

UNIVERSITÉ DU QUÉBEC

THÈSE PRÉSENTÉE
À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DU DOCTORAT EN GÉNIE ÉLECTRIQUE

PAR
BLÉ VÉLOU STÉPHANE

MODÉLISATION PARALLÈLE DES PHÉNOMÈNES NANOTHERMIQUES

OCTOBRE 2009

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

Résumé

L'étude des nanosystèmes a fait l'objet de plusieurs recherches cette dernière décennie. Cependant les travaux reliés au transfert de chaleur de ces nouveaux composants sont toujours au stade de développement.

Le but que nous visons à travers notre recherche, est le développement d'un outil de simulation et d'analyse électrothermique à l'échelle nanométrique afin de minimiser les pertes de puissance et d'optimiser le transfert de chaleur dans les nanosystèmes. Cela vise deux objectifs principaux à savoir une meilleure gestion du transfert de chaleur pour la fiabilité et la durabilité des composants et le deuxième, est de permettre lors de la génération de chaleur dans certaines applications biomédicales (traitement de cellules cancérogènes) une bonne estimation du champ de température.

Cet outil pour être efficace, devrait être capable de traiter une grande quantité de données en un temps acceptable. Pour répondre à cette problématique, nous avons décidé d'utiliser des algorithmes parallèles avec le standard MPI, puis implémenter le tout sur des calculateurs de haute performance (HPC) destinés plus particulièrement au traitement parallèle. Les systèmes de calcul que nous avons identifiés pour notre étude sont entre autres, les réseaux de calcul WestGrid, CLUMEQ et ACNET, faisant partie du Calcul Canada (www.c3.ca).

Summary

The study of nanosystems has been the subject of several searches in the last decade. But the works related to the heat transfer of these new components is still at the stage of development. The goal we seek through our research is to develop a tool to simulate and analyze electrothermal phenomena of nanoscale devices in order to minimize power loss and optimize heat transfer. This have two major objectives, first, a better management of heat transfer for the reliability and durability of components and the secondly, allow a good estimate of fields temperature when generating heat in some biomedical applications (treatment of cancer cell).

This tool to be effective should be able to handle a large amount of data in a reasonable time. To address this problem, we decided to use parallel algorithms with the MPI standard on high performance computing (HPC), designed specifically to parallel processing. The computing platforms that we have identified in our study are, WESTGRID, CLUMEQ and ACNET, all part of Canada Computation (www.c3.ca).

Avant-propos

Avant toute chose j'aimerais remercier le seigneur tout puissant de m'avoir permis de mener ce projet de recherche à terme.

J'aimerais aussi dire un merci à mon directeur de recherche le professeur Adam Skorek, pour m'avoir intégré dans son équipe de recherche au sein du Laboratoire de nanothermie de l'Université du Québec à Trois-Rivières. J'aimerais surtout lui dire merci pour son soutien inconditionnel sans relâche et sans faille, pour sa patience et ses encouragements. Je finirai tout simplement en disant merci à toute ma famille, mes ami(e)s ainsi que mes collègues pour leur soutien, leurs conseils et leurs encouragements. J'aimerais aussi dire merci à tous ceux ou toutes celles qui de près ou de loin ont contribué à la bonne marche de ce projet de thèse.

*« Tout ce que vous demanderez en priant,
croyez que vous l'avez reçu, et vous le verrez s'accomplir »*

Table des matières

Résumé.....	ii
Summary	iii
Avant-propos.....	iv
Table des matières.....	v
Liste des figures	viii
Chapitre 1 - Introduction.....	11
1.1 Problématique.....	12
1.2 Objectifs	14
1.3 Méthodologie de recherche	14
1.4 Infrastructure	21
Chapitre 2 - La nanotechnologie et les nanosciences.....	25
2.1 Les NEMS et les NOEMS.....	25
2.2 Les nanotubes de carbone.....	27
2.3 Les nanotransistors	30
2.4 Modélisation.....	32
2.5 Théories et lois de la physique quantique.....	33

2.5.1	Les Spins.....	33
2.5.2	Modèle d'Ising.....	36
2.5.3	L'équilibre thermique.....	39
Chapitre 3 - Développement de l'algorithme.....		41
3.1	Algorithme de Monte Carlo.....	43
3.2	Chaîne de Markov.....	44
3.3	Énumération.....	45
3.4	Parallélisation de l'algorithme.....	47
3.4.1	L'OpenMp.....	50
3.4.2	Le Standard MPI.....	52
3.4.3	Techniques de création d'algorithme parallèle.....	53
Chapitre 4 - Implémentation de l'algorithme.....		65
4.1	L'environnement de CLUMEQ.....	65
4.2	Décomposition de domaine.....	78
4.3	Étude paramétrique.....	79
4.3.1	Énergie du système.....	80
4.3.2	Chaleur spécifique.....	80
4.3.3	Conductivité thermique.....	81
Chapitre 5 - Validation.....		83

5.1	Introduction	83
5.2	Validation de l'algorithme.....	83
5.3	Validation des résultats de simulation avec la littérature	87
5.4	Validation des résultats de simulation comparaison avec nanoHub	91
5.4.1	Système 1	92
5.4.2	Système 2	93
5.4.3	Système 3	95
5.5	Évaluation des résultats	96
5.5.1	Résultat de la simulation du MOFET	97
5.5.2	Résultat de la simulation du système 2	99
5.5.3	Résultat de la simulation du système 3	101
5.6	Discussion.....	103
	Chapitre 6 - Conclusion	106
	Bibliographie.....	110

Liste des figures

Figure 1-1 Diagramme de la méthodologie de recherche	15
Figure 1-2 Quelques formes et matériaux possible de nanosystèmes [3].....	16
Figure 1-3 Modèle simple 2D de silicium [RP]	17
Figure 1-4 Spins voisins appartenant à des processeurs différents [RP]	20
Figure 1-5 Réseau global du WestGrid [22]	22
Figure 1-6 Réseau global du CLUMEQ [19].....	24
Figure 2-1 Réseau carré à 2D grille d'Ising [RP]	37
Figure 2-2 Réseau carré à 2D grille d'Ising [RP]	37
Figure 2-3 Alliage A, B sur réseau carré.....	38
Figure 3-1 Illustration de la hiérarchie du modèle de transport [10]	43
Figure 3-2 Infrastructure d'implémentation [100].....	48
Figure 3-3 Mémoire partagée [100].....	48
Figure 3-4 Mémoire distribuée [100].....	49
Figure 3-5 Processus d'exécution OpenMP.....	51
Figure 3-6 Décomposition par block [RP].....	55
Figure 3-7 Décomposition en bande [RP].....	56
Figure 3-8 Diagramme de communication pour une décomposition par bande [RP].....	57
Figure 3-9 Diagramme de communication pour une décomposition par block [RP].....	58

Figure 3-10 Interactions entre processeurs voisins [RP].....	59
Figure 3-11 : Répartition des données de calcul pour une décomposition en bande [RP].....	61
Figure 3-12 Transfert des mises à jour phase 1 [RP].....	62
Figure 3-13 Transfert des mises à jour phase 2 [RP].....	63
Figure 4-1 Démarrage de la connexion à CLUMEQ en utilisant le SSH [RP].....	66
Figure 4-2 Saisie des paramètres de connexion [RP].....	67
Figure 4-3 Saisie du mot de passe [RP]	68
Figure 4-4 Accès au nœud d'entrée (Stokes) [RP].....	69
Figure 4-5: Passage du nœud d'entré Stokes au nœud de soumission hn [RP]	70
Figure 4-6 Connexion effective au nœud hn [RP]	71
Figure 4-7 Répertoire utilisateur sur Stokes [RP].....	73
Figure 4-8 Exemple de compilation sur le nœud "hn" [RP].	74
Figure 4-9 Fichier de soumission pour l'exécution [RP].....	75
Figure 4-10 Soumission du fichier nano_1 pour l'exécution de l'application [RP].....	76
Figure 4-11 Visualisation de l'état des applications par leur numéro d'identification "Job id" [RP].	77
Figure 4-12 Exemple de distribution de particules dans un transistor [10]	78
Figure 4-13 Réseau 2D, géométrie simple [RP]	79
Figure 4-14 Décomposition 2D, d'un transistor à géométrie conventionnelle [16]	79
Figure 5-1 Évolution de la simulation vers l'équilibre thermique [RP].....	84
Figure 5-2: Évolution de la simulation vers l'équilibre thermique [RP]	86
Figure 5-3 Études paramétriques de la conductivité thermique de nanofils de silicium [RP].....	87
Figure 5-4 La conductivité thermique de nanofils de silicium [13].....	89

Figure 5-5: La conductivité thermique d'une mince couche de silicium [RP]	90
Figure 5-6 La conductivité thermique d'une mince couche de silicium [98]	91
Figure 5-7 Figure 43: MOSFET [16]	92
Figure 5-8 Dual-Gate FET [16]	94
Figure 5-9 MOSEFET SOI [16].....	95
Figure 5-10 Système 1, résultat avec Quamc2D [16]	97
Figure 5-11 Système 1 simulation avec BVSim_nano [RP].....	98
Figure 5-12 Système 2 résultat avec Quamc2D [16]	99
Figure 5-13 Système 2, simulation avec BVSim_nano [RP].....	100
Figure 5-14 Système 3 résultat avec Quamc2D [16]	101
Figure 5-15 Système 3, simulation avec BVSim_nano [RP].....	102

Chapitre 1 - Introduction

L'introduction présente l'idée directrice de la recherche. L'auteur y présente l'état de la Le 29 décembre 1959, le grand physicien américain Richard Feynman prononçait un discours resté fameux, intitulé « Il y a beaucoup d'espace en bas ». Devant un auditoire sidéré, Feynman, en génial visionnaire, imaginait et développait, avec un demi-siècle d'avance, les perspectives d'application des nanotechnologies. Feynman soulignait avec une assurance tranquille que " Les lois physiques autorisent a priori la manipulation et le positionnement contrôlé des atomes et des molécules, individuellement, "un par un". S'appuyant sur ce constat, il déclara " Pourquoi ne pourrait-on pas écrire les 24 volumes de l'encyclopédie Britannica sur une tête d'épingle? "

Les nanotechnologies sont l'art d'observer, de manipuler, de comprendre le fonctionnement de la matière à l'échelle de quelques nanomètres. Le préfixe « *nano* » est issu du Grec « *nanos* » qui signifie nain, petit.

Un nanomètre, correspond à un milliardième de mètre, c'est 50 000 fois plus petit que l'épaisseur d'un cheveu. L'œil humain est en mesure de distinguer des objets de taille de 10 000 nanomètres. Une chaîne de dix atomes d'hydrogène mis bout à bout ferait une taille de 1 nanomètre.

Les substances issues des nanotechnologies se caractérisent par leur petitesse (au moins dans une dimension), la nouveauté et, souvent, l'imprévisibilité de leurs propriétés.

On les produits sous une variété de formes : nano-particules, -tubes, -fibres, -fils, -sphères, -coquilles, -anneaux, -plaquettes, -feuilles, -texture de la surface d'objets massifs, etc.

Les nanotechnologies touchent des domaines d'application extrêmement variés et complètement distincts, la nanothermie en fait partie.

Les petites tailles affectent aussi bien le déplacement des charges que le flux de chaleur. Par conséquent toute conception de dispositif en nanotechnologie devrait se faire avec une connaissance approfondie du déplacement des charges et du flux de chaleur [12]. Ces dix dernières années plusieurs études ont traitées du sujet [11, 12, 13, 14,15]. Un défi majeur se pose au secteur de la nanothermie à savoir, la gestion de la chaleur dans les dispositifs de taille nanométrique (électronique et optoélectronique etc....) pour maintenir la fonctionnalité et la fiabilité. Il donne aussi de nouvelles possibilités telles que le stockage de données et la conversion de chaleur en énergie, et cela, par la manipulation et le contrôle des transporteurs de chaleur (électrons, photons, phonons) [25].

1.1 Problématique

La problématique principale de notre étude est, la modélisation des phénomènes électrothermiques [7,9]. Les outils dédiés à la simulation électrothermique à l'échelle macroscopique ainsi qu'à l'échelle microscopique, ne sont plus adéquats pour l'analyse aux échelles nanométriques parce que, les lois physiques utilisées dans la réalisation des outils macroscopiques et microscopiques ne sont plus applicables à l'échelle nanométrique. Une autre problématique est celle liée au temps de simulation et de validation de nos algorithmes et modèles, certaines simulations pouvant prendre jusqu'à plusieurs jours

d'exécution voir des mois d'exécution [24]. La validation des algorithmes, est une tâche très difficile mais nous l'espérons réalisable.

En effet, comme plusieurs études ainsi que plusieurs années de pratique en ingénierie l'ont démontrée, une bonne simulation numérique vaut mieux que plusieurs manipulations ou montages par « essai erreur ». Donc tout outil permettant de prévoir où d'avoir une idée du comportement d'un système lors de sa conception est le bien venu. En particulier lorsqu'il s'agit de composants difficilement manipulables tel que les nanosystèmes.

Les lois de la physique classique n'étant plus vérifiées à l'échelle nanométrique, il faut se tourner vers d'autres lois, celle de la physique quantique dont la loi d'onde de Schrödinger ainsi que la loi de Boltzmann nous semblaient les plus indiquées pour la modélisation des phénomènes nanométriques. La complexité de la résolution classique de ces équations, nous a conduit à considérer des méthodes statistiques de résolution. Nous nous sommes penchés sur les simulations de Monte Carlo dont celle dite de « Metropolis » qui s'avère très intéressante car intègre déjà la notion de température (elle utilise la loi de distribution de Boltzmann). L'utilisation de méthodes statistiques pose un problème majeur, celui de la représentativité de l'échantillonnage donc du temps de calcul et de support de stockage. Pour répondre à cette problématique, nous avons décidé d'utiliser (pour avoir un échantillonnage représentatif ainsi qu'un meilleur temps de calcul) des algorithmes parallèles [7,8] avec le standard MPI, puis implémenter le tout sur des calculateurs de haute performance (HPC) destinés plus particulièrement au traitement parallèle. Ces calculateurs ayant plusieurs nœuds de calcul, il nous sera donc possible de répartir les tâches de calcul entre ces différents nœuds ce qui permettra de réduire de façon considérable le temps de calcul et en même temps le problème de stockage.

1.2 Objectifs

Le but que nous visons est de développer un outil de simulation et d'analyse électrothermique des nanosystèmes afin de minimiser les pertes de puissance et d'optimiser le transfert de chaleur. Cela vise deux objectifs principaux à savoir un meilleur contrôle de la chaleur pour la fiabilité et la durabilité des nanosystèmes et le deuxième, est de permettre lors de la génération de chaleur dans certaines applications biomédicales (traitement de cellules cancérigènes) une bonne estimation du champ de température.

Cet outil pour être efficace, devrait être capable de traiter une grande quantité de données en un temps acceptable.

1.3 Méthodologie de recherche

La méthodologie utilisée consiste dans un premier temps, à déterminer un système de taille nanométrique puis, des lois de la physique et des équations mathématiques à même de satisfaire aux réalités de cette échelle de grandeur. Ensuite, il faut trouver des algorithmes réalistes et peu coûteux, qui nous permettront d'implémenter et de valider de façon pratique les différents modèles issus des lois physique et des équations mathématiques déterminées précédemment. Il faut aussi déterminer les différents supports informatiques qui pourraient nous permettre en temps réel de faire nos différentes simulations et validations pour finalement aboutir à l'outil désiré.

Le système que nous utilisons dans cette première étape est une surface carrée 2 D de silicium de dimension 10x10nm. Cela nous donnera une structure de 10 000 atomes à raison de 1/10 nm par atome.

L'objectif premier de cette recherche, est de développer un outil de prévision du comportement électrothermique des nanosystèmes. La conception de nanosystèmes étant difficilement réalisable vu leur petite taille, et le coût trop élevé de leur assemblage et mise en marche, il serait dommage qu'une fois mis en fonction, leur comportement observé (thermique) ne soit pas celui attendu. D'où notre proposition d'inscrire cet outil comme partie intégrante du processus de conception des nanosystèmes.

La démarche que nous proposons pour la réalisation de cet objectif est la suivante :

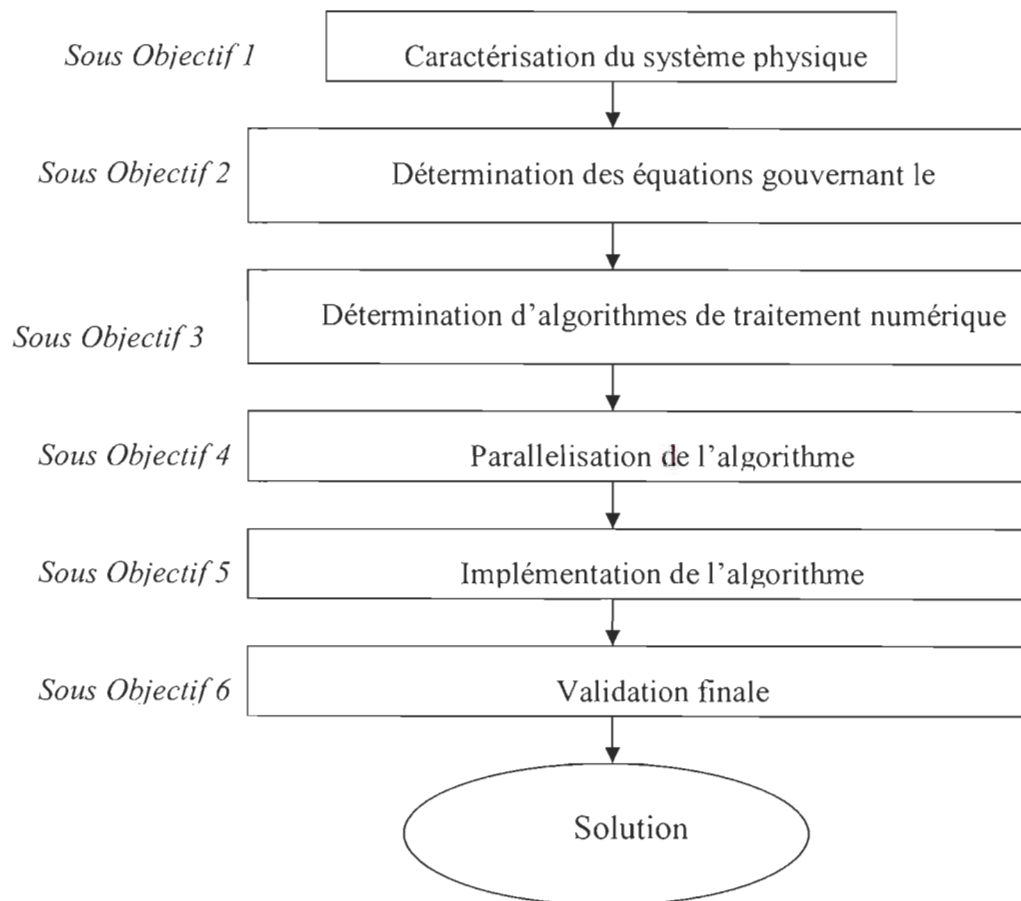


Figure 1-1 Diagramme de la méthodologie de recherche

Le sous objectif 1 : Détermination des différentes caractéristiques physiques des matériaux composant les systèmes. Nous avons établie l'ordre de grandeur des systèmes considérés à savoir, les systèmes de dimensions inférieures ou égales à 100nm. Les autres paramètres physiques que nous avons considérés, sont la forme des structures et la nature de sa composition. En ce qui concerne les formes elles peuvent être simples ou complexes. En ce qui concerne la nature de sa composition, nous entendons le ou les matériaux qui constituent le système. Ces matériaux peuvent aussi être de nature homogène ou pas. Toutes ces caractéristiques, nous permettrons de mieux cerner les systèmes à simuler. Nous présentons ci-dessous quelques formes et matériaux possibles de nanosystèmes.

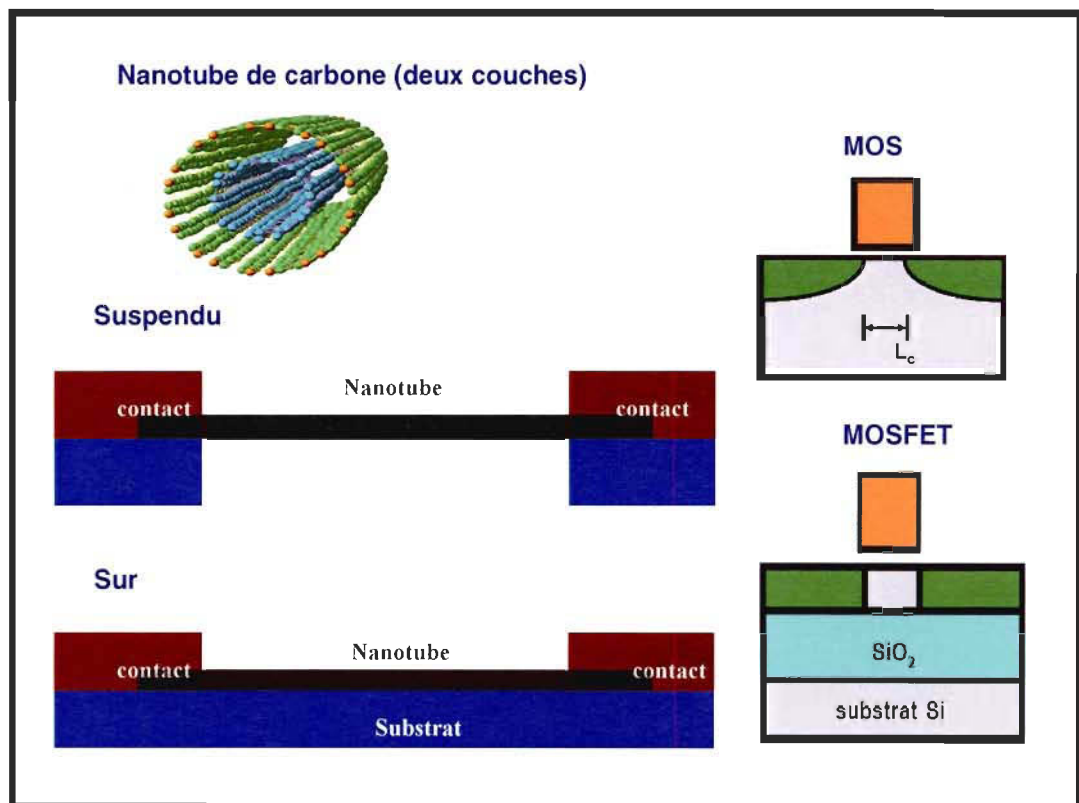


Figure 1-2 Quelques formes et matériaux possible de nanosystèmes [3]

Dans cette première étape, nous avons utilisé une surface 2D de silicium de dimension 10nm par 10nm.

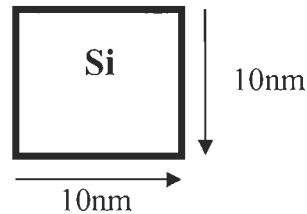


Figure 1-3 Modèle simple 2D de silicium [RP]

Le sous objectif 2 : Explorer les différentes lois et équations qui gouvernent le comportement thermique à l'échelle nanométrique. En effet, les phénomènes électrothermiques à l'échelle nanométrique, sont largement différents de ceux observés aux échelles macroscopiques ou microscopiques. Cela est sans surprise d'autant plus que les lois de la physique classique, ne sont plus appropriées pour l'explication de divers phénomènes à l'échelle nanométrique [4,21]. Donc c'est le même cas pour les phénomènes électrothermiques à cette échelle de grandeur. Les principales équations que nous avons explorées, sont l'équation de Schrödinger d'état d'un seul électron de Khon-Sham, l'équation de transport de Boltzmann (BTE), le modèle énergétique de Ising [2, 6], la loi de distribution de Boltzmann.

L'équation de Schrödinger s'écrit :

$$-\frac{\hbar^2}{2} \nabla [m^{-1}(r) \nabla \Psi(r)] + V(r) \Psi(r) = E \Psi(r) \quad 1.1$$

où m est la tension effective de la masse, \hbar est la constante de Planck réduite et Ψ et E respectivement la fonction d'onde et l'énergie.

L'équation de transport de Boltzmann, pour un état i d'un phonon sous une température ∇T est donnée par :

$$-V_{g,i} \cdot \nabla T \frac{\Delta n_i}{\Delta t} + \left(\frac{\Delta n_i}{\Delta t} \right) = 0 \quad 1.2$$

Où n_i est le nombre d'état d'occupation du phonon, t le temps et $V_{g,i}$ est la vitesse du groupe de phonon.

L'équation ci-dessous représente une forme généralisée de l'Hamiltonien du modèle d'Ising :

$$H = -J \sum_{\langle ij \rangle} s_i \cdot s_j \quad 1.3$$

où s_i et s_j sont des vecteurs représentant respectivement les spins du site i et du site j . Quand $J > 0$, les spins peuvent abaisser leur énergie E en se combinant les uns aux autres et le modèle est dit ferromagnétique. Quand $J < 0$, le modèle est dit anti-ferromagnétique.

La loi de distribution de Boltzmann énoncée comme suit, permet de trouver le système dans un état d'énergie compris entre E et $E+dE$:

$$p(E)dE = \frac{e^{-\beta E} \rho(E)dE}{Z} \quad 1.4$$

$\rho(E)$ Représente la densité d'état. Z est une fonction de partition pour un système à énergie continue.

La démarche réalisée à travers cet objectif, nous a permis d'avoir les outils et les informations nécessaires à la réalisation d'un modèle, pour la simulation numérique.

Le sous objectif 3 : Déterminer un algorithme simple qui nous permettrait d'associer les différentes équations explorées à l'étape précédente pour aboutir à un modèle simple. Un modèle qui nous permettrait après simulation de faire une analyse thermique de notre système.

Notre choix s'est porté sur les méthodes de Monte Carlo et en particulier celle dite de Metropolis. Cette méthode nous permet à travers le modèle énergétique d'Ising et la loi de distribution de Boltzmann, de déterminer pour un système à l'équilibre thermique [2] (pour une température donnée), la moyenne de certaines grandeurs physiques, telle que l'énergie et la magnétisation du système. La chaleur spécifique du système pouvant être déduite de l'énergie du système alors, nous pensons être sur la bonne voie quand à l'atteinte de notre objectif visé. Cependant cette méthode est beaucoup trop coûteuse en terme de temps et de support de stockage de données, d'où le sous objectif suivant.

Le sous objectif 4 : Trouver une manière ou un support informatique qui nous permettrait de réduire de façon appréciable, le temps de simulation de notre application. À cette attente nous avons proposé la parallélisation de notre algorithme afin de pouvoir le faire fonctionner sur des ordinateurs ou des calculateurs de hautes performances. Ces calculateurs ont la possibilité de faire la simulation sur plusieurs processeurs ou nœuds de calcul. La structure et la description de ces calculateurs se trouve sur le site du CLUMEQ [19], ou de WESTGRID [22], afin de permettre la communication entre processeur, de façon stable et efficace [20], nous utiliseront le MPI, pour la parallélisation de notre algorithme de l'étape précédente. Nous utiliserons aussi le langage C comme langage de base de l'implémentation. La technique de parallélisation qui se prête bien à notre application est la décomposition de domaine. Donc chaque processeur aura

à travailler sur un secteur bien précis de notre système. Une certaine synchronisation entre les différents processeurs sera nécessaire lors de l'échange d'information [24], sur l'état des spins voisins, qui appartiennent à des processeurs différents.

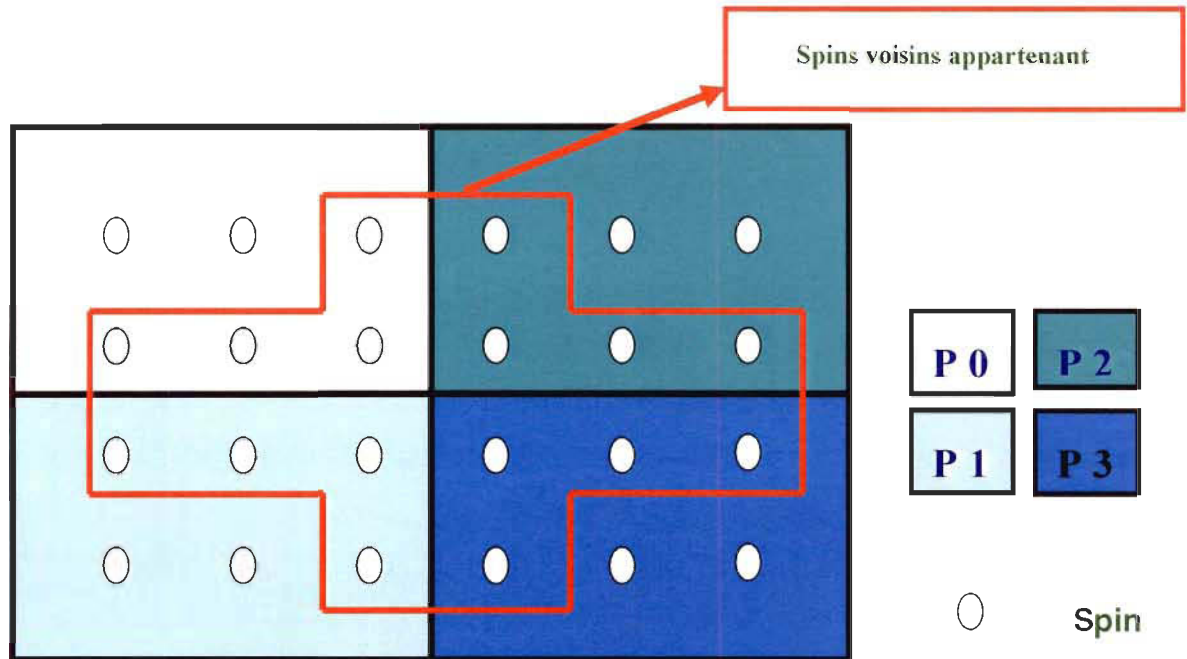


Figure 1-4 Spins voisins appartenant à des processeurs différents [RP]

Le sous objectif 5: Implémentation de l'algorithme sur les calculateurs en utilisant le standard MPI, la technique de parallélisation utilisée étant la décomposition de domaine, nous devons nous assurer, de la synchronisation de nos différents processeurs. Lors du calcul de l'énergie interne, il est nécessaire de connaître l'état des spins voisins. Si nous nous référons à la figure 4, nous constaterons que des spins voisins peuvent se retrouver sur des nœuds différents, donc la communication entre processeurs est inévitable.

1.4 Infrastructure

Nous utiliserons des calculateurs de hautes performances (HPC) comme infrastructure pour l'implantation de notre outil. Cela nous permettra de répondre aux problèmes posés par la quantité de données à traiter ainsi que par le temps de calcul.

En effet, ces calculateurs ayant plusieurs nœuds de calcul, il nous sera donc possible de répartir les tâches de calcul entre ces différents nœuds ce qui permettra de réduire de façon considérable le temps de calcul et en même temps le problème de stockage de données. Les calculateurs que nous avons identifiés pour notre étude sont entre autres, les réseaux de calcul WestGrid, CLUMEQ et ACNET, faisant partie du calcul Canadien [13].

Nous présentons ci-dessous, quelques ressources de WestGrid et de CLUMEQ.

- *Les ressources sur le réseau WestGrid,*

Le réseau WestGrid possède plusieurs plateformes de traitement pour des applications parallèles ainsi que des applications séquentielles.

Cortex:

Traitement parallèle avec mémoire partagé ou MPI. Il est convenable pour des traitements nécessitant une grande quantité de données cependant le temps d'exécution permis est limité. (24 heures). Cette plateforme possède des machines de 4, 32, ou 64 processeurs avec respectivement 16, 156, ou 256 GB de mémoire par nœuds de calcul.

Nexus:

Traitement parallèle avec mémoire partagé ou MPI. Il est convenable pour des traitements nécessitant une grande quantité de données cependant le temps d'exécution

permis est aussi limité. (24 heures). Cette plate forme possède des machines de 8 à 256 processeurs avec 8 à 256 GB de mémoire par nœuds de calcul.

Matrix:

Traitement parallèle avec le MPI. Cette plate forme possède 128 nœuds de calcul avec 2 GB de mémoire par processeur nœud. Elle possède aussi 832 GB d'espace de stockage pour les utilisateurs. Le réseau d'interconnexion des nœuds est très rapide, ce qui se prête bien aux traitements parallèles.

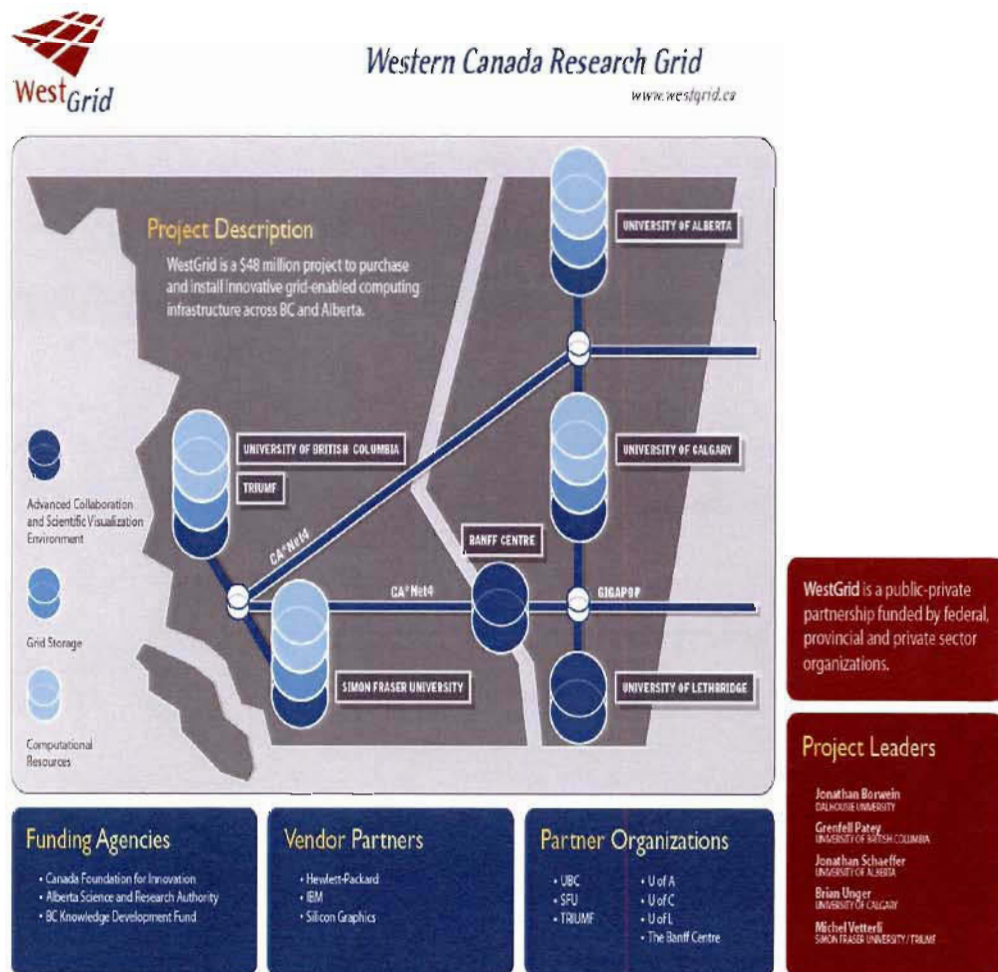


Figure 1-5 Réseau global du WestGrid [22]

- *Les ressources CLUMEQ*

Le réseau CLUMEQ possède deux plateformes principales de traitement pour des applications parallèles ainsi des applications séquentielles.

Beowulf Cluster :

Cette plateforme possède 256 processeurs avec 1.6 GHz et 1.5 GB de mémoire par processeur.

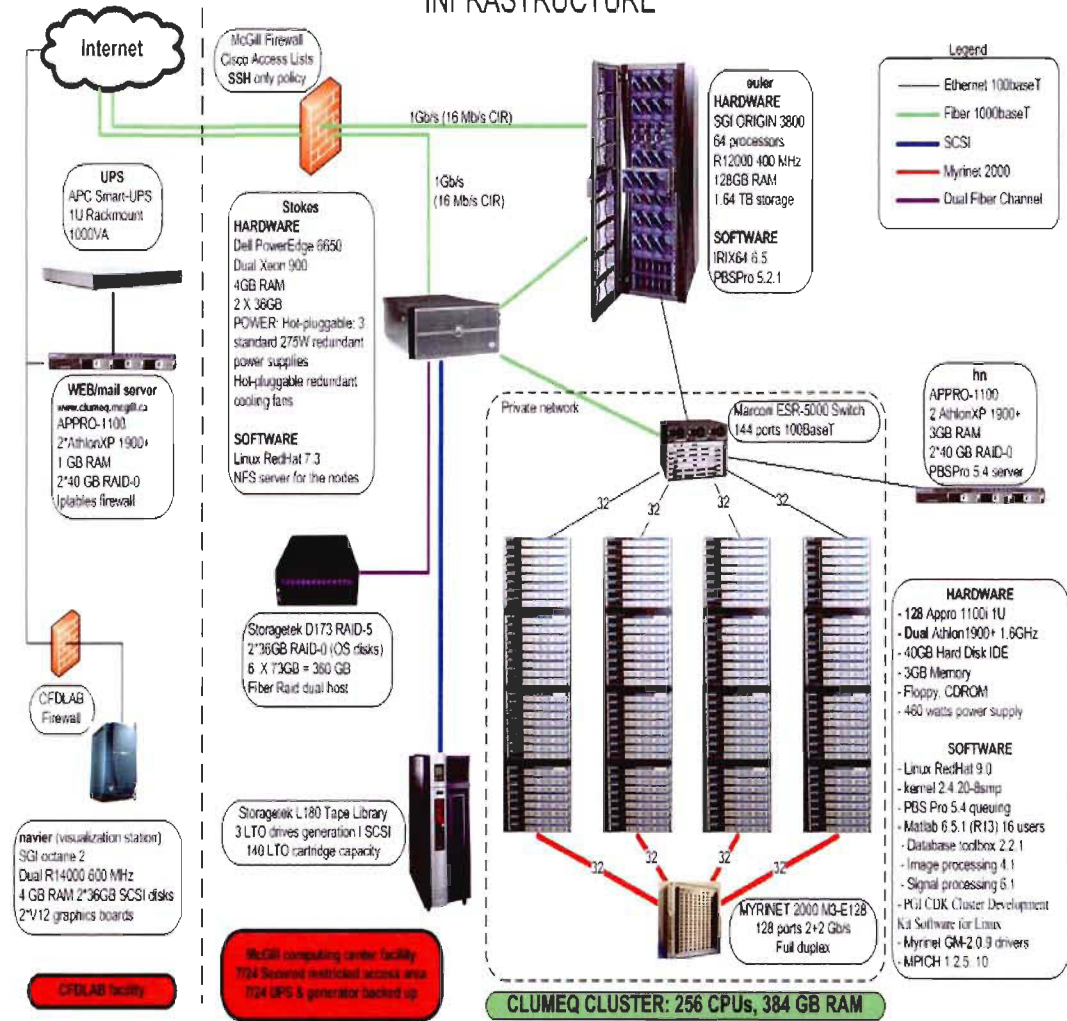
SGI Origin 3800 :

Cette plateforme possède 64 processeurs avec 600 MHz par processeur et 128 GB de mémoire au total.

À la page suivante, nous présentons une figure représentant le réseau global du CLUMEQ.

CLUMEQ

Consortium Laval UQAM McGill and Eastern Quebec INFRASTRUCTURE



01:12/2004 Rev.1.41

Figure 1-6 Réseau global du CLUMEQ [19]

Chapitre 2 - La nanotechnologie et les nanosciences

La nanotechnologie et les nanosciences couvrent un domaine très vaste d'applications et de phénomènes [23, 34], chacun d'entre eux pouvant faire l'objet de plusieurs thèses de doctorat. Nous avons choisi de faire un tour d'horizon de quelques uns de ces phénomènes à savoir, les NEMS (nano-electro-mechanical-systems), NOEMS (nano- opto-electro-mechanical-systems), les nanotransistors, les nanotubes et la nanofluidique

2.1 Les NEMS et les NOEMS

Les techniques utilisées dans le domaine de la microélectronique pour la fabrication des circuits intégrés ont beaucoup évoluées depuis l'introduction des techniques de photolithographie. Ces techniques ont énormément contribuées au développement d'objets micromécaniques, MEMS (micro-electro-mechanical-systems) et MOEMS (micro- opto-electro-mechanical-systems). L'attrait principal de ces techniques est leur capacité d'atteindre d'excellentes précisions et leur capacité de production simultanée, ce qui permet de nombreux exemplaires à moindre coût.

Les MEMS et les MOEMS ont des propriétés aussi bien mécaniques qu'électronique et/ou optique. Ils ont aussi les propriétés d'une taille de l'ordre du micron d'où leur utilisation dans une très grande variété d'applications [25,26].

L'avancée des techniques de miniaturisation utilisée dans la conception des MEMS, ont contribué grandement à faciliter la conception des NEMS et des NOEMS [25].

Les NEMS et des NOEMS sont des dispositifs analogues respectivement aux MEMS et aux MOEMS, mais à l'échelle du nanomètre. On parle de NEMS pour des structures mécaniques miniatures, réalisant des fonctions de capteur ou d'actionneur, dont au moins une des dimensions est de taille nanométrique. En raison de leur taille, la physique des NEMS et des NOEMS est différente de la physique "à l'échelle conventionnelle". Les forces surfaciques et les phénomènes quantiques deviennent prédominants.

Le développement des NEMS et des NOEMS est motivé par la double perspective d'un gain d'échelle et surtout de l'utilisation de phénomènes nouveaux spécifiques à l'échelle nanométrique. De plus en plus de travaux sont réalisés intégrant des nanotubes et nanofils (NNs) dans des structures NEMS (actuateurs, résonateurs,...).

La fabrication des NEMS peut être classée en deux catégories à savoir, l'approche 'Top-down' et l'approche 'Bottom-up' [25, 29, 31, 32]. L'approche 'Top-down' est basée sur la microfabrication, on part d'un matériau solide, puis on le découpe, on le transforme, etc., à l'échelle nanométrique. Cette approche inclue des technologies comme la nano-lithographie, nano-impression [31, 32].

Alors que l'approche 'Bottom-up' est une approche basée sur les techniques d'assemblage dans lesquelles on assemble la matière atome par atome pour former des particules ou des molécules qui sont par la suite intégrés dans des systèmes plus grands[29]. Ces stratégies incluent des techniques telles que l'auto-assemblage et la lithographie stylo à plume (dip-pen lithography) [30,31] qui est une technique de lithographie qui permet de reproduire le motif d'une structure nanométrique, en utilisant la pointe d'un microscope

à force atomique, pour déposer à la surface d'un matériau. Ces techniques peuvent produire des nano modèles réguliers à de grandes échelles. Avec la capacité de placer et orienter des objets nanométriques.

La manipulation nanorobotique est une technologie qui permet de structurer, de caractériser et d'assembler plusieurs types de nanosystèmes. En combinant les deux approches précédentes c'est à dire le 'Top-down' et le 'Bottom-up', une approche hybride nanorobotique basée sur la manipulation nanorobotique, peut fournir une troisième manière de fabriquer NEMS par des nanomatériaux ou des nanostructures [32]. Dans cette nouvelle approche, le 'Top-down' et le 'Bottom-up' peuvent être exécutés dans un ordre arbitraire.

2.2 Les nanotubes de carbone

Les nanotubes de carbone ont été découverts en 1991 par *Sumio Iijima*, de la compagnie japonaise NEC, au moyen d'un microscope électronique. Les chercheurs en avaient produit depuis de nombreuses années sans s'en rendre compte. Ce qui stimule les recherches sur les nanotubes de carbone est le fait qu'ils possèdent plusieurs propriétés fort intéressantes [39, 43, 44, 45]. Ainsi, les nanotubes de carbone sont très stables chimiquement, chaque atome de carbone étant lié à trois autres atomes de carbone. Les atomes forment un cylindre de quelques nanomètres seulement de circonférence, dont la longueur peut cependant atteindre plusieurs milliers de nanomètres. Les nanotubes de carbone peuvent même comporter plusieurs parois (cylindres) [43].

Les propriétés des nanotubes de carbone sont tout à fait étonnantes [25, 33]. On a calculé par exemple qu'un câble entièrement fabriqué de nanotubes de carbone

et mesurant 2,5 centimètres de diamètre (soit 10^{14} nanotubes parallèles) serait une centaine de fois plus résistant qu'un câble d'acier de la même dimension tout en étant six fois plus léger. Mais un problème subsiste : les nanotubes de carbone n'ont pas tendance à s'attacher entre eux, parce que pratiquement tous les électrons sont déjà liés à d'autres atomes de carbone [33]. Chaque nanotube est extrêmement résistant, tant à la pression qu'à l'étirement, et résiste très bien à la chaleur. En revanche, les nanotubes n'ont pas tendance à s'agglomérer [25]. Or, il est absolument nécessaire qu'ils tissent des liens très forts entre eux pour former un câble solide. Pour pallier à ce problème deux voies distinctes se dégagent pour permettre l'utilisation des nanotubes dans des matériaux. La première approche vise l'assemblage des nanotubes entre eux par des liaisons chimiques latérales [41] pour produire un matériau carboné renforcé, alors que la deuxième approche s'intéresse à l'inclusion de nanotubes dans des métaux ou des polymères pour renforcer ces derniers.

Les nanotubes ayant une très forte propension à dissiper la chaleur [39, 40, 41], ils constituent, théoriquement du moins, un matériau de choix pour des applications électroniques. Combinés à des métaux ou à des plastiques, ils pourraient également être employés pour dissiper la chaleur dans des moteurs, par exemple (ce qui permettrait l'emploi de plastiques, impossible actuellement). Des fabricants de prothèses médicales espèrent mettre à profit la résistance mécanique et la stabilité thermique des nanotubes pour remplacer les métaux et les polymères employés actuellement dans la fabrication de prothèses. Toujours dans le domaine médical, il semble qu'actuellement la société Honeywell étudie la possibilité d'utiliser des nanotubes pour fabriquer des muscles artificiels. Il serait également possible d'utiliser les nanotubes comme senseurs super

sensibles [44]. En effet, les nanotubes semi-conducteurs voient leur résistance électrique changer radicalement lorsqu'on les expose à des composés alcalins, des gaz halogènes ou d'autres gaz, ce qui soulève la possibilité de développer des senseurs chimiques améliorés. Des études sont en cours quant au stockage d'hydrogène dans les nanotubes, puisque ceux-ci sont creux. Reliés à la pointe d'un microscope à sonde balayante, des nanotubes peuvent augmenter la résolution latérale du microscope par un facteur de dix ou plus, ce qui permet de mieux voir des protéines et autres grandes molécules. On accorde une probabilité élevée à cette utilisation des nanotubes. Cependant, ce qui reçoit sans doute le plus d'attention des chercheurs est l'emploi des nanotubes en électronique. En effet, les nanotubes peuvent être soit conducteurs, soit semi-conducteurs selon leur chiralité (la façon dont leur feuille de graphite est enroulée) [33, 40], une propriété qui leur est exclusive par rapport aux autres matériaux (certains types de nanotubes deviennent même supraconducteurs à très basse température). Différents groupes de recherche ont réussi à créer des transistors à effet de champ (*field-effect transistor*, ou FET) [40, 41, 42] en reliant deux électrodes métalliques avec des nanotubes. Les électrons circulent d'une électrode à l'autre à l'intérieur d'un nanotube qui peut fonctionner à température ambiante et démontre des propriétés semblables à celles des transistors à base de silicium. De plus, étant donné sa petite taille, le nanotube de carbone utilise beaucoup moins d'électricité. Des théoriciens ont prédit que des interrupteurs basés entièrement sur les nanotubes pourraient fonctionner à des vitesses d'un térahertz ou plus, soit 1000 fois plus rapidement que les processeurs actuellement disponibles. Cependant comme nous l'avons mentionné précédemment, le grand problème est de trouver des moyens de manipuler ces nanotubes efficacement ou, à tout le moins, de trouver une méthode efficace d'assemblage.

Les nanotubes de carbone sont la matière première de nombreuses réalisations de NEMS [43, 48] (nano-moteur, oscillateur ultra haute fréquence, actuateurs, résonateurs).

Un des problèmes majeurs pour l'étude des nanotubes ainsi que pour leurs applications en électronique est le contrôle de leur configuration géométrique lors de la synthèse. Pour le moment, les échantillons issus des différentes méthodes de synthèse contiennent à la fois des nanotubes semi-conducteurs et des nanotubes conducteurs. Des études sur la séparation des nanotubes semi-conducteurs et des nanotubes conducteurs par électrophorèse ont été publiées [33,50], mais cette méthode, de part sa faible efficacité, reste à développer. En plus, la plupart des produits issus des différentes méthodes de synthèse contiennent des impuretés telles que des restes de catalyseurs, du carbone amorphe ou des fullerènes [46-51]. Des étapes de purifications sont donc nécessaires pour obtenir un matériau uniquement constitué de nanotubes [51].

2.3 Les nanotransistors

Plusieurs équipes de chercheurs se sont penchées sur des mécanismes susceptibles de miniaturiser les transistors ou d'entreposer de l'information : nous allons décrire ici quelques-uns de ces mécanismes qui ont fait l'objet de plusieurs publications [27, 28, 36, 38, 39, 40, 41, 46,47, 48, 49, 51,61], soit les dispositifs à effet tunnel résonnant, les points quantiques, les transistors à un seul électron ou encore, les fils quantiques. Les dispositifs à effet tunnel résonant font l'objet de recherches depuis plusieurs années. Deux types de mécanismes furent proposés, soit les diodes à effet tunnel résonant (RTD) [52, 53, 54] et les transistors à effet tunnel résonant (RTT) [27, 55, 56, 57, 58,59, 60, 64, 65, 67] (RTT, qui utilisent les RTD). Le principe de fonctionnement d'un RTD est le suivant : un matériau semi-conducteur donné, par exemple de l'arséniure de gallium (GaAs), est disposé de telle

façon qu'une sorte d'île est créée entre la source et le drain. Le matériau de cette île est conducteur et il est entouré d'un alliage de GaAs et d'arséniure d'aluminium (AlAs) qui laisse moins facilement passer les électrons. La seule façon pour les électrons de traverser les deux passages qui isolent l'île est d'avoir un niveau d'énergie identique à celui des électrons sur l'île. Lorsque le niveau d'énergie des électrons provenant de la source atteint celui des électrons sur l'île, on dit qu'ils sont en résonance et ils peuvent alors traverser la première barrière, atteindre l'île et traverser la deuxième barrière. Autrement dit, ce mécanisme assure que les électrons passent ou ne passent pas, ce qui est le propre d'un transistor. Il suffit donc d'augmenter le niveau d'énergie des électrons afin de les amener au niveau des électrons sur l'île pour que ceux-ci passent en continu. Des chercheurs ont créé un transistor hybride, soit un RTT, en incorporant un RTD à un circuit classique basé sur le silicium [35,65]. Un RTT permet de réduire le nombre de dispositifs nécessaires à l'implantation de fonctions logiques et donc d'augmenter la densité des circuits intégrés. Cependant, ce circuit hybride demeure limité par le problème de miniaturisation qui affecte la lithographie conventionnelle à base de silicium. C'est pourquoi d'autres avenues où l'on utilise des RTD dans des architectures de circuits différentes des circuits au silicium font l'objet de recherches qui visent à vraiment profiter des possibilités offertes par les dispositifs à effet tunnel résonant.

Les transistors à un seul électron (SET) fonctionnent un peu sur le même principe que les RTD [35, 58, 63, 66]. Une île contenant des électrons et isolée par un matériau semi-conducteur est située entre la source et la sortie. L'île est si petite qu'elle ne peut contenir qu'un nombre très limité d'électrons libres, ce qui bloque le passage des électrons provenant de la source. Ce phénomène, appelé blocage de Coulomb [36, 64], est dû à la force

de répulsion des électrons entre eux (en effet, ils sont tous chargés négativement). Une électrode est située à proximité de l'île. Il suffit d'augmenter la tension appliquée à cette électrode pour qu'un électron supplémentaire passe de la source à l'île et qu'un électron de l'île traverse vers la source, créant ainsi un faible courant continu. Il est donc possible de bloquer ou de laisser passer les électrons en modifiant la tension de l'électrode. La plupart des dispositifs de ce type ne fonctionnent présentement qu'à des températures très basses. En effet, l'énergie thermique des électrons, si elle est excessive, peut leur permettre de franchir le blocage de Coulomb et donc de sauter sur l'île, peu importe le niveau de tension appliqué à l'électrode. Plusieurs groupes s'emploient à mettre au point des dispositifs de ce type qui fonctionneraient à la température de la pièce [37, 67, 71].

Un autre exemple de circuit exploitant des effets quantiques et une architecture novatrice est basé sur l'utilisation de points quantiques (ou boîtes quantiques) [67, 68, 69, 70].

Les quelques exemples de nouvelles approches présentés ci-dessus ne constituent en rien une liste exhaustive. Bien d'autres recherches se poursuivent actuellement qui portent sur de nouveaux mécanismes et de nouvelles façons de concevoir les ordinateurs.

2.4 Modélisation

Actuellement le développement de nanosystèmes électromécaniques (NEMS) et des nanotransistors est motivé par la double perspective d'un gain d'échelle et surtout de l'utilisation de phénomènes nouveaux spécifiques à l'échelle nanométrique. De fait, de plus en plus de travaux sont réalisés intégrant des nanotubes et nanofils dans des structures NEMS (actuateurs, résonateurs,...). Cependant très peu de réalisations ont vu

réellement la mise en œuvre de nouveaux phénomènes liés à l'échelle nanométrique. La plupart des travaux se limitent jusqu'à présent à la reproduction à plus petite échelle de phénomènes couramment utilisés dans les systèmes micrométriques (MEMS). L'analyse et la simulation de phénomènes thermiques à l'échelle nanométrique, fait partie de ces phénomènes qui sont très peu abordés.

Notre travail se situe donc à ce niveau. Dans cette section, nous essayerons d'aborder certaines théories ou lois de la physique quantique qui nous permettra de modéliser et de simuler les phénomènes nanothermiques. Vu l'envergure du domaine de recherche lié à la nanotechnologie et à la diversité des matériaux, nous avons décidé de limiter notre étude à une surface de silicium (de taille nanométrique). Une généralisation pourra s'effectuer ultérieurement une fois la faisabilité de cette simulation établie.

2.5 Théories et lois de la physique quantique

Dans cette section, nous aborderons certaines expressions, concepts ou modèles physiques qui nous permettront de définir notre système ainsi que le phénomène que nous étudions.

L'élément quantique principal utilisé dans notre travail est le spin. Son utilisation nous permet de caractériser notre système ainsi que d'estimer certaines grandeurs physiques, telles que, l'énergie, la magnétisation, la température et la conductivité thermique.

2.5.1 Les spins

L'utilisation des spins dans l'étude des nanotechnologies est de plus en plus grandissante [72-86]. Son utilisation constitue à elle seule une technologie émergente qui exploite la propriété quantique du spin des électrons dans le but de stocker des informations

(prix Nobel de physique 2007) à savoir la spintronique ou électronique du spin. Nous essayerons ci-dessous de donner une brève définition du spin ainsi que quelques unes de ces propriétés qui nous seront utiles dans notre étude.

Le spin est une propriété quantique intrinsèque associée à chaque particule, qui est caractéristique de la nature de la particule, au même titre que sa masse et sa charge électrique [85, 86]. Elle permet de caractériser le comportement de la particule sous l'effet de la symétrie de rotation de l'espace. Le spin se manifeste comme un faible état d'énergie magnétique caractérisé par “*spin up*” et “*spin down*”.

Introduisons maintenant *l'hamiltonien* d'un système magnétique simple, constitué d'un ensemble de molécules ou d'atomes, dotés d'un moment magnétique intrinsèque, lié au spin et fixés au noeuds d'un réseau. L'énergie sera la somme de deux termes, l'un dû au paramagnétisme (l'interaction avec un champ extérieur) et l'autre dû au ferromagnétisme (lié aux interactions entre molécules liées) [74, 75, 77]. Le moment magnétique associé à un spin \vec{S} est donné par :

$$\vec{\mu} = -g\mu_B \vec{S} \quad \text{avec} \quad \mu_B = \frac{e\hbar}{2m} \quad 2.1$$

Dans l'équation ci-dessus, g est un facteur numérique proche de 1 et μ_B est le magnéton de Bohr.

En général le spin et les autres degrés de liberté de l'atome sont indépendants [72]. Ils sont seulement liés par les questions de symétrie de la fonction d'onde totale. L'énergie d'un ensemble de molécules dans un champ magnétique se lira donc :

$$\hat{H} = -\sum_{i=1}^N g\mu_B \vec{S}_i \cdot \vec{H} - \sum_{i=1}^N \sum_{j=1}^N J_{ij} \vec{S}_i \cdot \vec{S}_j \quad 2.2$$

Dans l'équation 2.2, \hat{H} correspond à l'hamiltonien, \vec{H} est le vecteur de champ magnétique extérieur, \vec{S}_i et \vec{S}_j représente respectivement le i-ème et le j-ème élément des site i et j. Le facteur J_{ij} donne la valeur du couplage d'échange entre les sites i et j. une valeur positive va favoriser les spins dans la configuration parallèles (en ce qui concerne l'énergie du système) alors qu'une configura antiparallèles des spins sera favorisée par une valeur négative du facteur J_{ij} [77, 78, 79].

L'interaction peut être à courte portée (premiers voisins sur un réseau cubique par exemple) ou à longue portée (tous les spins du réseau interagissent entre eux) ; on peut considérer uniquement la composante du spin dans la direction du champ (modèle dit d'Ising) ou toutes les composantes (modèle d'Heisenberg) [2], ou deux composantes (modèle XY) [2]. Nous nous intéressons ici au modèle d'Ising, et nous le considérerons à deux dimensions.

Ce modèle doit son succès à plusieurs raisons: la première est que beaucoup de problèmes peuvent être modélisés comme des ensembles de systèmes à deux possibilités [87-93]. Par exemple, le gaz sur réseau, où les particules peuvent se trouver aux noeuds d'un réseau et interagissent avec un premier voisin a une relation directe avec le modèle d'Ising (spin orienté vers le haut (-) noeud occupé, spin orienté vers le bas (-) noeud vide), de même un modèle d'alliage binaire avec des atomes Cu et Zn, etc. Comme les phénomènes ont un caractère d'universalité, on peut rendre compte de beaucoup de phénomènes en simplifiant au maximum et en ne gardant que ce qui est relevant pour l'approche du point critique. Une deuxième raison réside dans le fait que le modèle d'Ising

a été résolu exactement à 2 dimensions, sur un réseau carré et avec une interaction entre premiers voisins [87].

2.5.2 *Modèle d'Ising*

Le modèle d'Ising, dénommé d'après le physicien Ernst Ising, est un modèle de physique statistique [87, 88, 89]. Il a été utilisé pour modéliser différents phénomènes dans lesquels des effets collectifs sont produits par des interactions locales particules à deux états, comme le ferromagnétisme [1, 89, 90].

Le modèle d'Ising est un modèle sur réseau de moments magnétiques de particules ayant pour propriétés spécifique d'être toujours orientés suivant le même axe spatial et de ne prendre que deux valeurs possibles, $+M$ et $-M$. Ce modèle permet de décrire relativement simplement le magnétisme des matériaux ferromagnétiques présentant une anisotropie très forte avec une direction privilégiée très marquée. Une autre application du modèle d'Ising est la description des alliages binaires. Dans ce cas, les moments magnétiques $+M$ représentent une des espèces atomiques, et les moments magnétiques $-M$ représentent l'autre espèce atomique. L'ordre à longue distance du modèle d'Ising peut décrire une séparation de phase entre les deux espèces (dans le cas où la phase de basse température à tous les moments égaux à $-M$ ou $+M$) ou bien une phase ordonnée dans laquelle l'un des sous réseaux porte des atomes d'une espèce (moments $+M$) et l'autre sous réseau des atomes de l'autre espèce. La phase désordonnée du modèle d'Ising décrit respectivement un état où les deux espèces se mélangent ou un état où les sous-réseaux sont équivalents. Le second cas est appelé transition ordre-désordre. Cette version du modèle d'Ising est appelée modèle de Bragg et Williams (1936).

Modèle d'Ising appliqué à la magnétisation

Nous représentons ci-dessous, une structure 2 D. Chaque point du réseau est un atome, les flèches indiquent le moment magnétique.

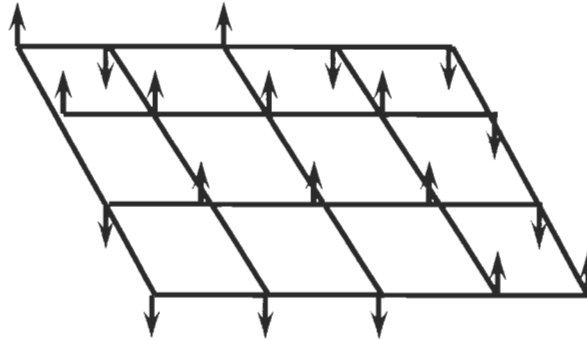


Figure 2-1 Réseau carré à 2D grille d'Ising [RP]

Chaque atome de la grille (représenté par les intersections de la grille) porte un moment magnétique (spin, représenté par les flèches orientées en haut et en bas) égal à : $\pm \mu_B$ (Magnéton de Bohr). Le moment magnétique total est peut s'écrire:

$$M = \mu_B \sum_{K=1}^N S_K \quad 2.3$$

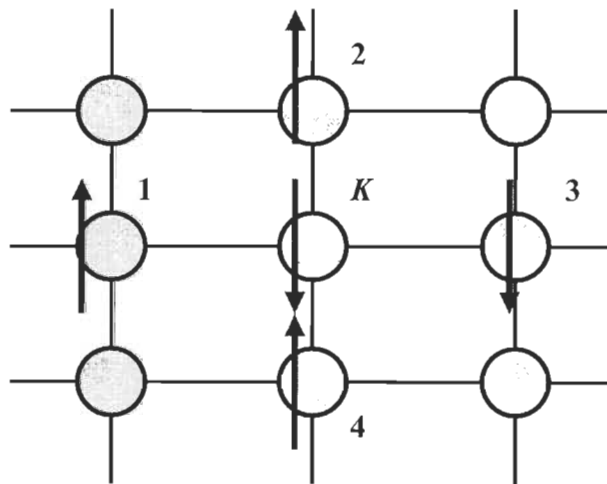


Figure 2-2 Réseau carré à 2D grille d'Ising [RP]

Les spins interagissent 2 à 2 entre premiers voisins uniquement [90, 91, 92, 93, 94]

L'énergie d'interaction est :

$$E = \frac{1}{2} \sum_{K=1}^N \sum_{\substack{j, \text{voisin} \\ \text{de } K}}^4 JS_j S_K - B \sum_{K=1}^N S_K \quad 2.4$$

J est la constante d'échange

$J < 0$: État fondamental ferromagnétique ($M \neq 0$)

Si $H=0$

$J > 0$: État fondamental Antiferromagnétique ($M = 0$)

Analogie avec les alliages binaires

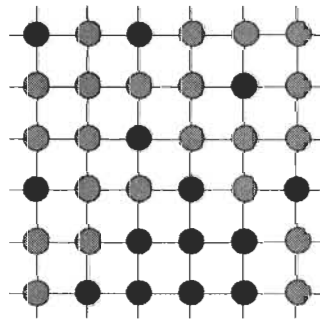


Figure 2-3 Alliage A, B sur réseau carré

La figure 2-3 représente un alliage A-B, sur un réseau carré, chaque atome interagit avec ses premiers voisins, avec des énergies : E_{AA} , E_{BB} , E_{AB} .

Alors l'énergie du système s'écrit :

$$E = \frac{1}{2} \sum_{K=1}^N \sum_{\substack{j, \text{voisin} \\ \text{de } K}}^4 E_{AA} C_j^A C_K^A + E_{BB} C_j^B C_K^B + E_{AB} (C_j^A C_K^B + C_j^B C_K^A) \quad 2.5$$

où $C_K^A = 1$ si l'atome K est de type A, et $= 0$ sinon.

$$C_K^B = 1 - C_K^A, \quad (C_K^A : \text{Nombre d'occupation})$$

En posant :

$$C_K^A = \frac{1+S_K}{2} \quad 2.6$$

et

$$C_K^B = \frac{1-S_K}{2} \quad (S_K = +1 \text{ si } A \text{ et } -1 \text{ si } B) \quad 2.7$$

On montre que l'énergie se réécrit :

$$E = \frac{1}{2} \sum_{K=1}^N \sum_{\substack{j, \text{voisin} \\ \text{de } K}}^4 \frac{1}{4} (E_{AA} + E_{BB} - 2E_{AB}) S_j S_K + (E_{AA} - E_{BB}) \sum_{K=1}^N S_K + Cste \quad 2.8$$

Si $J < 0$: État fondamental Ferromagnétique (c'est à dire soit totalement A, soit totalement B)

Si $J > 0$: État fondamental antiferromagnétique

2.5.3 L'équilibre thermique

L'orientation de l'ensemble des spins du système, nous permettra de connaître son état. Cependant cet exercice s'avère complexe dû au nombre élevé de spins à considérer pour

un système. Ce problème peu être résolu si nous considérons le système à l'équilibre thermique (le système est à la même température que son environnement et en tout point du système). En effet, La probabilité de distribution Boltzmann nous permet de connaître l'état du système lorsque celui-ci est à équilibre thermique.

Elle s'énonce comme suit :

$$P(S) = \frac{e^{-E(S)/k_B T}}{\sum_A e^{-E(A)/k_B T}} \quad 2.9$$

Dans cette équation, $P(S)$ est la probabilité de trouver le système dans un état S donné, T est la température à la quelle est soumis le système, K_B est la constante de Boltzmann, $E(s)$ est l'énergie du système à l'état S et la somme \sum_A représente la somme de tous les états possibles du système.

Chapitre 3 - Développement de l'algorithme

Dans notre démarche pour la simulation de transfert de chaleur dans les nanosystèmes, nous devrions trouver une méthode ou un algorithme qui nous donnerait la possibilité d'associer les particules, température et espace. Cela nous permettrait d'identifier, de calculer ou de déduire le transfert de chaleur dans le système sous certaines conditions. Plusieurs techniques et algorithmes sont présentés dans la littérature [10-18]. L'échelle de grandeur considérée nous exige de nouvelles considérations dans l'analyse des caractéristiques physiques des systèmes étudiés et des lois physique qui les gouvernent. L'une de ces lois est la distribution ou le transport des électrons ainsi que les interactions possibles avec leur environnement (avec les autres particules ou avec les frontières du système), la figure 3-1 ci-dessous nous donne une idée des modèles possibles pour l'analyse du transport des particules avec leur complexité et leur précision [10]. Nous pouvons y voir que les méthodes les plus appropriées pour cette échelle de grandeur, sont les méthodes basées sur l'approche quantique. Les méthodes les plus utilisées dans la liste de la figure 10, sont le modèle de transport de Boltzmann et le modèle de transport utilisant l'équation de Schrödinger. Cependant l'utilisation de ces techniques comporte des limites. La méthode de transport de Boltzmann a connu plusieurs dérivés, malgré cela l'approche demeure toujours la même et selon [10] son efficacité n'est effective qu'aux limites du classique. Donc lorsque nous tendrons de plus en plus vers

l'interprétation ou la simulation des applications quantiques, elle risque d'être de moins en moins efficace. En ce qui concerne le modèle de Schrödinger, il serait le plus efficace pour les applications quantiques. Cependant pour le type d'application qui nous concerne, son utilisation peut s'avérer un peu limitée dans la prise en compte de tous les événements qui pourraient se produire. En effet, l'équation de Schrödinger serait idéale pour la description complète d'une seule particule. Son utilisation serait compliquée pour la description d'un réseau de particules et de surcroît les différentes interactions entre ces particules d'une part et entre ces particules et leur environnement de l'autre. En plus il pourrait avoir plusieurs types de particules ce qui compliquerait encore plus la tâche.

Les modèles adéquats indiqués sur la figure 3-1, sont les modèles quantiques, ils permettent la prise en compte de la plus part des événements intervenant dans la description, la simulation et l'interprétation des grandeurs physiques recherchées lors de la simulation. Comme mentionné dans [10], un problème se pose cependant, c'est celui du temps de simulation et de la représentativité de l'échantillonnage. Cette préoccupation trouve solution, dans l'utilisation de calculateur de haute performance. Pour compléter un algorithme, la plupart des modèles de transport mentionnés plus haut sont combinés à une méthode de simulation. La plus utilisée de ces méthodes, est la simulation de Monte Carlo, nous en donnons une description dans la section ci-après.

Le modèle d'Ising pourrait être classé parmi les approches quantiques parce que faisant intervenir des particules (spin). Le modèle d'Ising permettra de décrire l'état énergétique de chaque particule après interaction avec ses premiers voisins.

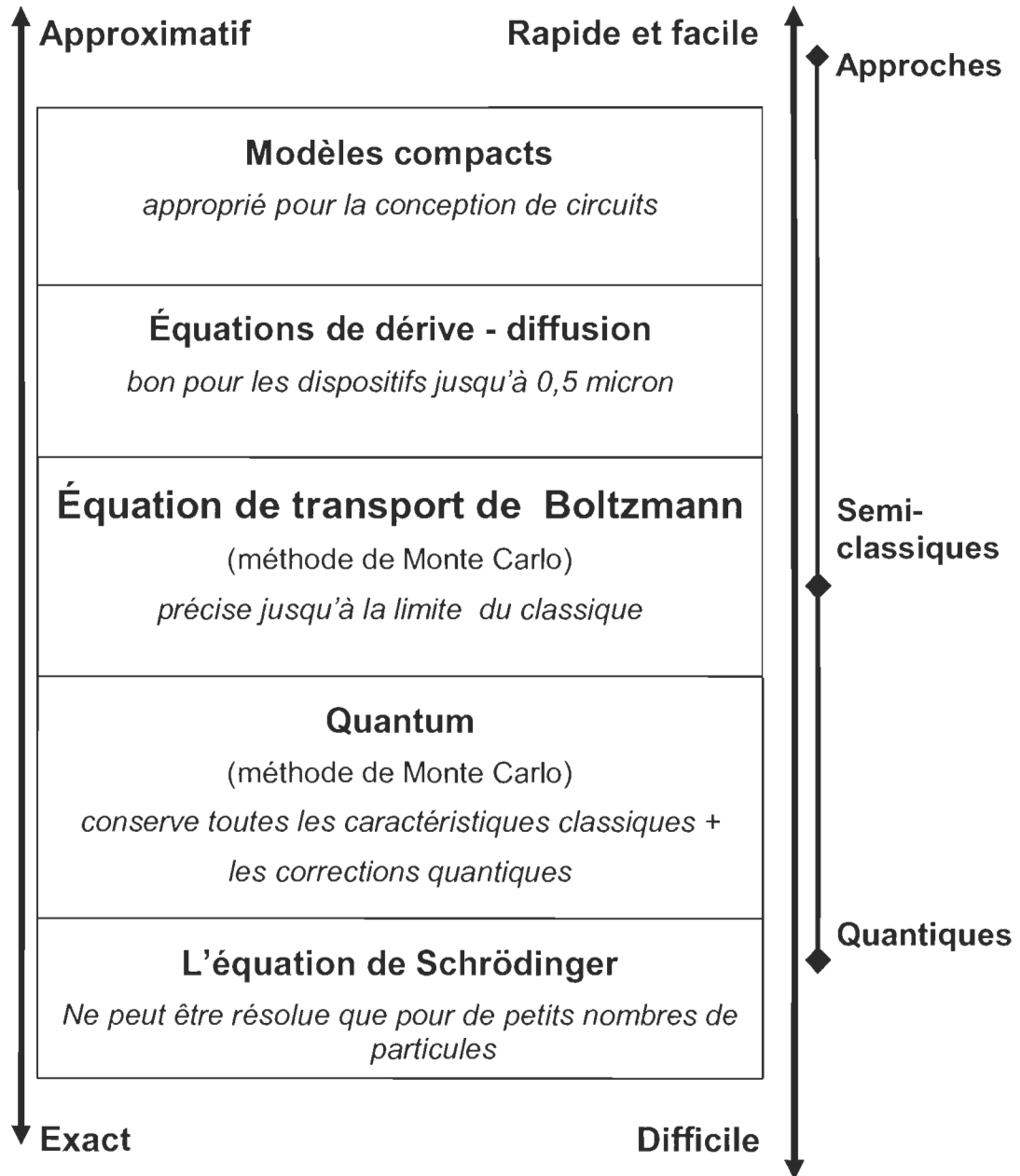


Figure 3-1 Illustration de la hiérarchie du modèle de transport [10]

3.1 Algorithme de Monte Carlo

L'algorithme de Monte Carlo est utilisé pour une multitude d'applications, elle tient son nom du célèbre casino de Monaco, et souligne le rôle important des décisions aléatoires

dans l'algorithme. La simulation de Monte Carlo de la "réalité" à travers un certain nombre de mesures aléatoire simple est souvent le seul moyen réaliste de la réalisation des calculs dans les grands systèmes complexes.

L'un des facteurs clé des méthodes numériques, telles que l'algorithme de Monte Carlo demeure l'échantillonnage. A titre d'exemple, si nous nous retrouvons avec un nanosystème de $N \times N$ sites (avec $N = 100$), nous aurons, 2^{10000} états possibles. Ce qui est un nombre spectaculairement grand, d'où l'importance de prendre une matrice d'échantillonnage de dimension assez grande sinon l'on risquera de ne pas parcourir tous les états. On parle dans ce cas d'échantillonnage important.

3.2 Chaîne de Markov

Le processus de Markov est le mécanisme qui génère un état n du système à partir d'un autre m connu. L'état généré n'est pas toujours le même, il parcourt le système à la recherche de nouveaux états avec une probabilité de transition $P(m \rightarrow n)$ sur lesquels il impose deux conditions:

- a) elles ne varient pas avec le temps.
- b) elles dépendent uniquement des propriétés du système sur les états m et n .

Ceci traduit le fait que la probabilité de transition $P(m \rightarrow n)$ d'un état m à un autre état n du processus de Markov est toujours constante et devra satisfaire la relation de fermeture

$$P(m \rightarrow n) = 1$$

Dans la simulation de Monte Carlo, nous utiliserons à répétition le processus de Markov pour générer la chaîne de Markov de nouveaux états.

L'algorithme de Monte Carlo utilise le processus de Markov pour choisir les états utilisés. Il est généralement utilisé lorsqu'on veut partir de n'importe quel état du système et générer une suite de configurations de certains états précis. Pour être complet, le processus de Markov devra être ergodique.

Une chaîne de Markov est dite ergodique ou irréductible si tout état est atteignable depuis tout autre état. Elle est dite régulière s'il existe une puissance p^k de sa matrice de transition P dont tous les éléments sont strictement positifs. La particularité des chaînes régulières est que l'on peut aller de n'importe quel état vers n'importe quel autre état en un nombre fixé k de pas, où k est indépendant de l'état de départ. Pour les chaînes ergodiques, on demande simplement que tout état soit atteignable depuis tout autre, mais le nombre de pas n'est pas nécessairement fixé.

3.3 Énumération

En 1953, Nicholas Metropolis et al. [1] ont proposé une nouvelle procédure d'échantillonnage qui incorpore la température du système. Cela se fait en utilisant la loi de distribution de Boltzmann qui permet de facilement calculer la valeur moyenne d'un système. Cette méthode modifiée de Monte Carlo est connue sous l'appellation de Monte Carlo Metropolis. L'espace des configurations du système utilisé dans la méthode de Metropolis selon [5], est une séquence aléatoire d'états générés de leur état précédent (chaîne de Markov). La probabilité gouvernant la transition d'un état à l'autre se fait selon la différence d'énergie entre l'état initial et l'état final. Cette différence d'énergie peut se calculer en utilisant le modèle de Ising présenté précédemment.

Sur le modèle appliqué à la magnétisation :

- à $T = 0 \text{ K}$, le système est dans son état fondamental
- à $T \neq 0 \text{ K}$, par agitation thermique, le système passe par une succession d'états S qui, à l'équilibre, apparaissent avec la probabilité de Boltzmann :

$$P(S) = \frac{1}{Z} \exp\left(-\frac{E(S)}{k_B T}\right) \quad 3.1$$

Où E est l'énergie du système, k_B la constante de Boltzmann, T la température et Z est la fonction de partition. Pour un système à spectre d'énergie continu, la façon évidente de la généraliser est la suivante :

$$Z = \int e^{-\beta E} \rho(E) dE \quad 3.2$$

où $\beta = \frac{1}{k_B T}$,

L'algorithme de Metropolis peut être décrit comme il suit [5] :

(1) On part d'une configuration initiale (exemple : spin répartis aléatoirement)

(2) On choisit un spin k au hasard et on considère son renversement $S_k \rightarrow -S_k$

(3) On calcule la variation d'énergie ΔE correspondante

Si $\Delta E < 0$, on accepte le renversement

Si $\Delta E > 0$, on accepte avec une probabilité $P = \exp\left(-\frac{\Delta E}{k_B T}\right)$

(Cela veut dire qu'on tire un nombre aléatoire w entre 0 et 1

Si $w < p$, on accepte, sinon on refuse)

(4) On repart en (2) et ainsi de suite.

Le calcul de ΔE se fait localement avec

$$\Delta E = -2JS_K \sum_{\substack{j, \text{voisin} \\ \text{de } K}} S_j \quad 3.3$$

L'évaluation de la chaleur spécifique ainsi que de la magnétisation peut se faire avec les équations suivantes [2]:

$$c = \frac{k_B \beta^2}{N} \left(\langle E^2 \rangle - \langle E \rangle^2 \right) \quad 3.4$$

C , N , représentent respectivement, la chaleur spécifique, le nombre de spin ou nombre de site. Dans notre cas N sera égal à 10 000, donc si nous voulons parcourir tous les états possibles de notre modèle, nous devons faire au minimum $N \times N$ itérations. Ce chiffre sera plus élevé si le modèle devient complexe ou si la taille du composant augmente. Par exemple si nous avons une dimension de 50nm de longueur, nous nous retrouverons avec un nombre de site qui serait de 250 000 et un nombre d'itération de $62\,500 \times 10^6$.

3.4 Parallélisation de l'algorithme

En quoi consiste le calcul parallèle? Il sert faire coopérer plusieurs processeurs pour réaliser un calcul. Il présente certains avantages tels que la rapidité d'exécution et la taille de la mémoire. Pour un nombre N_p de processeurs, le temps de calcul sera en théorie divisé par $n=N_p$ et la taille de la mémoire sera en général n fois plus grande. Cependant certaines difficultés existent, la gestion du partage des tâches et la gestion de l'échange d'information pour les tâches non indépendantes [100].

Nous ne pouvons parler de parallélisation sans jeter un coup d'œil sur les différents modèles possibles (matériels et logiciels).

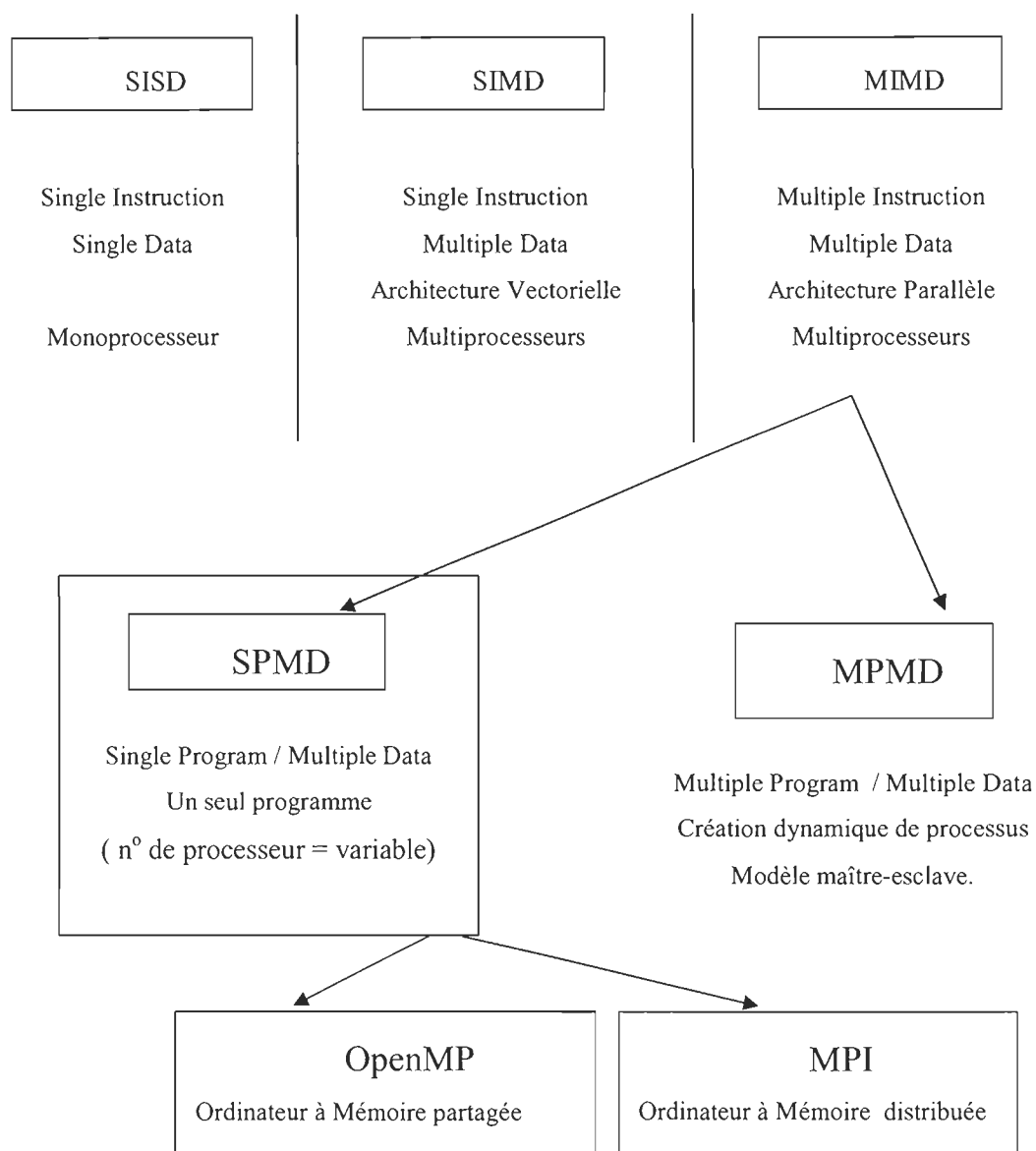


Figure 3-2 Infrastructure d'implémentation [100]

Nous présentons ci-dessous un aperçu des architectures à mémoire partagée et à mémoire distribuée.

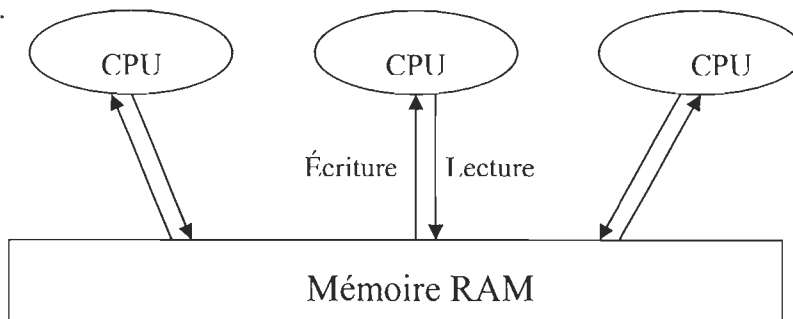


Figure 3-3 Mémoire partagée [100]

Dans une architecture à mémoire partagée, tous les processeurs ont accès à l'ensemble de la mémoire. Ils peuvent donc y lire et écrire simultanément des données, cela s'avère très important car il permet de sauver du temps (le temps lié à la communication entre processeurs). Cependant une attention particulière doit être portée aux conflits entre différentes tâches. Les architectures à mémoires partagées sont coûteuses.

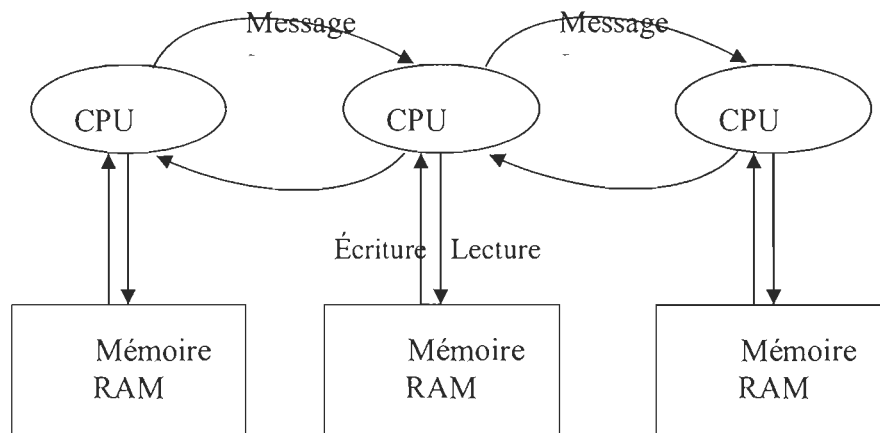


Figure 3-4 Mémoire distribuée [100]

Dans les architectures à mémoire distribuée, chaque processeur possède sa propre mémoire, aucun autre processeur ne peut accéder à ces données. Cette configuration nécessite absolument une communication inter processeur, pour des applications qui nécessitent un échange d'information sur les données. Il faudra donc être capable de gérer l'échange de messages, cela pourrait engendrer des surcoûts. Contrairement aux architectures à mémoire partagée, c'est une architecture bon marché avec un nombre de processeurs pouvant être illimité.

Ces dernières années les outils permettant de supporter la programmation parallèle sur plusieurs processeurs ont beaucoup évolué. La parallélisation peut être effectuée à divers niveaux:

Langages:

- Compositionnel C++ (CC++)
- Fortran M
- High performance Fortran (HPF)

Bibliothèques:

- Message Passing Interface (MPI)
- Parallel Virtual Machine (PVM) : obsolète

Directives de compilation:

- OpenMP
- Instructions spécifiques constructeur.

Compilateurs:

- Intel Fortran/C compilé : gratuit.

Les deux standards les plus utilisés sont le MPI et l'OpenMP, le MPI étant dédié aux architectures à mémoire distribuée et l'OpenMP aux architectures à mémoire partagée.

3.4.1 L'OpenMp

L'OpenMP est un ensemble de directives de compilation pour paralléliser un code sur une architecture SMP (interfaces Fortran, C et C++).

Le compilateur interprète les directives OpenMP.

Modèles d'exécution OpenMP:

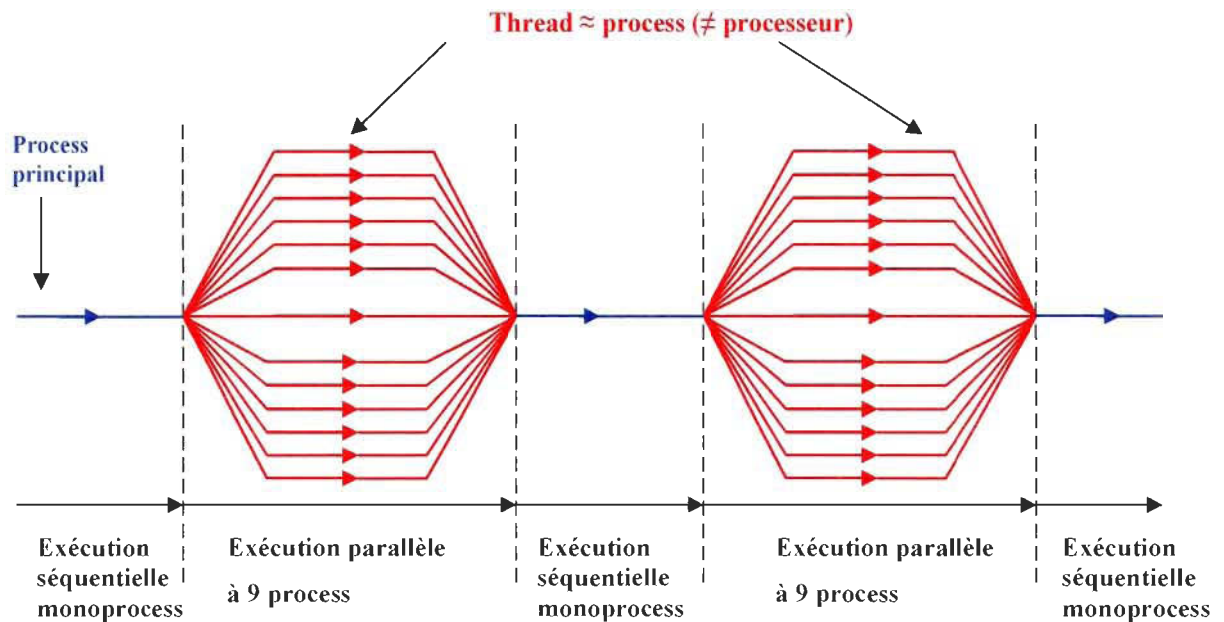


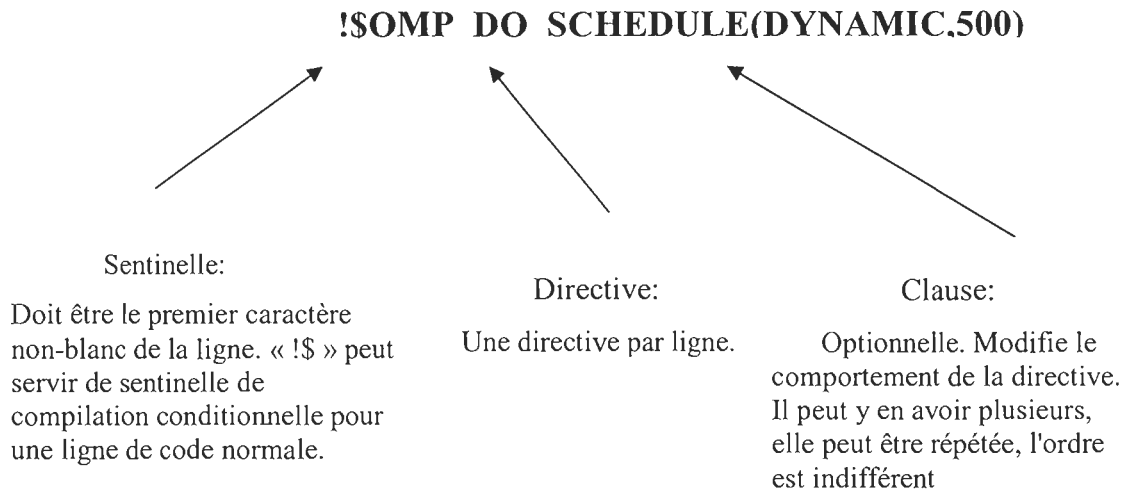
Figure 3-5 Processus d'exécution OpenMP

Voici quelques points de repère pour faire fonctionner un code OpenMP :

- Écrire le code séquentiel (tout ou partie)
- Déboguer
- Se placer sur une machine SMP multiprocesseur
- Insérer les directives OpenMP
- Compiler avec l'option appropriée
- Lancer le programme
- Déboguer
- Évaluer les performances, en particulier le "speedup" :
 - Temps exécution N procs/ temps exécution 1 proc

Exemples de syntaxe d'une directive OpenMP :

Pour f90 :



Pour C/C++ :

```
#pragma_omp_for_schedule(dynamic,500)
```

3.4.2 Le Standard MPI

Le MPI, est une norme définissant une bibliothèque de fonctions, utilisable avec les langages C/C++ et Fortran. Il permet d'exploiter des ordinateurs distants ou multiprocesseurs par passage de messages. Il est devenu *de facto* un standard de la communication pour des nœuds exécutant des programmes parallèles sur des systèmes à mémoire distribuée. Le MPI a été écrit pour obtenir de bonnes performances aussi bien sur des machines massivement parallèles à mémoire partagée que sur des clusters d'ordinateurs hétérogènes à mémoire distribuée. Il est grandement disponible sur de très nombreux matériels et systèmes d'exploitation. Ainsi, MPI possède l'avantage par rapport aux plus vieilles bibliothèques de passage de messages d'être grandement portable (car MPI a été implanté sur presque toutes les architectures de mémoires) et rapide (car chaque implantation a été optimisée pour le matériel sur lequel il s'exécute).

Avec le MPI, on définit une seule zone parallèle dans le programme, souvent l'ensemble du code. Les parties du code avant et après la zone parallèle sont locales sur chaque processeur donc pas de communications possible.

Dans chaque programme MPI, il faut inclure la bibliothèque MPI en utilisant :

Mpif.h en fortran et mpi.h en C/C++.

Le début de la zone parallèle se fait par :

- call MPI_INIT(err) en fortran,
- MPI_Init(); en C et
- MPI::Init(); en C++

Fonction collective (appelée par tous les process)

La fin de la zone parallèle se fait par :

- call MPI_FINALIZE(err) en Fortran,
- MPI_Finalize (); en C et
- MPI::Finalize (); en C++

Fonction collective (appelée par tous les process)

3.4.3 *Techniques de création d'algorithme parallèle*

Il existe trois classes primaires de technique de création d'algorithme parallèle [2], à savoir, la parallélisation triviale, la décomposition de fonction et la décomposition de domaine.

La technique triviale ne s'applique qu'à des problèmes qui peuvent être séparés en des tâches complètement indépendantes pouvant être exécutées par des processeurs isolés ne nécessitant aucune communication inter processeur.

Les deux autres techniques s'adressent à des problèmes beaucoup plus compliqués. La technique de décomposition de fonction quant à elle, permet de diviser une tâche en d'autres beaucoup plus simples, telle l'évaluation d'une fonction complexe. Chaque unité de calcul va assurer un type de calcul différent sur l'ensemble des données.

En ce qui concerne la décomposition de domaine, elle permet au programme de se concentrer sur différente région du système simulé. Toutes les unités de calcul effectuent les mêmes calculs sur la partie de données qui leur est attribuée. L'accès aux données des autres unités de calcul se fait par échange de messages.

Une décomposition idéale serait qu'il n'y ait aucun besoin de communication entre les différents domaines. Ce serait des domaines statiques à charge constante et équilibrée. Cela n'est pas du parallélisme. Dans une décomposition satisfaisante, le temps de communication est inférieur au temps de calcul. Le temps de redéfinition dynamique des domaines est aussi inférieur au temps de calcul.

La possibilité et efficacité d'une décomposition en domaines dépendent du problème-algorithme. La simulation de Monte-Carlo pour certains problèmes avec équilibrage de charge simple serait un exemple de décomposition simple et efficace.

Il y'a deux moyens bien connus de décomposer un modèle 2D pour le traitement parallèle à savoir, la décomposition par bande et la décomposition en block.

Pour un modèle 2D de dimension $S \times S$ et un nombre P de processeurs, allant de P_0, P_1, \dots, P_{p-1} , la méthode de décomposition par block est définie comme la partition du modèle en P sous-domaines contiguës de taille chacun $\frac{S}{\sqrt{P}} \times \frac{S}{\sqrt{P}}$, avec le processeur P_i assigné au site (j,k) pour :

$$(i \bmod \sqrt{P}) \cdot \frac{S}{\sqrt{P}} \leq j < (i \bmod \sqrt{P} + 1) \cdot \frac{S}{\sqrt{P}} \quad 3.5$$

et

$$(\lfloor i / \sqrt{P} \rfloor) \cdot \frac{S}{\sqrt{P}} \leq k < (\lfloor i / \sqrt{P} \rfloor + 1) \cdot \frac{S}{\sqrt{P}} \quad 3.6$$

La figure 3-6 présente un exemple de la décomposition par block pour 6 processeurs.

0	2	4
1	3	5

Figure 3-6 Décomposition par block [RP]

Pour le même modèle une décomposition par bande serait, la partition du modèle en P sous-domaines contiguës de taille chacun $\frac{S}{P} \times S$ ou $(S \times \frac{S}{P})$, avec le processeur P_i assigné au site (j, k) pour $0 \leq j < S$ avec

$$i. \frac{S}{P} \leq j < (i+1). \frac{S}{P} \quad 3.7$$

et

$$i. \frac{S}{P} \leq k < (i+1). \frac{S}{P} \quad 3.8$$

(ou $0 \leq k < S$).

La figure 3-7 présente un exemple de la décomposition par bande pour six processeurs.

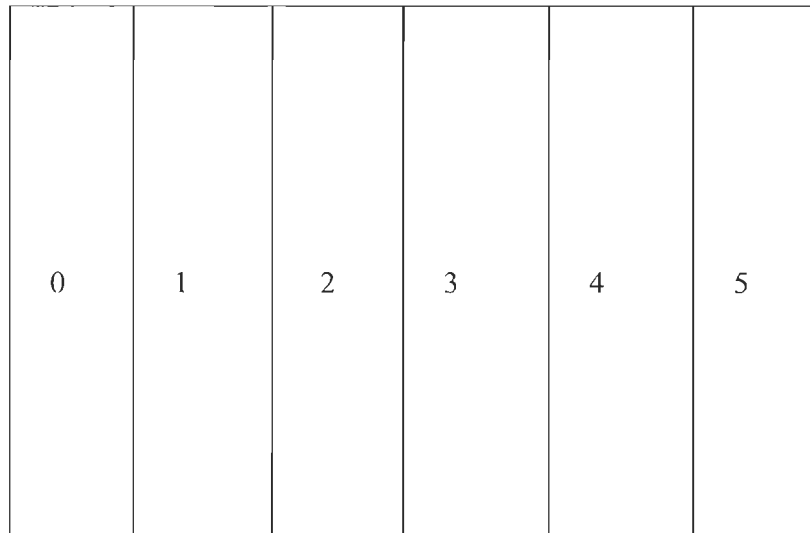


Figure 3-7 Décomposition en bande [RP]

La décomposition peut aussi se faire de façon horizontale.

Sur une architecture à mémoire distribuée, les unités de calculs doivent échanger des messages. Le coût dépend du réseau de communication et de l'implémentation MPI. Selon [95] il vaut mieux envoyer un gros message que plein de petits messages. Il faut donc grouper les communications.

Les deux techniques énoncées ci-dessus ont des modes de communication différents, figure 3-8 nous donne un aperçu du transfert de données dans le cas d'une décomposition en bande alors que la figure 3-9 nous montre le transfert de données dans le cas d'une décomposition en block.

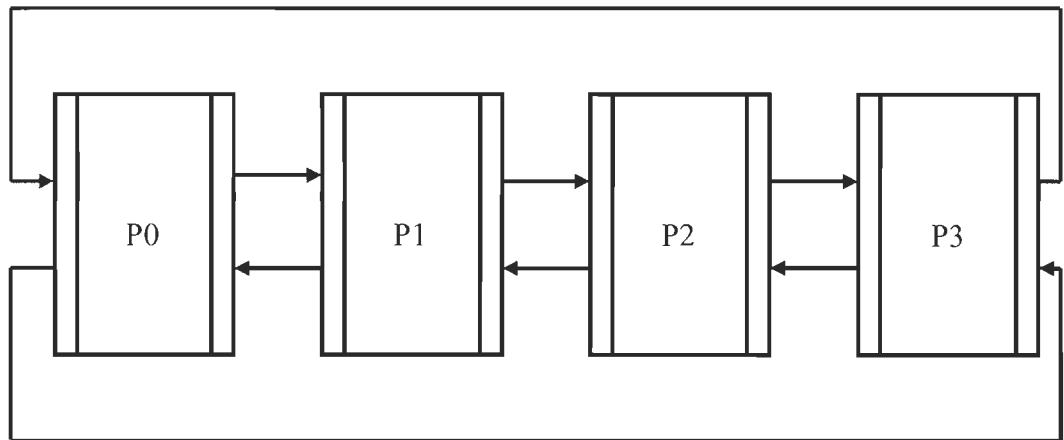


Figure 3-8 Diagramme de communication pour une décomposition par bande [RP]

Après chaque pas de Monte Carlo, les processeurs échangent des données afin de mettre à jour leur valeur frontière. Dans le cas du modèle d'Ising, les spins étant influencés par l'état de leurs voisins, il faudra alors échanger entre processeurs voisins, les états des spins à la frontière du sous-domaine.

Dans la décomposition par bande, chaque processeur échange une fois avec son voisin lors de la mise à jour des données après une itération. Il ne peut qu'avoir un maximum de deux échanges dans le réseau quelque soit le nombre de processeur. Alors que dans la décomposition par block (figure ci-dessous), chaque processeur échange deux fois avec son voisin, le nombre de communication est dans ce cas deux fois plus que dans le cas précédent. Cependant, le volume de données transféré dans le cas de la décomposition par bande, est beaucoup plus élevé.

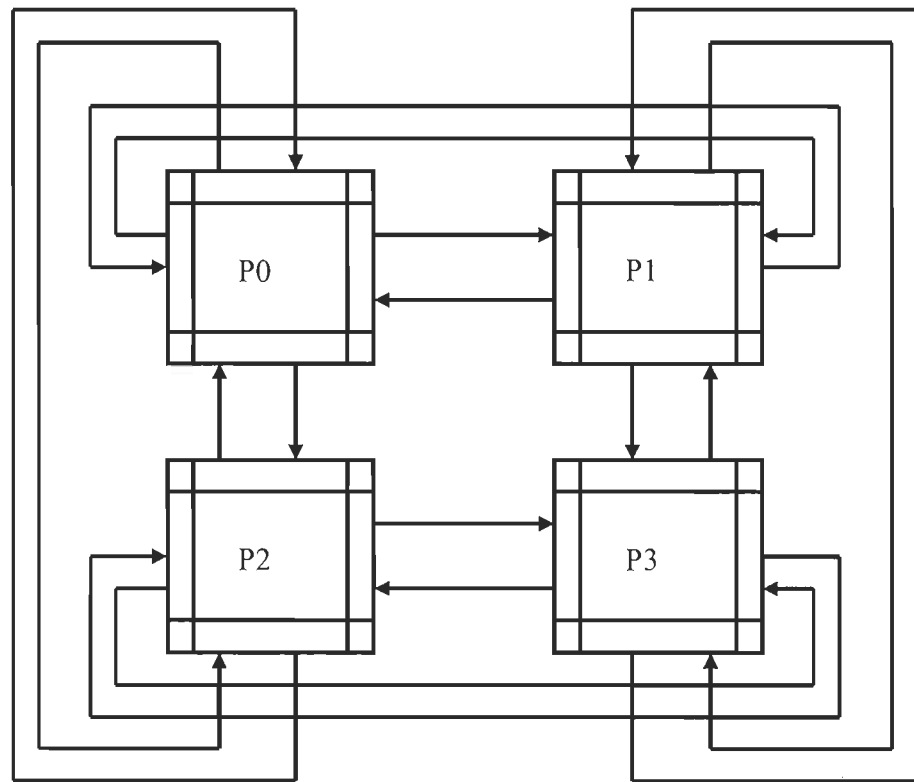


Figure 3-9 Diagramme de communication pour une décomposition par block [RP]

3.4.3.1 Application au modèle d'Ising

Le fait de paralléliser l'algorithme peut permettre d'inverser plusieurs spins simultanément. La condition de balance implique que des spins voisins ne peuvent être

inversés au même moment. Donc il faudra en tenir compte lors de la parallélisation de l'algorithme de Métropolis.

Pour ce faire, les processeurs devraient éviter d'inverser les spins à leur (sous-domaine) frontières commune de façon simultanée. La solution proposée dans [95] est de diviser une étape d'itération en 2 phases.

À la fin de chaque étape, la communication et la synchronisation sera effectuée entre les processeurs voisins. Au début de chaque étape, les données nécessaires pour l'exécution de ce stade doivent être disponibles pour le processeur.

Lors de la repartitions des données, chaque processeur reçoit des données supplémentaires des processeurs voisins. Ces données sont essentiellement des données limites du sous-domaine voisin. La figure 3-10 donne un aperçu de cette pratique pour deux processeurs voisins.

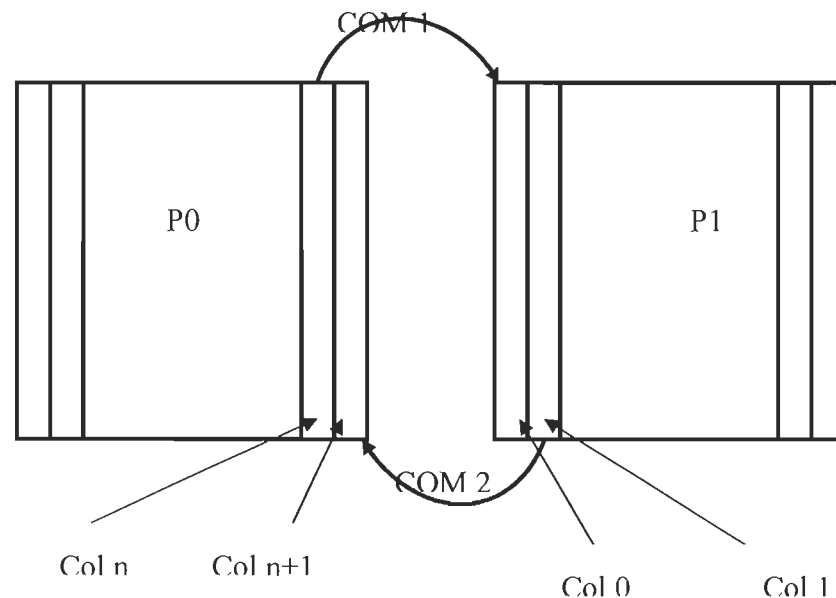


Figure 3-10 Interactions entre processeurs voisins [RP]

Dans la configuration présentée à la figure 3-10, $Col\ n$ (colonne n) contient les derniers éléments du sous domaine traité par le processeur P_0 et $Col\ n+1$ (colonne $n+1$) contient les premiers éléments du sous-domaine traité par le processeur P_1 . Alors que $Col\ 0$ (colonne 0) contient les derniers éléments du sous-domaine traité par le processeur P_0 et $Col\ 1$ (colonne 1) contient les premiers éléments du sous-domaine traité par le processeur P_1 . Après chaque itération, les données sont mises à jour à travers des messages représentés sur la figure par $COM\ 1$ et $COM\ 2$. Ces données sont nécessaires pour que les processeurs locaux calculent la variation d'énergie due à l'inversion des spins voisins (localisé dans le sous domaine voisin).

En raison de la condition de balance, une fois qu'un processeur P_i envoie ses données limites à ses voisins, il ne devrait plus les remettre à jour, à moins que ses voisins ne soient plus en train d'utiliser ces données. Cela conduit à la nécessité d'une synchronisation et de la communication. Nous décrivons ci-dessous la démarche liée à la décomposition par bande.

Description de la démarche liée à la décomposition par bande

Prenons un modèle 2D de dimension $S \times S$ reparté sur 4 processeurs P_0, P_1, P_2, P_3 . Chaque processeur reçoit une portion de $S/4$ lignes comme montrées sur la figure ci-dessous. Le fait que chacun des processeurs ait seulement deux voisins simplifie énormément le problème.

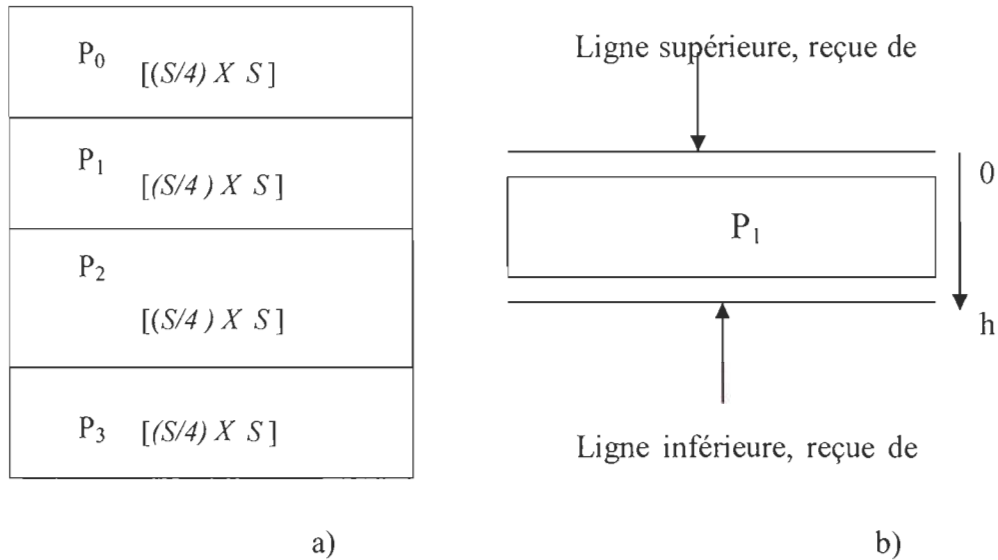


Figure 3-11 : Répartition des données de calcul pour une décomposition en bande [RP]

La figure 3-11 a) représente la répartition des données de calcul pour chaque processeur. La figure 3-11 b) présente l'ajout de deux lignes supplémentaires à chaque matrice de donnée.

Comme mentionné plus haut, pour respecter la condition de balance lors de la parallélisation de l'algorithme, une étape d'itération de Monte Carlo doit se faire en deux phases afin que des spins voisins ne soient pas inversés simultanément.

Nous prenons le processeur P_1 à titre d'exemple (tous les processeurs effectuent la même démarche).

Au cours de la phase 1, le processeur P_1 met localement à jour les lignes à 1 ($h-2$) dans son propre bloc. Après la mise à jour locale, elle enverra les données de la ligne 1 à son voisin supérieur (P_0), tout en recevant les données de son voisin inférieur P_2 , et stocke les données reçues dans la ligne frontalière (ligne basse) de la matrice.

Au cours de la phase 2, le processeur P1 met localement à jour la ligne du bas (h-1). Après la mise à jour locales, il envoie les données de la ligne du bas à son voisin inférieur qui est P2, tout en recevant les données de son voisin supérieur P0, et stocke les données reçues dans la lignes frontière (ligne haute) de la matrice.

Phase 1

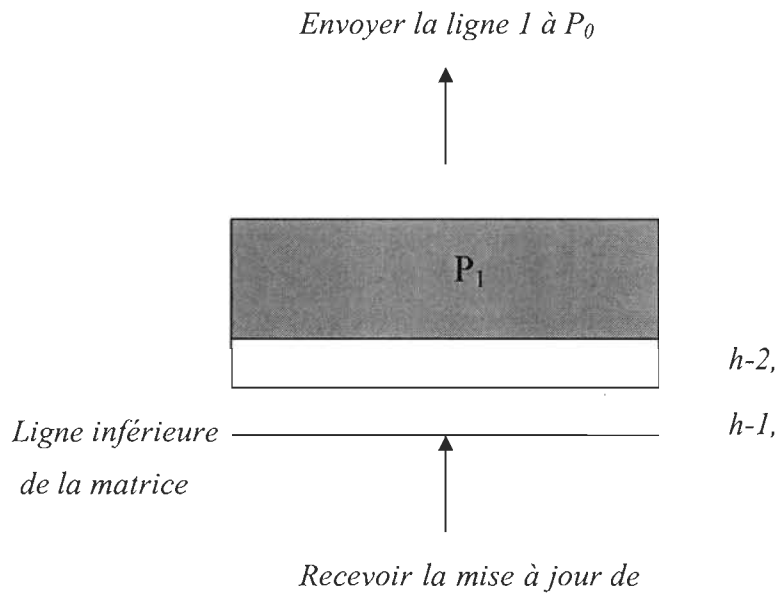


Figure 3-12 Transfert des mises à jour phase 1 [RP]

La partie ombragée de la figure 3-12 représente la région dans laquelle P0 exécute sa mise à jour locale. C'est dans l'ensemble $\{(1, 0), (h-2, S-1)\}$

Phase 2

Sur la figure 3-13, P1 exécute sa mise à jour locale sur la dernière ligne des données. C'est dans l'ensemble $\{(h-1, 0), (h-1, S-1)\}$.

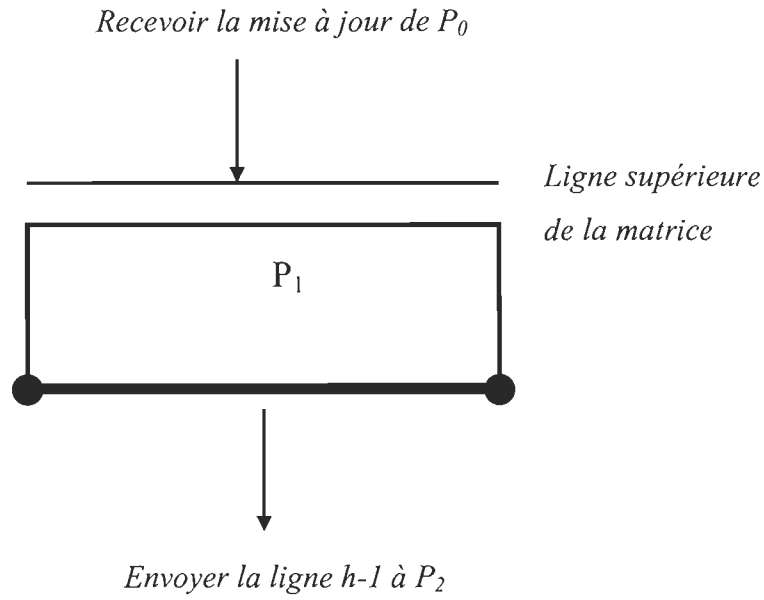


Figure 3-13 Transfert des mises à jour phase 2 [RP]

3.4.3.2 Organisation générale de l'algorithme

Nous présentons ci-dessous, sans toute fois rentrer dans les détails, une organisation générale de notre algorithme avec les grandes lignes.

Initialisation

- Caractéristiques (*in*)
 - Matériaux (*in*)
 - Potentiel d'interaction (*in*)
 - Nombre de particules (*in*)
- Géométrie (*in*)
 - Complexité (*in*)

Décomposition

- Bande
 - Nombre de processeurs disponibles (*in*)
 - Nombre de domaines (*out*)
 - Nombre de sous domaines (*out*)
 - Charge des nœuds (*condition*)
 - Équilibrage des nœuds (*condition*)
 - Zones tampons (*out*)

Monte Carlo

- Domaine (*in*)
- Pas de calcul Monte Carlo (*in*)
- Mise à jour (*in*)
- Synchronisation (*condition*)

Calcul

- Équilibre thermique (*condition*)
- Énergie (*out*)
- Chaleur spécifique (*out*)
- Conductivité thermique (*out*)

Résultats

- Partiels (*in*)
- Total (*out*)

Chapitre 4 - Implémentation de l'algorithme

La problématique principale de notre travail comme citée dans le chapitre 1, est de permettre une bonne évaluation du comportement thermique de nanosystème, dans un environnement de simulation nous permettant de réaliser un gain de temps et une facilité de stockage des données de calcul.

Le comportement thermique le plus rependu dans la littérature et de loin le plus significatif dans la caractérisation thermique d'un système, est la conductivité thermique. L'évaluation de ce comportement passe par la connaissance de certaines grandeurs physiques, à savoir la capacité thermique du système, l'énergie globale du système ainsi que la densité des particules du système. Nous présentons ci-dessous l'environnement de simulation utilisé, ainsi que la conductivité thermique.

4.1 L'environnement de CLUMEQ

Le CLUMEQ fait partie des réseaux du calcul à haute performance du Canada. C'est un environnement qui possède plusieurs machines (comme présenté au chapitre 1). Le nœud d'entré de l'environnement est la machine Stokes (stokes.clumeq.mcgill.ca), elle est entièrement dédiée à cette tâche. Cependant elle peut permettre de voir les travaux en cours d'exécution et ceux dans la file d'attente. Cela permet de savoir l'état de notre application qui peut être soit en cours d'exécution, en attente ou supprimé.

Elle permet de voir le temps d'exécution en cours. La connexion à Stokes se fait par le protocole de communication SSH (Secur Shell). Le SSH permet aussi le transfert de nos données vers Stokes, et de Stokes vers notre poste de travail. La facilité d'utilisation de ce protocole nous permet de porter toute notre attention sur notre application. Par les figures ci-dessous nous montrons la démarche pour se connecter au nœud Stokes de CLUMEQ.

Étape 1

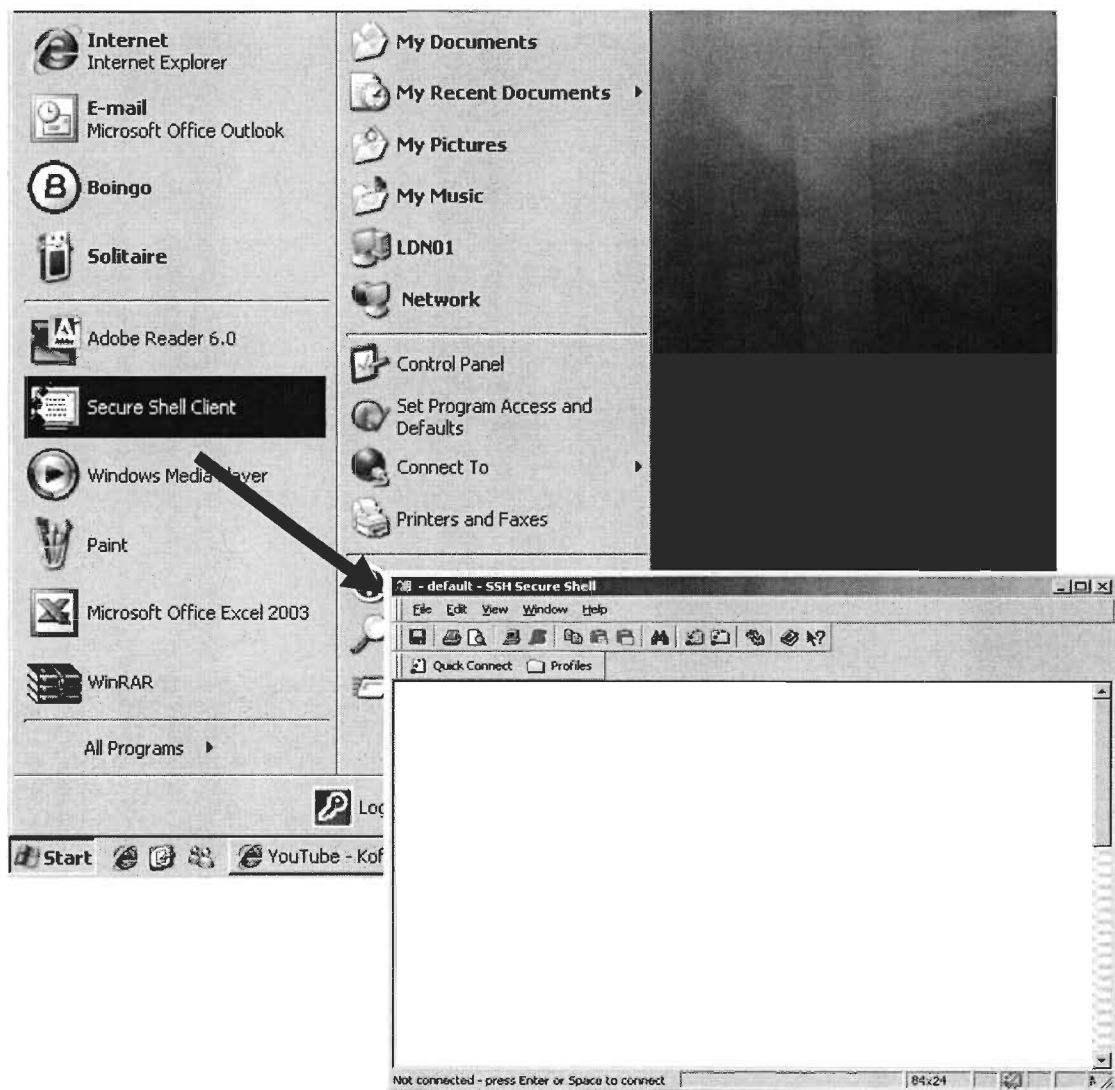


Figure 4-1 Démarrage de la connexion à CLUMEQ en utilisant le SSH [RP]

La figure 4-1 présente l'étape 1, à cette étape nous allons tout simplement chercher le SSH (précédemment installé) sur notre poste de travail.

Étape 2

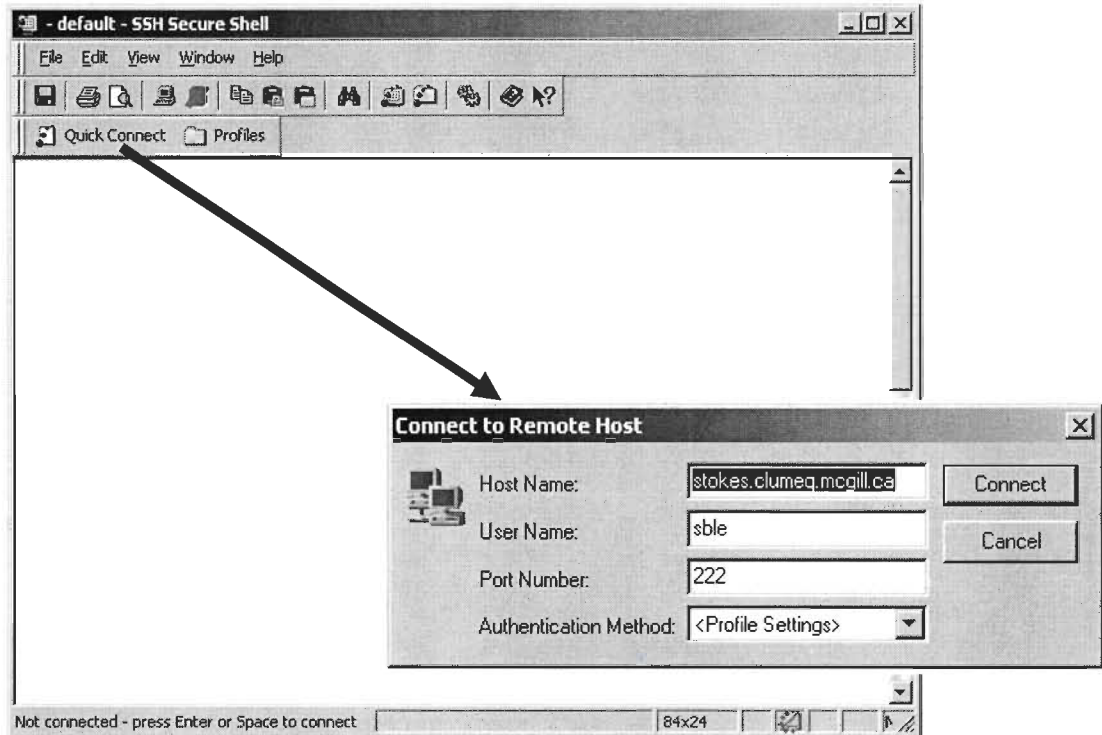


Figure 4-2 Saisie des paramètres de connexion [RP]

Le “*shell*” étant ouvert à partir de l'étape 1, nous pouvons maintenant nous connecter à stokes. Dans l'étape 2 présentée par la figure 4-2, comme montré sur la figure, nous avons utilisé l'option “*Quick Connect*”. Pour le faire, nous devrions auparavant saisir dans le “*Profiles*” le nom du serveur auquel nous voulons nous connecter. Dans notre cas, le nom du serveur d'entrée sur CLUMEQ est “*stokes.clumeq.mcgill.ca*”. Nous rentrons ensuite notre nom d'utilisateur ainsi que le numéro du port de connexion (222).

Étape 3

La connexion à Stokes, nécessite un mot de passe comme montré sur la figure 4-3. Lorsque nous finissons d'entrer les informations requises pour l'étape 2 et que nous lançons la connexion, une fenêtre nous apparaît pour la saisie de notre mot de passe. Une fois cela complété, nous pouvons accéder au nœud d'entrée de CLUMEQ.

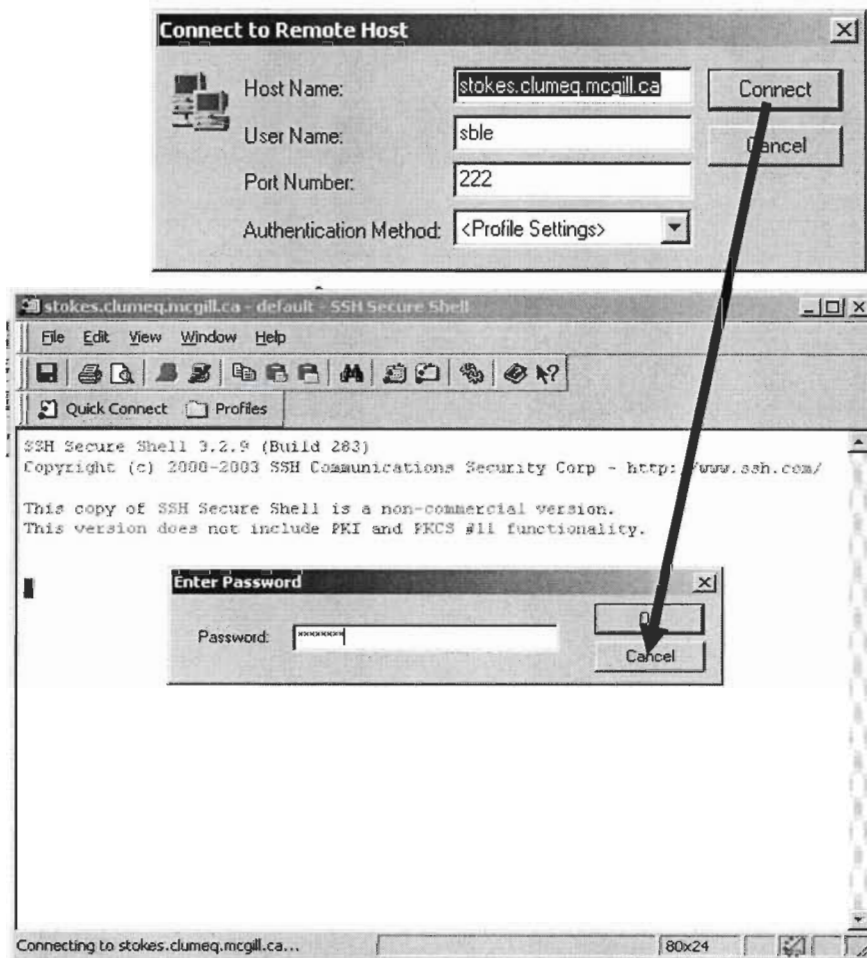


Figure 4-3 Saisie du mot de passe [RP]

Étape 4

Sur la figure 4-4, la connexion à CLUMEQ (par Stokes) est effective. Le nœud Stokes n'étant pas un nœud de soumission, nous devons nous rediriger vers un autre nœud pour soumettre (compiler, déboguer et exécuter) notre application. Le nœud "hn" est celui qui nous permettra de soumettre notre application. Pour y accéder, nous utilisons toujours le protocole de communication SSH, comme montré sur la figure 4-5.

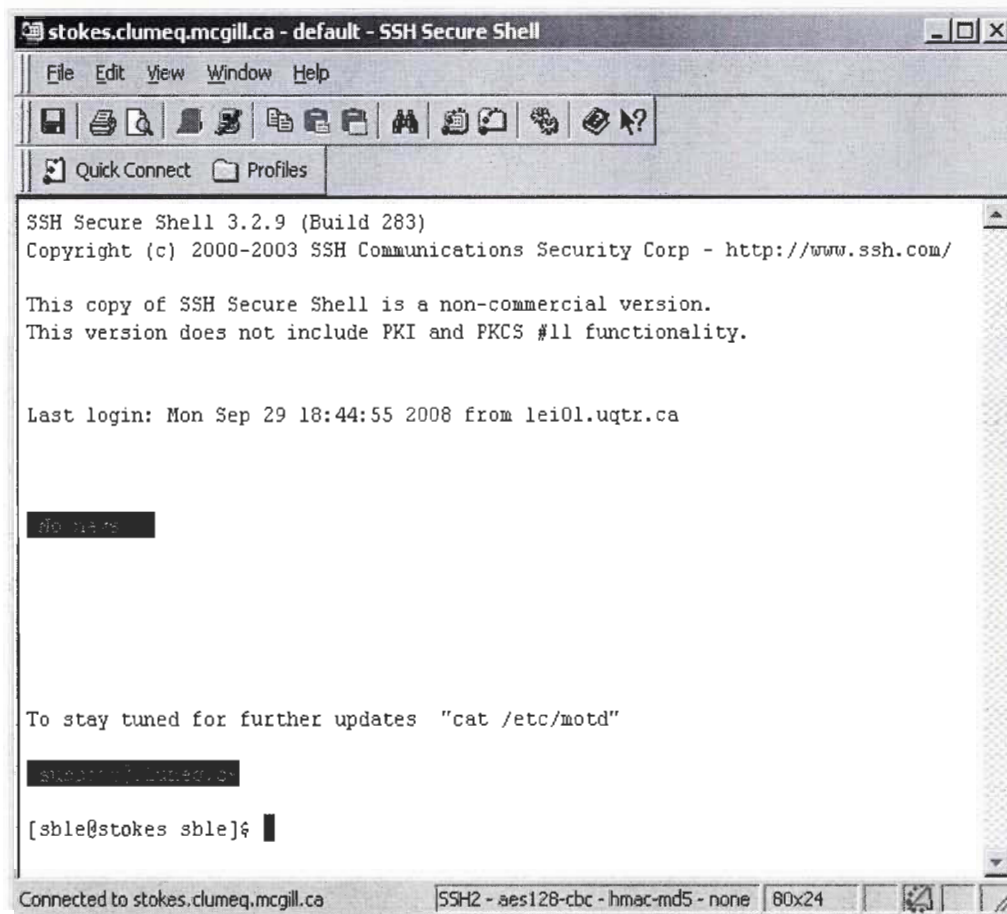


Figure 4-4 Accès au nœud d'entrée (Stokes) [RP]

Étape 5

À l'étape 5 nous redirigeons vers le nœud *“hn”*, par la commande *“ssh”*. Un mot de passe est aussi requis, cependant nous utilisons le même mot de passe.

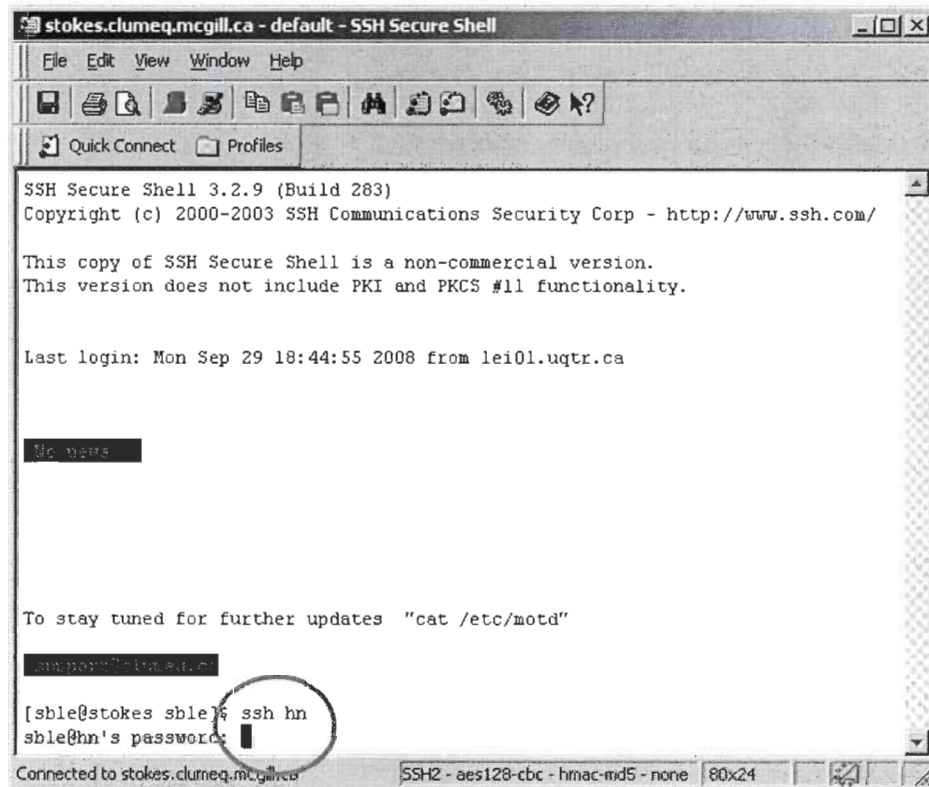


Figure 4-5: Passage du nœud d'entrée Stokes au nœud de soumission hn [RP]

Dans la figure 4-6, nous nous sommes pris à deux reprises pour accéder au nœud *“hn”*. Le premier essai ayant échoué parce que le mot de passe n'était pas le bon. Le deuxième essai a été concluant, à l'étape 5 nous avons donc accédé au nœud *“hn”*. Nous avons tapé la commande *“qstat”* pour lister toutes les applications encore d'exécution ou dans la file d'attente. Nous pouvons y voir, le numéro d'identification de l'application, le nom de l'application, le nom de l'utilisateur, le statut de l'application

(en cours d'exécution "R" ou en attente "Q") ainsi que le type d'application (multiprocesseur "multi" ou un processeur "single").

The screenshot shows an SSH terminal window titled "stokes.clumeq.mcgill.ca - default - SSH Secure Shell". The window has a menu bar (File, Edit, View, Window, Help) and a toolbar. A red circle highlights the "Repertoire de l'utilisateur" icon in the toolbar. The terminal output shows the user logging in, receiving a "No news" message, and being prompted to stay tuned for updates. The user then attempts to connect to a host named "hn" using the command "ssh hn". This attempt fails with the message "Permission denied, please try again." and is circled in red, with an arrow pointing to the text "Premier essai". The user then runs the command "qstat", which is also circled in red, with an arrow pointing to the text "Deuxième essai". The output of "qstat" is a table showing system jobs.

```

stokes.clumeq.mcgill.ca - default - SSH Secure Shell
File Edit View Window Help
Repertoire de l'utilisateur
Quick Connect Profiles

SSH Secure Shell 3.2.9 (Build 283)
Copyright (c) 2000-2003 SSH Communications Security Corp - http://www.ssh.com/

This copy of SSH Secure Shell is a non-commercial version.
This version does not include PKI and PKCS #11 functionality.

Last login: Mon Sep 29 18:44:55 2008 from lei01.uqtr.ca

No news

To stay tuned for further updates "cat /etc/motd"
support@clumeq.mcgill.ca

[sble@stokes sble]$ ssh hn
sble@hn's password:
Permission denied, please try again.
sble@hn's password:

No news

To stay tuned for further updates "cat /etc/motd"
support@clumeq.mcgill.ca

[sble@hn sble]$ qstat
Job id      Name          User          Time Use S Queue
-----
290556.stokes  protMem3rdstep  elvis         00:00:24 Q multi
290955.stokes  script.x        mathieug      09:35:39 R single
290956.stokes  script.x        mathieug      09:35:44 R single
290957.stokes  script.x        mathieug      09:35:43 R single
290958.stokes  script.x        mathieug      09:35:30 R single
290959.stokes  script.x        mathieug      09:34:22 R single

```

Figure 4-6 Connexion effective au nœud hn [RP]

Étape 6

Étant maintenant sur le nœud “*hn*”, nous pouvons soumettre notre application. Mais avant faudrait que notre application soit mise dans le répertoire qui nous ai alloué sur stokes. Nous utilisons toujours le protocole SSH, pour le transfère des données.

La figure 4-7 nous montre le répertoire ainsi que certains fichiers nécessaires à la compilation et à l’exécution de notre application. Sans toutefois les nommer tous, nous présenterons quelques uns d’entre eux.

L’item “*Application*” correspond au fichier contenant l’application que nous désirons exécuter. Dans le cas présent, le fichier application est “*projet4_N1.c*”. Toutes les modifications au fichier peuvent ce faire directement dans le répertoire utilisateur. La compilation peut aussi se faire directement sur le nœud “*hn*”, avec la commande suivante : “*mpicc projet4_N1.c -lm*”, dans le cas présent l’instruction “*mpicc*” est utilisée pour compiler les applications en C et C++, utilisant le standard MPI. Le nœud “*hn*” nous offre aussi la possibilité de compiler des applications en Fortran ainsi que des applications qui utilisent le standard OpenMP.

La figure 4-8 nous montre deux exemples de compilation, une réussie et une non réussie. Pour une compilation non réussie, les erreurs nous apparaissent tout de suite après l’instruction de compilation. Alors que dans le cas d’une compilation réussie, rien n’apparaît, le curseur est prêt à prendre une autre commande. Après la compilation, le fichier “*a.out*” (voir *fichier machine* sur la figure 4-7) est crée pour les instructions machines.

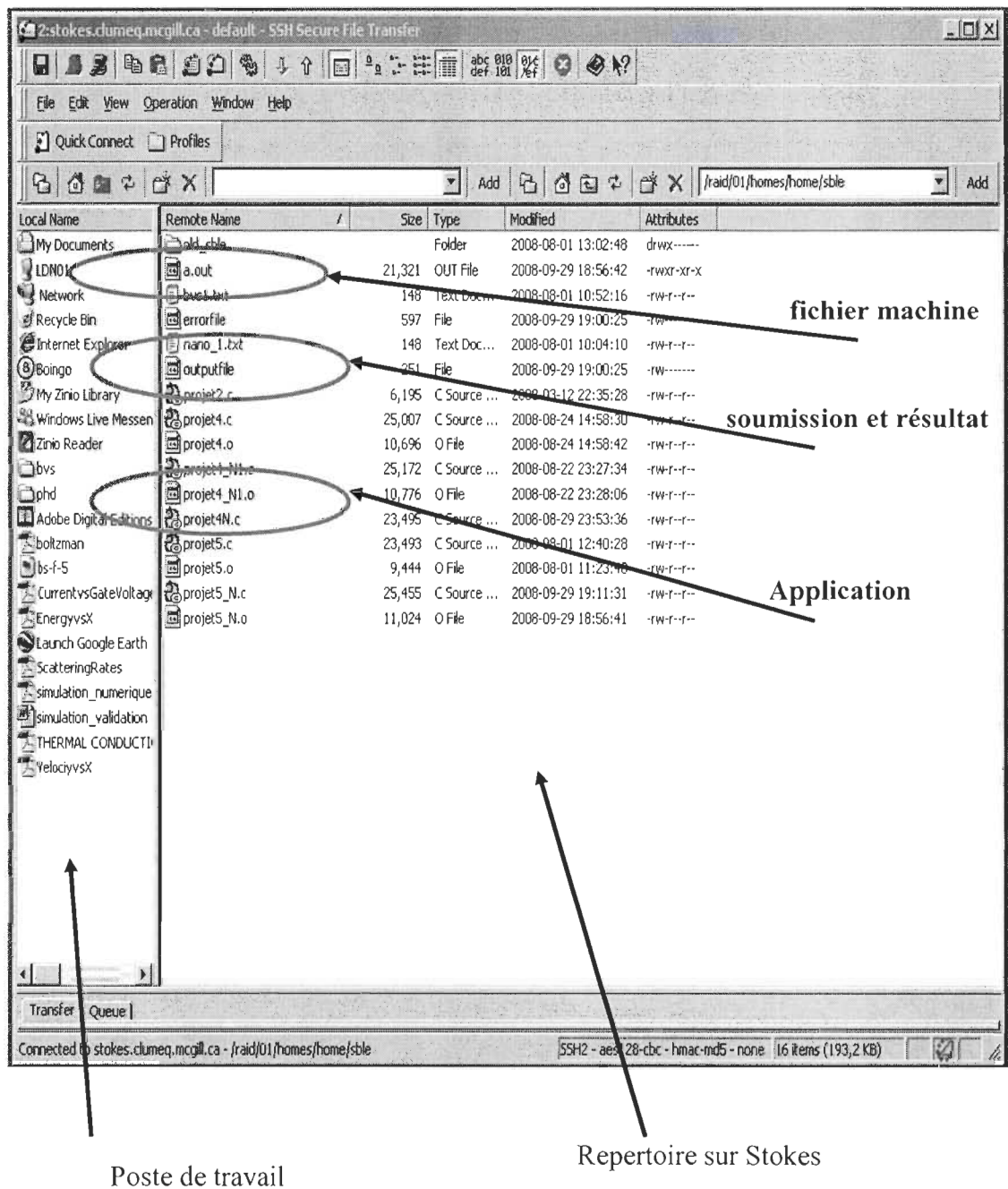


Figure 4-7 Répertoire utilisateur sur Stokes [RP]

Pour exécuter notre application, il nous faut d'abord créer un fichier d'instruction PBS (voir *soumission* sur la figure 4-7) dans lequel nous passerons un certains nombre d'information

concernant le type d'application à savoir, le nombre de processeur à utiliser pour l'exécution, le standard à utiliser ainsi que le chemin du fichier machine. La figure 4-9, nous montre un exemple de fichier de soumission.

```

1:stokes.clumeq.mcgill.ca - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles

Permission denied, please try again.
sble@hn's password:

Welcome to Stokes

To stay tuned for further updates "cat /etc/motd"
support@clumeq.ca

[sble@hn sble]$ mpicc projet5_N.c -lm
projet5_N.c: In function `main':
projet5_N.c:377: parse error before "printf"
[sble@hn sble]$
[sble@hn sble]$
[sble@hn sble]$ mpicc projet5_N.c -lm
[sble@hn sble]$

```

Connected to stokes.clumeq.mcgill.ca SSH2 - aes128-cbc - hmac-md5 - none 80x24

Figure 4-8 Exemple de compilation sur le nœud “hn” [RP].

La section encerclée en pointillé (rouge) sur la figure 4-8 correspond à la compilation non réussie, après les corrections nous avons à nouveau lancé la compilation. La section encerclée en trait plein (vert) correspond à la compilation réussie.

```

nano_1 - Notepad
File Edit Format View Help
#PBS -l mem=256mb
#PBS -l nodes=2:ppn=2 ← 1
#PBS -o outputfile ← 2
#PBS -e errorfile ← 3
#PBS -V
mpirun -np 4 -machinefile $PBS_NODEFILE /home2/sble/a.out

```

The image shows a nano_1 - Notepad window with a submission file. The file content is as follows:

```

#PBS -l mem=256mb
#PBS -l nodes=2:ppn=2 ← 1
#PBS -o outputfile ← 2
#PBS -e errorfile ← 3
#PBS -V
mpirun -np 4 -machinefile $PBS_NODEFILE /home2/sble/a.out

```

Hand-drawn annotations in the image include:

- Arrows pointing from the numbers 1, 2, and 3 to the lines: `#PBS -l nodes=2:ppn=2`, `#PBS -o outputfile`, and `#PBS -e errorfile` respectively.
- Arrows pointing from the numbers 4, 5, and 6 to the arguments of the `mpirun` command: `-np 4`, `-machinefile`, and `$PBS_NODEFILE` respectively.
- A bracket under the `-machinefile $PBS_NODEFILE /home2/sble/a.out` part of the `mpirun` command, with an arrow pointing from the number 6 to it.

Figure 4-9 Fichier de soumission pour l'exécution [RP]

Sur la figure 4-9 nous montrons les éléments importants pour l'exécution d'une application préalablement compilée dont le fichier machine se trouve être "*a.out*". L'item 1 correspond à la déclaration PBS du nombre de processeur utilisé pour l'exécution de l'application.

"*nodes=2:ppn=2*", dans cette instruction, "*nodes=2*" correspond au nombre de nœud de calcul et "*ppn=2*" correspond au nombre de processeur pour chaque nœud. Cela donne dans le cas présent un total de 4 processeurs, donc l'application ici pourra être décomposée en quatre tâches distinctes lors de son exécution.

L'item 2 "*outputfile*", correspond à la déclaration PBS du fichier des résultats de simulation.

L'item 3 "*errorfile*", correspond à la déclaration PBS du fichier généré en cas d'erreur lors de la simulation.

L'item 4 "*mpirun*", correspond à la commande d'exécution du standard MPI. Le 5, "*np 4*" est le nombre de processeur et le 6, "*machinefile \$PBS_NODEFILE /home2/sble/a.out*" est le chemin d'accès au fichier machine. Donc l'item 5 et 6 sont les paramètres de l'item 4.

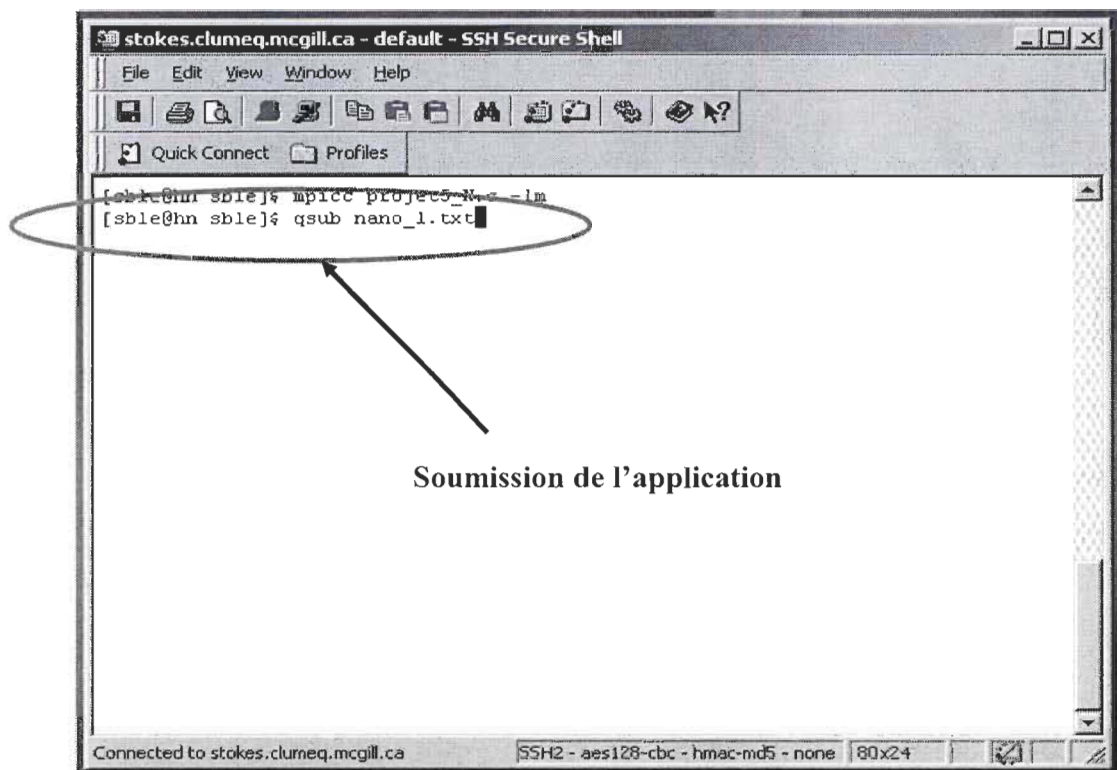


Figure 4-10 Soumission du fichier nano_1 pour l'exécution de l'application [RP]

Après la soumission de l'application présentée à la figure 4-10, le serveur lui affecte un numéro d'identification "*Job id*". Ce numéro peut ensuite servir pour voir l'évolution de l'exécution de l'application comme dans la figure-11, en utilisant la commande PBS "*qstat*".

```
[sble@hn sble]$ qstat
```

Job id	Name	User	Time Use	S	Queue
290556.stokes	protMem3rdstep	elvis	00:00:24	Q	multi
290955.stokes	script.x	mathieug	09:35:39	R	single
290956.stokes	script.x	mathieug	09:35:44	R	single
290957.stokes	script.x	mathieug	09:35:43	R	single
290958.stokes	script.x	mathieug	09:35:30	R	single
290959.stokes	script.x	mathieug	09:34:22	R	single

Figure 4-11 Visualisation de l'état des applications par leur numéro d'identification "*Job id*" [RP].

Nous avons présenté dans cette section, l'environnement de travail sur le réseau CLUMEQ que nous avons utilisé dans notre recherche. Dans notre démarche, nous avons utilisé le compilateur GNU C avec le langage de programmation C et le standard MPI. Cependant, il est possible d'utiliser d'autres compilateurs ou d'autres langages de programmation tel que Fortran et MatLab et il est aussi possible d'utiliser un standard autre que le MPI.

Il existe comme nous l'avons mentionné au chapitre 1, d'autres réseaux de calcul haute performance. Les plus importants au Canada, sont regroupés au sein du C3 (www.c3.ca). Les environnements de simulation sont beaucoup semblables à quelque variations près d'un réseau à l'autre et quelque fois au sein du même réseau tout dépendant de la machine utilisée et du type d'application (multiprocesseurs ou simple processeur).

4.2 Décomposition de domaine

Pour tirer profit de ces réseaux de calcul, il était nécessaire pour nous de décomposer notre application en domaine afin de les répartir sur plusieurs processeurs. Ce qui allégerait le nombre de particule par processeur, la figure 4-12 nous montre un exemple de distribution de particules dans un transistor.

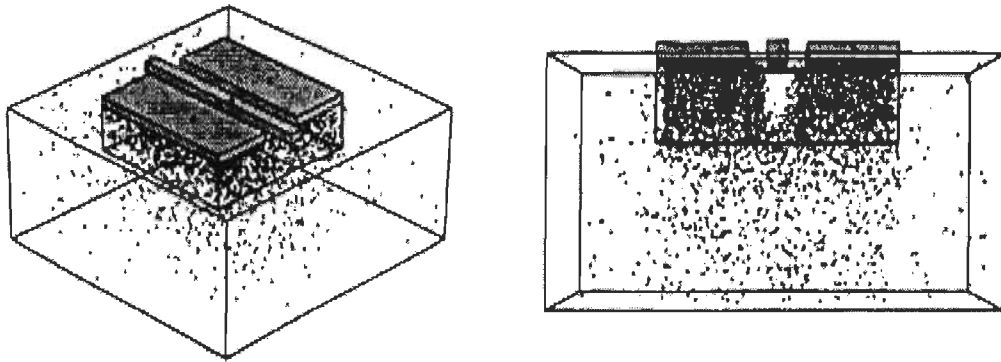


Figure 4-12 Exemple de distribution de particules dans un transistor [10]

Pour un réseau simple 2D comme montré à la figure 4-13, la décomposition se fait en répartissant les différents éléments du réseau en fonction du nombre de processeurs tout en s'arrangeant pour conserver les liens entre les particules voisines. Cela nous facilitera la tâche lors de l'échange d'information entre les processeurs sur l'état des particules voisines qui n'appartiennent pas au même processeur. Pour des géométries moins simple, le réseau peu être décomposé en fonction de l'activité (fonctionnalité de chaque section possible), comme le montre la figure 4-14.

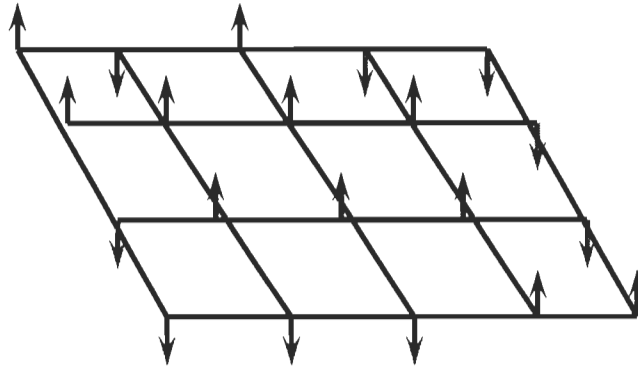


Figure 4-13 Réseau 2D, géométrie simple [RP]

La décomposition de la figure 4-14 se fait en fonction de chaque section du transistor. Le dopage de chaque section pouvant être différent les unes des autres (la concentration ainsi que le type de particule). Chaque section ainsi décomposée pourra être affectée à un nœud de calcul ou à un processeur.

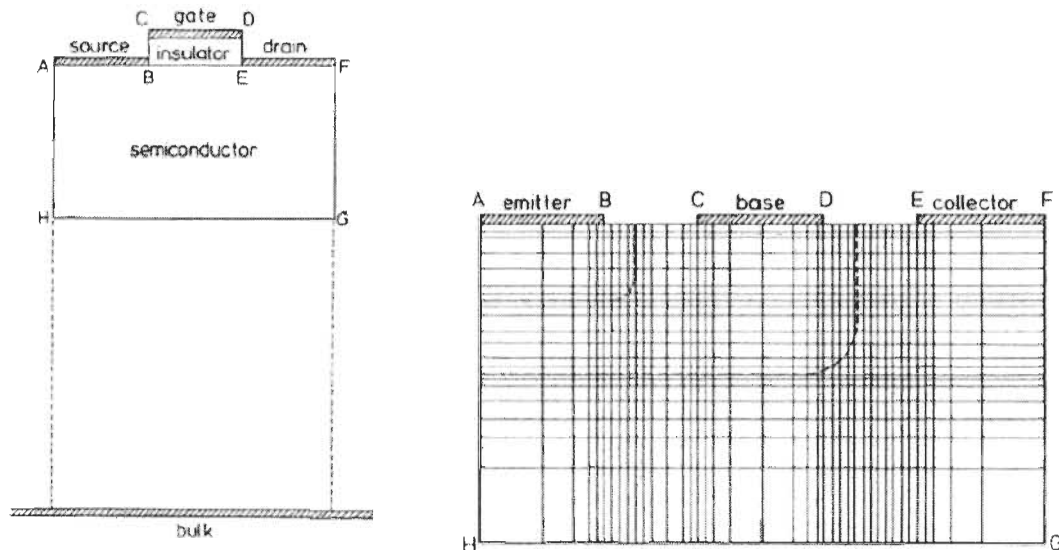


Figure 4-14 Décomposition 2D, d'un transistor à géométrie conventionnelle [16]

4.3 Étude paramétrique

Lorsque notre modèle est décomposé et repartie sur les processeurs, nous avons maintenant la possibilité de calculer plusieurs grandeurs physiques.

Parmi ces valeurs, les plus utiles pour notre étude sont, l'énergie du système, la chaleur spécifique ainsi que la conductivité thermique.

4.3.1 Énergie du système

Le modèle énergétique d'Ising, nous permet de calculer l'énergie du système.

Cette énergie est calculée à partir des interactions entre les différentes particules, en utilisant l'algorithme de Métropolis. À partir de cette énergie, nous pouvons calculer d'autres grandeurs physiques. L'équation suivante, nous permet de calculer l'énergie totale du système [1] :

$$E = \frac{1}{2} \sum_{K=1}^N \sum_{\substack{j, \text{voisin} \\ de K}}^4 JS_j S_K - B \sum_{K=1}^N S_K \quad 4.1$$

4.3.2 Chaleur spécifique

La chaleur spécifique d'un matériau est la quantité de chaleur nécessaire pour élever d'un degré centigrade (ou Kelvin) une masse d'un kg de ce matériau. Intuitivement, la chaleur spécifique rend compte de la capacité d'un matériau à stocker la chaleur.

La chaleur spécifique, peut être calculée à partir de l'énergie du système obtenue du modèle énergétique d'Ising par l'algorithme de Métropolis. L'équation nous permettant son calcul est la suivante [1] :

$$c = \frac{k_B \beta^2}{N} \left(\langle E^2 \rangle - \langle E \rangle^2 \right) \quad 4.2$$

4.3.3 Conductivité thermique

La conductivité thermique est une grandeur physique caractérisant le comportement des matériaux lors du transfert thermique par conduction. Elle représente la quantité de chaleur transférée par unité de surface et par une unité de temps sous un gradient de température. En général, la conductivité thermique va de pair avec la conductivité électrique. A l'échelle classique, la conductivité thermique est déterminée avec la loi de Fourier.

$$\mathbf{q} = -k\nabla T \quad 4.3$$

Dans cette équation, k représente la conductivité thermique, \mathbf{q} le flux thermique et ∇T le gradient de température.

Cependant à l'échelle nanométrique, la loi de Fourier n'est plus applicable [103]. L'étude paramétrique que nous avons effectuée porte sur l'évolution de la conductivité thermique en fonction de la taille du système. Pour le calcul de la conductivité thermique nous utilisons la variation de l'énergie du système en fonction de la surface et de du temps pour une température donnée. La relation de Green-Kubo, utilisée dans plusieurs études portant sur les méthodes de simulation utilisant la dynamique moléculaire [97, 101, 102, 103] peut nous permettre de calculer la conductivité thermique. En adaptant cette équation à notre démarche,

$$k = \frac{1}{k_B T^2} \int dr \int dt \langle j_x(r, t) \cdot j_x(0, 0) \rangle \quad 4.4$$

Dans cette équation, k représente la conductivité thermique, \mathbf{j}_x le courant dans la direction x , t le temps du pas de simulation, et \mathbf{r} est la position de la particule.

$$j(r, t) = \frac{d}{dt} \sum r_i(t) \cdot E_i(t) \quad 4.5$$

r_i est la position de la particule i et E_i est l'énergie du site i calculée avec le modèle énergétique de Ising. En récrivant l'équation 4.4, nous pouvons mieux l'adapter à notre démarche.

$$k = \frac{1}{k_B T^2 A} \langle E \rangle^2 \quad 4.6$$

Dans l'équation 4.6, A est la section du système étudié et $\langle E \rangle^2$ l'énergie moyen du système au carré.

Nous avons simulé quatre nanostructures de silicium de différents diamètres à savoir, 22nm, 37nm, 50nm, et de 70nm afin de vérifier l'effet de la température et de la taille des structures, sur la conductivité thermique. Les résultats de simulation sont présentés dans le chapitre suivant.

Chapitre 5 - Validation

5.1 Introduction

La validation de notre travail s'est effectuée en trois points, dans un premier temps, nous avons vérifié la convergence de l'algorithme ensuite nous avons simulé la conductivité thermique de nanostructures de silicium de diamètres différents (22nm, 37nm, 50nm, et de 70nm). Les résultats de ces simulations ont été comparés à des études existantes, dans la littérature [13, 98]. Enfin, nous avons simulé trois nanosystèmes, ce sont des nanotransistors de caractéristiques, de tailles et de géométries différentes à savoir, un MOSEFT conventionnel, un Dual-Gate FET (une FET à double grille) et MOSFET SOI. Ces nanotransistors ont aussi été simulés avec le logiciel Quamc2D disponible sur le site nanoHub.org [16].

5.2 Validation de l'algorithme

Dans un premier temps il nous fallait valider nos algorithmes de calcul, il s'agissait de vérifier après un certain temps de calcul que nous atteignons l'équilibre thermique.

Cette validation a été faite avec CLUMEQ (un seul processeur) et avec MATLAB.

Les différentes figures ci-dessous montrent l'évolution des particules en fonction du temps de calcul. Le temps ici fais allusion au pas de calcul de Monté Carlo par site, en d'autres termes notre analyse permet nous permet d'évaluer le nombre d'itérations

nécessaires pour atteindre l'équilibre thermique. Rappelons ici que c'est lorsque le système atteint l'équilibre thermique que nous pouvons calculer les différentes grandeurs physiques dont nous avons besoins [2]. Il est à noter que nous avons utilisé un seul processeur lors de cette évaluation pour une raison très simple. En effet le processeur utiliser nous permettra de déterminer (sans tenir compte de la durée de la simulation en temps réel) le nombre de pas de Monte Carlo nécessaires pour que le système simulé atteigne l'équilibre thermique. Nous avons utilisé deux échantillons de tailles différentes à savoir, 10x10 et 100x100 particules. Cette estimation du nombre de pas de Monte Carlo nous permettra de connaître le minimum d'itération à imposer dans l'algorithme ainsi que le maximum de particules (données) à imposer pour le respect de cette condition (l'équilibre thermique).

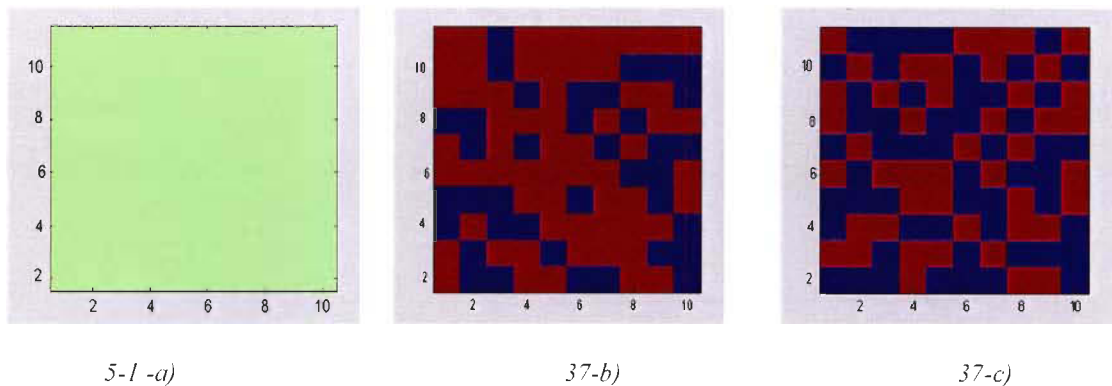
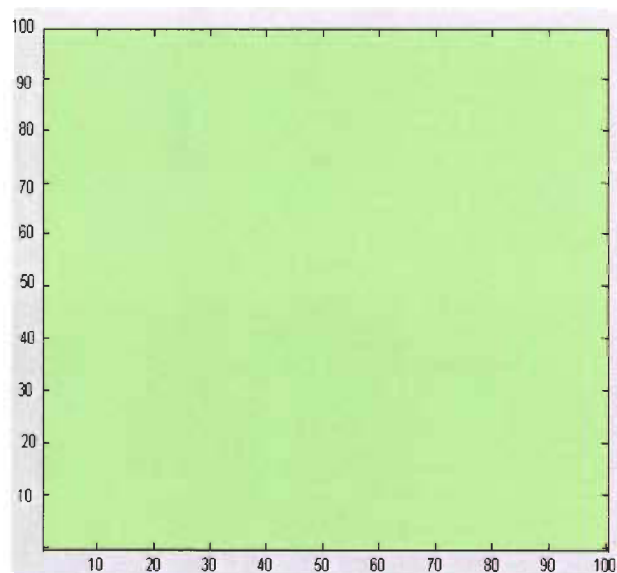


Figure 5-1 Évolution de la simulation vers l'équilibre thermique [RP]

La figure 5-1 montre l'état du système à différents temps de simulation (pas de Monte Carlo), exactement à $t_1=0$, $t_2=1000$ pas et à $t_3=2500$ pas pour le premier échantillon de particules (10x10).

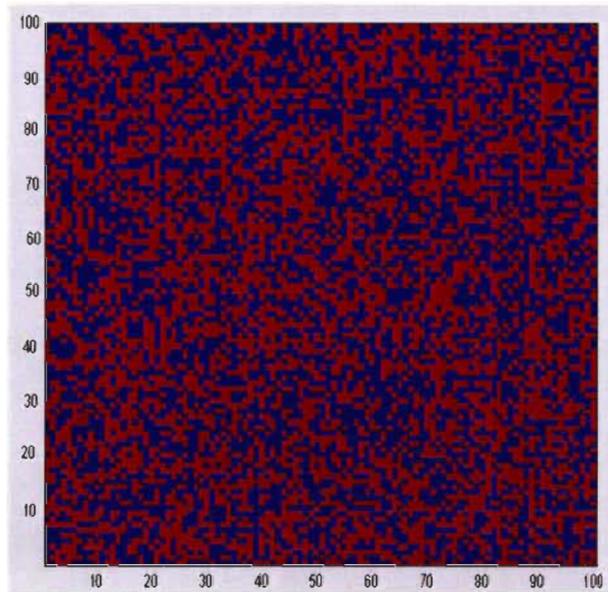
Au temps t_1 représenté par la figure 5-1 a), nous imposons le même sens à toutes les particules (spins), à une température $T=0K$ que nous considérons dans le cas présent comme étant l'état fondamental de ce système. Lorsque nous soumettons le système à une perturbation thermique, la température $T \neq 0K$, les particules vont tendre vers l'équilibre thermique. Les figures 5-1 b), et 5-1 c) représentent respectivement l'état du système au temps t_2 et au temps t_3 . Au temps t_2 nous voyons une réorganisation des spins (-1 et 1), avec une prédominance pour les spins positifs en rouge. Au temps t_3 on voit une évolution plus nette de l'état du système vers l'équilibre thermique.

La figure 5-2 montre l'état du système à différents temps de simulation, exactement à $t_1=0 \text{ pas}$, $t_2=8000 \text{ pas}$ pour le deuxième échantillon de particules (100x100). Au temps t_1 représenté par la figure 5-2 a), nous imposons le même sens à toutes les particules, à une température $T=0$ que nous considérons dans le cas présent comme étant l'état fondamental de ce système. Lorsque nous soumettons le système à une perturbation thermique, la température $T \neq 0$, les particules vont tendre vers l'équilibre thermique.



5-2 a)

La figure 5-2 b), représente l'état du système au temps t_2 . Nous voyons ici qu'après la réorganisation des spins (-1 et 1), l'état du système temps vers l'équilibre thermique.



5-2 b)

Figure 5-2: Évolution de la simulation vers l'équilibre thermique [RP]

L'utilisation de ces deux échantillons nous a permis de mettre en relief le nombre de pas de Monte Carlo versus la taille de l'échantillon pour l'atteinte de l'équilibre thermique. L'estimation du temps Monte Carlo dans le deuxième échantillon nous permettra d'imposer le nombre minimal d'itérations et le nombre maximal de données pour chaque processeur lors de simulations sur plusieurs nœuds de calculs.

5.3 Validation des résultats de simulation avec la littérature

Cette validation nous permet de comparer nos résultats de simulation avec des résultats trouvés dans la littérature [13, 98, 96].

La figure 5-3, présente notre simulation de la conductivité thermique de quatre nanofils de silicium de diamètres différents (22nm, 37nm, 50nm, et 70nm). La simulation a été réalisée dans deux buts, dans un premier temps vérifier l'effet de la température et de la taille sur la conductivité thermique. Cette figure nous permet de comparer nos résultats de simulation avec des données expérimentales de [96] et de simulation [13].

En effet lorsque nous analysons nos résultats de simulations sur la figure 5-3, nous voyons que plus le diamètre du nanofils diminue, plus la conductivité thermique baisse. En ce qui concerne la température nous observons une variation de la conductivité thermique.

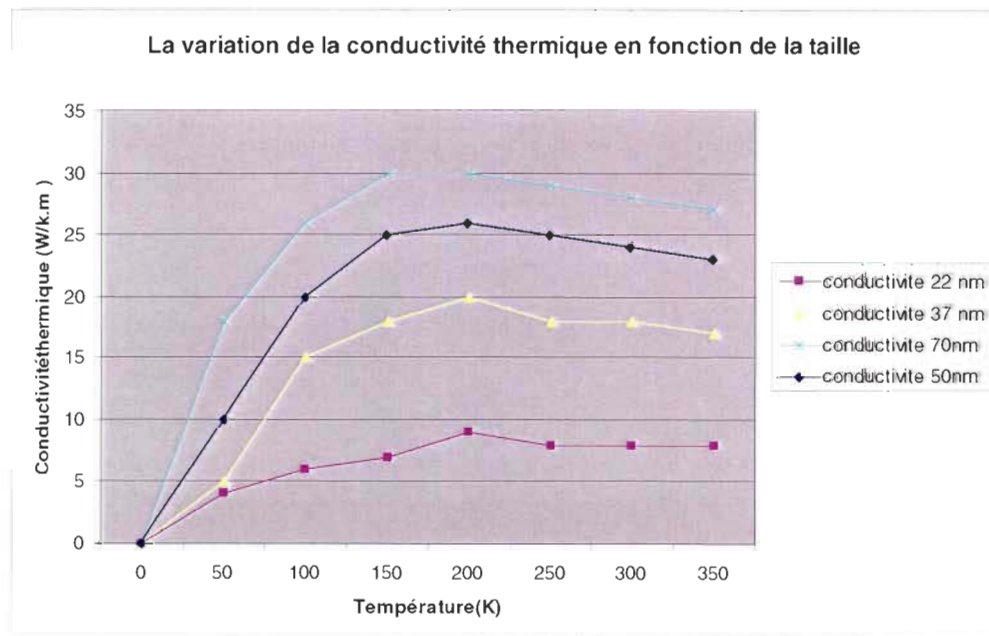


Figure 5-3 Études paramétriques de la conductivité thermique de nanofils de silicium [RP]

La comparaison des résultats (de notre simulation) présentés à la figure 5-3 avec ceux présentés à la figure 5-4, nous permet de valider nos résultats de simulation.

La figure 5-4 provient de [13], elle représente les résultats de simulations de la conductivité thermique de trois nanofils de silicium de diamètres différents (22nm, 37nm, 115nm). Elle présente aussi sur le même graphique, les résultats de mesures expérimentales de la conductivité thermique de ces trois nanofils de silicium. Les courbes présentées sous forme de traits sont les résultats de la simulation réalisée dans [13] alors que celles sous forme de symboles sont les résultats expérimentaux de Majumdar A. et al [96].

L'analyse des courbes de 37nm montre une ressemblance entre nos résultats et les courbes de la figure 5-4, avec de petites différences.

Cependant il existe une différence un peu marquée avec la courbe expérimentale et de 22nm. Cette différence est due à la non prise en compte effective des conditions aux frontières dans notre système car dans notre simulation, nous utilisons des conditions virtuelles en plus nous n'avons pas pris en compte l'effet des impuretés à basse température [13].

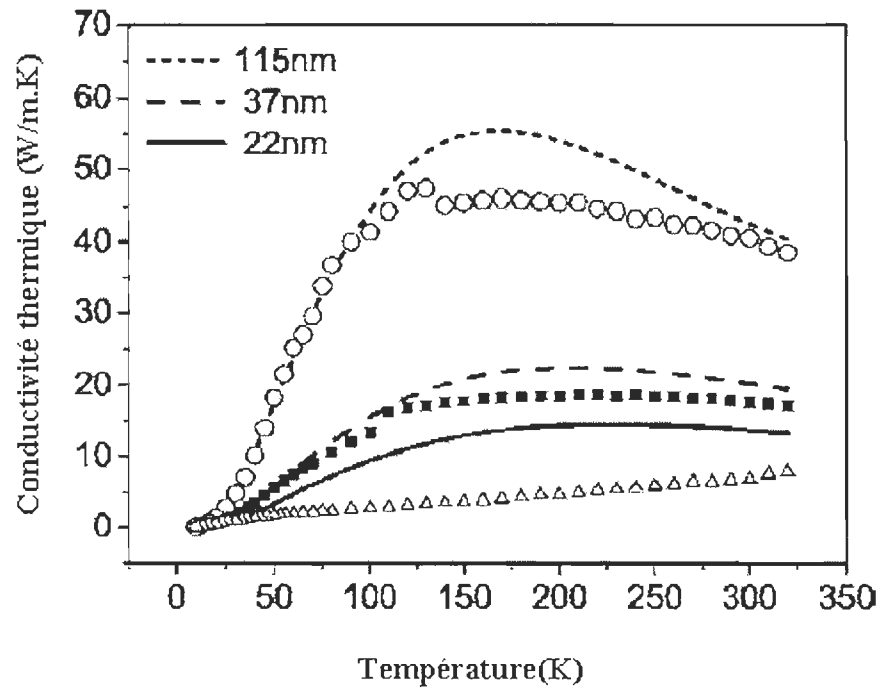


Figure 5-4 La conductivité thermique de nanofils de silicium [13]

La figure 5-5 représente, la simulation d'une mince couche de silicium dont le diamètre est de 50nm. Nous évaluons ici, La conductivité thermique, à des températures allant de 300K à 500K.

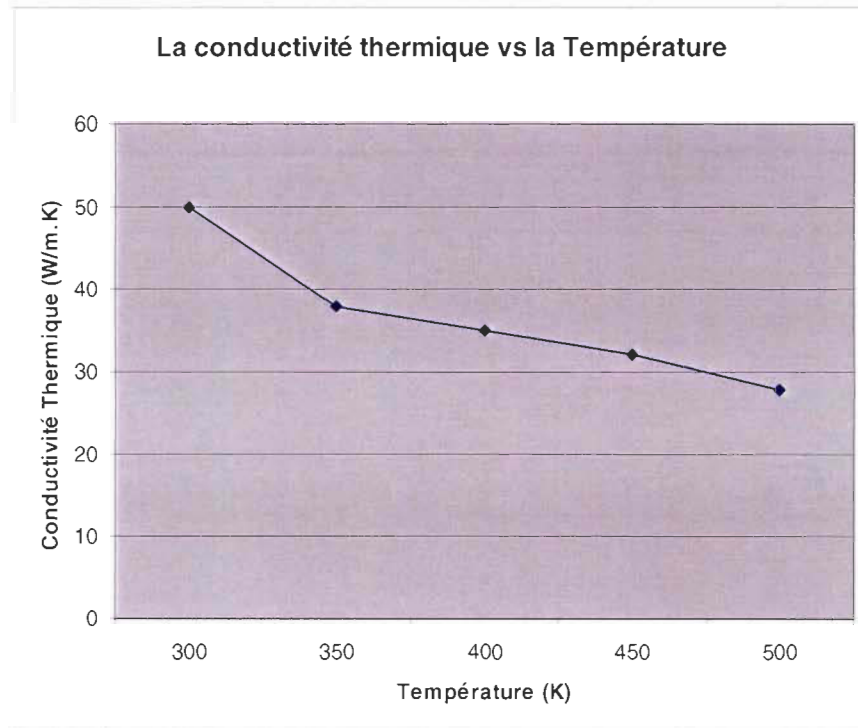


Figure 5-5: La conductivité thermique d'une mince couche de silicium [RP]

L'observation que l'on peut faire ici est que lorsque la température augmente, la conductivité thermique baisse. Nous avons comparé la courbe résultante de notre simulation, à la courbe 2 de la figure 5-6 afin de valider notre simulation. La courbe 2 de la figure 5-6 représente le résultat de la simulation d'une mince couche de silicium de 50nm de diamètre réalisé par Asheghi M. et al [98]. Les deux courbes sont semblables avec une différence de $\pm 5\%$, ce qui nous conforte dans notre démarche de validations. La courbe 1 de la figure 5-6 nous permet tout simplement de mettre en relief, l'effet de la taille des composants sur leurs caractéristiques à l'échelle nanométrique. En effet on constate ici qu'il y a une baisse importante de la conductivité thermique lorsque la taille du système baisse.

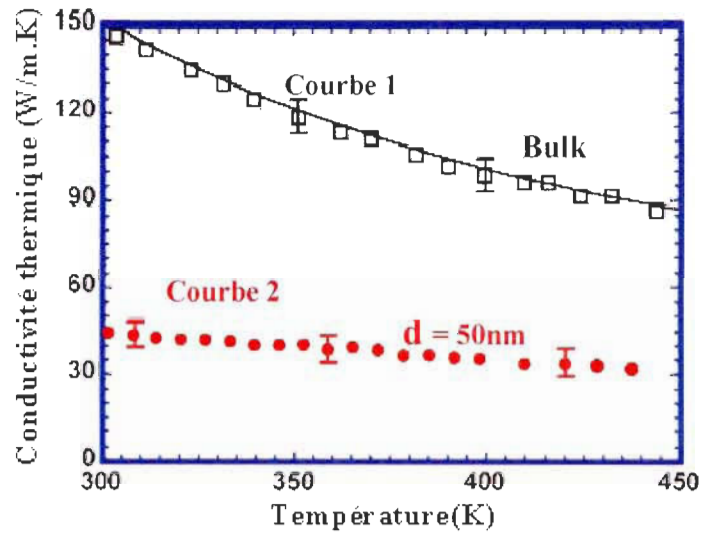


Figure 5-6 La conductivité thermique d'une mince couche de silicium [98]

5.4 Validation des résultats de simulation comparaison avec nanoHub

Dans cette section, nous comparons nos résultats de simulation avec des résultats de simulations obtenus avec le logiciel Quamc2D [16]. Ce logiciel a été réalisé par l'équipe du professeur Dragica Vasileska de l'université de l'Arizona et mis à la disposition du publique sur le site nanoHub.org. C'est site qui a environ 90 000 usagers, il met leur disposition un espace d'échange sur les nanosciences, des formations ainsi que plusieurs outils de simulation numérique des phénomènes et des systèmes à l'échelle nanométrique. Il permet aussi aux usagers d'y publier leurs travaux et aussi de faire valider leurs outils de simulation par un grand nombre de personnes.

Nous avons utilisé pour valider notre simulation, trois nanotransistors à savoir, un MOSFET, un Dual-Gate FET et un MOSFET SOI.

Nous présentons dans les sections qui suivent, les trois nanotransistors ainsi que leurs caractéristiques utilisées pour les différentes simulations. Et par la suite, nous faisons la comparaison des différents résultats obtenus lors de cette validation.

5.4.1 Système 1

La figure 5-7 montre le système 1, c'est un MOSFET conventionnel de 25nm, pour le simuler, nous avons utilisé les caractéristiques présentées dans le tableau 5.1

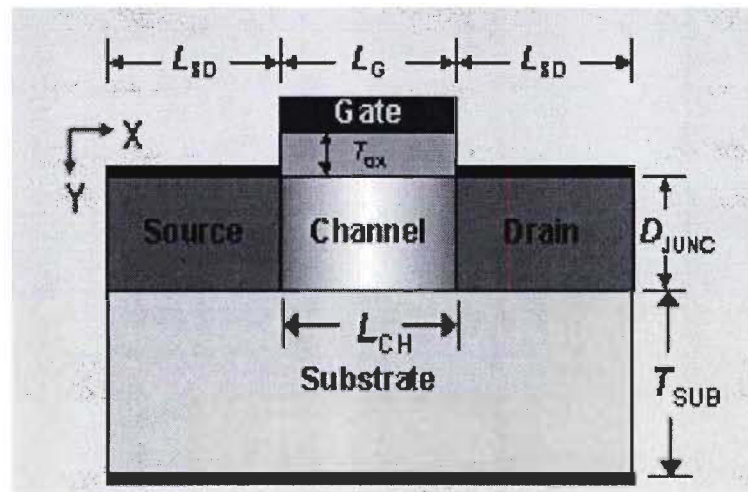


Figure 5-7 Figure 43: MOSFET [16]

La simulation porte sur l'énergie des électrons dans le transistor. Pour réaliser la simulation, nous avons divisé le système en trois domaines, la source, le drain et le canal. Nous avons utilisé les dimensions indiquées pour les différents domaines ainsi que leur dopage. La simulation que nous avons réalisée s'est faite de façon statique chaque particule est restée propre à son domaine. Il faut aussi noter que nous n'avons pas pris en compte, les

conditions frontières de chaque domaine. Les résultats de la simulation sont présentés à la section suivante sous forme de graphique, sur les figures 5-10,5-11.

Tableau 5-1 : Les caractéristiques de simulation du MOSFET

<u>DIMENSIONS</u>		<u>DOPAGE</u>	
L_G	= 25 nm	N_D	= 10^{19} cm^{-3}
L_{CH}	= 25 nm	N_S	= $2 \cdot 10^{26} \text{ cm}^{-3}$
L_{SD}	= 25 nm	N_C	= $-5.0 \cdot 10^{24} \text{ cm}^{-3}$
T_{SUB}	= 60 nm	Bulk	= $-5.0 \cdot 10^{22} \text{ cm}^{-3}$
T_{OX}	= 1.5 nm	<u>Température Ambiante</u> = 300K	
MS_{AL}	= 0.5 nm	<u>Nombre de maille:</u>	
MS_{AD}	= 0.5 nm	T_{OX}	= 10

Le tableau 5.1 contient les dimensions utilisées pour la simulation du système 1, le dopage des différentes sections du transistor, la température ambiante ainsi que le nombre de maille.

5.4.2 Système 2

Le système 2 est un Dual-Gate FET présenté à la figure 5-8, pour le simuler, nous avons utilisé les caractéristiques présentées dans le tableau 5.2.

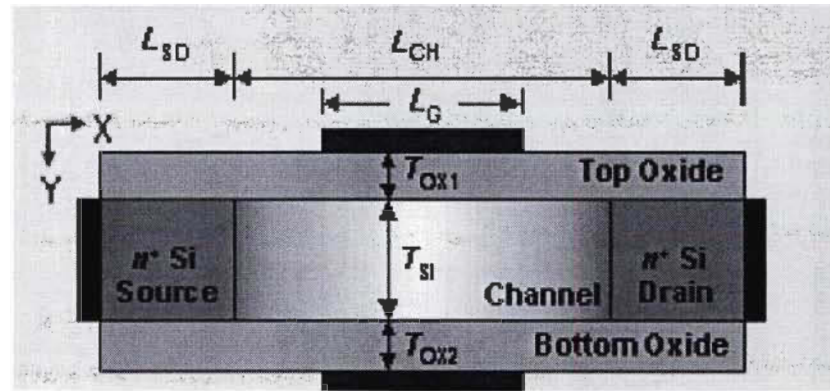


Figure 5-8 Dual-Gate FET [16]

Pour réaliser la simulation, nous avons divisé le système en trois domaines, la source, le drain et le canal.

Tableau 5-2: Les caractéristiques de simulation du Dual-Gate FET

<u>DIMENSIONS</u>		<u>DOPAGE</u>	
L_G	= 9 nm	N_D	= $2.e^{26} \text{ cm}^{-3}$
L_{CH}	= 17 nm	N_S	= $2.e^{26} \text{ cm}^{-3}$
L_{SD}	= 6 nm	N_C	= $-1.e^0 \text{ cm}^{-3}$
T_{SI}	= 3 nm	<u>Température Ambiante</u> = 300K	
T_{OX1}	= 1 nm	<u>Nombre de maille:</u>	
T_{OX2}	= 1 nm	T_{OX1}	= 10
MS_{AL}	= 0.3 nm	T_{OX2}	= 10
MS_{AD}	= 0.1 nm		

Le tableau 5.2 contient les dimensions utilisées pour la simulation du système 2, le dopage des différentes sections du transistor, la température ambiante ainsi que le nombre de maille.

Les résultats de la simulation sont présentés à la section suivante sous forme de graphique, sur les figures 5-12,5-13.

5.4.3 Système 3

Le système 3 est un MOSFET SOI, pour le simuler, nous avons utilisé les caractéristiques présentées dans le tableau 5.3

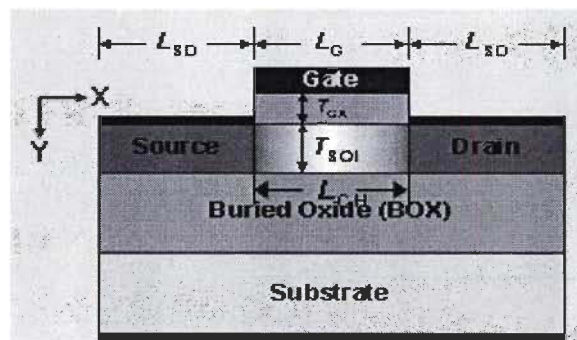


Figure 5-9 MOSEFET SOI [16]

Tableau 5-3 : Les caractéristiques de simulation du MOSFET SOI

<u>DIMENSIONS</u>		<u>DOPAGE</u>	
L_G	= 25 nm	N_D	= $2.e^{25} \text{ cm}^{-3}$
L_{CH}	= 25 nm	N_S	= $2.e^{25} \text{ cm}^{-3}$
L_{SD}	= 25 nm	N_C	= $-1.e^{23} \text{ cm}^{-3}$
T_{SOI}	= 5 nm	<u>Température Ambiante</u> = 300K	
T_{OX}	= 1.5 nm		
B_{OX}	= 150 nm	<u>Nombre de maille:</u>	
Substrat	= 90 nm		
MS_{AL}	= 0.5 nm		
MS_{AD}	= 0.5 nm		
		T_{OX}	= 10
		B_{OX}	= 15

Le tableau 5.3 contient les dimensions utilisées pour la simulation du système 3, le dopage des différentes sections du transistor, la température ambiante ainsi que le nombre de maille. Les résultats de la simulation sont présentés à la section suivante sous forme de graphique, sur les figures 5-14,5-15.

5.5 Évaluation des résultats

Nous présentons dans cette section, les résultats des différentes simulations réalisées avec Quamc2D [16] et BVSim_nano [RP]. Les simulations effectuées, évaluent l'énergie des électrons à travers le nanotransistor.

5.5.1 Résultat de la simulation du MOFET

La figure 5-10 est le résultat de la simulation du MOSFET réalisée avec le logiciel Quamc2D. Elle présente la répartition de l'énergie dans le nanotransistor.

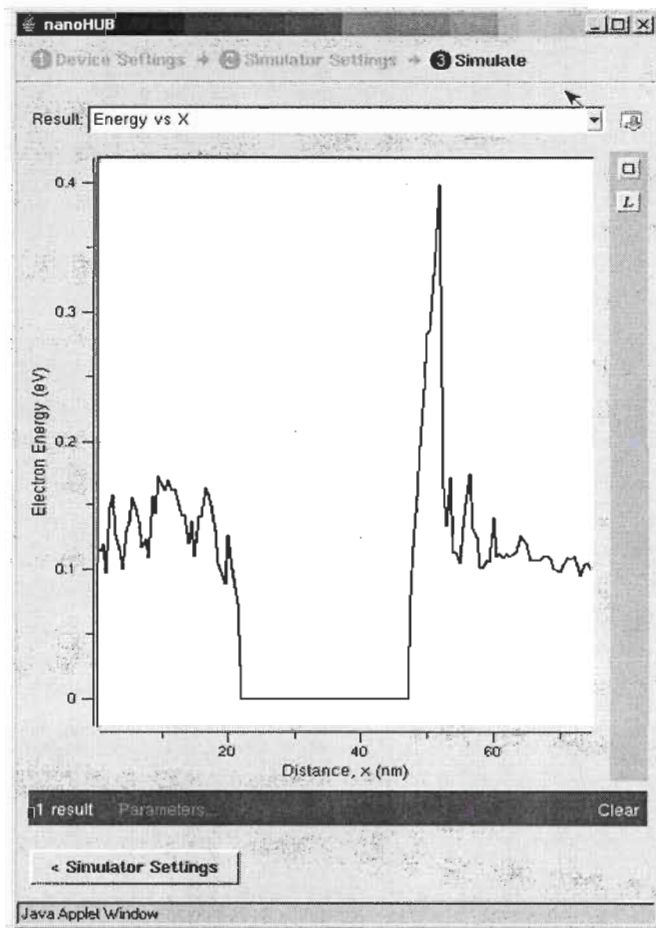


Figure 5-10 Système 1, résultat avec Quamc2D [16]

La figure 5-11 est le résultat de la simulation du MOSFET réalisée avec le logiciel *BVSim_nano* [RP].

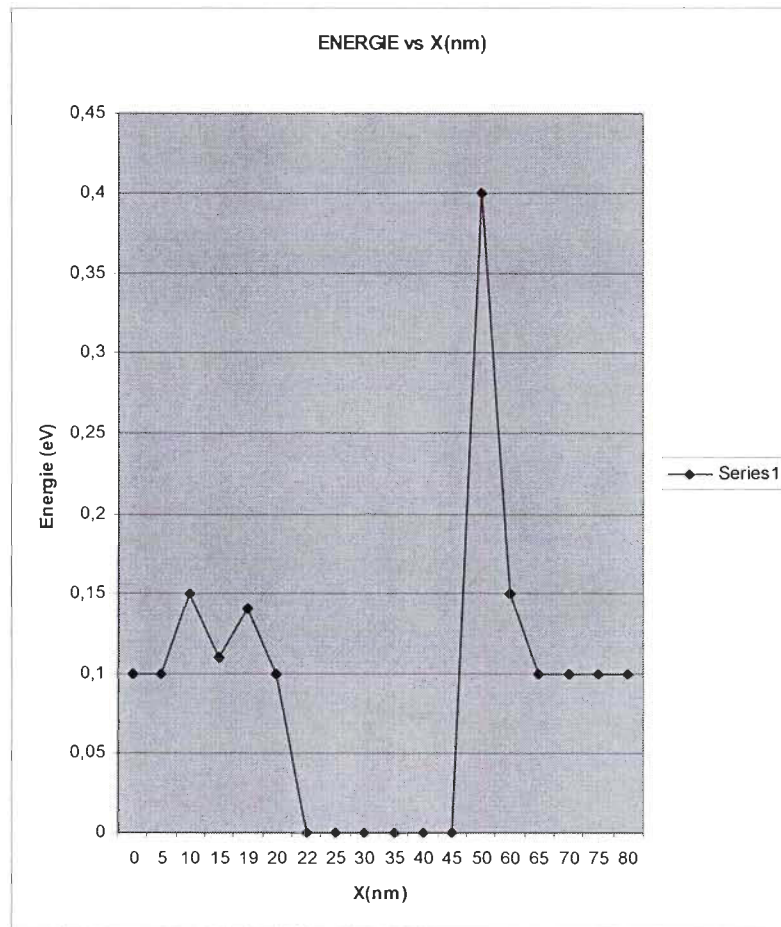


Figure 5-11 Système 1 simulation avec *BVSim_nano* [RP]

Nous remarquons une grande similitude entre les figures 5-10 et 5-11, avec de faibles différences.

5.5.2 Résultat de la simulation du système 2

La figure 5-12 est le résultat de la simulation du Dual-Gate FET réalisée avec le logiciel Quamc2D.

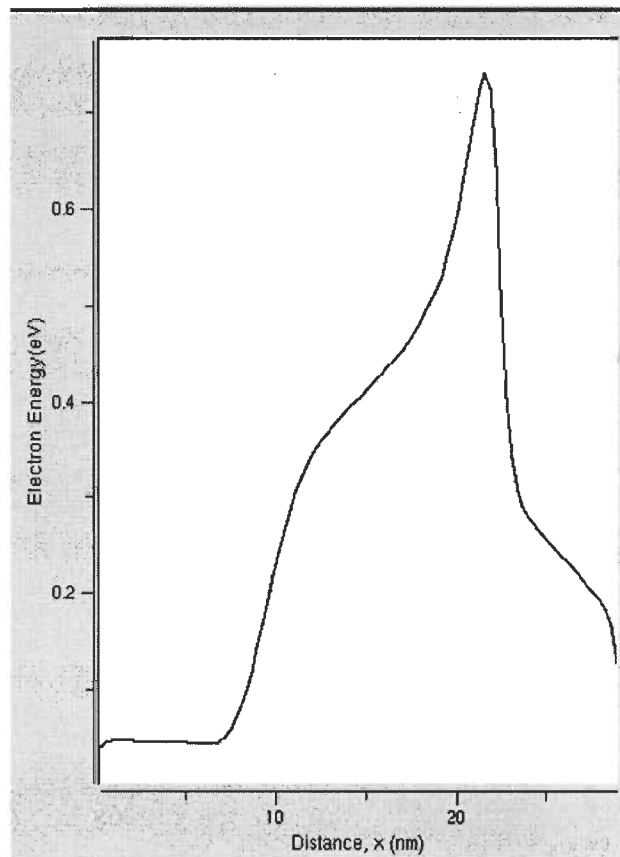


Figure 5-12 Système 2 résultat avec Quamc2D [16]

La figure 5-12 est le résultat de la simulation du Dual-Gate FET réalisée avec le logiciel *BVSim_nano* [RP].

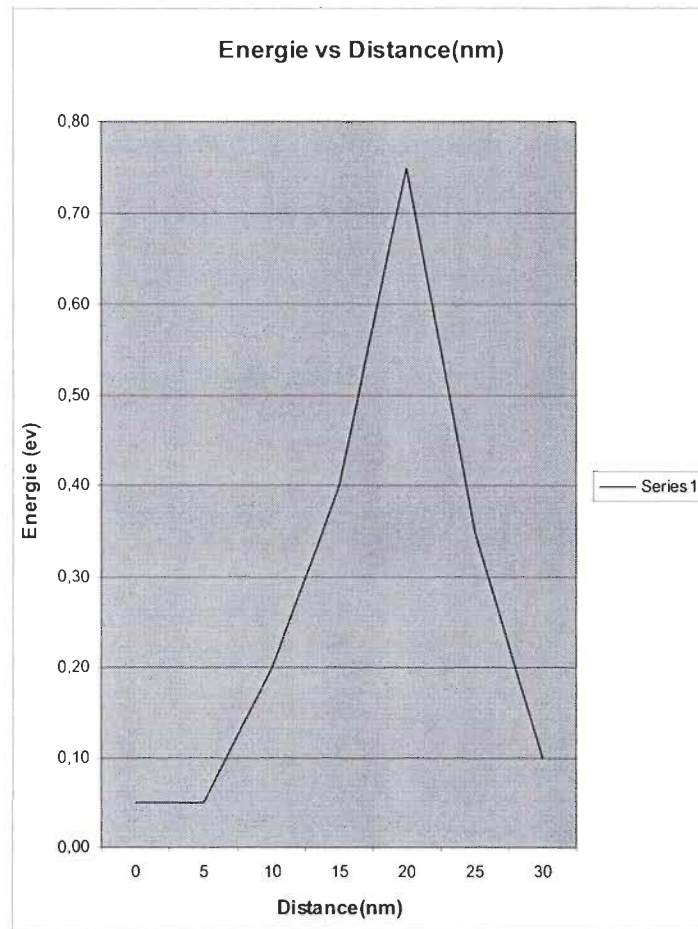


Figure 5-13 Système 2, simulation avec BVSim_nano [RP]

Nous remarquons une grande similitude entre les figures 5-12 et 5-13, avec de faibles différences.

5.5.3 Résultat de la simulation du système 3

La figure 5-14 est le résultat de la simulation du MOSFET SOI réalisée avec le logiciel Quamc2D

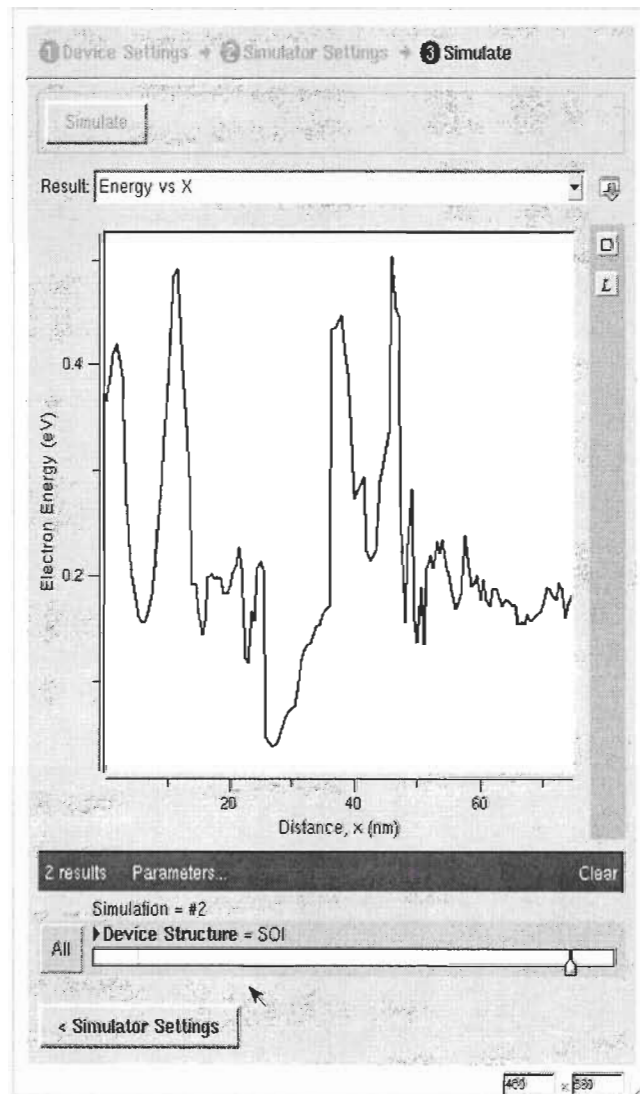


Figure 5-14 Système 3 résultat avec Quamc2D [16]

La figure 5-15 est le résultat de la simulation du MOSFET SOI réalisée avec le logiciel *BVSim_nano* [RP].

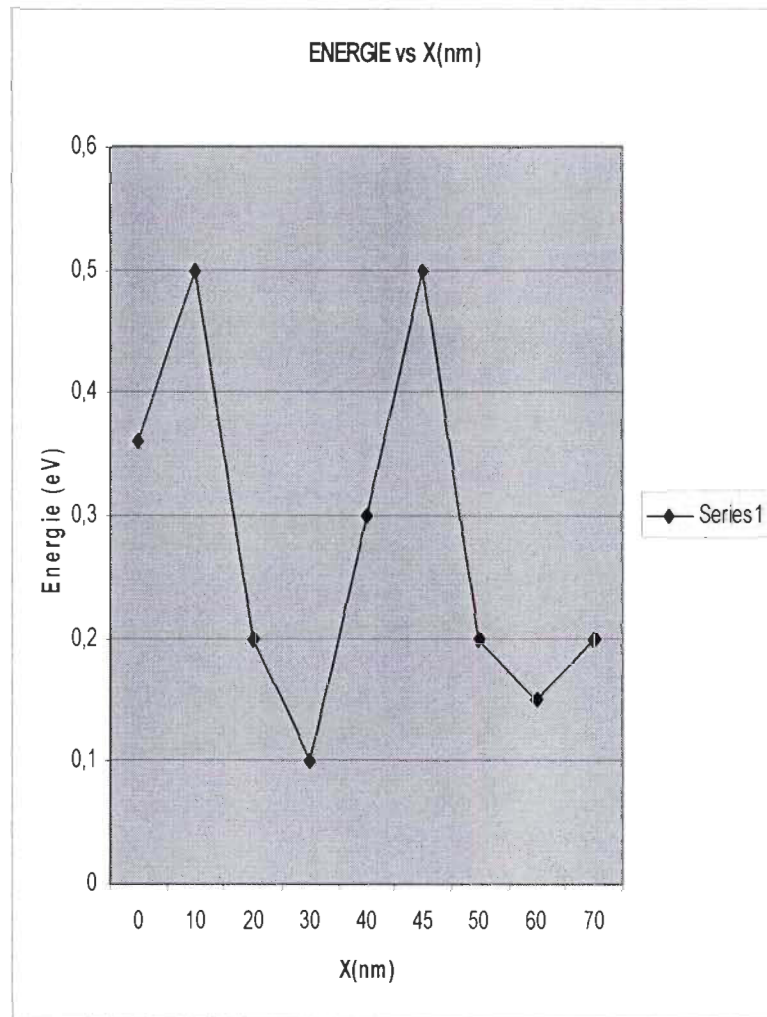


Figure 5-15 Système 3, simulation avec *BVSim_nano* [RP]

La différence est beaucoup prononcée entre des figures 5-14 et 5-15, malgré cette différence, l'allure demeure toujours la même ainsi que les différents sommets.

La différence entre ces résultats se situe au niveau des conditions frontières non prises en compte dans notre simulation, et surtout par la mobilité des particules. Dans notre

simulation, nous simulons les interactions entre toutes les particules du réseau cependant, nous ne prenons pas en compte le déplacement des particules entre les différents nœuds de calcul.

5.6 Discussion

La validation de notre étude c'est par plusieurs simulations et comparaisons avec d'une part des données expérimentales présentées dans la littérature et d'autre part, avec des données de simulations provenant d'un logiciel.

Cependant, la première validation que nous avons fait consistait tout simplement un moyen de nous de la convergence de notre algorithme vers l'équilibre thermique.

En fait nous avons étudié deux échantillons l'un de 10x10 et l'autre de 100x100 pour nous permettre de savoir le temps Monte Carlo nécessaire pour atteindre l'équilibre thermique. Le temps déterminé dans le cas présent 8000 *pas*, sera fixé comme pré-condition à l'atteinte de l'équilibre thermique. Cette façon de faire peut être coûteuse si l'échantillon est de petite taille.

Nous avons par la suite faite une étude paramétrique laquelle nous avons vérifié l'effet de la taille des composants sur la conductivité thermique à l'échelle nanométrique. Nous avons observé que plus la taille baisse, moins bonne est la conductivité thermique. Nous avons utilisé comme composants, quatre nanofils de silicium de diamètres différents (22nm, 37nm, 50nm, et 70nm). Nous avons ensuite comparé les résultats de notre simulation avec des données expérimentales [96] et de simulation [13]. De cette comparaison, nous avons remarqué que la prise en compte ou non d'impuretés dans les matériaux pourrait influencer fortement la simulation.

Nous avons par la suite simulé une mince couche de silicium de 50nm de diamètre à température allant 300K à 5000K. Le résultat de cette simulation a été comparé à l'étude réalisé par Asheghi M. et al [98].

Pour terminer notre étape de la validation, nous avons comparé nos résultats de simulations avec des résultats obtenus avec le logiciel Quamc2D [16]. Nous avons utilisé comme modèles, un MOSFET, un Dual-Gate FET et un MOSFET SOI. La comparaison des résultats des deux premiers modèles est très proche. Le dernier modèle présente des divergences. La différence entre ces résultats se situe au niveau de la non prise en compte des conditions frontières dans notre simulation, et surtout par la mobilité des particules entre les différents domaines de calcul.

Les simulations et les validations effectuées nous permettent d'affirmer qu'une fois optimisé, l'outil (*BVSim_nano*) développé pourrait être très utile dans le processus de conception et de développement des nanosystèmes. Il permettra de faire l'analyse électrothermique du système et ou de certains composants critiques du système (exemple de la couche de silicium dans le MOSEFET SOI), afin de donner une bonne indication dans le choix des matériaux et même dans le choix de la géométrie du nanosystème.

Plusieurs travaux traitent des phénomènes électrothermiques à l'échelle nanométrique. Cependant tous les modèles proposés utilisent comme paramètres de calcul, la vitesse des particules, le temps de relaxation et le libre parcours moyen des particules.

Le modèle que nous proposons n'utilise aucun de ces paramètres, il se base uniquement sur l'interaction entre les particules et leur potentiel d'échange pour évaluer l'énergie du système et par la suite, la conductivité thermique ainsi que la chaleur spécifique.

Cette différence est très importante car la vitesse des particules, le temps de relaxation et le libre parcours moyen des particules sont fortement dépendant de la taille des systèmes. Leur utilisation pourrait donc s'avérer problématique lorsque la taille des systèmes (ce qui est de plus en plus le cas) est inférieure à ces paramètres. C'est essentiellement pour cette raison que lors de notre démarche, nous avons choisis des paramètres qui ne dépendent pas de la taille des systèmes. En plus de cet aspect (très important), nous avons proposé une architecture parallèle en utilisant la norme MPI.

Sans avoir fait l'évaluation de l'apport de la parallélisation de notre application, nous restons convaincus qu'il serait de loin plus performant qu'un code séquentiel.

Chapitre 6 - Conclusion

L'étude des nanosystèmes a fait l'objet de plusieurs recherches cette dernière décennie. Cependant les travaux reliés à la chaleur dégagée par ces nouveaux composants sont minimes.

Le but de nos travaux de recherche, vise à fournir à l'industrie et au milieu académique, un outil d'évaluation et de simulation du flux thermique au sein des nanosystèmes.

La méthodologie utilisée consiste dans un premier temps, à déterminer des lois de la physique et des équations mathématiques à même de satisfaire aux réalités de cette nouvelle échelle de grandeur. Ensuite, il nous fallait trouver des algorithmes réalistes et peut coûteux, qui nous permettrons d'implémenter et de valider de façon pratique les différents modèles issus des lois physique et des équations mathématiques déterminer précédemment. Il nous fallait aussi déterminer, les différents supports informatiques qui pourraient nous permettre en temps réel de faire nos différentes simulations et validations pour finalement aboutir à l'outil désiré.

La problématique principale de notre étude, est liée à la gestion de la chaleur dans les nanosystèmes. En plus, les outils dédiés à la simulation thermique des macro systèmes ainsi que des microsystèmes, ne sont plus adéquat pour l'analyse des nano système. Les lois physiques utilisées dans la réalisation de ces outils ne sont plus vérifiées à l'échelle nanométrique. Une autre problématique est celle liée au temps de simulation

et de validation de nos différentes évaluations, certaines simulations pouvant prendre jusqu'à plusieurs jours d'exécution.

En effet, comme plusieurs études ainsi que plusieurs années de pratique en ingénierie l'ont démontrée, une bonne simulation numérique vaut mieux que plusieurs manipulations ou montages par « essai erreur ». Donc tout outil permettant de prévoir où d'avoir une idée du comportement d'un système lors de sa conception est le bien venu. En particulier lorsqu'il s'agit de composants difficilement manipulables tel que les nanosystèmes.

Les lois de la physique classique n'étant plus vérifiées à l'échelle nanométrique, il faut se tourner vers d'autres lois, celle de la physique quantique dont la loi d'onde de Schrödinger ainsi que la loi de Boltzmann nous semblaient les plus indiqués pour la résolution de notre problème. La complexité de la résolution classique de ces équations, nous a conduit à considérer des méthodes statistiques de résolution. Nous nous sommes penché sur les simulations de Monte Carlo dont celle dite de « Métropolis » qui s'avère très intéressante car intègre déjà la notion de température (elle utilise la probabilité de Boltzmann). L'utilisation de méthode statistique pose un problème majeur, celui de la représentativité de l'échantillonnage donc du temps de calcul et de support de stockage. Pour répondre à cette problématique, nous avons décidé d'utiliser (pour avoir un échantillonnage représentatif ainsi qu'un meilleur temps de calcul) des algorithmes parallèles avec la norme MPI, puis implémenter le tout sur des calculateurs de haute performance (HPC). Ces calculateurs ayant plusieurs nœuds de calcul, il nous sera donc possible de répartir les tâches de calcul entre ces différents nœuds ce qui permettra de réduire de façon considérable le temps de calcul et en même temps le problème de stockage. Les HPC que nous avons identifiés pour notre étude sont entre autres,

le WestGrid, le CLUMEQ et l'ACNET. Parmi tous ces réseaux de calcul, nous avons utilisé le réseau CLUMEQ pour faire nos simulations. Après avoir fait un tour d'horizon et analysé nos besoins, nous avons trouvé que tous les réseaux étaient en mesure de satisfaire nos demandes donc notre choix n'est en rien un quelconque avantage du réseau CLUMEQ sur les autres plus haut mentionnés.

À travers cette thèse, nous avons utilisé le modèle d'Ising comme modèle de base pour notre démarche. Cela nous a permis de prendre en compte toutes les interactions entre les différentes particules des systèmes que nous avons simulés. Nous avons relevé le défi de la représentativité des échantillons lors de la simulation des systèmes, par la parallélisation. En effet, nous avons utilisé la décomposition en domaines pour diviser nos différentes tâches (les différentes sections du système) afin de les répartir sur plusieurs nœuds de calcul.

Nous avons fait plusieurs simulations, dont les résultats sont présentés au chapitre 5 de cette thèse. À travers ce chapitre, nous avons comparés nos résultats à ceux de la littérature et à ceux obtenus avec le logiciel Quamc2D. Les résultats sont très intéressants et très encourageants. Cependant, malgré ces résultats des étapes restent à franchir pour aboutir à un outil entièrement satisfaisant. Les étapes telle que la prise en compte effective des conditions physiques réelles des systèmes seraient un apport non négligeable lors du couplage des nanosystèmes avec d'autres systèmes physiques de tailles différentes.

Aussi la prise en compte de la différence entre les particules pourrait nous permettre une meilleure appréciation de leur effet dans certaines conditions particulières comme par l'exemple le comportement des impuretés dans un réseau de silicium à basse température.

Dans notre approche, nous simulons les interactions entre toutes les particules du réseau cependant, nous ne prenons pas en compte le déplacement des particules entre les différents nœuds de calcul, cela peut aussi être à l'origine de certaines disparités entre les résultats que nous avons obtenus et ceux obtenus avec le logiciel Quamc2D.

Toutes ces raisons qui ont été à l'origine des différences lors de la validation de notre application et les autres travaux feront l'objet de notre orientation future.

Bibliographie

- [1] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth. A.H. Teller and E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys.* 21 (1953) 1087-1092.
- [2] M. E. J. Newman, G. T. Barkema, "Monte Carlo Methods in Statistical Physics", *Oxford University press, Oxford, 1999.*
- [3] Ken Goodson, "Microscale thermal Engineering of Electronic Systems", Thermosciences Division Mechanical Engineering Departement, *Standford University 2003.*
- [4] A. W. Skorek , S. V. Blé, A. Gryko-Nikitin , J. Nazarko, " Nanothermal Management in Nanoelectronics Systems", *University of Quebec at Trois-Rivières, Bialystok Technical University*, conference NSTI Nanotech 2007, Santa Clara, USA, May 20-24, 2007
- [5] D. P. Landau, K. Binder, "A Guide to Monte Carlo Simulations in Statistical Physics", *Cambridge University press, Cambridge, 2000.*
- [6] J. Shumway: "All-Electron Path Monte Carlo Simulations of Small Atoms and Molecules", *Computer Simulations Studies in Condensed Matter Physics XVII, Springer Verlag, Berlin, 2006, 181-195.*
- [7] A. Skorek: "A Parallel Approach for Electrothermal Analysis of Power Electronics Devices", *Proceedings of the INTERPACK.97, The Pacific RIM/ASME International, Intersociety Electronic & Photonic Packaging Conference and Exhibition, Hawaii, USA, June 15-19, 1997,*
- [8] B. Butrylo, A.Jordan. A.Skorek, "A Genetically Optimized Electrothermal Parallel Algorithm", *Proceedings if the INTERPACK.99, The Pacific RIM/ASME International, Intersociety Electronic & Photonic Packaging Conference and Exhibition, Hawaii, USA, June 13-20, 1999*
- [9] J. Gołębiowski, J.W. Peterson: "Stationary Thermal Field in Long Duct of Electrical Heating System", *Archiv fur Electrotechnics, Springer Verlag, Berlin, Vol. 79, No 1*
- [10] Computational electronic, D. Vasileska, S.M Goodnick/Materials Science and Engineering R 38, 2002

- [11] Rodrigo Escobar, Brian Smith, Cristina Amon : “ Lattice Boltzmann Modeling of Subcontinuum Energy Transport in Crystalline and Amorphous Microelectronic Devices” , Transactions of the ASME Vol. 128, JUNE 2006.
- [12] G. Chen, D. Borca-Tasciuc, R. G. Yang: ‘‘ Nanoscale Heat Transfer ’’ Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, Encyclopedia of Nanoscience and Nanotechnology Edited by H. S. Nalwa Volume X: Pages (1–30)
- [13] Yunfei Chen Deyu Li, Jennifer R. Lukes, Arun Majumdar: “Monte Carlo Simulation of Silicon Nanowire Thermal Conductivity”, Journal of Heat Transfer OCTOBER 2005, Vol. 127 / 1135.
- [14] Gianluca Fiori, Marco G. Pala, and Giuseppe Iannaccone: “Three-Dimensional Simulation of Realistic Single Electron Transistors”, IEEE Transactions on Nanotechnology, Vol. 4, No. 4, 2005.
- [15] E.Pop, D.Mann, K.Goodson and H.Dai, "Electrical and Thermal Transport in Metallic Single-Wall Carbon Nanotubes on Insulating Substrates,« *Journal of Applied Physics*, vol.101, p.093710, 2007.
- [16] <http://www.NanoHub.org>, en date de mars 2009
- [17] V. Perebeinos, P. Avouris, “Impact Excitation by Hot Carriers in Carbon Nanotubes”, *Physical Review Letters* 74, p.121410, 2006.
- [18] ONGGUI YANG AND GANG CHEN, “Thermal conductivity modeling of periodic two-dimensional nanocomposites”, PHYSICAL REVIEW B 69, 195316, 2004)
- [19] <http://www.clumeq.mcgill.ca>, en date de mars 2009
- [20] <http://www.llnl.gov/computing/tutorials/mpi>, en date de mars 2009
- [21] Rodrigo Escobar, Brian Smith, Cristina Amon, “Lattice Boltzmann Modeling of Subcontinuum Energy Transport in Crystalline and Amorphous Microelectronic Devices”, JUNE 2006 Transactions of the ASME.
- [22] <http://www.c3.ca.>, en date de mars 2009
- [23] Nanotechnologies et environnement : Bénéfices et préoccupation- conférence du 8 décembre 2005.
- [24] Fabien Banse, Jean-Luc Dekeyser, Renaud Fauquembergue, “ parallélisation d’une méthode de Monte Carlo pour la simulation de semi-conducteurs”, Laboratoire d’informatique Fondamentale de Lille (LIFL), février 1999.
- [25] Springer Handbook of Nano-Technology, Bhushan Editor, 2004.
- [26] J. Bryzek, S. Roundy and al, “Marvelous MEMS”, IEEE Circuit and Devices Magazine, 2006.

- [27] J. Ping Sun, Georges I. Haddad and al., “Resonant Tunneling Diodes: Models and Properties”, Proceedings of IEEE, Vol. 86 No. 4, 1998.
- [28] P. Mazumder, S. Kulkarni, M. Bhattacharya, J. P. Sun, and G. I. Haddad, “Digital circuit applications of resonant tunneling devices,” this issue, pp. 664–686.
- [29] Aristides A. G. Requicha, “Nanorobots, NEMS, and Nanoassembly”, Proceedings of IEEE, Vol. 91 No. 11 , 2003
- [30] S. Hong and C.H. Mirkin, “A nanoplotter with both parallel and serial writing capabilities,” *Science*, vol. 288, pp. 1808-1811 2000.
- [31] Chris Phoenix, “Studying Molecular Manufacturing”, IEEE Technology and Society Magazine, 2004.
- [32] “Des nanotechnologies aux application”, Clefs CEA, No52, 2005.
- [33] Carbon Nanotubes: Synthesis, Structure, Properties, and Applications, Dresselhaus, M.S., Dresselhaus, G., Avouris, Ph., Springer-Verlag, Berlin, 2001.
- [34] L.X. Dong, “Nanorobotic Manipulations of Carbon Nanotubes”, Ph.D Dissertation of Nagoya University, March 2003.
- [35] Victor V. Zhirnov , James A. et al, “Emerging Research logic devices”, IEEE Circuit and Devices Magazine, 2005.
- [36] Katsuhiko Degawa, Takafumi Aoki, Tatsuo Higuchi, “A Single-Electron-Transistor Logic Gate Family and its Application”, IEEE Proceedings of the 34th International Symposium on Multiple-Valued Logic (ISMVL’04).
- [37] Masumi Saitoh, Hidehiro Harata, Toshiro Hiramoto “Room-Temperature Demonstration of Integrated Silicon Single-Electron Transistor Circuits for Current Switching and Analog Pattern Matching” IEDM 04.
- [38] Xiaohui Tang; Baie, X.; Bayot, V.; Van de Wiele, F.; Colinge, J.P., “An SOI single-electron transistor”, SOI Conference, 1999. Proceedings. 1999 IEEE International 4-7 Oct. 1999 Page(s):46 - 47
- [39] Jun Xu, T.S. Fisher, “Enhanced thermal contact conductance using carbon nanotube arrays”, Inter Society Conference on Thermal Phenomena, 2004
- [40] R. Martel, V. Derycke, J. Appenzeller, S. Wind, and Phi. Avouris, “Carbon Nanotube Field-Effect Transistors and Logic Circuits”, IEEE
- [41] A.-S. Teh, K.B.K. Teo, D.G. Hasko, H. Ahmed W.I. Milne, G.A.J. Amarahnga, “Carbon Nanotube Lateral Field Emission Devices”, IEEE 2003

- [42] Joerg Appenzeller, Joachim Knoch, Richard Martel, Vincent Derycke, Shalom J. Wind, and Phaedon Avouris, "Carbon Nanotube Electronics", IEEE Transactions on Nanotechnology, Vol. 1, No. 4, 2002.
- [43] Robert Vajtai, Bingqing Wei, Yung Joon Jung, Anyuan Cao, Sujit K. Biswas, Ganapathiraman Ramanath, and Pulickel M. Ajayan, "Building and Testing Organized Architectures of Carbon Nanotubes " IEEE Transactions on Nanotechnology, Vol. 2, No. 4, 2003.
- [44] Ju-Hyung Yun¹, Han Chang-Soo¹, Joondong Kim¹, Jin-Won Song¹, Dong-Hun Shin¹, and Young-Geun Park¹, "Fabrication of Carbon Nanotube Sensor Device by Inkjet Printing", Proceedings of the 3rd IEEE Int. Conf. on Nano/Micro Engineered and Molecular Systems, 2008.
- [45] Yi Huang and Wen-Yan Yin, "Performance Predication of Carbon Nanotube Bundle Dipole Antenna", Proceedings of Asia-Pacific Microwave Conference 2007.
- [46] Banoo, K., Jung-Hoon Rhew; Lundstrom, M.; Chi-Wang Shu; Jerome, J.W. "Simulating quasi-ballistic transport in Si nanotransistors", Computational Electronics, IWCE Glasgow 2000.
- [47] Svizhenko, A., Anantram, M.P., "The effect of scattering on drive current of nanotransistors", Device Research Conference, 2002.
- [48] Engstrom, K. E.; Kinaret, J. M., "Performance Gain Through Dynamic Control of Device Geometry: Nanoelectromechanical Carbon Nanotube-Based Switch", Electron Device Letters, IEEE Volume 27, 2006.
- [49] Svizhenko, A., Anantram, M.P., "Role of scattering in nanotransistors", IEEE Transactions on Electron Devices, Volume 50, 2003.
- [50] Zhibin Ren, Venugopal, R., Datta, S., Lundstrom, M., Jovanovic, D., Fossum, J, "The ballistic nanotransistor: a simulation study", Electron Devices Meeting, 2000. IEDM Technical Digest, 2000.
- [51] Ravariu, C., "A NOI-nanotransistor", Semiconductor Conference, 2005. CAS 2005 Proceedings, Volume 1.
- [52] Velling P., Agethen M., Prost W., Janssen G., Bertenburg R.M., Tegude F.J., "An intrinsically coupled HBT/RTD device enabling an adjustable peak-current-density" Device Research Conference, 2001.
- [53] Sing-Rong Li, Mazumder P., Chua L.O., "On the implementation of RTD based CNNs" Circuits and Systems, 2004. ISCAS '04.
- [54] Jian Ping Sun, Haddad G.I., Mazumder P., Schulman J.N., "Resonant tunneling diodes: models and properties", Proceedings of the IEEE, Vol 86, 1998.

- [55] Lind E., Lindstrom P., Nauen A., Wernersson L.-E., “Resonant tunneling permeable base transistor based pulsed oscillator”, Device Research Conference, 2004.
- [56] Pacha C., Goser K., Brennemann A., Prost W., “A threshold logic full adder based on resonant tunneling transistors”, Solid-State Circuits Conference, 1998.
- [57] Ando Y., Contrata W., Maruhashi, K., Miyamoto H., “Microwave noise properties for resonant tunneling transistors (RTTs)”, Compound Semiconductors, IEEE International Symposium on 8-11 Sept. 1997 Page(s):609 - 612 .
- [58] Kolagunta V.R., Janes D.B., Melloch M.R., Youtsey C., “Quasi one-dimensional confinement in double-well sidewall gated resonant tunneling transistors”, Compound Semiconductors, IEEE International Symposium on 8-11 Sept. 1997.
- [59] Lind E., Lindstrom P., Wernersson L.-E., “Resonant tunneling permeable base transistors with high transconductance”, Electron Device Letters, IEEE Volume 25, Issue 10, Oct. 2004.
- [60] Albrecht T., Guckian A., Ulstrup J., Vos J.G., “Transistor effects and in situ STM of redox molecules at room temperature”, Nanotechnology, IEEE Transactions on Volume 4, 2005.
- [61] Shiou-Ying Cheng, “In_{0.49}Ga_{0.51}/GaAs superlatticed resonant-tunneling transistor (SRTT)”, Indium Phosphide and Related Materials, International Conference on 14-18 May 2000.
- [62] Kyung Rok Kim, Dae Hwan Kim et al., “Silicon-based field-induced band-to-band tunnelling effect transistor”, Electron Device Letters, IEEE Volume 25, June 2004.
- [63] Katayama Y., Laux S.E., “Simulation study of tunneling devices with quantum confinement in source and drain”, Device Research Conference, 2004.
- [64] Ning Deng, Liyang Pan, Lei Zhang, Peiyi Chen, “Ge quantum dot memory realized with vertical Si/SiGe resonant tunneling structure Junction Technology”, IWJT '04.
- [65] Ting D.Z.-Y., Cartoixa X., Chang, Y.-C., “Nonmagnetic resonant tunneling spin devices”, Compound Semiconductors. International Symposium on 25-27 Aug. 2003.
- [66] Lageweg C., Cotofana S., Vassiliadis S., “Single electron encoded logic memory elements”, IEEE-NANO. Third IEEE Conference on Volume 1, 12-14 Aug. 2003.
- [67] Phillips J., Kamath K., Brock K.T., Bhattacharya P., “Room temperature self-organized quantum dot transistors”, Device Research Conference Digest, 1998. 56th Annual 22-24 June 1998.
- [68] Kim D.H., Lee J.D., Park B.G., Lee H.G., “Silicon single electron switch with an electrically formed quantum dot”, Device Research Conference Digest, 1998. 56th Annual 22-24 June 1998.

- [69] Melnikov D.V., Kim J., Zhang L.-X., Leburton J.-P., “Three-dimensional self-consistent modelling of spin-qubit quantum dot devices”, *Circuits, Devices and Systems*, IEE Proceedings, Volume 152, 2005.
- [70] Arakawa Y., “Prospects of Quantum Dots for Advanced Nanophotonic Devices”, *Lasers and Electro-Optics - Pacific Rim*, 2007.
- [71] Morris J.F., Wu F., Radehaus C., Hietschold M., Henning A., Hofmann, K., Kiesow A., “Single electron transistors: modeling and fabrication”, *Solid-State and Integrated Circuits Technology*, 2004.
- [72] Otani Y., Kimura T., “Spin Current and Spin Hall Effect in Metallic Nanostructures”, *Magnetics*, IEEE Transactions on Volume 44, Issue 7, Part 2, July 2008 Page(s):1911 – 1915
- [73] Pramanik S., Bandyopadhyay S., Cahay, M., “Spin transport in nanowires”, *Nanotechnology*, 2003. IEEE-NANO 2003. 2003 Third IEEE Conference on Volume 1, 12-14 Aug. 2003 Page(s):87 - 90 vol.2
- [74] Waag A., “Magnetic materials for spin electronics”, *Optoelectronic and Microelectronic Materials and Devices*, 2000. COMMAD 2000. Proceedings Conference on 6-8 Dec. 2000 Page(s):246 – 252
- [75] Samarth N., Awschalom D., “Spin control in semiconductors: Variations on a theme”, *Physics of Semiconductor Devices*, 2007. IWPSD 2007. International Workshop on 16-20 Dec. 2007 Page(s):8 – 10
- [76] Li F.X., Tabib-Azar M., Mann J.A., “Theoretical analysis of single electron spin surface detections” *Nano/Micro Engineered and Molecular Systems*, 2008. NEMS 2008. 3rd IEEE International Conference on 6-9 Jan. 2008 Page(s):1071 – 1074
- [77] Stres S., Pestotnik R., “Study of possibilities for a spin flip in high energy electron ring HERA”, *Nuclear Science*, IEEE Transactions on Volume 53, Issue 2, April 2006 Page(s):484 - 490
- [78] Stres S., “Spin flip in high energy electron rings”, *Nuclear Science Symposium Conference Record*, 2007. NSS '07. IEEE Volume 2, Oct. 26 2007-Nov. 3 2007 Page(s):1687 – 1690
- [79] Das Sarma S., Fabian J., Xuedong Hu, Zutic I., “Theoretical perspectives on spintronics and spin-polarized transport”, *Magnetics*, IEEE Transactions on Volume 36, Issue 5, Part 1, Sept 2000 Page(s):2821 – 2826.
- [80] Demokritov S.O., Hillebrands B., Slavin A.N., “Confined dynamic excitations in structured magnetic media”, *Magnetics*, IEEE Transactions on Volume 38, Issue 5, Part 1, Sept. 2002 Page(s):2502 – 2507.

- [81] Saikin S., Min Shen, Cheng M.-C., “Study of spin-polarized transport properties for spin-FET design optimization”, *Nanotechnology*, IEEE Transactions on Volume 3, Issue 1, March 2004 Page(s):173 – 179.
- [82] Ansermet J.-P., “Thermodynamic Description of Spin Mixing in Spin-Dependent Transport”, *Magnetics*, IEEE Transactions on Volume 44, Issue 3, March 2008 Page(s):329 – 335.
- [83] Johnson M., “Optimized Device Characteristics of Lateral Spin Valves”, *Electron Devices*, IEEE Transactions on Volume 54, Issue 5, May 2007 Page(s):1024 – 1031.
- [84] Fert A., George J.-M., Jaffres H., Mattana R., “Semiconductors Between Spin-Polarized Sources and Drains”, *Electron Devices*, IEEE Transactions on Volume 54, Issue 5, May 2007 Page(s):921 – 932.
- [85] Sugahara S., “Spin MOSFETs as a basis for integrated spin-electronics”, *Nanotechnology*, 2005. 5th IEEE Conference 11-15 July 2005 Page(s):142 - 145 vol.1.
- [86] Lo C.K., Huang Y.W., Yao Y.D., Huang, D.R.; Huang, J.H., “Spin transistor for magnetic recording”, *Magnetics*, IEEE Transactions on Volume 41, Issue 2, Feb. 2005 Page(s):892 – 895.
- [87] Pascal Viot, “Simulation numérique en physique statistique”, Laboratoire de Physique Théorique de la Matière Condensée.
- [88] Santos E.E., Gayathri Muthukrishnan, “Efficient simulation based on sweep selection for 2-D and 3-D Ising spin models on hierarchical clusters”, *Parallel and Distributed Processing Symposium*, 2004. Proceedings 18th International 26-30 April 2004 Page 229.
- [89] Nakamura Y., Tucker J.W., “Monte Carlo study of a mixed spin-1 and spin-3/2 Ising ferromagnet”, *Magnetics*, IEEE Transactions on Volume 38, Issue 5, Part 1, Sept. 2002 Page(s):2406 – 2408.
- [90] David D., Diosan L., Dumitrescu D., “A New Nature-Inspired Computation Model Ising Model with Rays”, *Symbolic and Numeric Algorithms for Scientific Computing*, 2005. Seventh International Symposium on 25-29 Sept. 2005 Page(s):315 – 320
- [91] Yuliang Wang, Lieberman M., “Thermodynamic behavior of molecular-scale quantum-dot cellular automata (QCA) wires and logic devices”, *Nanotechnology*, IEEE Transactions on Volume 3, Issue 3, Sept. 2004 Page(s):368 – 376.
- [92] Xuekun Kou, Novotny M.A., Rikvold P.A., “Simulation of magnetization switching in biaxial single-domain ferromagnetic Particles”, *Magnetics*, IEEE Transactions on Volume 36, Issue 1, Part 2, Jan. 2000 Page(s):231 - 240

- [93] Bondarenko A.N., Lomov A.V., “Phase transition in fractals ferromagnetics models”, Science and Technology, 2003. Proceedings KORUS 2003. The 7th Korea-Russia International Symposium on Volume 3, 28 June-6 July 2003 Page(s):65 - 70 vol.3
- [94] Bowler K.C., Pawley G.S., “Molecular dynamics and Monte Carlo simulations in solid-state and elementary particle physics”, Proceedings of the IEEE Volume 72, Issue 1, Jan. 1984 Page(s):42 – 55
- [95] Loeliger Hans-Andrea, Molkarai Mehdi, “Simulation-based estimation of the partition function and the information rate of two-dimensional models”, Information Theory, 2008. ISIT 2008. IEEE International Symposium on 6-11 July 2008 Page(s):1113 – 1117.
- [96] Li D., Wu Y., Kim P., Shi L., Yang P., and Majumdar A., 2003 “Thermal conductivity of individual Silicon Nanowires”, Appl. Phys. Lett., 83.
- [97] http://rsc.anu.edu.au/~evans/EM_Book/Chapter_4.pdf , en date de mars 2009
- [98] Liu W., Etesam-Yazdani K., Hussin R., Asheghi M., “Modeling and Data for Thermal Conductivity of Ultrathin Single-Crystal SOI Layers at High Temperature”, IEEE Transactions on electron devices, Volume 53, NO. 8, august 2006.
- [99] M. G. Holland, “Analysis of lattice thermal conductivity,” Phys. Rev., vol. 132, no. 6, pp. 2461–2471, Dec. 1963.
- [100] Advanced computer architecture and parallel processing Par Hesham El-Rewini, Mostafa Abd-El-Barr.
- [101] S. Volz, J.B Saulnier, G. Chen, P. Beauchamp, “Computation of thermal conductivity of Si/Ge superlattices by molecular dynamics techniques”, Microelectronics Journal 31 (2000) 815–819
- [102] T. Luo, J. R. Lloyd “Equilibrium Molecular Dynamics Study of Lattice Thermal Conductivity/Conductance of Au-SAM-Au Junctions”, <http://arxiv.org/ftp/arxiv/papers/0905/0905.2154.pdf>, en date de mars 2009
- [103] Z. Yao, J.S. Wang, B. Li, G.R. Liu, “Thermal conduction of carbon nanotubes using molecular dynamics”, PHYSICAL REVIEW B 71, 085417 (2005)