

HABILITATION À DIRIGER DES RECHERCHES

AU TITRE DE L'ÉCOLE DOCTORALE
DE MATHÉMATIQUES ET D'INFORMATIQUE

Par **Olivier Coulaud**

Contributions algorithmiques pour les simulations complexes en physique des matériaux

Soutenue et présentée publiquement le : 29 novembre 2013

Après avis des rapporteurs :

Daniel Bouche Directeur de Recherche, CEA
Jocelyne Erhel Directrice de Recherche, INRIA
Olivier Pironneau Professeur, Université Pierre et Marie Curie

Devant la commission d'examen composée de :

Isabelle Baraille Professeur, Université de Pau et des Pays de l'Adour
Jean-Paul Caltagirone Professeur, Université de Bordeaux 1
Jean Roman Professeur, Institut Polytechnique de Bordeaux en détachement Inria.

Résumé

Avec l'accroissement de la puissance des ordinateurs, les simulations numériques de phénomènes physiques deviennent de plus en plus complexes. Cette complexité provient d'une part, de l'ajout de modèles physiques plus compliqués pour mieux représenter la physique et d'autre part, par une discrétisation plus fine du problème conduisant à des problèmes de grande taille. Une difficulté supplémentaire s'ajoute avec la complexification des ordinateurs notamment par des mémoires et des unités de calculs hiérarchiques. Il est donc nécessaire de redessiner les algorithmes, les schémas numériques et les logiciels pour les rendre plus efficaces sur ces architectures émergentes. Le travail présenté dans ce document correspond à deux axes de recherche. Le premier porte sur le développement et la parallélisation de la méthode des multipôles rapides pour calculer rapidement des interactions de paires. Le deuxième axe concerne le couplage de méthodes et de codes en physique des matériaux à échelle atomique ainsi que le pilotage de ces simulations.

Abstract

Because computer performance is always increasing, the numerical simulations of physical phenomena become much harder. Where does this complexity come from? On the one hand, for a higher fidelity of the solution problem, more advanced physical models are introduced; on the other hand, the size of the discretization is shrinking which leads to very large-scale numerical problems. In addition the computers themselves become more complex, for instance with hierarchical memories and hierarchical process units. Hence the need to redesign algorithms, numerical schemes and software to make them more efficient and effective on emerging architectures.

The work layout in this document is divided in two mainlines. First, the development and the parallel design of the fast multipole method to compute pair-wise interactions. Then the coupling of models/methods and codes in material physics at atomistic scale, and the computational steering of these simulations as well.

Remerciements

Combien de fois ai-je entendu tout au long de ces dernières années entre Nancy et Bordeaux : « alors ton HDR, c'est pour quand ? ». Et bien ça y est, je l'ai enfin rédigée.

Je tiens avant toute chose à remercier Daniel Bouche, Jocelyne Ehrel et Olivier Pironneau qui ont accepté d'évaluer mon travail.

Merci à Daniel d'avoir initié la collaboration avec le CEA et notamment mon activité sur la propagation des fissures.

Je remercie Isabelle Baraille que j'ai connu au travers de l'ANR NOSSI et qui m'a fait l'amitié d'être dans mon jury et qui patiemment a tenté de m'expliquer les subtilités de la chimie quantique.

Merci à Jean Roman, non seulement d'avoir accepté de présider ce Jury, mais aussi de m'avoir donné la possibilité de migrer vers le Sud-Ouest avant l'implantation d'Inria à Bordeaux.

Un remerciement spécial pour Jean-Paul Caltagirone qui a été mon directeur de thèse, il y bien longtemps et qui m'a donné le goût pour la recherche en calcul scientifique sans perdre de vue les applications et la validation expérimentale.

Un grand merci aux membres du projet ScAlAplix puis de Hiepac pour les fructueuses discussions que nous avons eu tout au long de ces années. Merci à Luc Giraud pour ses conseils avisés et pour m'avoir poussé à finaliser ce mémoire.

Merci aux doctorants, post-doctorants, ingénieurs et stagiaires qui ont travaillé avec moi et sans qui ce travail n'aurait pas pu prendre toute cette importance. Je ne citerai que Pierre Fortin, Guillaume Anciaux, Aurélien Esnard, Bérenger Bramas, mais vous êtes nombreux à m'avoir accompagné . . .

Ma dernière pensée se tourne vers ma famille, tout particulièrement mon épouse Martine, mes enfants Jérémie, Thibault et Guillaume. C'est grâce à leurs encouragements et leur patience que maintenant je peux dire « c'est fait ! ».

Table des matières

1	Méthode des multipôles rapide	3
1.1	Présentation	3
1.2	Formulation matricielle de la FMM	5
1.2.1	Accélération de l'opérateur M2L	5
1.2.2	Formulation adaptative	6
1.3	Parallélisme	8
1.3.1	Parallélisme hybride	9
1.3.2	Parallélisation en tâches	10
2	Simulations complexes en physique des matériaux	13
2.1	Simulation de fissures	14
2.2	Algorithmes pour le calcul de spectres optiques dynamiques de molécules dans leur environnement	14
2.2.1	Accélération de la TD-DFT	15
2.2.2	Algorithme de couplage	20
2.3	Pilotage de simulations	22
3	Perspectives	25
3.1	Méthode multipôle rapide	25
3.1.1	Vers un parallélisme massif	25
3.1.2	Nouveaux champs d'applications	26
3.2	Simulation à grande échelle en physique des matériaux	26
3.2.1	Dynamique des dislocations	26
3.2.2	Calcul rapide de valeurs propres	27

Le dénominateur commun de mon activité de recherche, depuis le début de ma carrière est le développement de schémas et d'algorithmes numériques pour accélérer les calculs dans les applications complexes issues de la physique. Ces applications ont évolué en fonction de mes collaborations. Durant les premières années de ma recherche, je me suis intéressé au magnétoformage puis sous l'impulsion de Gilles Kahn au domaine de la chimie computationnelle que j'ai étendu par la suite à la physique des matériaux. Dans toutes ces applications, j'essaye de contribuer sur toute la chaîne allant de la modélisation à l'implémentation efficace sur des machines parallèles. Cela se traduit par deux types de contributions : la première concerne l'amélioration des modèles, le développement d'algorithmes hiérarchiques ou multi-échelles. Le deuxième aspect porte sur le développement et l'implémentation efficace d'algorithmes parallèles en lien avec ces applications.

Mon activité de recherche en calcul scientifique a toujours été à l'intersection de deux domaines : mathématiques appliquées et informatique mais toujours en lien avec des applications. Depuis une dizaine d'années avec mon arrivée à Bordeaux, j'ai recentré mon activité sur l'étude et le développement de modèles et de méthodes hiérarchiques pour les simulations de grande taille en physique des matériaux. Du fait de la taille des applications visées, l'objectif de mes travaux consiste à améliorer les méthodes pour les rendre plus efficaces sur les architectures parallèles émergentes.

J'ai choisi de ne présenter dans ce document qu'une partie de mes activités de recherche depuis que j'ai intégré l'équipe SATANAS. J'ai sélectionné deux axes les plus représentatifs de mes travaux. Le premier concerne l'algorithmique pour le calcul rapide des interactions entre paires par la méthode des multipôles rapide. Cette activité est alimentée par les applications sur lesquelles je travaille. Au démarrage, ces travaux étaient issus des problèmes en chimie computationnelle et en astrophysique puis se sont étendus récemment à l'élasticité pour le calcul des forces en dynamique des dislocations. Le deuxième axe concerne les simulations complexes, notamment multi-échelles, en physique des matériaux. Les travaux dans cette thématique évoluent très fortement en fonction des collaborations que je tisse au travers de thèses, de contrats et de projets (ACI, ANR, ...). De plus, elles conduisent à des travaux plus numériques et algorithmiques ainsi qu'au développement d'outils pour faciliter la mise au point de ces simulations complexes.

Chapitre 1

Méthode des multipôles rapide

1.1 Présentation

Le calcul rapide des interactions entre paires de particules est central dans beaucoup d'applications provenant de domaines très différents. On peut citer les simulations en astrophysique, en physique des matériaux, en dynamique moléculaire mais aussi en acoustique, en électromagnétisme, ... Il s'agit d'évaluer le potentiel ou la force sur toutes les particules à chaque pas de temps. Cela conduit à évaluer la somme suivante :

$$V_i = \sum_{j=1}^N K(x_i, x_j) w_j \quad \forall i = 1, \dots, N, \quad (1.1)$$

où N est le nombre de particules, x_j est la position d'une particule, w_j sa grandeur physique (par exemple la charge) et $K(.,.)$ un potentiel de paire. Ce potentiel est en général une fonction de Green qui décroît en fonction de la distance. Nous ne nous intéresserons ici qu'à des potentiels non oscillants.

Depuis le milieu des années 80, de nombreux algorithmes ont été proposés pour réduire la complexité quadratique de l'évaluation directe de la somme [10, 37, 36, 71]. L'introduction de méthodes hiérarchiques basées sur des structures en arbre ou en octree a considérablement réduit la quantité de calcul nécessaire pour simuler ces systèmes pour une précision donnée. Deux catégories de méthodes ont été développées, à savoir la méthode basée sur des algorithmes en arbre développée à l'origine par J. Barnes and P. Hut [10] ayant une complexité en $O(N \ln(N))$ et la méthode des multipôles rapide par L. Greengard [37] avec une complexité linéaire.

Du fait de la décroissance du potentiel à l'infini, on le décompose en un champ proche et un champ lointain. Cette séparation est basée sur l'utilisation d'un octree ou d'un arbre. Le champ proche est calculé directement via l'expression (1.1). On parlera de calcul direct tandis que le champ lointain est approché. Il existe deux grandes classes de méthodes pour l'approcher. La première, la plus utilisée, consiste à faire des développements d'ordre P du potentiel. Deux développements sont généralement considérées : le développement de Taylor [9, 71, 75, 35] très utilisé en astrophysique et le développement en harmoniques sphériques [37, 61, 25, 15, 41]. Ces développements sont spécifiques à chaque noyau et doivent être établis pour chaque nouveau potentiel. La deuxième classe [5, 79, 31] est basée sur une interpolation du potentiel sur un ensemble de points. L'algorithme hiérarchique ainsi construit est indépendant du potentiel.

L'implémentation efficace de ces méthodes et leur parallélisation est un enjeu important dans différents domaines comme l'astrophysique, la dynamique moléculaire, la physique des matériaux, mais aussi tous les problèmes issus des formulations intégrales (électrostatique, acoustique, électromagnétisme, élasticité, ...). De nombreux travaux [24, 38, 15, 40, 39] ont porté sur

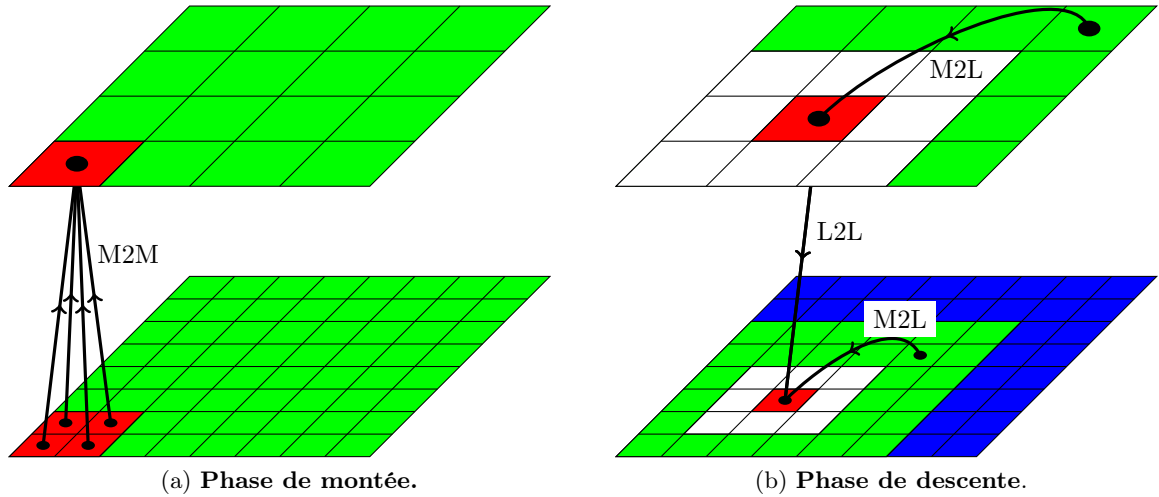


FIGURE 1.1 – Parcours de l’octree en $2D$. Les opérateurs du champ lointain (M2M, M2L et L2L) à des niveaux au-dessus du niveau feuille sont représentés.

l’amélioration de ces méthodes pour diminuer la complexité des opérateurs impliqués et accroître la scalabilité des implémentations parallèles de la méthode [9, 36, 68, 73, 79, 49, 42, 12, 50, 43, 14, 19, 21, 51].

L’algorithme pour évaluer le champ lointain est basé sur le parcours de l’octree et se décompose en deux parties : une phase montante et une phase descendante. Dans la phase montante Figure (1.1a), on part des feuilles (i.e. les cellules en bas de l’octree) pour calculer via l’opérateur **Particule-to-Multipole** (P2M) des multipôles sur la cellule. Il peut être vu comme une approximation de la grandeur physique impliquée dans le problème (masse, charge, ...) sur la cellule. Une fois ces multipôles construits, on remonte niveau par niveau dans l’arbre en appliquant l’opérateur **Multipole-to-Multipole** (M2M) pour accumuler les développements des cellules filles à la cellule père. Lorsqu’on arrive au sommet de l’arbre, tous les multipôles sont disponibles et on peut commencer la phase de descente pour évaluer le champ lointain dû aux multipôles. Pour cela, on utilise un critère de bonne séparation ; dans notre cas, deux cellules sont dites bien séparées si elles ne sont pas voisines. En l’absence de conditions aux limites périodiques et à cause de ce critère on ne remonte pas jusqu’à la racine mais on s’arrête au niveau 2 de l’arbre (grille du haut de la Figure (1.1)). Pour chaque cellule on évalue via l’opérateur **Multipole-to-Local** (M2L) les développements locaux dues aux cellules bien séparées des voisins du père, *c.-à-d.* les cellules vertes dans la Figure (1.1b). Ces cellules sont stockées dans une liste d’interaction de la cellule source. En dimension 3, pour un octree uniforme et sans cellule vide, il y a 189 cellules dans cette liste. Une fois les coefficients locaux évalués sur tous les niveaux supérieurs, l’opérateur **Local-to-Local** (L2L) translate ces coefficients à ses cellules filles. Ces deux étapes de la phase descendante, s’appliquent jusqu’au niveau des feuilles. La dernière étape consiste à appliquer l’approximation du champ lointain sur les particules via l’opérateur **Local-to-Particule** (L2P). Le champ proche correspond à l’interaction entre la cellule et les boîtes non séparées (interaction entre la cellule rouge et les cellules blanches dans la Figure (1.1b)). Il est évalué par l’opérateur **Particule-to-Particule** (P2P) par un calcul direct des interactions entre les boîtes voisines via l’expression (1.1).

Pour que l’algorithme soit efficace, la hauteur de l’octree est choisie pour que les temps de calcul entre le champ proche et le champ lointain soient équilibrés. Dans le calcul du champ

lointain, l'opérateur M2L est celui qui consomme le plus de temps.

L'utilisation des développements de Taylor d'ordre P conduit à considérer $(P + 1)^3$ termes pour les développements multipolaires et locaux tandis que si on utilise les développements en harmoniques sphériques il n'y a que $(P + 1)^2$ termes. C'est pourquoi lorsque l'on a besoin d'une précision élevée, *c.-à-d.* P grand, on préfère utiliser les développements en harmoniques sphériques. Dans les sections suivantes, nous nous plaçons dans le cadre des développements en harmoniques sphériques.

1.2 Formulation matricielle de la FMM

La complexité de l'algorithme original, [37], est en $O(P^4N)$ et le coût de l'opérateur M2L est dominant par rapport aux autres opérateurs impliqués dans le calcul du champ lointain. De nombreuses améliorations pour diminuer la complexité de cet opérateur ont été proposées dans la cadre des approximations en harmoniques sphériques. Ces améliorations ne portent que sur la dépendance en P dans la complexité de la méthode. On peut citer la méthode FFT par bloc [24, 38], les rotations [15, 40] et les ondes planes [39, 15]. La FFT par blocs conduit à des instabilités numériques, les ondes planes ramènent la complexité en $O((P^3 + P^2)N)$ mais introduit une autre approximation qu'il faut maîtriser. Dans [15], les paramètres des ondes planes pour conserver la précision de la méthode des multipôles sont donnés pour trois précisions : 3, 6 et 9 décimales. Pour terminer, l'utilisation des rotations dans les harmoniques sphériques ramène la complexité en $O(P^3N)$. Cette dernière optimisation s'applique aussi aux opérateurs M2M et L2L.

1.2.1 Accélération de l'opérateur M2L

En algèbre linéaire, il est bien connu que pour faire des produits matrice-vecteur, matrice-matrice efficaces, il faut utiliser des versions blocs plutôt que scalaires de ces opérateurs. En utilisant les opérations matricielles basées sur les bibliothèques BLAS (Basic Linear Algebra Subprograms, voir [52, 23, 22]) on peut atteindre des performances proches de la performance crête sur les processeurs modernes.

Dans l'article « *High Performance Blas Formulation of the Multipole-to-Local Operator in the Fast Multipole Method* », [17], nous présentons différentes formulations matricielles de l'opérateur M2L et des algorithmes pour utiliser des produits matrice-vecteur ou matrice-matrice de façon à maximiser l'utilisation des processeurs.

La méthode originale [37] ne permet pas d'écrire les opérateurs sous forme de produit matrice-vecteur. C'est pourquoi, nous avons considéré un formalisme plus simple développé par exemple dans [25], [75]. L'opérateur de transfert M2L entre les développements multipôles et les développements locaux peut s'écrire sous la forme matricielle suivante :

$$\begin{bmatrix} L_0^0 \\ L_1^0 \\ L_1^1 \\ L_2^0 \\ L_2^1 \\ L_2^2 \end{bmatrix} = \begin{bmatrix} O_0^0 & O_1^1 & O_1^0 & O_1^{-1} & O_2^2 & O_2^1 & O_2^0 & O_2^{-1} & O_2^{-2} \\ O_1^0 & O_2^1 & O_2^0 & O_2^{-1} & 0 & 0 & 0 & 0 & 0 \\ O_1^{-1} & O_2^0 & O_2^{-1} & O_2^{-2} & 0 & 0 & 0 & 0 & 0 \\ O_2^0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ O_2^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ O_2^{-2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} M_0^0 \\ M_1^{-1} \\ M_1^0 \\ M_1^1 \\ M_2^{-2} \\ M_2^{-1} \\ M_2^0 \\ M_2^1 \\ M_2^2 \end{bmatrix} \quad (1.2)$$

où M_ℓ^m sont les développements multipolaires pour $l = 0, \dots, P$ et $-\ell \leq m \leq \ell$ et L_ℓ^m les développements locaux mais en ne considérant que les indices positifs. On note T_{M2L} la matrice

dans l'expression ci-dessus. Comme on peut le voir, la matrice T_{M2L} est creuse. L'objectif est de découper le produit matrice-vecteur creux ou le produit matrice-matrice en plusieurs produits de même type dense. Pour des raisons de portabilité et d'évolution, nous avons choisi de découper ce produit en produits denses et quelques produits creux plutôt que de faire du déroulement de boucle, du blocage de boucle, ...

Le premier algorithme est basé sur des produits matrice-vecteur. Il consiste à découper la matrice en bloc dense. Pour cela, nous avons proposé une stratégie qui consiste à isoler le plus gros bloc dense (bloc gris foncé dans la Figure (1.2)) dans la matrice et à itérer de manière récursive dans les sous matrices restantes. Il faut remarquer que la matrice est identique pour

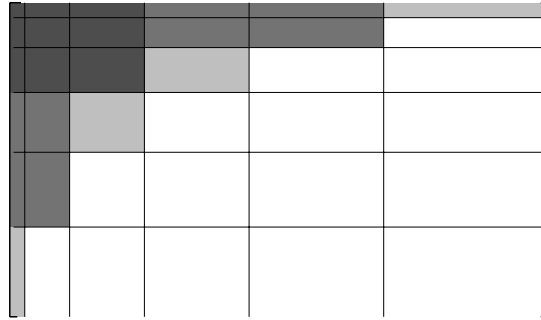


FIGURE 1.2 – Découpage bloc de la matrice de transfert T_{M2L} pour $P = 5$.

tous les couples de cellules ayant le même vecteur reliant le centre de ces cellules. On peut ainsi regrouper des vecteurs de multipôles pour construire en un produit matrice-matrice les développements locaux associés. L'idée est de regrouper le maximum de développements pour une matrice donnée. Ce regroupement se fait niveau par niveau en fonction du type des cellules et à condition que la liste d'interactions soit complète. Plusieurs algorithmes sont détaillés en fonction du stockage des développements pour diminuer le surcoût de la copie. La meilleure solution consiste à stocker côte à côte les développements des cellules qui ont le même indice de Morton (local) ligne par ligne. Ceci est fait à chaque niveau de l'octree. Le nombre de vecteurs que l'on peut faire ensemble augmente lorsque la taille de l'arbre augmente. Ce stockage est appelé *raw storage*. Nous utilisons le même algorithme que précédemment concernant la matrice.

Cette approche basée sur l'utilisation soit de produit matrice-matrice soit matrice-vecteur donne des temps de calculs, dans la plupart des cas, meilleurs que les approximations comme la FFT, les rotations et les ondes planes. Cette approche marche bien pour des distributions uniformes et pour les précisions requises en astrophysique et en dynamique moléculaire (entre 10^{-2} et 10^{-7}).

Les améliorations à l'aide des FFT par bloc et des rotations ont été intégrées dans le code FMB (Fast Multipole with Blas) pour les comparer avec notre approche. Pour les ondes planes nous avons utilisé un code tiers.

1.2.2 Formulation adaptative

La méthode des multipôles rapide (FMM) basée sur un arbre uniforme, *c.-à-d.* les feuilles sont toutes au même niveau, conduit pour des distributions de particules non uniformes comme en astrophysique pour la simulation de galaxie à un coût prohibitif en mémoire. Dans ce cas, beaucoup de cellules sont présentes dans l'arbre mais vides. Pour éviter ce stockage inutile, des méthodes appelées adaptatives ont été introduites [13, 15, 61].

Pour construire l'octree, on fixe le nombre de particules maximal, s_{max} , dans une feuille. Lorsque ce nombre est atteint on découpe la cellule en 8 fils et les particules sont distribuées

entre ces fils. Deux cellules voisines ne sont donc plus forcément de même taille. Ce type d'octree impose d'introduire cinq types de listes d'interactions (Figure 1.3) contre deux dans le cas uniforme (en comptant la liste des voisines pour l'opérateur P2P). Cette approche a été développée

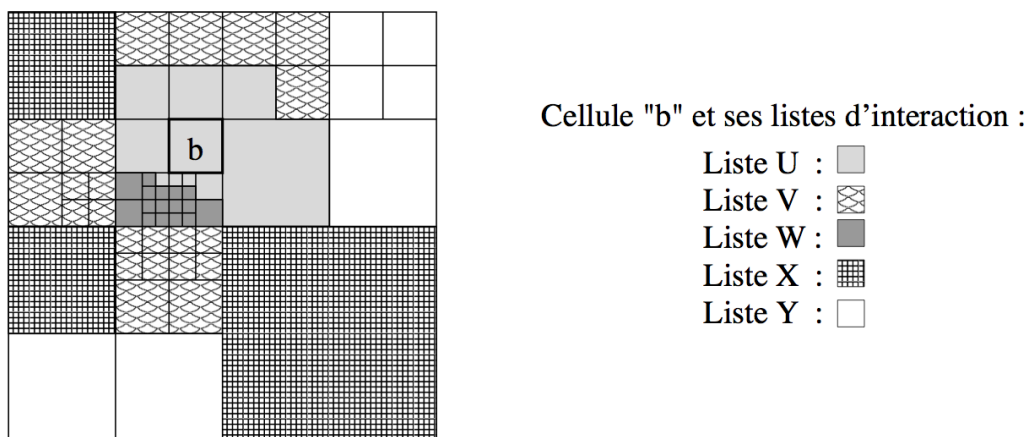


FIGURE 1.3 – Listes d'interaction de la FMM adaptative pour un quadtree (d'après l'exemple donné dans [13])

dans [13] et améliorée par Cheng *et al.* [15] avec l'introduction d'autres opérateurs (M2P et P2L) lorsque le nombre de particules est faible dans une feuille. Toutefois, cette approche ne garantit pas une complexité linéaire. On peut construire des distributions conduisant à une complexité en $O(N \ln(N))$.

Une autre approche a été introduite dans [61, 64] où au lieu de fixer le nombre de particules par feuille, on se donne la hauteur de l'octree. Nous obtenons ainsi une octree creux *i.e.* avec des cellules vides. Deux améliorations ont été introduites : les cellules adaptatives et un critère pour utiliser les opérateurs M2P et P2L. Cette méthode est très compétitive par rapport à la méthode précédente ; de plus, la complexité linéaire est garantie quelle que soit la distribution de particules [61].

Dans l'article « *High performance BLAS formulation of the multipole-to-local operator in the fast multipole method* », [17], nous présentons une adaptation de la méthode introduite dans [61] à notre formalisme matriciel avec notamment de nouvelles structures de données pour garder un compromis entre vitesse d'exécution et encombrement mémoire. Nous avons introduit un octree avec indirections, Figure (1.4), constitué d'un assemblage d'octree de hauteur fixe (hauteur 3 dans cette figure). De plus, nous introduisons une détection des zones uniformes pour optimiser les calculs. Les principaux résultats obtenus sont :

- Une structure de données efficaces. L'octree déséquilibré est découpé en plusieurs sous-octree de taille constante h . Nous obtenons ainsi un assemblage de sous-octree comme le montre la Figure (1.4).

A chacune de ces niveaux d'indirection $h, 2h, 3h, \dots$, on stocke dans un tableau linéaire les adresses des sous-octree en fonction de l'indice de Morton de leur racine.

- Sous certaines hypothèses sur la distribution, nous pouvons améliorer notre formulation BLAS de l'opérateur M2L. Nous recherchons des zones cubiques ou rectangulaires avec des cellules ayant à la fois des cellules sources et cibles ; ces zones sont appelées « *zones uniformes* ». Dans ces zones, le coût additionnel des copies pour permettre d'utiliser les BLAS peut être caché par le gain que l'on obtient. De plus, les matrices peuvent être pré-calculées. La détection de ces zones se fait en deux étapes. Dans la phase montante, les cellules ayant 8 enfants avec à la fois des développements sources et cibles, ont un numéro $1 +$ le numéro le plus petit des cellules filles. Cette phase peut se faire en même temps

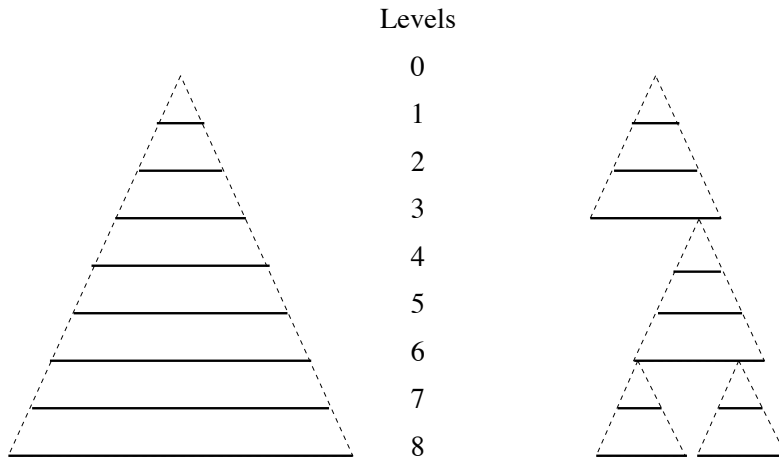


FIGURE 1.4 – Octree classique (à gauche) et octree avec indirection (à droite).

que la phase montante de l’algorithme FMM. Plus le label sera grand, plus profonde sera la zone uniforme. Dans la phase de descente, on parcourt les cellules voisines pour augmenter ces zones uniformes. Une fois cette phase terminée, on pourra effectuer la descente de l’algorithme FMM en utilisant le formalisme BLAS dans ces zones détectées.

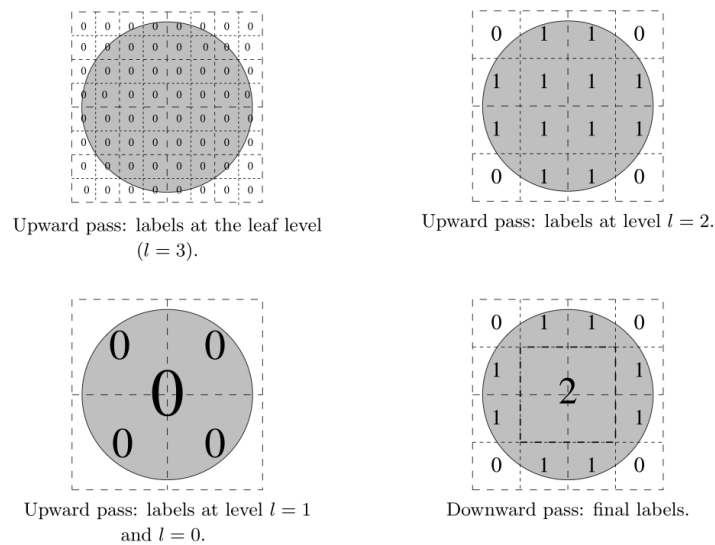


FIGURE 1.5 – Détection de zone uniforme

- Les résultats montrent que lorsque la zone uniforme est assez grande en pratique supérieure à 5, les gains en temps de calcul peuvent aller jusqu’à 50%.

1.3 Parallélisme

Historiquement, la première décomposition proposée pour les algorithmes à N-corps était le bipartitionnement récursif (ORB) du domaine géométrique pour l’algorithme de Barnes et Hut [54, 71]. D’autres décompositions [68, 72] ont été introduites qui s’appuient directement sur la numérotation des boîtes de l’octree à l’aide des courbes de remplissage de l’espace (Morton ou Hilbert) et de fonctions de coûts. Dans [68, 72, 64] il a été montré que ces dernières décomposi-

tions sont plus efficaces que la décomposition ORB aussi bien pour des architectures à mémoire partagée et à mémoire distribuée.

1.3.1 Parallélisme hybride

Dans l'article « *Hybrid MPI-Thread Parallelization of the Fast Multipole Method* » [16], nous présentons une parallélisation hybride de la FMM pour à la fois les distributions uniformes et non uniformes pour l'équation de Laplace. Nous considérons une numérotation 1D des boîtes issue de la numérotation de Morton avec des fonctions coûts.

Nos principales contributions sont :

- Dans le cas d'une parallélisation mémoire partagée, le choix s'est porté sur l'utilisation des threads Posix. La distribution des cellules se fait niveau par niveau et suivant l'indice de Morton en pondérant les cellules par un coût. Ce dernier est construit à partir des coûts du calcul du champ proche et de ceux des différents opérateurs du champ lointain. Nous avons ainsi un bon équilibrage dans tout l'octree. Il reste à gérer les conflits d'écriture au sein d'une cellule lorsqu'elle intervient dans un opérateur P2P et M2L entre deux cellules de deux threads différents. Pour cela un mécanisme d'exclusion mutuelle est mis au niveau des cellules avec des files d'attente FIFO pour traiter les calculs bloqués. Pour limiter les dépendances entre les calculs et minimiser le temps d'attente des threads, le champ lointain (opérateur P2P) est évalué juste après la phase de montée dans l'octree et avant la phase de descente. Dans le cas du formalisme basé sur les BLAS, nous avons choisi du fait de la faible taille des matrices non pas d'utiliser une version multithreadée du produit matrice-matrice ou matrice-vecteur mais de considérer que chaque thread fait un produit matrice-matrice.
- Dans la version hybride, le coût des communications impose de n'avoir plus qu'une seule distribution des cellules dans l'octree. Les feuilles sont distribuées suivant l'indice de Morton et en considérant le nombre de particules dans une feuille comme fonction de coût. Les cellules père sont attribuées aux processeurs qui possèdent le premier fils dans la numérotation de Morton. Ainsi les communications pour les opérateurs P2M, M2M, L2L et L2P sont minimisées. Contrairement au cas multithreadé on n'utilise pas le principe d'interaction mutuelle entre deux feuilles appartenant à deux processeurs différents. Les communications des différentes phases sont entrelacées avec le calcul. Nous obtenons ainsi l'algorithme suivant :

Algorithme 1 : Recouvrement calcul communications

- Step 1** : Construction et envoi des messages pour le calcul direct
 - Step 2** : Phase de montée (P2M et M2M) (incluant les communications M2M)
 - Step 3** : Terminaison des communications pour le calcul direct et traitement des messages
 - Step 4** : Construction et envoi des messages pour le calcul des interactions
 - Step 5** : Calcul direct (P2P)
 - Step 6** : Terminaison des communications pour les listes d'interactions
 - Step 7** : Phase de descente
 - Step 8** : Évaluation
-

Les communications sont non bloquantes. Côté réception, un thread est dédié à la réception au plus tôt des messages, le « dépaquetage » se fait par un thread de calcul.

- L'efficacité est plus faible que dans le cas de la mémoire partagée. Pour récupérer cette efficacité, nous avons notamment travaillé sur l'ordonnancement des calculs au sein d'un nœud mémoire partagée pour maximiser le taux d'occupation des processeurs par du calcul en développant une version hybride mixant l'approche distribuée avec l'approche mémoire partagée. Sur 128 processeurs, nous obtenons une efficacité de 85% (resp. 75%)

pour des distributions uniformes (resp. non uniforme) jusqu'à 100 millions de particules.

Ces travaux sont issus en grande partie de la thèse de P. Fortin [33] financée par un contrat CNRS/Région et ont donné lieu à un prototype FMB (Fast Multipole with Blas). Ce prototype a permis dans le cadre d'une action de développement (ADT) Inria de développer le code ScalFMM. Il a été réalisée pour assurer une plus grande portabilité et générique tant en terme de noyaux que de parallélisation. Ce logiciel est diffusé en Open Source à <http://scalfmm-public.gforge.inria.fr/>.

1.3.2 Parallélisation en tâches

Les travaux ci-dessus ont tous été réalisés pour des développements en harmoniques sphériques du champ lointain. Dans le cadre de l'équipe associée FastLA avec Stanford et Berkeley, nous avons intégré la méthode *black-box* introduite par Fong et Darve [31] dans ScalFMM. Cette méthode basée sur l'interpolation de Chebyshev permet d'être indépendante du noyau et renforce le caractère générique de notre bibliothèque. Dans cette section, nous présentons une parallélisation mémoire partagée efficace pour les nœuds many-core.

Dans « *Optimized M2L Kernels for the Chebyshev Interpolation based FastMultipole Method* », [59], nous proposons une amélioration (produit tensoriel, symétries, ...) des opérateurs M2M, L2L et M2L qui rendent cette méthode aussi compétitive que la méthode classique basée sur des développements en harmoniques sphériques. La parallélisation à l'aide des threads Posix est très intrusive dans l'algorithme et les évolutions ainsi que les améliorations des algorithmes sont délicates. De part le caractère mature des compilateurs basés sur les directives OpenMP et avec l'arrivée de la programmation par tâche dans le standard OpenMP, la parallélisation par directives s'est imposée comme choix.

Dans le rapport « *Task-based FMM for Multicore Architectures*, [2] », soumis à SISC, nous explorons dans un premier temps différents modèles de parallélisation à l'aide d'OpenMP. Puis nous présentons une approche au-dessus du moteur d'exécution StarPU.

Concernant le modèle OpenMP, trois approches sont proposées. La première concerne une parallélisation uniquement sur les boucles des cellules, niveau par niveau. La deuxième approche consiste à remplacer le parallélisme par boucle par un parallélisme par tâche, niveau par niveau. Contrairement à la première approche, pour diminuer le surcoût de la gestion, on augmente la granularité des tâches. Chaque tâche ne traitera plus une cellule, mais un groupe de cellules voisines au sens de la numérotation de Morton. À partir de cette version en tâche, on propose une parallélisation qui permet d'entrelacer les calculs du champ proche avec ceux du champ lointain. Cette dernière version peut être jusqu'à trois fois plus rapide lorsque le coût de calcul du champ lointain est dominant. Toutefois, les synchronisations inhérentes à l'approche par niveau deviennent importantes lorsque le calcul du champ lointain domine fortement celui du champ proche.

Le modèle OpenMP (version 3.1) présente plusieurs limitations pour que l'on puisse augmenter la scalabilité de notre approche. Tout d'abord, il n'est pas possible de pouvoir exprimer la dépendance entre les tâches pour casser ces synchronisations. Ensuite, on ne peut pas donner des priorités entre les tâches. Pour terminer, le standard OpenMP ne supporte pas les accélérateurs. C'est pourquoi l'utilisation d'un moteur d'exécution va nous permettre de gérer les dépendances entre les données et de pouvoir utiliser les accélérateurs (GPU).

Pour cela, l'algorithme FMM est exprimé sous forme d'un « *directed acyclic graph* » (DAG), ce qui permet d'explicitement les dépendances de données entre les tâches. L'ordre d'insertion des tâches est important, pour aider l'ordonnanceur à minimiser les temps d'attente. Pour pouvoir lancer les tâches au plus tôt, nous avons modifié l'ordonnanceur de StarPU pour introduire des priorités différentes aux tâches. Celles-ci peuvent dépendre de leur position dans l'octree ainsi que du hardware sur laquelle elle peut être exécutée.

Cette approche nous permet dans le cas uniforme d'avoir un gain de temps de calcul entre 10% et 120% sur une machine à 160 cœurs et de pouvoir utiliser le même algorithme sur une machine avec 12 cœurs et 3 GPUs. Dans le cas non-uniforme avec un ratio de 1 à 100 sur le nombre de particules entre les cellules, les temps de calcul peuvent être divisés par 2.

Chapitre 2

Simulations complexes en physique des matériaux

En raison de l'augmentation de la puissance de calcul disponible, de nouvelles applications dans les nano sciences et la physique des matériaux apparaissent. On peut citer l'étude des propriétés de nouveaux matériaux (peintures écologiques, matériaux photovoltaïques, biocapteurs,...), l'étude des défauts dans les matériaux, ... Ces simulations sont largement utilisées par les chimistes et les physiciens pour étudier ces matériaux. Ces calculs peuvent aller jusqu'à simuler des systèmes de plusieurs centaines de millions d'atomes, pour des échelles de temps pouvant aller jusqu'à quelques micro secondes. Toutefois, plus la simulation implique un nombre important d'atomes, moins la durée du phénomène simulée sera grande, sauf en dégradant la complexité des calculs (potentiel) et donc la précision des résultats. Aussi, si nous devons augmenter la précision, il faut réduire le coût du calcul. Pour cela, deux approches peuvent être envisagées. La première approche consiste à améliorer les modèles, les schémas et les algorithmes notamment en introduisant plus de parallélisme. Cette approche, que je qualifierais de classique, permet d'améliorer la rapidité des codes sans diminuer le volume d'inconnues que l'on traite. Il peut s'avérer nécessaire d'utiliser d'autres approches pour traiter des problèmes plus importants. Dans ce cas, on « *simplifie* » le modèle initial en introduisant des modèles multi-échelles. De nombreuses applications en physique des matériaux couplent plusieurs modèles [55], comme le couplage mécanique quantique et mécanique moléculaire (QM/MM) [80], ou le couplage entre des modèles moléculaires et des modèles mésoscopiques ou continus. Ces couplages permettent de traiter par exemple des solides ou des molécules dans leur environnement (cadre de l'ANR NOSSI). Beaucoup de phénomènes macroscopiques (comme la couleur du matériau) dépendent de phénomènes à des échelles plus petites. Simuler tout le matériau au niveau le plus fin n'est pas réaliste. La plupart du temps, le niveau le plus fin est seulement nécessaire là où le phénomène que l'on étudie survient. Par exemple, dans une simulation de propagation d'une fissure, loin de la pointe, *c.-à-d.* des discontinuités, la solution est « lisse » et nous pouvons utiliser un modèle plus grossier pour la calculer. L'idée est donc de limiter la simulation au niveau le plus fin à un sous-ensemble du domaine et de la coupler avec un niveau macroscopique. Cela implique que les simulations atomiques doivent être accélérées par plusieurs ordres de grandeur.

Les simulations multi-échelles intervenant dans beaucoup de domaines applicatifs, je me suis focalisé sur l'analyse, le développement et l'implémentation efficace de ces méthodes pour des problèmes en physique des matériaux en fonction de mes collaborations. Je montre des résultats pour deux applications différentes. Dans la première section, je présente des résultats issus de la thèse de G. Anciaux sur la propagation de fissures dans les matériaux. Ce travail a été réalisé dans le cadre d'un contrat avec le CEA-DAM-Ile de France. Dans la deuxième section, nous présentons des améliorations algorithmiques pour calculer des spectres optiques dynamiques des

molécules ou des solides dans leur environnement, réalisé dans le cadre de l'ANR CIS NOSSI.

2.1 Simulation de fissures

Dans cette section nous présentons les principaux résultats issus de la thèse de G. Anciaux [3], sur la propagation de fissures dans les matériaux. Ce travail a été réalisé en collaboration avec le CEA. L'objectif était de développer un simulateur multi-échelles pour étudier différentes hypothèses sur l'apparition de fissures dans le hublot et les lentilles du laser Megajoule. La taille de ces fissures était de l'ordre du micro-mètre, ce qui interdisait toute approche par dynamique moléculaire. De plus, le comportement des fissures dans les matériaux utilisés était mal connu, ce qui empêchait d'utiliser des approches en mécanique continue avec des éléments finis enrichis comme X-FEM [8]. D'où la nécessité de considérer une approche multi-échelles entre un modèle atomique et un modèle d'élasticité continu. De nombreuses approches existaient avec recouvrement ou sans recouvrement [70, 74, 20, 77, 56, 60].

Dans le papier « *High Performance Multiscale Simulation for Crack propagation* » [4], nous présentons une implémentation parallèle d'un couplage entre un modèle atomique et un modèle continu pour étudier la propagation d'une fissure. Le simulateur développé est basé sur la méthode « *bridging zone* » introduite par Belytschko and Xio [77]. Les avantages de cette approche sont nombreux. Tout d'abord, l'utilisation d'une zone de recouvrement évite de raffiner le maillage à l'échelle atomique. Ensuite, il permet d'amortir plus facilement les ondes hautes fréquences issues de la partie atomique. Pour terminer, l'algorithme permet de calculer de manière concurrente les déplacements atomiques et continus. Le recollement est réalisé en imposant une contrainte entre les positions atomiques et continues.

Nous présentons une approche parallèle du couplage à partir des codes parallèles LAMMPS, [62], et Stamp (code développé au CEA) avec un code éléments finis que nous avons développé, basé sur la bibliothèque LibMesh, [45]. Nous détaillons une structure de données efficace pour mettre en relation les atomes et les éléments finis dans la zone de recouvrement afin d'accélérer le temps de calcul de la partie recollement. Le code d'élasticité utilise une approche Lagrangienne (*c.-à-d.* le maillage est fixe dans le temps) tandis que les codes de dynamique moléculaire utilisent une approche Eulérienne. Nous avons dû développer un algorithme efficace pour garantir la cohérence de la structure lors de la migration des atomes entre deux processeurs. Pour terminer, l'approche a été validée sur un exemple en dimension 2 de propagation d'une fissure.

Dans la thèse de G. Anciaux [3], nous avons aussi amélioré la méthode de couplage pour diminuer la réflexion des ondes émises dans la partie moléculaire et défini une interface générique de notre coupleur, appelé LibMultiScale, pour pouvoir coupler facilement différents codes de dynamique moléculaire.

2.2 Algorithmes pour le calcul de spectres optiques dynamiques de molécules dans leur environnement

Les matériaux hybrides constitués de molécules organiques au sein d'une structure inorganique sont des systèmes intéressants pour beaucoup d'applications comme par exemple les peintures stables, la dépollution de l'air, ... Pour étudier, par exemple les propriétés optiques dynamiques de matériaux hybrides, il est nécessaire de coupler trois modèles. Un modèle atomique (MM) pour simuler l'évolution en temps du système complet (molécule cible et environnement), un modèle quantique (DFT) pour la molécule qui nous intéresse et un modèle (TD-DFT) pour calculer la réponse linéaire des états excités de cette molécule. Cela représente en général le couplage de trois codes parallèles MD/ DFT/TD-DFT.

Le développement de nouveaux outils pour étudier ces matériaux a été l'enjeu de l'ANR Calcul Intensif et Simulation « NOSSI : Nouveaux Outils de Simulation pour les Surfaces et les Interfaces ». Ce code MM/DFT/TD-DFT permet de déterminer le spectre de polarisabilité du colorant (molécule organique, comme l'indigo) en présence de la structure (système inorganique, par exemple une argile comme la palykorskite). Il repose sur le couplage de trois logiciels parallèles : le code SIESTA [7] (code utilisant des pseudo-potentiels et des orbitales atomiques numériques à support fini) pour le calcul quantique et le code DL_POLY 2 [32] pour la dynamique moléculaire classique. Les transitions électroniques du colorant sont déterminées par le code FAST développé dans le cadre de cette ANR.

Dans ce couplage, je me suis intéressé d'une part, au schéma de couplage entre le modèle quantique (DFT) et un modèle atomique (MM) et d'autre part, à l'accélération du code FAST.

Ces travaux sont en cours d'intégration dans la version développement du logiciel de chimie quantique SIESTA¹.

2.2.1 Accélération de la TD-DFT

Dans cette partie, je ne développerai pas la méthode de théorie de la fonctionnelle dépendante du temps (TD-DFT) ni la théorie de la réponse linéaire. J'indique juste que la méthode de la réponse linéaire appliquée à la méthode de la TD-DFT permet de calculer les transitions électroniques et les forces d'oscillateurs d'une molécule. Je renvoie le lecteur aux références suivantes [58, 57] pour plus de détails.

Dans le formalisme d'une combinaison linéaire d'orbital atomique (LCAO), la fonction réponse de la densité s'écrit dans la base des orbitales atomiques de Kohn-Sham

$$\chi^0(r, r', \omega) = \sum_{E, F} (n_F - n_E) \frac{\Psi_E(r) \Psi_F(r) \Psi_F(r') \Psi_E(r')}{\omega - (E - F) + i\epsilon} \quad (2.1)$$

où ϵ est un paramètre de régularisation, E (resp. F) l'énergie d'une orbitale occupée (resp. vide), Ψ_E (resp. Ψ_F) sa fonction d'état, $n_{i=E, F}$ le facteur d'occupation. En se ramenant aux orbitales localisées, le spectre de polarisabilité, P , pour la fréquence, ω , est constitué de la somme de trois termes chacun étant la solution des équations de Petersilka-Gossmann-Gross appliquées aux dipôles. Nous obtenons le système suivant à résoudre pour chaque fréquence

$$\begin{cases} P(\omega) = \sum_{i=1}^3 \langle d_i, X_i(\omega) \rangle \\ (1 - \chi^0(\omega)\Sigma) X_i(\omega) = \chi^0(\omega)d_i \end{cases} \quad (2.2)$$

où d_i est le dipôle dans la direction i , $\langle ., . \rangle$ représente le produit scalaire.

La résolution de ce système présente plusieurs difficultés. Tout d'abord, si M est le nombre d'orbitales atomiques, comme on considère le produit de ces fonctions, (cf. Equation (2.1)), la matrice que l'on construit est de taille $M^2 \times M^2$. Ces produits de fonctions ne forment pas une base et donc nous avons beaucoup de fonctions colinéaires que l'on pourrait supprimer. Pour terminer, lorsque le paramètre de régularisation, ϵ , est très petit, ce système peut devenir très mal conditionné lorsqu'une fréquence est proche de la différence $E - F$. Cela conduit dans certains cas à des valeurs non physiques de la polarisabilité.

Dans les sections suivantes, je présente des améliorations algorithmiques pour accélérer les calculs de ces spectres de polarisabilité.

1. <http://www.icmab.es/dmmis/leem/siesta>

Construction rapide de fonction de réponse de Kohn-Sham.

Dans l'article intitulé « *Fast Construction of the Kohn-Sham Response Function for Molecules* », [47], nous proposons une compression des produits d'orbitales moléculaires pour diminuer la taille des matrices à construire. Cette compression se fait en considérant des couples d'atomes ayant une intersection non vide des supports des orbitales associées à ces atomes. On construit la matrice associée aux produits de ces fonctions pour la norme coulombienne. Un algorithme de type SVD, nous permet de supprimer toutes les dépendances entre ces fonctions. Dans la suite de cette publication, nous explicitons un algorithme qui permet de calculer rapidement pour toutes les fréquences, ω , le coefficient $\chi_{0,i,j}(\omega)$ de la matrice de réponse. Le système linéaire est résolu par une méthode de Lanczos bi-orthogonale. L'avantage de cette méthode est que l'inverse du premier terme de la matrice tridiagonale est la polarisabilité à un facteur près. Cette approche nécessite de stocker une matrice $N^2 \times N_\omega$, où N est le nombre de produits que l'on garde, N_ω le nombre de fréquences. Pour finir, nous présentons des résultats pour les molécules d'indigo, fullerene C_{60} et les comparons avec des logiciels du domaine.

Algorithme itératif parallèle rapide

Dans l'approche précédente, la construction de la matrice pleine, χ_0 , pour toutes les fréquences conduit à un coût mémoire inenvisageable pour les molécules cibles, même en la distribuant sur plusieurs machines pour pouvoir la stocker en mémoire. Pour aller vers de grosses molécules ou des calculs intégrés dans des itérations d'une dynamique quantique comme pour le calcul de spectre dynamique, nous avons proposé, dans l'article « *A Parallel Iterative Method for Computing Molecular Absorption Spectra* », [48], d'une part une méthode itérative et d'autre part, une parallélisation hybride nous permettant de traiter rapidement de gros systèmes.

Le système linéaire (2.2) est résolu maintenant par la méthode de GMRES plus robuste que la méthode de Lanczos. Dans le cas des méthodes itératives, il est juste nécessaire de savoir faire un produit matrice-vecteur avec la matrice χ_0 . Aussi, nous proposons un algorithme qui évalue à la volée un tel produit et qui a l'avantage de ne pas stocker les coefficients de cette matrice. On peut montrer que la complexité obtenue est seulement cubique par rapport au nombre d'atomes du système.

Dans la deuxième partie de l'article, nous présentons une parallélisation mémoire partagée puis hybride pour accélérer les calculs. La parallélisation mémoire partagée est basée sur le standard OpenMP et on considère un parallélisme de boucles. Pour construire la matrice Σ , nous trions les couples d'atomes en fonction du nombre d'orbitales que l'on a gardé pour que la charge de calcul soit décroissante au cours des itérations de la boucle. De cette façon, les temps de synchronisation à la fin des boucles parallèles sont réduits.

Pour terminer, de nombreux résultats numériques sur des molécules variées sont présentés. Pour des molécules comme le benzène ou l'indigo, la scalabilité de la méthode donne de très bons speedups. La parallélisation hybride permet de calculer le spectre de polarisabilité de très grosses molécules (Chlorophyll-a, PCBM). Toutefois, pour ces grosses molécules avec une structure 3d comme le fullerene C_{60} la parallélisation choisie conduit à une saturation de la bande passante mémoire et à une perte de scalabilité.

Algorithme adaptatif

Dans l'approche ci-dessus, la distribution des fréquences est uniforme et le spectre complet est calculé. Pour les chimistes, les spectres ainsi obtenus ne sont pas significatifs ; il faut calculer les transitions électroniques ω_k^0 et la force d'oscillateur A_k associée. Pour cela, nous approchons

le spectre calculé par une somme de Lorentzienne normalisée

$$f_i(B, A_y, \omega_y^0) = B + \sum_{k=1}^N \frac{\epsilon}{\pi} \frac{A_k}{(\omega_i - \omega_k^0)^2 + \epsilon^2} - Y_i,$$

où nous devons déterminer N le nombre de transitions électroniques, ω_k^0 la position de la $k^{\text{ième}}$ transition et A_k son intensité. Ces coefficients sont calculés par une méthode de moindres carrés à l'aide du schéma de Levenberg-Marquardt de la GNU scientific library [1]. Comme le nombre de raies N est petit (moins de 5 en pratique), il est important pour bien évaluer les coefficients de concentrer les points vers les transitions. Pour des distributions uniformes de fréquences seulement un très faible pourcentage de ces points donne des valeurs non nulles de la polarisabilité. Cela nécessite donc d'utiliser beaucoup de points équi-répartis de façon en avoir suffisamment proches des transitions pour bien les approcher. Nous avons développé l'algorithme adaptatif suivant, qui permet de regrouper les points là où le spectre varie beaucoup.

La Figure (2.1) montre l'influence du paramètre de régularisation sur la forme du spectre de la molécule d'indigo. Cette courbe montre que pour les transitions les plus fortes en intensité, il n'est pas nécessaire de considérer ce paramètre trop petit. De plus, il nous précise la largeur de la Lorentzienne et le nombre de points que l'on doit calculer pour bien la représenter. L'algorithme 2

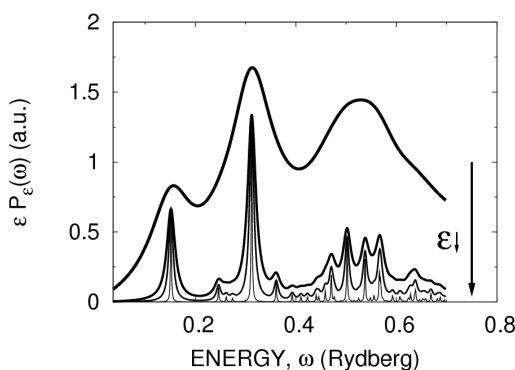


FIGURE 2.1 – Influence du paramètre de régularisation sur le spectre sur la molécule d'indigo.

détaille les différentes étapes de l'algorithme adaptatif que nous avons mis en place. On démarre un calcul de spectre à partir d'une distribution uniforme de fréquences.

Algorithme 2 : Algorithme Adaptatif.

Initialisation ($p = 1$) :

begin

 Start with a small number N_1 of uniformly distributed frequency points

$$\omega_j^1 = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \times \frac{j-1}{N_p}$$

 and a not too small value of $\epsilon = \epsilon_1$.

for $it = 1, iterMax$ **do**

 Compute P on the current ω_j .

Build the new distribution :
begin

 Define a set of uniformly spaced $U_i = i/N_p, i = 1, \dots, N_p$ in $[0,1]$.

step Dist

 Find ω_i such that $J(\omega_i) \leq U_i < J(\omega_{i+1})$

 The new improved frequency is $\omega_i = J_{\text{approx}}^{-1}(U_i)$
step Fit

 Perform the least squares fit and determine the transitions Ω_k and associated oscillator strengths f_k .

Check the convergence :

$$\max(\max_{k < N_L} \|\Omega_k^{it} - \Omega_k^{it-1}\|, \max_{k < N_L} \|f_k^{it} - f_k^{it-1}\|) < \mu$$

Reduce the regularization parameter if necessary.

Pour calculer la nouvelle distribution (step Dist) de points, nous construisons la fonction intégrale normalisée de P, J , qui est une fonction positive et croissante. La Figure (2.2) explicite le

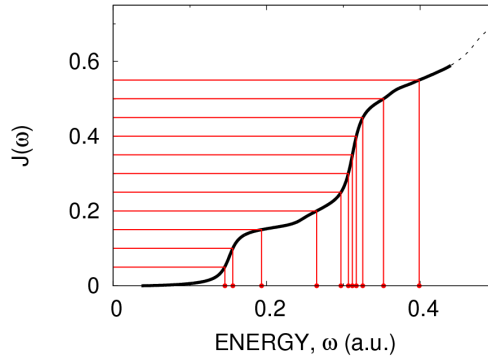


FIGURE 2.2 – Intégrale normalisée et calcul de la distribution non uniforme des fréquences.

calcul de cette distribution à partir d'une distribution uniforme de points de l'intervalle $[0, 1]$ de l'axe des ordonnées. On observe que les points vont naturellement se concentrer vers les pics du spectre (*c.-à-d.* les variations fortes de la courbe de l'intégrale normalisée). La distribution non uniforme de points est donnée en calculant les antécédents par J de la distribution uniforme. Dans l'étape du fit, (step Fit), on détermine d'abord le nombre de Lorentziennes impliquées dans le spectre en calculant le nombre de pics dans la courbe. La convergence est obtenue lorsque la condition suivante est satisfaite :

$$\max(\max_k (|\omega_k^{n+1} - \omega_k^n|), \max_k (|A_k^{n+1} - A_k^n|)) < \mu,$$

où n est le numéro de l'itération courante.

Cet algorithme permet plusieurs stratégies. On peut étant donné un premier calcul garder le même nombre de points dans la phase adaptative et obtenir un calcul plus précis ou partir d'un faible nombre de points et l'augmenter dans la phase adaptative pour atteindre rapidement la convergence. Nous considérons que le minimum de points pour bien représenter une Lorentzienne doit être de 4 sur un intervalle de longueur 2ϵ centré sur le pic. Avec ce choix, le nombre minimal de points pour bien approcher le spectre sur l'intervalle $[\omega_{min}, \omega_{max}]$ est

$$N_{\text{freq}} = 1 + \frac{3}{2\epsilon}(\omega_{max} - \omega_{min}). \quad (2.3)$$

Pour illustrer les performances de l'algorithme, on détermine les transitions dans le spectre visible de l'indigo pour des fréquences allant de $0.02Ry$ à $0.4Ry$. La Table (2.1) présente les résultats obtenues pour les stratégies suivantes

test 1 : Le nombre de points est constant et est donné par la formule (2.3). Ensuite, on améliore le placement des points jusqu'à convergence.

test 2 et 3 : Même approche que dans le *test 1* ; mais on démarre avec moins de points.

test 4 : On démarre avec le même nombre de points que dans le *test 3* : mais à chaque itération on augmente de 25% le nombre de points.

test 5 : On démarre avec un paramètre de régularisation plus grand, que l'on fait converger en deux itérations vers le paramètre final. À chaque itération, le nombre de points augmente de 25%.

	Test 1	test 2	test 3	test 4	test 5
$\epsilon^{\text{initial}}$ (Ry)	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	2×10^{-2}
ϵ^{final} (Ry)	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}
$N_{\text{freq}}^{\text{initial}}$	571	285	101	101	50
$N_{\text{freq}}^{\text{final}}$	571	285	101	198	255
N_{iter}	2	2	5	4	5
N_{eval}	1142	570	505	583	663
N_{trans}	7	7	4	6	7
Err_{ms}	3.0×10^{-4}	3.4×10^{-4}	1.7×10^{-3}	5.9×10^{-3}	4.2×10^{-4}
δf_{max}	2.2×10^{-4}	2.1×10^{-5}	1.0×10^{-4}	8.9×10^{-4}	6.2×10^{-4}

TABLE 2.1 – Résultats des différentes stratégies. N_{iter} nombre d'itérations pour un critère d'arrêt de $5.0 \cdot 10^{-2}$, $N_{\text{freq}}^{\text{initial}}$ (resp. $N_{\text{freq}}^{\text{final}}$) le nombre initial (resp. final) de fréquences, N_{eval} le nombre d'évaluations de \hat{P} ; Err_{ms} l'erreur obtenue par la méthode des moindres carrés à convergence; δf_{max} le maximum de la variation entre les forces d'oscillateur entre les deux dernières itérations à convergence.

Dans cet exemple, la meilleure stratégie est de partir d'un nombre de points de 50% voire 80% plus petit que le nombre minimal requis et d'améliorer le placement de ces points. Ces tendances sont identiques sur d'autres jeux de molécules.

Toutes ces améliorations nous permettent de calculer le spectre pendant une itération du code quantique SIESTA.

2.2.2 Algorithme de couplage

Le système que l'on considère est constitué d'une molécule organique et d'une structure inorganique. Deux cas peuvent se produire dépendant du fait que la molécule quantique touche ou ne touche pas la structure inorganique. Dans le cas où il n'y a pas de contact, nous avons deux domaines disjoints et il n'y a pas d'interface. Dans l'autre cas, la molécule organique se greffe sur la structure inorganique, il faut couper une liaison pour séparer les deux modèles. De nombreuses méthodes existent pour couper cette liaison. On peut citer la méthode de l'atome de liaison, de frontière et des orbitales gelées. Pour plus de détails, je renvoie le lecteur sur l'article de synthèse [80]. Dans le cas des pseudo-potentiels utilisés dans le code SIESTA, la coupure se fait via l'atome sur mesure [76].

Dans la suite, nous considérons que la molécule de colorant ne touche pas la structure inorganique (argile ou silice) : cela n'a pas d'importance pour l'algorithme de couplage que nous avons mis en place.

Le calcul quantique (code SIESTA) détermine l'énergie et les forces appliquées sur les atomes de la partie QM, lorsqu'ils sont soumis à un potentiel électrique dû à l'environnement, à savoir les atomes de la partie MM. Le code SIESTA est basé sur des pseudo-potentiels avec des bases localisées (support compact), et le calcul du potentiel d'Hartree nécessite de résoudre sur une grille cartésienne un problème de Poisson non périodique.

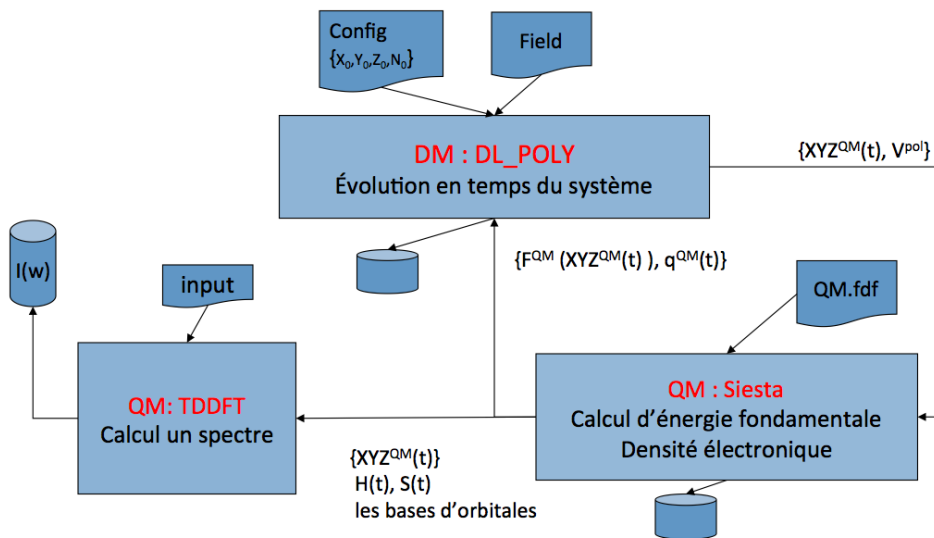


FIGURE 2.3 – Échanges des données dans le couplage.

La Figure (2.3) présente les différentes grandeurs qui sont échangées dans le couplage. Dans notre approche, le déplacement de tous les atomes (MM et QM) est fait par la dynamique moléculaire. Nous nous focaliserons uniquement sur le couplage QM/MM. A chaque itération, la première étape consiste côté dynamique moléculaire à évaluer le potentiel de polarisation de la molécule quantique, V^{ext} dû aux atomes de la partie MM, sur une grille grossière de la boîte quantique. Ce potentiel ainsi que la position des atomes quantiques, XYZ^{QM} sont envoyés au code quantique. Ce dernier interpole le potentiel de polarisation sur sa grille puis calcule l'état fondamental et les forces associées à chaque atome de la molécule ainsi polarisée. De plus, nous construisons des charges ponctuelles sur chaque atome à partir de la densité électronique. Pour cela, nous avons intégré dans SIESTA une méthode d'ajustement du potentiel électrostatique (ESP) [11, 34]. À la fin d'une itération quantique, le code SIESTA renvoie à la dynamique moléculaire les forces, $F^{QM}(XYZ^{QM})$, et des charges ponctuelles, q^{QM} , sur les

atomes quantiques. Côté dynamique moléculaire, après avoir calculé les forces atomiques en évitant de compter deux fois les contributions venant des atomes quantiques, on intègre en temps le système.

Le code SIESTA étant un code pour la physique des matériaux, seules les conditions aux limites périodiques y sont disponibles. Comme la dynamique moléculaire utilise des conditions aux limites périodiques, il fallait construire une boîte quantique aussi grande que la boîte de dynamique moléculaire pour limiter les effets de la périodicité [67]. Nous avons décidé de ne pas suivre cette approche et d'introduire dans le code SIESTA un solveur non périodique pour résoudre le problème de Poisson avec des conditions aux limites de Dirichlet. Ce solveur nécessite donc de connaître le potentiel sur la surface de la boîte quantique. Comme la densité ρ est connue, la solution du problème de Poisson, V , s'écrit $V(x) = \int_{\Omega} \frac{\rho(y)}{\|x-y\|} dy$. Nous développons cette expression à l'aide des harmoniques sphériques (comme dans la méthode FMM) pour pouvoir l'évaluer rapidement. Cette approximation fonctionne bien tant que le support de la densité est loin du bord. L'équation de Poisson est discrétisée par la méthode de différences finis avec un schéma compact d'ordre 4. Nous utilisons la bibliothèque hypre [29, 30] pour résoudre ce système à l'aide soit d'une méthode multigrille soit par un gradient conjugué préconditionné par multigrille.

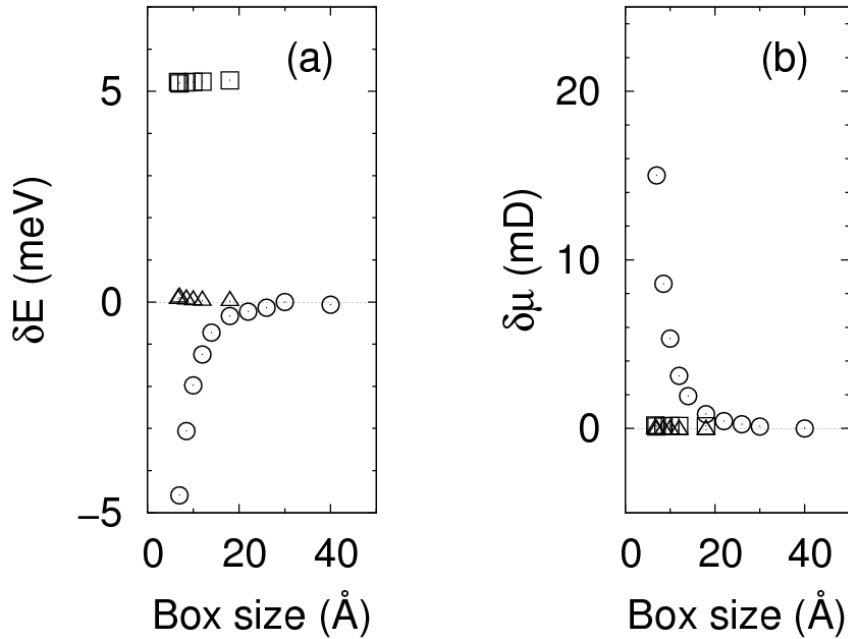


FIGURE 2.4 – Convergence des propriétés physiques d'une molécule d'eau par rapport à la taille de la boîte de simulation : (a) Énergie ; (b) Moment du dipôle Électrique. Symboles : o, simulation périodique simulation, utilisant le solveur FFT standard ; □ (resp. △) simulation non-périodique simulation, utilisant un stencil à 7 points du deuxième ordre (reps. le stencil compact à 16 points du quatrième ordre).

La Figure (2.4) présente, pour une molécule d'eau, l'influence de la taille de la boîte sur l'énergie de Kohn-Sham par la méthode *FFT* pour les schémas d'ordre 2 et d'ordre 4. L'énergie obtenue par une simulation périodique converge lorsque la taille de la boîte augmente vers l'énergie non périodique obtenue par le schéma d'ordre 4. On se rend compte que le schéma d'ordre 2 n'est pas assez précis pour obtenir la bonne valeur de l'énergie. Ces résultats sont en accord avec ceux de [6].

Les résultats présentés dans cette section sont détaillés dans le rapport [18].

2.3 Pilotage de simulations

Mettre au point des simulations complexes (multi-échelles, multi-physiques) est un processus long et délicat. Cela est dû d'une part, à la complexité des modèles utilisés et d'autre part, au fait que l'on couple des codes entre eux comme on l'a vu au chapitre précédent. Nous nous sommes rendu compte qu'il était important, dans les phases de développement et de validation sur des petits modèles, d'avoir un retour visuel immédiat des résultats intermédiaires de la simulation, sans attendre les points de sauvegarde de la simulation. Les outils existants, comme par exemple CUMULVS [46], VASE [44], soit ne proposent qu'un point d'instrumentation soit ne fonctionnent que pour des applications séquentielles. Ce n'est pas suffisant dans les simulations complexes où on visualise pas à pas, à la façon d'un débogueur, l'évolution de grandeurs physiques. C'est pourquoi je me suis intéressé très tôt, avec les Actions de Recherche Collaborative Simbio et Couplage puis l'ACI GRID EPSN, au développement d'une plateforme permettant de visualiser en direct les données de la simulation, puis d'interagir avec elles en changeant des paramètres comme le pas de temps, la fréquence sauvegarde, la température, ... Cette simulation pouvant être séquentielle, parallèle voire distribuée. Le client de visualisation est généralement séquentiel mais peut être parallèle pour traiter de gros volumes de données ou des murs d'écrans. Les principaux problèmes que l'on a abordés au travers des thèses de A. Esnard [27] et de N. Richart [66] sont la modélisation du flot d'exécution de la simulation, pour définir un système de datation pour se repérer dans la simulation ; la coordination efficace des opérations de pilotage en parallèle ou d'extractions des données ; puis le problème de la redistribution des données entre deux groupes de processeurs très différents.

Dans l'article « *A time-coherent model for the steering of parallel simulations* », [28], nous présentons notre modèle de pilotage pour des simulations numériques parallèles. Ce modèle est décomposé en deux parties ; la première concerne la représentation du flot d'exécution et la seconde la description des données de la simulation. Le flot d'exécution est capturé par un graphe de tâches hiérarchiques basé sur des tâches simples : *basic task*, *loop task*, *conditional task*. Avec ces trois types de tâches nous sommes capables de bien représenter les simulations. De plus, sur ces tâches, nous affectons un des états suivants aux données : lecture, écriture, modification, release ; cela nous permet d'extraire les données durant le traitement de la tâche par la simulation. Grâce à ce modèle, nous proposons un système de date de façon à se repérer dans les simulations parallèles de type *Single Program Multiple Data* (SPMD). En s'appuyant sur ce modèle, nous avons proposé un algorithme de coordination des requêtes qui garantit la cohérence des traitements de pilotage et de steering sans synchroniser la simulation.

Dans la thèse de A. Esnard [27], un modèle en couche a été proposé pour définir une bibliothèque de redistribution efficace. Il est basé sur une description générique des données qui permet de décrire simplement comment la donnée est distribuée sur les processeurs et comment elle est stockée en mémoire. Ensuite, grâce à cette abstraction, on peut construire la matrice de communication. La dernière couche concerne la couche de communication qui effectuera le transfert.

Coté client de pilotage, dans l'article « *A Steering Environment for Online Parallel Visualization of Legacy Parallel Simulations* », [26], nous présentons la plateforme EPSN développée au dessus du middleware CORBA ainsi qu'une amélioration du système de requêtes pour garantir la cohérence des données côté visualisation parallèle. Cette approche a été validée à partir du code d'astrophysique Gadget2, [69], avec un client parallèle utilisant un mur de 4 écrans.

Dans la publication « *Toward a computational steering environment for legacy coupled Simulations* » [65], nous avons étendu les travaux précédents aux simulations couplées de type

Multiple-SPMD comme pour le simulateur de propagation de fissures et de type client/serveur apparaissant dans le couplage QM/MM. Je décris seulement les principaux résultats et pour plus de détails, je renvoie le lecteur à la thèse financée dans le cadre d'un contrat Inria/Région, de Nicolas Richart [66]. Les deux principales contributions sont un nouveau modèle pour décrire le flot d'exécution de la simulation et un algorithme de coordination pour extraire de manière cohérente et efficace les données de la simulation. La principale difficulté avec l'approche dite « SPMD » est d'une part, d'assurer la cohérence du système de date entre les différentes simulations car nous n'avons pas la même représentation de leur flot d'exécution et d'autre part, de gérer des données distribuées entre les codes. Pour résoudre le premier problème, nous introduisons une abstraction du flot d'exécution du code couplé et définissons deux nouvelles tâches : *remote task* et *concurrent task*. La tâche *remote task* permet de modéliser l'appel d'une tâche sur un code distant (approche client/serveur). La tâche *concurrent task* précise que des tâches concurrentes s'exécutent sur des codes différents mais en même temps. Ces deux notions suffisent pour étendre notre système de date aux simulations couplées. Le modèle de donnée a été étendu pour prendre en compte les données partagées par plusieurs codes. Nous avons pu valider notre approche sur le simulateur de propagation de fissure (section 2.1).

Tous ces travaux ont été implémentés dans le logiciel EPSN². Nous avons développé des greffons QT qui pour des applications de visualisation basée sur le système QT nous permettent d'utiliser rapidement EPSN au sein de ces environnements. Nous les avons intégrés dans le logiciel Open Source de visualisation scientifique parallèle ParaView ainsi que dans des outils graphiques développés dans les ANRs MASSIM et NOSSI.

2. <http://www.labri.fr/projet/epsn>

Chapitre 3

Perspectives

Pour les prochaines années, mes perspectives de recherche s'articuleront autour de deux grandes thématiques. La première concerne le développement algorithmique autour de la méthode multipôle rapide. Le deuxième domaine concerne le développement de schémas et algorithmes pour développer des simulations complexes passant à l'échelle sur plusieurs centaines de milliers de cœurs en physique des matériaux.

3.1 Méthode multipôle rapide

Il s'agit de travailler principalement dans deux directions. La première consiste à étendre notre méthode pour d'autres domaines applicatifs comme la dynamique des dislocations, les noyaux dépendant du temps, et de l'utiliser pour faire des produits matrice-vecteur pour résoudre rapidement des systèmes linéaires issus d'équations intégrales. La deuxième direction consiste à développer une méthode multipôle efficace au dessus d'un moteur d'exécution en vue d'implantations efficaces sur un très grand nombre de cœurs.

Je participe pour les aspects multipôles au montage d'un projet de centre d'excellence européen autour du calcul à haute performance. Une partie des travaux présentés ci-dessous se feront en collaboration avec les partenaires de ce futur centre d'excellence Franco-Allemand sur le HPC.

3.1.1 Vers un parallélisme massif

Les travaux que j'ai présentés sur la parallélisation de la FMM (section 1.3.2) sur un nœud de calcul à l'aide d'un moteur d'exécution permettent d'obtenir de très bonnes efficacités tout en ayant une parallélisation d'assez haut niveau sur des architectures complexes. Pour aller vers un parallélisme massif, nous devons travailler dans plusieurs directions. Tout d'abord, la méthode que nous avons parallélisée est basée sur un octree uniforme, il est important d'étendre cette approche à un octree non uniforme [50, 51]. Les difficultés sont multiples car il y a plusieurs listes d'interaction par cellule. Ces listes font intervenir des cellules sur plusieurs niveaux de l'arbre et conduisent à des dépendances plus compliquées que dans l'approche uniforme. L'insertion des tâches que nous avons mise en place dans le cas uniforme ne marche plus. Le deuxième axe est de comprendre comment utiliser un moteur d'exécution, dans le cas distribué, pour gérer les communications entre les nœuds. Outre la gestion des communications, le degré de parallélisme sur un nœud sera moins important que dans l'approche mémoire partagée et un paramétrage adapté sera nécessaire. Pour aller vers une FMM générique, nous devons aussi définir un mécanisme basé sur un modèle de coût pour adapter automatiquement la granularité du calcul

en fonction des noyaux de calculs, comment agencer les calculs pour maximiser l'utilisation des unités de calculs (CPU, accélérateurs (CELL, GPU, MIC, ...)).

La mise en œuvre d'une méthode multipôle rapide générique passant à l'échelle sur des milliers de cœurs est l'enjeu de cette recherche.

3.1.2 Nouveaux champs d'applications

Avec des implémentations très efficaces, de nouveaux champs d'applications s'ouvrent à ces méthodes. Tout d'abord, l'introduction de nouveaux noyaux nécessite de redéfinir tous les opérateurs, la méthode Black-Box introduite par E. Darve [31], s'affranchit de cela et ne demande qu'une méthode pour calculer le noyau sur des points. Cette approche devrait nous permettre dans le cadre de l'équipe associée FastLA, de développer une méthode efficace pour les noyaux isotropes mais surtout anisotropes intervenant dans la dynamique des dislocations (ANR OP-TIDIS).

Les équations intégrales sont un autre vaste champ d'application de la FMM ; les domaines d'applications sont variés (électrostatique, acoustique, électromagnétisme, élasticité, ...). La méthode FMM intervient pour effectuer rapidement des produits matrice-vecteur au sein d'une méthode itérative. Cette approche permettrait de faire un lien entre mes travaux et les travaux autour des solveurs au sein du projet HiePACS. Une difficulté dans ces calculs provient souvent du fait que le potentiel calculé sur la surface doit être ensuite évalué sur un ensemble de points dans le domaine délimité par cette surface. L'introduction d'une approche type FMM devrait conduire à des optimisations très importantes et rendre cette approche compétitive par rapport aux approches classiques.

Un dernier axe concerne l'extension à des noyaux dépendant du temps intervenant dans les équations paraboliques ou dans les équations acoustiques. Dans ce cas, une quatrième dimension intervient dans « octree » généralisé conduisant à une algorithmique encore plus riche quant à la construction et à la mise à jour des coefficients. Ce travail se fait dans le cadre de la thèse de B. Bramas qui rentre dans le cadre d'une collaboration entre ASTRIUM-EADS et Inria avec le soutien de la région Aquitaine.

3.2 Simulation à grande échelle en physique des matériaux

Cette partie concerne à la fois le développement de nouvelles méthodes numériques et d'algorithmes parallèles pour pouvoir simuler des problèmes en physique des matériaux sur un très grand nombre de cœurs. Deux grandes applications sont abordées : la première concerne la dynamique des dislocations (CEA Saclay), la deuxième est le calcul des fréquences de vibrations de grosse molécules (IPREM¹).

3.2.1 Dynamique des dislocations

Suite à nos travaux autour de la propagation des fissures et de la FMM nous avons été approchés pour améliorer les simulations en dynamique des dislocations (échelle mésoscopique). La dynamique des dislocations (DD) est un domaine où la simulation est en plein essor et les enjeux sont très importants pour étudier le comportement mécanique des matériaux en s'appuyant sur des bases physiques. Par exemple, dans le domaine du nucléaire l'étude de la ténacité des matériaux irradiés est crucial pour pouvoir prolonger la durée de vie des centrales. Pour pouvoir utiliser les simulations comme un outil de prédiction, les codes doivent pouvoir traiter un voire

1. Institut pluridisciplinaire de recherche sur l'environnement et les matériaux - Université de Pau et des Pays de l'Adour.

deux ordres de grandeur en terme de dislocations avec une physique plus riche. Développer l’algorithmique pour atteindre ces objectifs sur les clusters de multicoeurs est l’objectif de l’ANR Optidis.

Les dislocations sont discrétisées par des éléments finis 1d et elles interagissent entre elles d’une part, par les forces à longue portée et d’autre part, aussi par les collisions entre lignes de dislocations. Ces caractéristiques font que ces simulations sont très riches en terme de schémas, d’algorithmes et d’analyse numérique.

Un pas de temps d’une simulation se décompose en trois phases de calcul : 1) le calcul des forces entre toutes les dislocations, 2) le calcul des vitesses au travers d’une loi de mobilité et 3) le traitement des collisions et du remaillage.

A ma connaissance, personne ne s’est penché sur l’analyse numérique des méthodes utilisées et de nombreux problèmes se posent. On peut citer par exemple les questions suivantes. Comment déterminer la CFL dans les schémas d’intégration en temps en fonction du problème traité ? Quels critères/estimateurs pour remailler les lignes de dislocations ? Quelle est la précision du calcul ? Comment utiliser des schémas à pas-multiple pour diminuer le coût d’évaluation des forces. Ces études sont primordiales pour développer un code à la fois performant et robuste.

Un autre problème est le calcul des forces qui peut être vu comme un problème à N-corps et conduit à l’utilisation de méthodes rapides pour les calculer dans le cas isotrope [53, 81, 78]. Dans le cas anisotrope, tout reste à faire et l’utilisation de méthode générique comme l’approche basée sur des interpolations du noyau [31] semble très prometteuse (cf section 3.1).

Pour terminer, la parallélisation de ces algorithmes sur des machines hybrides est un enjeu important et la DD présente deux particularités en terme de parallélisme conduisant à des problèmes de recherche intéressants. Concernant la parallélisation, il y a deux grandes phases de calculs qui s’opposent en terme de stratégie de parallélisation à savoir, le calcul des forces basé sur une méthode multipôle et le calcul des réactions basé sur un automate cellulaire.

La deuxième difficulté concerne le caractère très dynamique de la simulation. En effet, les dislocations naissent et meurent induisant des déséquilibres de charge pouvant être très importants. Une simulation démarrant avec quelques dislocations peut atteindre quelques millions de dislocations. Cette dynamique conduit à des stratégies différentes en terme de choix d’algorithmes, d’équilibrage, ... En effet, la simulation débute par des distributions de dislocations fortement non uniformes pour arriver soit à des distributions uniformes en fin de simulation soit à ce que les dislocations soient localisées dans des « zones de clivage ». Développer des algorithmes qui s’adaptent à ces évolutions est un des enjeux de ces simulations et va tirer les développements autour de la FMM.

3.2.2 Calcul rapide de valeurs propres

Depuis que j’ai abordé le domaine de la chimie quantique, l’accélération des méthodes pour calculer des valeurs propres d’une matrice a toujours été un sujet qui m’intéressait, car c’est la brique de base et souvent le goulot d’étranglement de beaucoup de problèmes en chimie. Suite à l’ANR NOSSI, nous avons rencontré des problèmes de calculs de valeurs et de vecteurs propres sur des matrices de très grandes tailles pouvant aller jusqu’à plusieurs millions d’inconnues. La résolution de l’équation de Schrödinger vibrationnelle conduit à un problème aux valeurs propres pour des matrices creuses de grande taille. Seul un sous ensemble des valeurs propres est intéressant pour la modélisation du spectre. En effet, seules les valeurs propres dont les vecteurs propres associés ont de grandes composantes suivant des directions prédéfinies doivent être conservées car celles-ci correspondent à des transitions vibrationnelles actives détectables expérimentalement. Cette contrainte inhabituelle n’est pas prise en compte dans les techniques standard de résolution ; la solution actuelle consiste à diagonaliser la matrice c’est à dire calculer toutes les valeurs propres, puis sélectionner a posteriori les valeurs propres pertinentes. Cette

diagonalisation constitue à l'heure actuelle un goulot d'étranglement majeur qui limite la taille des molécules (~ 20 atomes) que peuvent étudier les chimistes à l'aide de ces codes.

L'objectif est de lever un verrou en proposant une méthode en rupture; nous étudierons d'une part une nouvelle méthode numérique qui calcule uniquement les valeurs propres cibles pour la simulation et d'autre part, réaliserons une implantation de ces nouveaux algorithmes innovants sur les machines many-cœurs hétérogènes émergentes. Dans cette optique, l'idée est de généraliser la méthode du polynôme noyau (KPM) développée par Silver et al [63] pour déterminer des propriétés d'un Hamiltonien dans des systèmes de grande taille en physique des matériaux. La densité des états est approchée par un développement en polynôme de Chebyshev et les moments sont calculés par des produits matrice vecteur. En collaboration avec Y. Saad (Université de Minnesota), nous avons commencé à étendre cette méthode. Cette idée est nouvelle et semble prometteuse pour déterminer la partie du spectre pertinente. De plus, pour aborder des problèmes de grande taille cette méthode sera implantée en utilisant un paradigme de programmation par tâches mis en œuvre sur un moteur d'exécution. Ce dernier permet d'abstraire les caractéristiques et la complexité des calculateurs cibles. Une telle approche permet de développer des algorithmes qui peuvent s'adapter rapidement aux architectures des calculateurs émergents telles que les nœuds de calcul hétérogènes comportant des accélérateurs GPU et des Xeon Phi.

Une partie de ce travail est financée dans le cadre d'un projet région, dont je suis le porteur, autour des nouveaux algorithmes pour des plateforme émergentes.

Bibliographie

- [1] GSL - GNU Scientific Library. <http://www.gnu.org/software/gsl>.
- [2] Emmanuel Agullo, Bérenger Bramas, Olivier Coulaud, Eric Darve, Matthias Messner, and Toru Takahashi. Task-based FMM for Multicore Architectures. Rapport de recherche RR-8277, INRIA, Mar 2013.
- [3] G. Anciaux. *Simulation multi-échelles des solides par une approche couplée dynamique moléculaire/éléments finis. De la modélisation à la simulation haute performance*. PhD thesis, Université Bordeaux 1, 2007.
- [4] G. Anciaux, O. Coulaud, and J. Roman. High Performance Multiscale Simulation for Crack Propagation. In *Parallel Processing Workshops, 2006. ICPP 2006 Workshops. 2006 International Conference on*, pages 480–488, 2006.
- [5] Christopher R. Anderson. An implementation of the fast multipole method without multipoles. *SIAM Journal on Scientific and Statistical Computing*, 13(4) :923–947, 1992.
- [6] E Artacho, E Anglada, O Diéguez, J D Gale, A García, J Junquera, R M Martin, Pablo Ordejón, J M Pruneda, Daniel Sánchez-Portal, and José M Soler. The SIESTA method ; developments and applicability. *Journal of Physics : Condensed Matter*, 20(6) :64208, 2008.
- [7] Emilio Artacho, Daniel Sánchez-Portal, Pablo Ordejón, Alberto Garcia, and José M. Soler. Linear-scaling ab-initio calculations for large and complex systems. *physica status solidi (b)*, 215(1) :809–817, 1999.
- [8] I. Babuska and J. M. Melenk. The partition of unity method. *International Journal for Numerical Methods in Engineering*, 40(4) :727–758, 1997.
- [9] J. E. Barnes. An efficient n-body algorithm for a fine-grain parallel computer. In P. Hut and S. L. W. McMillan, editors, *The Use of Supercomputers in Stellar Dynamics*, volume 267 of *Lecture Notes in Physics*, Berlin Springer Verlag, page 175, 1986.
- [10] Josh Barnes and Piet Hut. A hierarchical $o(n \log n)$ force-calculation algorithm. *Nature*, 324 :446–449, 1986.
- [11] Christopher I Bayly, Piotr Cieplak, Wendy Cornell, and Peter A Kollman. A well-behaved electrostatic potential based method using charge restraints for deriving atomic charges : the RESP model. *The Journal of Physical Chemistry*, 97(40) :10269–10280, 1993.
- [12] Robert G. Belleman, Jeroen Bédorf, and Simon F. Portegies Zwart. High performance direct gravitational n-body simulations on graphics processing units ii : An implementation in cuda. *New Astronomy*, 13(2) :103 – 112, 2008.
- [13] J. Carrier, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm for particle simulations. *SIAM Journal on Scientific and Statistical Computing*, 9(4) :669–686, 1988.
- [14] Aparna Chandramowlishwaran, Kamesh Madduri, and Richard Vuduc. Diagnosis, tuning, and redesign for multicore performance : A case study of the fast multipole method. In *Proceedings of the 2010 ACM/IEEE conference on Supercomputing*, 2010.

- [15] H. Cheng, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm in three dimensions. *Journal of Computational Physics*, 155(2) :468 – 498, 1999.
- [16] O. Coulaud, P. Fortin, and J. Roman. Hybrid mpi-thread parallelization of the fast multipole method. In *Parallel and Distributed Computing, 2007. ISPDC '07. Sixth International Symposium on*, page 52, 2007.
- [17] O. Coulaud, P. Fortin, and J. Roman. High performance blas formulation of the multipole-to-local operator in the fast multipole method. *Journal of Computational Physics*, 227(3) :1836 – 1862, 2008.
- [18] Olivier Coulaud, Patrice Bordat, Pierre Fayon, Vincent Lebris, Isabelle Baraille, and Ross Brown. Extensions of the Siesta DFT Code for Simulation of Molecules. Rapport de recherche RR-8221, INRIA, February 2013.
- [19] Felipe A. Cruz, Matthew G. Knepley, and L. A. Barba. Petfmm—a dynamically load-balancing fast multipole library. *Int. J. Numer. Meth. Engng.*, 85(4) :403–428, 2011.
- [20] W A Curtin and Ronald E Miller. Atomistic / continuum coupling in computational materials science. *Modelling Simul. Mater. Sci. Eng.*, 11, 2003.
- [21] Eric Darve, Cris Cecka, and Toru Takahashi. The fast multipole method on clusters, multi-core processors, and graphics processing units. *Comptes Rendus Mécanique*, 339(2-3) :185–193, 2011.
- [22] J. J. Dongarra, Jeremy Du Croz, Sven Hammarling, and I. S. Duff. A set of level 3 basic linear algebra subprograms. *ACM Trans. Math. Softw.*, 16(1) :1–17, March 1990.
- [23] Jack J. Dongarra, Jeremy Du Croz, Sven Hammarling, and Richard J. Hanson. An extended set of FORTRAN Basic Linear Algebra Subprograms. *ACM Transactions on Mathematical Software*, 14(1) :1–17, March 1988.
- [24] W. Elliott and J. Board, Jr. Fast fourier transform accelerated fast multipole algorithm. *SIAM Journal on Scientific Computing*, 17(2) :398–415, 1996.
- [25] Michael A. Epton and Benjamin Dembart. Multipole translation theory for the three-dimensional laplace and helmholtz equations. *SIAM Journal on Scientific Computing*, 16(4) :865–897, 1995.
- [26] A. Esnard, N. Richart, and O. Coulaud. A steering environment for online parallel visualization of legacy parallel simulations. In *Distributed Simulation and Real-Time Applications, 2006. DS-RT'06. Tenth IEEE International Symposium on*, pages 7–14, 2006.
- [27] Aurélien Esnard. *Analyse, conception et réalisation d'un environnement pour le pilotage et la visualisation en ligne de simulations numériques parallèles*. These, Université Sciences et Technologies - Bordeaux I, December 2005. mention très honorable.
- [28] Aurélien Esnard, Michaël Dussere, and Olivier Coulaud. A time-coherent model for the steering of parallel simulations. In *in Europar 2004*, pages 90–97. Springer Verlag, 2004.
- [29] R. D. Falgout and J. E. Jones. Multigrid on massively parallel architectures. In E. Dick, K. Rienslagh, and J. Vierendeels, editors, *Multigrid Methods VI*, volume 14 of *Lecture Notes in Computational Science and Engineering*, pages 101–107, Berlin, 2000. Springer. Proc. of the Sixth European Multigrid Conference held in Gent, Belgium, September 27-30, 1999. Also available as LLNL technical report UCRL-JC-133948.
- [30] R. D. Falgout, J. E. Jones, and U. M. Yang. The design and implementation of *hypre*, a library of parallel high performance preconditioners. In A. M. Bruaset and A. Tveito, editors, *Numerical Solution of Partial Differential Equations on Parallel Computers*, pages 267–294. Springer-Verlag, 2006. Also available as LLNL technical report UCRL-JRNL-205459.

- [31] William Fong and Eric Darve. The black-box fast multipole method. *Journal of Computational Physics*, 228(23) :8712 – 8725, 2009.
- [32] T. R. Forester and W. Smith. *The dl_poly 2.0 User Manual*. Daresbury Laboratory, CCLRC, Daresbury, Warrington, U.K., 2007.
- [33] Pierre Fortin. *Algorithmique hiérarchique parallèle haute performance pour les problèmes à N-corps*. PhD thesis, Bordeaux 1 University, 2006.
- [34] Michelle Miller Francl, Christina Carey, Lisa Emily Chirlian, and David M Gange. Charges fit to electrostatic potentials. II. Can atomic charges be unambiguously fit to electrostatic potentials? *Journal of Computational Chemistry*, 17(3) :367–383, 1996.
- [35] Yuhong Fu, Kenneth J. Klimkowski, Gregory J. Rodin, Emery Berger, James C. Browne, Jürgen K. Singer, Robert A. Van De Geijn, and Kumar S. Vemaganti. A fast solution method for three-dimensional many-particle problems of linear elasticity. *International Journal for Numerical Methods in Engineering*, 42(7) :1215–1229, 1998.
- [36] L. Greengard and W.D. Gropp. A parallel version of the fast multipole method. *Computers and Mathematics with Applications*, 20(7) :63 – 71, 1990.
- [37] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2) :325 – 348, 1987.
- [38] L. Greengard and V. Rokhlin. On the efficient implementation of the fast multipole algorithm. Research Report YALEU/DCS/RR-602, YALE, 1988.
- [39] Leslie Greengard and Vladimir Rokhlin. A new version of the fast multipole method for the laplace equation in three dimensions. *Acta Numerica*, 6 :229–269, 1997.
- [40] Nail A. Gumerov and Ramani Duraiswami. Recursions for the computation of multipole translation and rotation coefficients for the 3-d helmholtz equation. *SIAM J. on Scientific Computing*, 25 :1344–1381, 2003.
- [41] Nail A. Gumerov and Ramani Duraiswami. Comparison of the efficiency of translation operators used in the fast multipole method for the 3d laplace equation. Technical Report UMIACS 4701, University of Maryland, Department of Computer Science, 2005.
- [42] Nail A. Gumerov and Ramani Duraiswami. Fast multipole methods on graphics processors. *Journal of Computational Physics*, 227(18) :8290 – 8313, 2008.
- [43] Tsuyoshi Hamada, Tetsu Narumi, Rio Yokota, Kenji Yasuoka, Keigo Nitadori, and Makoto Taiji. 42 tflops hierarchical n-body simulations on gpus with applications in both astrophysics and turbulence. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, SC '09, pages 62 :1–62 :12, New York, NY, USA, 2009. ACM.
- [44] D. Jablonowski, J. Bruner, B. Bliss, and R. Haber. VASE : The Visualization and Application Steering Environment. In *Proceedings of Supercomputing'93*, pages 560–569, 1993.
- [45] B. S. Kirk, J. W. Peterson, R. H. Stogner, and G. F. Carey. libMesh : A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations. *Engineering with Computers*, 22(3–4) :237–254, 2006. <http://dx.doi.org/10.1007/s00366-006-0049-3>.
- [46] J. A. Kohl and P. M. Papadopoulos. CUMULVS : Providing fault-tolerance, Visualization, and Steering of Parallel Applications. *International Journal of Supercomputer Applications*, 11 :224–235, 1997.
- [47] Peter Koval, Dietrich Foerster, and Olivier Coulaud. Fast construction of the Kohn-Sham response function for molecules. *Physica Status Solidi B.*, 247 :1841–1848, October 2010. Trends in Nanotechnology (TNT2009), Barcelona, SPAIN, SEP 07-11, 2009.

- [48] Peter Koval, Dietrich Foerster, and Olivier Coulaud. A parallel iterative method for computing molecular absorption spectra. *Journal of Chemical Theory and Computation*, 6(9) :2654–2668, July 2010.
- [49] Jakub Kurzak and B. Montgomery Pettitt. Massively implementation of a fast multipole method for distributed memory machines. *Journal of Parallel and Distributed Computing*, 65(7) :870 – 881, 2005.
- [50] Ilya Lashuk, Chandramowliswaran Aparna, Harper Langston, Tuan-Anh Nguyen, Rahul Sampath, Aashay Shringarpure, Richard Vuduc, Lexing Ying, Denis Zorin, and George Biros. A massively parallel adaptive fast-multipole method on heterogeneous architectures. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, SC '09*, pages 58 :1–58 :12, New York, NY, USA, 2009. ACM.
- [51] Ilya Lashuk, Aparna Chandramowliswaran, Harper Langston, Tuan-Anh Nguyen, Rahul Sampath, Aashay Shringarpure, Richard Vuduc, Lexing Ying, Denis Zorin, and George Biros. A massively adaptive fast multipole method on heterogeneous architectures. *Commun. ACM*, 55(5) :101–109, May 2012.
- [52] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh. Basic linear algebra subprograms for fortran usage. *ACM Trans. Math. Softw.*, 5(3) :308–323, September 1979.
- [53] Richard Lesar and J M Rickman. Multipole expansion of dislocation interactions : Application to discrete dislocations. *Phys. Rev. B*, 65(14) :144110, 2002.
- [54] Pangfeng Liu and Sandeep N. Bhatt. Experiences with n-body simulation. In *Proceedings of the sixth annual ACM symposium on Parallel algorithms and architectures, SPAA '94*, pages 122–131, New York, NY, USA, 1994. ACM.
- [55] Benzhuo Lu, Xiaolin Cheng, and J. Andrew McCammon. New-version-fast-multipole-method accelerated electrostatic calculations in biomolecular systems. *Journal of Computational Physics*, 226(2) :1348 – 1366, 2007.
- [56] Gang Lu and Efthimios Kaxiras. Overview of Multiscale Simulations of Materials. In Michael Rieth Schommers and Wolfram, editors, *Handbook of Theoretical and Computational Nanotechnology, Vol X*, volume X, chapter 22, pages 1–33. 2005.
- [57] M.A.L. Marques, C.A. Ullrich, F. Nogueira, A. Rubio, K. Burke, and E.K.U. Gross. *Time-Dependent Density Functional Theory*, volume 706 of *Lecture Notes in Physics*. Springer, 2006.
- [58] Richard M. Martin. *Electronic Structure : Basic Theory and Practical Methods (Vol 1)*. Cambridge University Press, April 2004.
- [59] Matthias Messner, Bérenger Bramas, Olivier Coulaud, and Eric Darve. Optimized M2L Kernels for the Chebyshev Interpolation based Fast Multipole Method, 2012, <http://arxiv.org/abs/1210.7292>.
- [60] Ronald E Miller and E B Tadmor. A unified framework and performance benchmark of fourteen multiscale atomistic/continuum coupling methods. *Modelling and Simulation in Materials Science and Engineering*, 17(5) :053001, July 2009.
- [61] K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White. Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory. *Journal on Scientific Computing*, 15(3) :713–735, 1994.
- [62] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *J Comp Phys*, 117(1-19), 1995.
- [63] A. F. Voter R. N. Silver, H. Roeder and J. D. Kress. Kernel Polynomial Approximations for Densities of States and Spectral Functions. *Journal of Computational Physics*, 124(1) :115–130, 1996.

- [64] Theodore William Rankin. *Efficient Parallel Implementations of Multipole Based N-Body Algorithms*. PhD thesis, Duke University, 1999.
- [65] N. Richart, A. Esnard, and O. Coulaud. Toward a computational steering environment for legacy coupled simulations. In *Proc. Sixth Int. Symp. Parallel and Distributed Computing ISPDC '07*, page 43, 2007.
- [66] Nicolas Richart. *Conception et mise en œuvre d'une plate-forme de pilotage de simulations numériques parallèles et distribuées*. Thèse, Université de Bordeaux 1, janvier 2010.
- [67] C Sanz-Navarro, R Grima, A García, E Bea, A Soba, J Cela, and Pablo Ordejón. An efficient implementation of a QM–MM method in SIESTA. *Theoretical Chemistry Accounts : Theory, Computation, and Modeling (Theoretica Chimica Acta)*, 128(4-6) :825–833, 2010.
- [68] J. P. Singh, C. Holt, J. L. Hennessy, and A. Gupta. A parallel adaptive fast multipole method. In *Proceedings of the 1993 ACM/IEEE conference on Supercomputing*, Supercomputing '93, pages 54–65, New York, NY, USA, 1993. ACM.
- [69] V. Springel. The cosmological simulation code gadget-2. *Monthly Notices of the Royal Astronomical Society*, 364(4) :Page 1105–1134, Dev 2005.
- [70] E B Tadmor and Ronald E Miller. The Quasicontinuum Method : Overview , applications and current. *Journal of Computer-Aided Materials Design*, 9 :203–239, 2002.
- [71] M. S. Warren and J. K. Salmon. Astrophysical n-body simulations using hierarchical tree data structures. In *Proceedings of the 1992 ACM/IEEE conference on Supercomputing*, Supercomputing '92, pages 570–576, Los Alamitos, CA, USA, 1992. IEEE Computer Society Press.
- [72] M. S. Warren and J. K. Salmon. A hashed oct-tree n-body algorithm. In *Proceedings of the 1993 ACM/IEEE conference on Supercomputing*, Supercomputing '93, pages 12–21, New York, NY, USA, 1993. ACM.
- [73] Michael S. Warren and John K. Salmon. A portable parallel particle program. *Computer Physics Communications*, 87(1–2) :266 – 290, 1995. <ce :title>Particle Simulation Methods</ce :title>.
- [74] E Weinan, Bjorn Engquist, Xiantao Li, and Weiqing Ren. The Heterogeneous Multiscale Method : A Review. *Communications in Computational Physics*, 2(3) :Communications in Computational Physics, Vol. 2, N, 2007.
- [75] Christopher A. White, Benny G. Johnson, Peter M. W. Gill, and Martin Head-Gordon. The continuous fast multipole method. *Chemical Physics Letters*, 230(1-2) :8 – 16, 1994.
- [76] C. Xiao and Y. Zhang. Design-atom approach for the quantum mechanical/molecular mechanical covalent boundary : A design-carbon atom with five valence electrons. *J. Chem. Phys.*, 127, 2007.
- [77] S.P. Xiao and T. Belytschko. A bridging domain method for coupling continua with molecular dynamics. *Computer Methods in Applied Mechanics and Engineering*, 193(17-20) :1645–1669, May 2004.
- [78] Jie Yin, D M Barnett, S P Fitzgerald, and Wei Cai. Computing dislocation stress fields in anisotropic elastic media using fast multipole expansions. *Modelling and Simulation in Materials Science and Engineering*, 20(4) :045015, June 2012.
- [79] Lexing Ying, George Biros, Denis Zorin, and Harper Langston. A new parallel kernel-independent fast multipole method. In *Proceedings of the 2003 ACM/IEEE conference on Supercomputing*, SC '03, pages 14–, Washington, DC, USA, 2003. IEEE Computer Society.

- [80] Rui Zhang, Bogdan Lev, Javier Eduardo Cuervo, Sergei Yu Noskov, and Dennis R Salahub. Chapter 10 - A Guide to QM/MM Methodology and Applications. In John R Sabin and Erkki Brändas, editors, *Combining Quantum Mechanics and Molecular Mechanics. Some Recent Progresses in QM/MM Methods*, volume 59 of *Advances in Quantum Chemistry*, pages 353–400. Academic Press, 2010.
- [81] Degang Zhao, Jingfang Huang, and Yang Xiang. A new version fast multipole method for evaluating the stress field of dislocation ensembles. *Modelling and Simulation in Materials Science and Engineering*, 18(4) :45006, 2010.

Travaux autour de la FMM

- [1] O. Coulaud, P. Fortin, and J. Roman, “High performance BLAS formulation of the multipole-to-local operator in the fast multipole method,” *Journal of Computational Physics*, vol. 227, no. 3, pp. 1836–1862, 2008. <http://www.sciencedirect.com/science/article/pii/S0021999107004354>
- [2] O. Coulaud, P. Fortin, and J. Roman, “High performance BLAS formulation of the adaptive Fast Multipole Method,” *Mathematical and Computer Modelling*, vol. 51, no. 3–4, pp. 177–188, 2010. <http://www.sciencedirect.com/science/article/pii/S0895717709002647>
- [3] O. Coulaud, P. Fortin, and J. Roman, “Hybrid MPI-Thread Parallelization of the Fast Multipole Method,” in *Parallel and Distributed Computing, 2007. ISPDC '07. Sixth International Symposium on*, 2007, p. 52. <https://hal.inria.fr/inria-00131001v2/document>
- [4] E. Agullo, B. Bramas, O. Coulaud, E. Darve, M. Messner, and T. Takahashi, “Task-Based FMM for Multicore Architectures,” *SIAM Journal on Scientific Computing*, vol. 36, no. 1, pp. C66–C93, Feb. 2014. <http://epubs.siam.org/doi/abs/10.1137/130915662>

Travaux autour des simulations complexes en physique des matériaux

- [1] G. Anciaux, O. Coulaud, and J. Roman, “High performance multiscale simulation of crack propagation,” in *Parallel Processing Workshops, 2006. ICPP 2006 Workshops. 2006 International Conference on*, 2006, p. 8 pp. –480. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1690735>
- [2] P. Koval, D. Foerster, and O. Coulaud, “Fast construction of the Kohn-Sham response function for molecules,” *Physica Status Solidi B.*, vol. 247, pp. 1841–1848, 2010. onlinelibrary.wiley.com/doi/10.1002/pssb.200983811/pdf
- [3] P. Koval, D. Foerster, and O. Coulaud, “A Parallel Iterative Method for Computing Molecular Absorption Spectra,” *Journal of Chemical Theory and Computation*, vol. 6, no. 9, pp. 2654–2668, 2010. <http://pubs.acs.org/doi/abs/10.1021/ct100280x>

Travaux autour du pilotage de simulation

- [1] O. Coulaud, M. Dussere, and A. Esnard, "Toward a distributed computational steering environment based on CORBA," in *Parallel Computing - Software Technology, Algorithms, Architectures and Applications*, vol. 13, F. J. P. G.R. Joubert W.E. Nagel and W. V Walter, Eds. North-Holland, 2004, pp. 151–158. <https://hal.inria.fr/inria-00357468/document>
- [2] A. Esnard, M. Dussere, and O. Coulaud, "A time-coherent model for the steering of parallel simulations," *Euro-Par 2004 Parallel Processing*, pp. 90–97, 2004. http://link.springer.com/chapter/10.1007/978-3-540-27866-5_12
- [3] A. Esnard, N. Richart, and O. Coulaud, "A Steering Environment for Online Parallel Visualization of Legacy Parallel Simulations," in *Distributed Simulation and Real-Time Applications*, 2006. DS-RT'06. Tenth IEEE International Symposium on, 2006, pp. 7–14. <https://www.computer.org/csdl/proceedings/ds-rt/2006/2697/00/26970007.pdf>
- [4] N. Richart, A. Esnard, and O. Coulaud, "Toward a Computational Steering Environment for Legacy Coupled Simulations," in *Proc. Sixth Int. Symp. Parallel and Distributed Computing ISPDC '07*, 2007, p. 43. <https://www.computer.org/csdl/proceedings/ispdc/2007/2936/00/29360043.pdf>