre de Nusselt :

$$Nu = \frac{hL}{k} \quad (6.3)$$

où  $L$  est la dimension caractéristique de la surface d'échange et  $k$  la conductivité thermique de l'air. Pour une surface rectangulaire, la dimension caractéristique  $L$  est donnée par la relation  $L = \frac{A}{P}$  où  $A$  est l'aire de la surface et  $P$  son périmètre. La complexité de l'écoulement d'air au-dessus du système étudié ne permet pas de calculer analytiquement le nombre de Nusselt, qui est obtenu à l'aide de la formule empirique proposée par Holman (2002) :

$$Nu = C Ra^m \quad (6.4)$$

Les constantes  $C$  et  $m$  dépendent de la géométrie de la surface d'échange et de la nature de l'écoulement d'air au contact de la surface. Le nombre de Rayleigh  $Ra$  est défini comme le produit des nombres de Grashof  $Gr$  et de Prandtl  $Pr$ , qui peuvent être calculés à partir des propriétés de l'air. Le coefficient de transfert  $h$  du modèle thermique est finalement estimé à  $h = 7.38 \text{ W/m}^2 \cdot ^\circ\text{C}$ .

La figure 6.7 présente le modèle thermique proposé pour le système étudié.

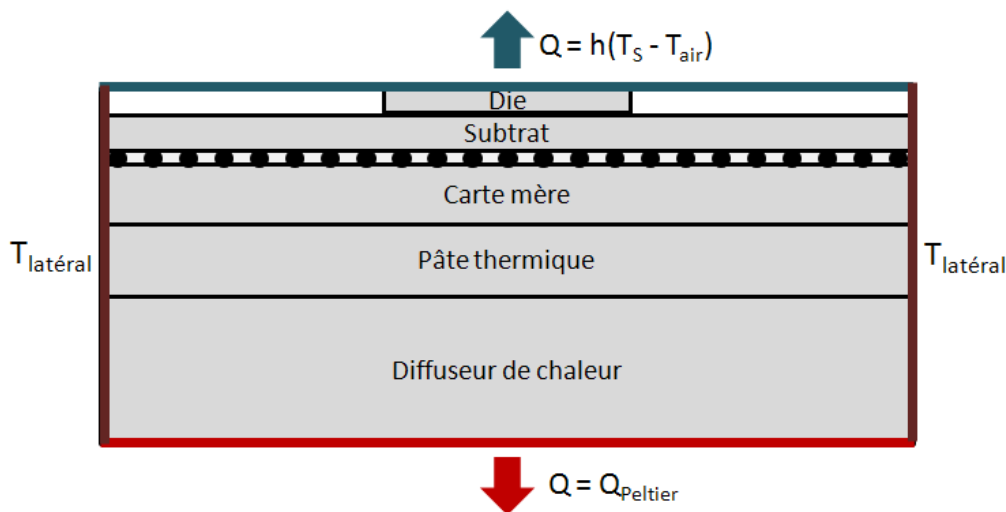


Figure 6.7 Modèle thermique proposé

Au niveau de la face inférieure, la chaleur est absorbée par la cellule de Peltier. La condition aux limites imposée sur la face inférieure est donc un flux constant égal à celui prélevé par la cellule de Peltier.

### 6.2.3 Implémentation du modèle

Pour valider le modèle thermique proposé, indépendamment des méthodes de discrétisation et de résolution employées, deux implémentations du modèle sont réalisées. La première implémentation du modèle est réalisée avec le logiciel ICTherm qui utilise un maillage orthogonal et les méthodes de résolution proposées au chapitre 4. La deuxième implémentation du modèle est réalisée avec le logiciel COMSOL COMSOL qui utilise un maillage tétraédrique et la méthode de résolution multi-grille.

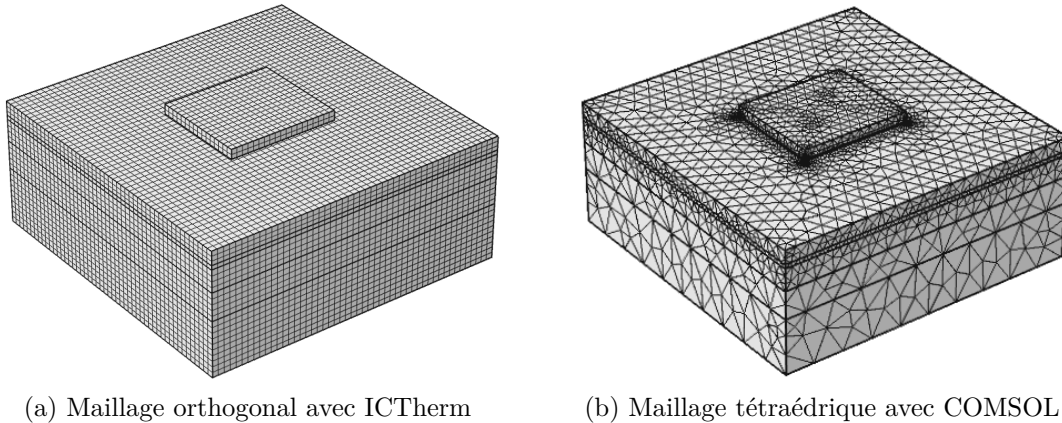


Figure 6.8 Discretisation du système étudié

La figure 6.8 présente les maillages orthogonal et tétraédrique, utilisés respectivement par ICTherm et COMSOL pour discrétiser le système étudié.

## 6.3 Validation du modèle thermique

Le modèle thermique est validé en comparant les résultats obtenus par simulation à l'aide des logiciels ICTherm et COMSOL, aux mesures de température effectuées sur le système réel. La température de surface du processeur Atom est relevée<sup>†</sup> pour deux scénarios de dissipation de puissance, présentés dans le tableau 6.2.

<sup>†</sup> Les mesures de température ont été réalisées par Sami Riahi, étudiant du laboratoire MIST, Ecole Polytechnique Montreal

Tableau 6.2 Scénarios de puissance considérés pour la validation du modèle thermique

|           | Scénario 1 | Scénario 2 |
|-----------|------------|------------|
| Activité  | 90%        | 100%       |
| Puissance | 9 W        | 10 W       |

Ces deux scénarios correspondent respectivement à une activité de 90% et 100% dans le processeur Atom, contrôlé à l'aide du benchmark Parboil.

La figure 6.9 présente les cartes thermiques du régime permanent obtenues pour le scénario 2 par imagerie infrarouge et par simulation.

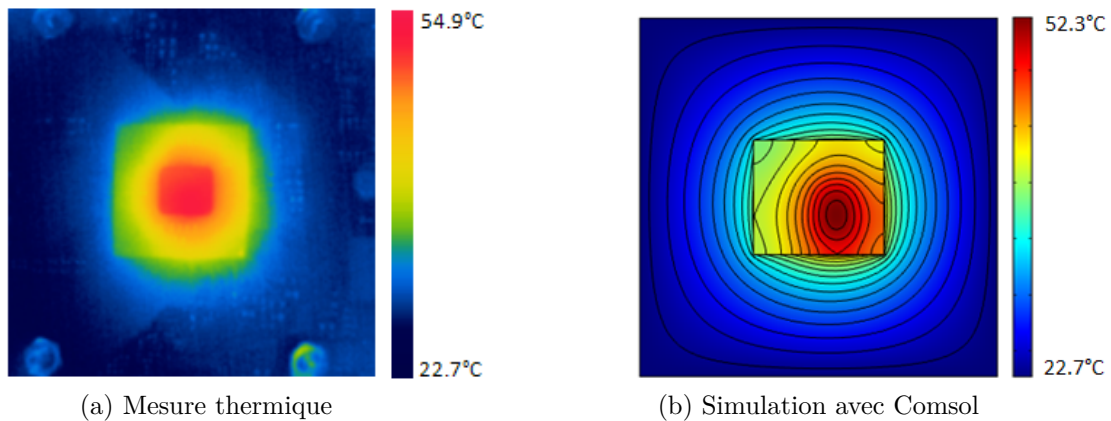


Figure 6.9 Cartes thermiques en régime permanent pour le scénario 2

La figure 6.10 présente la comparaison entre les températures maximales en régime permanent obtenues par imagerie infrarouge et par simulation.

Pour les deux scénarios de puissance considérés, la différence entre la valeur de température maximale mesurée et les valeurs obtenues par simulation est inférieure à 6%. Cette erreur est tout à fait acceptable compte tenu du fait que la simulation thermique est utilisée dans une phase d'avant-projet où les détails de l'architecture finale ne sont pas encore connus. Il faut également rappeler que la caméra thermique utilisée est précise à  $\pm 2^{\circ}\text{C}$  après avoir été calibrée. Les résultats obtenus par simulation sont donc peut être plus précis qu'ils paraissent. La différence entre les simulations à l'aide d'ICTherm et Comsol est inférieure à 1%. Elle peut s'expliquer par la différence dans la méthode de maillage employée par ces deux logiciels. Concernant les performances, la simulation du régime permanent avec ICTherm dure  $1.43 \pm 0.1 \text{ sec}$  contre  $46 \pm 0.2 \text{ sec}$  avec le logiciel Comsol. Il à noter que les deux logiciels n'utilisent pas le même maillage, ni la même méthode de calcul. Néanmoins, ce travail montre que

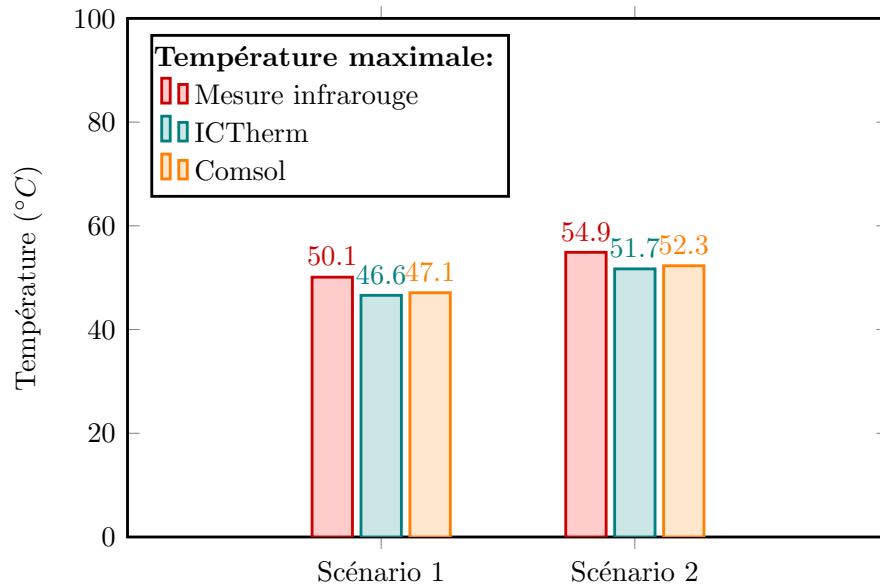


Figure 6.10 Cartes thermiques en régime permanent

pour une précision comparable, le logiciel ICTherm offre un gain de performance significatif par rapport au logiciel Comsol (accélération de la simulation d'un facteur  $\times 32$  dans ces expérimentations).

#### 6.4 Bilan

Ce travail valide la méthodologie utilisée pour faire la simulation thermique de circuits intégrés, en comparant les résultats obtenus par simulation à des mesures thermiques sur un processeur réel. Les expérimentations effectuées montrent que l'utilisation de modèles thermiques simplifiés de circuits intégrés peut fournir une bonne estimation de la température réelle pour autant que les transferts de chaleur prédominants soient identifiés et modélisés correctement. Ce travail compare également entre eux, les deux logiciels de simulation thermique Comsol et ICTherm. Pour une précision comparable, le logiciel ICTherm offre un gain de performance de plus d'un ordre de grandeur par rapport au logiciel Comsol dans les expérimentations réalisées.

## CHAPITRE 7 CONCLUSION

### 7.1 Synthèse des travaux

La simulation thermique est une étape indispensable pour la conception de circuits intégrés modernes. Elle permet de prévenir les problèmes de dissipation thermique et de limiter les risques liés à une température trop élevée. La simulation thermique d'un circuit intégré en cours de conception est notamment effectuée lors des phases de placement et de routage, pour décider de l'architecture du circuit et lors de la phase de test pour valider l'architecture finale. La simulation thermique d'un circuit vise à reproduire son comportement thermique afin de s'assurer que la chaleur produite durant son fonctionnement pourra être dissipée dans l'environnement. La simulation thermique des circuits intégrés modernes nécessite de résoudre des problèmes de transferts de chaleur très complexes. Les circuits modernes ont une structure hétérogène, tridimensionnelle, intégrant des systèmes de refroidissement sophistiqués. Les méthodes analytiques ne permettent pas de résoudre des problèmes de transfert de chaleur aussi complexes. Les méthodes numériques fournissent un outil de simulation puissant et flexible, mais aussi très coûteux en calcul. Il est courant de devoir résoudre des problèmes contenant des centaines de milliers, voire des millions de variables. Les simulateurs thermiques existants basés sur les méthodes numériques requièrent de longs temps de calcul et une consommation de mémoire importante. Cet inconvénient majeur ralentit considérablement la conception des circuits intégrés ou contraints à choisir des solutions thermiques non optimales.

Cette thèse propose deux méthodes de calcul numérique efficaces pour la simulation thermique des circuits intégrés. La première méthode est destinée à la simulation du régime transitoire, tandis que la deuxième méthode cible le régime permanent.

La première méthode proposée simule le régime transitoire en résolvant à chaque pas de temps l'équation d'Euler arrière. Cette équation a l'avantage d'être numériquement stable, c'est-à-dire qu'elle donne des résultats précis et stables quel que soit le pas de temps utilisé, permettant ainsi de simuler des systèmes à dynamique lente comme rapide. Pour simuler rapidement le régime transitoire, la méthode proposée combine une technique de décomposition des opérateurs à un schéma itératif. La technique de décomposition des opérateurs permet de découpler le problème de transfert de chaleur tridimensionnel en trois problèmes unidimensionnels, résolus successivement. Chacun de ces problèmes unidimensionnels est résolu efficacement par l'algorithme de Thomas, qui possède une complexité linéaire. Le découplage des opérateurs introduit cependant une erreur additionnelle, appelée « erreur de décomposition », qui peut donner lieu à des résultats imprécis lorsque de grands pas de temps

sont employés. Un schéma itératif basé sur la résolution d'une équation du point fixe est proposé pour corriger l'erreur de décomposition. Le fort couplage thermique existant dans le circuit sur l'axe vertical est exploité afin de développer un schéma itératif convergeant rapidement. L'erreur de décomposition est ainsi corrigée sans entraîner de pénalité excessive sur les performances.

La deuxième méthode proposée simule le régime permanent en résolvant une équation matricielle linéaire symétrique ou asymétrique (lorsque le circuit contient un fluide de refroidissement). Afin de simuler efficacement le régime permanent, la méthode proposée combine une méthode de projection, un préconditionnement incomplet LU et une décomposition de domaine. L'originalité de la méthode proposée réside dans la combinaison de la méthode de projection préconditionnée à une décomposition de domaine basée sur la matrice de Schur. La décomposition de domaine est effectuée en utilisant un coloriage rouge-noir des variables, qui permet de calculer explicitement la matrice de Schur. Le système réduit associé à la matrice de Schur est ensuite résolu à l'aide de la méthode de projection préconditionnée.

Les méthodes de calcul proposées sont évaluées sur trois problèmes thermiques réalistes et variés : un circuit 3D utilisant des connexions verticales, un circuit 3D multicore refroidi par circulation de liquide et un téléphone intelligent. Les méthodes proposées sont comparées à des méthodes de calcul classiques bien éprouvées : la factorisation LU pour matrices creuses, la méthode du gradient conjugué et la méthode GMRES. Les résultats expérimentaux montrent que les méthodes proposées accélèrent la simulation du régime transitoire d'un facteur  $\times 10$  à  $\times 35$ , et la simulation du régime permanent d'un facteur  $\times 3.6$  à  $\times 5.1$ . La consommation de mémoire est réduite d'un facteur  $\times 7$  à  $\times 110$  pour la simulation du régime transitoire et reste sensiblement équivalente pour la simulation du régime permanent.

Le développement des méthodes de calcul proposées a donné lieu aux publications suivantes :

1. A. Fourmigue, G. Beltrame, G. Nicolescu, "Efficient Transient Thermal Simulation of 3D ICs with Liquid-Cooling and Through Silicon Vias". In *Proc. DATE*, 2014
2. A. Fourmigue, G. Beltrame, G. Nicolescu, "Explicit Transient Thermal Simulation of Liquid-Cooled 3D ICs", In *Proc.*, DATE, 2013.
3. A. Fourmigue, G. Beltrame, G. Nicolescu, E.M. Aboulhamid, "A linear-time approach for the transient thermal simulation of liquid-cooled 3D ICs". In *Proc. CODES+ISSS*, 2011
4. A. Fourmigue, G. Beltrame, G. Nicolescu, E.M. Aboulhamid, I.O'Connor, "Multi-Granularity Thermal Evaluation of 3D MPSoC Architectures". In *Proc. DATE*, 2011.

et à la demande de brevet suivante :

1. A. Fourmigue, G. Beltrame, G. Nicolescu, "Systems and Method for Thermal Modeling of Electronic Devices", US Patent 61931106, 2014.

La méthodologie utilisée pour modéliser le comportement thermique des circuits intégrés est validée en comparant les résultats obtenus par simulation à des mesures thermiques effectuées sur un circuit physique réel. Un système composé d'une carte mère contenant un processeur Intel Atom, d'une cellule de refroidissement de Peltier et d'une caméra thermique permet de relever la température de surface du processeur Atom durant son fonctionnement par imagerie infrarouge. Un modèle thermique du système est réalisé à l'aide du logiciel commercial Comsol et du logiciel ICTherm développé dans le cadre de cette thèse. Les résultats obtenus par simulation sont conformes aux mesures effectuées sur le processeur réel, avec une erreur maximale inférieure à 6% de la température maximale du processeur.

## 7.2 Limitations des méthodes proposées

Les méthodes proposées pour la simulation thermique des circuits intégrés ont deux limitations principales. Premièrement la structure du circuit et le pas de temps ne doivent pas changer au cours de la simulation (transitoire). Deuxièmement, le maillage utilisé pour construire le modèle numérique doit être orthogonal.

La première limitation concerne seulement la simulation du régime transitoire. La structure du circuit et le pas de temps doivent être invariables au cours de la simulation. Tout changement de la structure du circuit affecte les matrices de conductance et de capacitance. De même, tout changement du pas de temps modifie l'équation d'Euler arrière et implique de calculer de nouveau les factorisations des systèmes tridiagonaux. Cette limitation n'est pas une contrainte majeure. La structure physique du circuit n'est pas censée être modifiée durant son fonctionnement et les simulations à pas de temps variable ne présentent pas vraiment d'intérêt pour analyser le comportement thermique du circuit.

La deuxième limitation porte sur la géométrie du maillage utilisé pour discrétiser le problème thermique et concerne donc à la fois le régime transitoire et le régime permanent. Les méthodes proposées sont applicables uniquement à des maillages orthogonaux. Étant donné la géométrie rectangulaire des circuits intégrés, cette limitation n'est généralement pas un inconvénient majeur. L'utilisation d'un maillage orthogonal peut toutefois conduire à des problèmes contenant plus de variables que nécessaire. Si un composant particulier du circuit nécessite d'être discrétisé avec une résolution élevée, le raffinement local du maillage se propage dans les directions  $x, y, z$  du fait de l'alignement des cellules. À précision égale, un maillage orthogonal conduit généralement à un problème plus volumineux qu'un maillage non structuré composé de cellules tétraédriques ou pyramidales. Toutefois, pour la simulation thermique des circuits intégrés, le sur-dimensionnement des problèmes dû à l'utilisation d'un maillage orthogonal est largement compensé par l'efficacité des méthodes de calcul proposées.

### 7.3 Travaux futurs

Plusieurs travaux futurs sont possibles afin d'apporter une solution aux limitations évoquées et développer des simulateurs thermiques plus performants.

Un premier travail possible porte sur l'extension des méthodes proposées à des maillages non structurés. L'intérêt d'un tel travail est d'exploiter la plus grande précision des maillages non structurés (à nombre égal de cellules) afin de résoudre des problèmes matriciels plus petits. Les méthodes itératives, par relaxation et par projection, sont directement applicables aux maillages non structurés. La décomposition de domaine peut également être effectuée sur un maillage non structuré, même si le coloriage rouge-noir proposé n'est plus possible. La technique de décomposition des opérateurs est applicable en théorie seulement aux maillages orthogonaux, afin d'obtenir des systèmes découplés qui soient tridiagonaux. La matrice de conductance peut cependant encore être décomposée sur un maillage non structuré. Les équations découplées ne sont plus tridiagonales, ce qui ne signifie pas pour autant qu'elles ne peuvent plus être résolues de façon efficace. La meilleure manière de découpler ces équations et de les résoudre est à déterminer.

Un deuxième travail possible consiste à appliquer les méthodes de calcul proposées pour générer des modèles d'ordre réduit. La réduction d'ordre est une technique couramment utilisée en mécanique des structures ou en génie électrique afin de réduire la taille d'un modèle tout en préservant le comportement de certaines variables d'intérêt. Le modèle réduit comporte généralement un nombre de variables très inférieur au modèle original et la simulation est souvent accélérée de plusieurs ordres de grandeur. Une erreur répandue consiste à croire que la réduction d'ordre est un moyen peu onéreux de résoudre des modèles de grande taille. La réduction d'ordre est en réalité un processus très coûteux en calcul, impliquant de multiples résolutions des mêmes équations de thermiques que celles considérées dans cette thèse. Un grand nombre de techniques de réduction d'ordre existent, donnant chacune un compromis différent entre précision et performance. Ces différentes techniques de réduction d'ordre doivent être explorées, ainsi que la façon d'appliquer les méthodes proposées pour générer efficacement des modèles d'ordre réduit.



## RÉFÉRENCES

“ANSYS”, [www.ansys.com](http://www.ansys.com), Juil. 2013.

“Intel Atom D2700”, [www-ssl.intel.com/content/www/us/en/processors/atom/atom-d2000-n2000-vol-1-datasheet.html](http://www-ssl.intel.com/content/www/us/en/processors/atom/atom-d2000-n2000-vol-1-datasheet.html), Oct. 2014.

C. Auth, A. Blattner, D. Bergstrom, et M. Brazier, “A 22nm high performance and low-power CMOS technology featuring fully-depleted tri-gate transistors, self-aligned contacts and high density MIM capacitors”, dans *VLSI Technology (VLSIT), 2012 Symposium on*, June 2012, pp. 131–132.

S. C. Chapra et R. P. Canale, *Numerical Methods for Engineers*, 6e éd. New York, NY, USA : McGraw-Hill, Inc., 2009.

Y.-K. Cheng, C.-H. Tsai, C.-C. Teng, et S.-M. Kang, “Thermal Simulation for VLSI Systems”, dans *Electrothermal Analysis of VLSI Systems*. Boston : Kluwer Academic Publishers, 2000.

“COMSOL”, [www.comsol.com](http://www.comsol.com), Juil. 2013.

“Core i5”, <http://www.intel.com/design/corei5/documentation.htm>, Sep. 2014.

T. A. Davis et E. Palamadai Natarajan, “Algorithm 907 : KLU, A Direct Sparse Solver for Circuit Simulation Problems”, *ACM Trans. Math. Softw.*, vol. 37, no. 3, pp. 36 :1–36 :17, Sep. 2010.

J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, et J. W. H. Liu, “A supernodal approach to sparse partial pivoting”, *SIAM Journal on Matrix Analysis and Applications*, vol. 20, pp. 720–755, 1999.

J. Douglas et J. Gunn, “A General Formulation of Alternating Direction Methods. Part I. Parabolic and Hyperbolic Problems.” *Numerische Mathematik*, vol. 6, pp. 428–453, 1964.

D. Dutoit, C. Bernard, S. Cheramy, F. Clermidy, Y. Thonnart, P. Vivet, C. Freund, V. Guerin, S. Guilhot, S. Lecomte, G. Qualizza, J. Pruvost, Y. Dodo, N. Hotelier, et J. Michailos, “A 0.9 pj/bit, 12.8 gbyte/s wideio memory interface in a 3d-ic noc-based mp soc”, dans *VLSI Technology (VLSIT), 2013 Symposium on*, June 2013, pp. C22–C23.

Z. Feng et P. Li, “Fast thermal analysis on GPU for 3D-ICs with integrated microchannel cooling”, dans *Proc. Int. Conf. Computer-Aided. Design (ICCAD)*, Nov. 2010, pp. 551–555.

“Caméra Flir, Serie A”, [www.flir.com](http://www.flir.com), Oct. 2014.

“FloTHERM”, [www.mentor.com/products/mechanical/products/flotherm](http://www.mentor.com/products/mechanical/products/flotherm), Juil. 2013.

B. Goplen et S. Sapatnekar, “Placement of thermal vias in 3D ICs using various thermal objectives”, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 4, pp. 692–709, apr. 2006.

H. Y. Hanhua Qian, Chip-Hong Chang, “An efficient channel clustering and flow rate allocation algorithm for non-uniform microfluidic cooling of 3D integrated circuits”, *Integration, the VLSI Journal*, vol. 46, no. 1, pp. 57–68, 2013.

M. R. Hestenes et E. Stiefel, *Methods of conjugate gradients for solving linear systems*. National Bureau of Standards Washington, DC, 1952, vol. 49.

J. P. Holman, *Heat Transfer*, 9e éd. New York : Mc Graw Hill, 2002.

J. Howard et al., “A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling”, *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 1, pp. 173–183, jan. 2011.

W. Huang, S. Ghosh, S. Velusamy, et K. Sankaranarayanan, “HotSpot : A Compact Thermal Modeling Methodology for Early-Stage VLSI Design”, *IEEE Trans. VLSI Syst.*, vol. 14, no. 5, pp. 501–513, Mai 2006.

Y. J. Kim, Y. K. Joshi, A. G. Fedorov, Y.-J. Lee, et S. K. Lim, “Thermal Characterization of Interlayer Microfluidic Cooling of Three-Dimensional IC With Non-Uniform Heat Flux”, *ASME Conference Proceedings*, vol. 2009, no. 43499, pp. 1249–1258, 2009.

J.-M. Koo, S. Im, L. Jiang, et K. E. Goodson, “Integrated Microchannel Cooling for Three-Dimensional Electronic Circuit Architectures”, *Journal of Heat Transfer*, vol. 127, no. 1, pp. 49–58, 2005.

P. Lascaux et R. Théodor, *Analyse numérique matricielle appliquée à l’art de l’ingénieur*, série Analyse numérique matricielle appliquée à l’art de l’ingénieur. Masson, 1987, no. vol.2.

J. Lau, “Evolution, challenge, and outlook of TSV, 3D IC integration and 3D silicon integration”, dans *Advanced Packaging Materials (APM), 2011 International Symposium on*, 2011, pp. 462–488.

Y.-M. Lee, T.-H. Wu, P.-Y. Huang, et C.-P. Yang, “NUMANA : A hybrid numerical and analytical thermal simulator for 3D ICs”, dans *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, 2013, pp. 1379–1384.

P. Li, L. Pileggi, M. Asheghi, et R. Chandra, “IC thermal simulation and modeling via efficient multigrid-based approaches”, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 9, pp. 1763–1776, Sep. 2006.

S. Liu, X. Xu, R. Xie, G. Zhang, et B. Li, “Anomalous Heat Conduction and Anomalous Diffusion in Low Dimensional Nanoscale Systems”, *The European Physical Journal B*, vol. 85, no. 10, pp. 1–20, 2012.

X.-X. Liu, Z. Liu, S.-D. Tan, et J. Gordon, “Full-chip thermal analysis of 3D ICs with liquid cooling by GPU-accelerated GMRES method”, dans *Quality Electronic Design (ISQED), 2012 13th International Symposium on*, march 2012, pp. 123–128.

G. H. Loh, Y. Xie, et B. Black, “Processor Design in 3D Die-Stacking Technologies”, *IEEE Micro*, vol. 27, no. 3, pp. 31–48, may 2007.

Z. Luo, H. Cho, X. Luo, et K. il Cho, “System thermal analysis for mobile phone”, *Applied Thermal Engineering*, vol. 28, no. 14–15, pp. 1889 – 1895, 2008.

J. Meng, K. Kawakami, et A. K. Coskun, “Optimizing Energy Efficiency of 3D Multicore Systems with Stacked DRAM Under Power and Thermal Constraints”, dans *Proceedings of the 49th Annual Design Automation Conference*, série DAC '12, 2012, pp. 648–655.

H. Mizunuma, Y.-C. Lu, et C.-L. Yang, “Thermal Modeling and Analysis for 3D ICs With Integrated Microchannel Cooling”, *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 30, no. 9, pp. 1293 –1306, sept. 2011.

E. Monier-Vinard, C. Dia, V. Bissuel, O. Daniel, et N. Laraqi, “Extension of the DELPHI methodology to Dynamic Compact Thermal Model of electronic component”, dans *Thermal Investigations of ICs and Systems (THERMINIC), 2011 17th International Workshop on*, Sept 2011, pp. 1–6.

D. Oh, C.-P. Chen, et Y.-H. Hu, “Efficient Thermal Simulation for 3D IC With Thermal

Through-Silicon Vias”, *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 31, no. 11, pp. 1767–1771, 2012.

R. Patti, “Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs”, *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1214–1224, jun. 2006.

V. F. Pavlidis et E. G. Friedman, “Thermal Management Techniques”, dans *Three-Dimensional Integrated Circuit Design*, V. F. Pavlidis et E. G. Friedman, édés. Morgan Kaufmann, 2009, pp. 99–134.

H. Rosten et C. Lasance, “DELPHI : The Development of Libraries of Physical Models of Electronic Components for an Integrated Design Environment”, dans *Model Generation in Electronic Design*, série Current Issues in Electronic Modeling, J.-M. Bergé, O. Levia, et J. Rouillard, édés. Springer US, 1995, vol. 1, pp. 63–89.

Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2e éd. Philadelphia, PA, USA : Society for Industrial and Applied Mathematics, 2003.

C. Santos, P. Vivet, D. Dutoit, P. Garrault, N. Peltier, et R. Reis, “System-level thermal modeling for 3d circuits : Characterization with a 65nm memory-on-logic circuit”, dans *3D Systems Integration Conference (3DIC), 2013 IEEE International*, Oct 2013, pp. 1–6.

S. Shidore, “Compact thermal modeling in electronics design”, <http://www.electronics-cooling.com/2007/05/compact-thermal-modeling-in-electronics-design>, Mai 2007.

C. B. Sobhan et G. P. Peterson, *Microscale and Nanoscale Heat Transfer : Fundamentals and Engineering Applications*. Boca Raton, Fla. : CRC Press, 2008.

“SOLIDWORKS”, [www.solidworks.com](http://www.solidworks.com), Juil. 2013.

A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunswiler, et D. Atienza, “3D-ICE : Fast Compact Transient Thermal Modeling for 3D ICs with Inter-Tier Liquid Cooling”, dans *Proc. Int. Conf. Computer-Aided. Design (ICCAD)*, San Jose, USA, Nov. 2010, pp. 463–470.

J. A. Stratton, C. Rodrigues, I. J. Sung, N. Obeid, L. W. Chang, N. Anssari, G. D. Liu, et W. W. Hwu, “Parboil : A revised benchmark suite for scientific and commercial throughput computing”, *Center for Reliable and High-Performance Computing*, 2012.

V. Székely, A. Poppe, M. Rencz, M. Rosental, et T. Teszéri, “THERMAN : a thermal simulation tool for IC chips, microstructures and PW boards”, *Microelectronics Reliability*,

vol. 40, no. 3, pp. 517 – 524, 2000.

A. Vassighi et M. Sachdev, “Power, Junction Temperature, and Reliability”, dans *Thermal and Power Management of Integrated Circuits*, A. Chandrakasan, éd. New-York : Springer US, 2006, pp. 13–49.

A. Vincenzi, A. Sridhar, M. Ruggiero, et D. Atienza, “Accelerating thermal simulations of 3D ICs with liquid cooling using neural networks”, dans *Proc. of the Great Lakes Symposium on VLSI*, série GLSVLSI’12. ACM, 2012, pp. 15–20.

T.-Y. Wang et C. C.-P. Chen, “3D Thermal-ADI : a linear-time chip level transient thermal simulator”, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 21, no. 12, pp. 1434–1445, dec. 2002.

N. Weste et D. Harris, *CMOS VLSI Design : A Circuits and Systems Perspective*, 4e éd. New York : Addison Wesley, 2010.

Y. Zhan et S. Sapatnekar, “High-Efficiency Green Function-Based Thermal Simulation Algorithms”, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 9, pp. 1661–1675, Sep. 2007.

S. Zimmermann, I. Meijer, M. K. Tiwari, S. Paredes, B. Michel, et D. Poulikakos, “Aquasar : A hot water cooled data center with direct energy reuse”, *Energy*, vol. 43, no. 1, pp. 237 – 245, 2012.