

Extraction des capacités par méthode multipolaire rapide à multi-niveaux adaptatifs

Sommaire

1	Introduction	73
2	Théorie sur la décomposition du potentiel et du champ électrique en harmoniques sphériques	74
2.1	Polynômes de Legendre	74
2.2	Harmoniques sphériques	75
2.3	Théorèmes sur la décomposition multipolaire du potentiel dû à une distribution de charges ponctuelles	76
2.4	Applications des harmoniques sphériques à l'électrostatique	85
3	Partitionnement : Construction d'un octree adaptatif	90
3.1	Principe du multi-niveaux à niveau constant	90
3.2	Partitionnement multi-niveaux et interactions lointaines	91
3.3	Principe du multi-niveaux adaptatif	92
3.4	Listes d'interaction avec le cube c : \mathcal{U}_c , \mathcal{V}_c , \mathcal{W}_c et \mathcal{X}_c	93
3.5	Adaptation de l'octree pour considérer la surface des éléments	95
4	Intégration : Construction de la "matrice" d'interaction avec la FMM 97	97
4.1	FMM et méthodes de collocation ou de Galerkin à l'ordre 0	97
4.2	Calcul des interactions proches	97
4.3	Calcul des interactions lointaines	98
4.4	Calcul du produit $[P/En].q$	98
5	Résolution : Calcul des capacités à l'aide de solveur itératif	99
5.1	Introduction du choix de solveur itératif	99

Chapitre III. Extraction des capacités par méthode multipolaire rapide à multi-niveaux adaptatifs

5.2	Solveur itératif GMRES	100
5.3	Solveur itératif GMRES(m)	105
5.4	Solveur itératif BiCGSTAB	105
5.5	Préconditionnement à gauche	108
5.6	Résolution à plusieurs seconds membres	108
6	Validations et performances	109
6.1	Intégration avec la FMM	109
6.2	Résolution avec solveur itératif	115
7	Calcul de la matrice de capacités avec la FMM du capteur à courant de Foucault	121
8	Conclusion du chapitre	122

Résumé

Dans ce troisième chapitre, on présentera une méthode numérique d'approximation d'interactions lointaines, la méthode multipolaire rapide à multi-niveaux adaptatifs (Adaptive Mutli-Level Fast Multipole Method ou AMLFMM). Elle permet d'accélérer le calcul du potentiel et du champ normal pour des interactions lointaines par le biais d'approximations en fonction de la distance entre chaque élément. Par conséquent, cela permet de gagner énormément en temps de calcul et en place mémoire tout en maîtrisant la précision des calculs. L'adaptation de la méthode originale à notre problème et l'utilisation du solveur itératif GMRES(m) permettent de traiter des grands problèmes en très peu de temps et avec très peu de place mémoire.

1 Introduction

Grâce aux travaux de Greengard et al. [47, 48], la méthode multipolaire rapide - Fast Multipole Method (FMM) - apparaît dans les années 80 pour évaluer plus rapidement le potentiel électrique créé par des nuages de charges ponctuelles. Par la suite, White et Naibors au MIT ont appliqué cette méthode au calcul des capacités parasites des micro-processeurs dans les années 90 [32, 33, 49]. Enfin, cette méthode est très largement utilisée dans la communauté des hautes fréquences par les antennistes. Effectivement, dans ces applications l'air est prédominant et rend quasiment impossible l'utilisation de méthodes comme la méthode des éléments finis à cause du maillage de l'air.

Le coût de calcul des interactions passe d'une loi en $\mathcal{O}(N^2)$ pour la méthode en interaction totale présentée précédemment à une loi en $\mathcal{O}(N \log N)$. Ce résultat est visuel comme le montre la figure III.1.

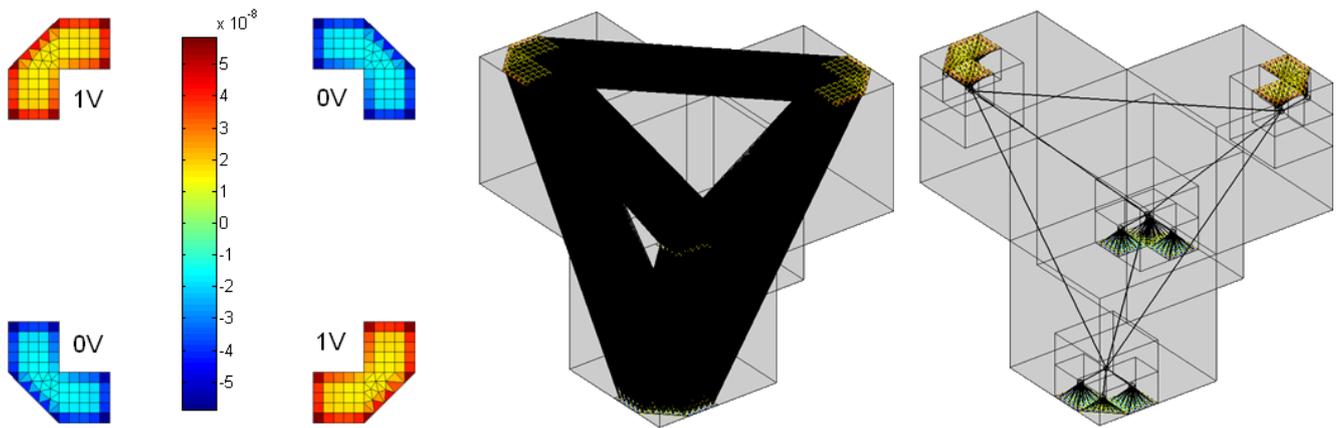


Figure III.1 – Valeurs des charges surfaciques pour 4 conducteurs surfaciques (à gauche) ; visualisation des interactions lointaines avec la méthode en interaction totale (au milieu) et de même avec la FMM (à droite)

Dans cette figure, on a considéré 4 conducteurs surfaciques, chacun est maillé en 75 éléments. Au milieu de la figure sont représentés toutes les interactions lointaines. Chaque élément interagit avec tous les autres. On rappelle que pour chaque interaction, un coefficient sera calculé et placé dans la matrice d'interaction.

A droite, sont représentés les interactions lointaines avec la FMM. On remarque que certains chemins d'interaction sont communs à plusieurs éléments. Ces chemins sont créés en fonction de l'éloignement des éléments à l'aide d'un découpage en cube de différents niveaux (appelé partitionnement). A chacun des chemins d'interaction correspond cette-fois ci un développement multipolaire tronqué à un certain degré (défini dans la section suivante) permettant d'approximer le potentiel (ou le champ). Ce choix de décomposition (ou de degré de troncature) dépend de la distance entre le point de calcul du potentiel et la source. La Fast Multipole Method permet de contrôler cette notion d'éloignement à l'aide du partitionnement.

En réalité, on ne construit pas de matrice d'interaction avec la FMM comme dans le chapitre précédent II. Un ensemble de matrices d'interaction proches et de vecteurs d'interactions lointaines est intégré et directement utilisé pour évaluer le potentiel et le champ normal. On n'obtient plus de vraie matrice, c'est pourquoi il est nécessaire d'utiliser un solveur itératif pour résoudre le problème électrostatique. Ce solveur présenté plus loin (section 5.3) cherche une solution approchée du problème en faisant des produits "[P/En].q (on rappelle que [P/En] est la matrice d'interaction de la méthode intégrale en interaction totale).

2 Théorie sur la décomposition du potentiel et du champ électrique en harmoniques sphériques

D'un point de vue plus théorique, la FMM repose sur des décompositions en harmoniques sphériques du potentiel qui sont solutions de l'équation de Laplace. Une fonction est dite harmonique si son laplacien est nul. En physique, on retrouve ces fonctions en acoustique, en géophysique, en cristallographie, en physique quantique et dans notre cas, théorie du potentiel newtonien...

Les harmoniques sphériques sont des fonctions harmoniques particulières qui s'expriment à l'aide de polynômes de Legendre.

2.1 Polynômes de Legendre

Les polynômes de Legendre sont les solutions de l'équation différentielle de Legendre suivante :

$$\frac{d}{dx} \left[(1-x^2) \frac{dy}{dx} \right] + n(n+1)y = 0 \quad (\text{III.1})$$

On peut exprimer les polynômes de Legendre comme suit :

$$L_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} [x^2 - 1]^n \quad (\text{III.2})$$

Et enfin, on définit les fonctions propres $P_n^m(x)$ à partir des polynômes de Legendre $L_n(x)$:

$$P_n^m(x) = (-1)^m [1-x^2]^{m/2} \frac{d^m L_n(x)}{dx^m} \quad (\text{III.3})$$

On verra dans la section suivante que les coefficients des harmoniques sphériques, exprimés en coordonnées sphériques, utilisent ces fonctions propres avec pour variable $\cos \theta$. Dans ce cas particulier, ces fonctions propres peuvent être également obtenues par une récurrence d'ordre 2 :

III.2 Théorie sur la décomposition du potentiel et du champ électrique en harmoniques sphériques

$$\begin{cases} P_m^m(\cos \theta) &= \frac{(2m)!}{2^m m!} \cdot (-\sin \theta)^m \\ P_{m+1}^m(\cos \theta) &= (2m+1) \cdot \cos \theta \cdot P_m^m(\cos \theta) \\ P_n^m(\cos \theta) &= \frac{1}{n-m} \left[(2n-1) \cdot \cos \theta \cdot P_{n-1}^m(\cos \theta) - (n+m-1) P_{n-2}^m(\cos \theta) \right] \end{cases} \quad (\text{III.4})$$

Le tableau III.1 présente les premières fonctions propres jusqu'au degré 4. On peut remarquer que $P_n^{-m} = (-1)^{m+1} P_n^m$.

$P_0^0 = 1$	$P_3^3 = -15 \sin^3(\theta)$	$P_4^4 = 105 \sin^4(\theta)$
$P_1^1 = \sin(\theta)$	$P_3^2 = 15 \sin^2(\theta) \cos(\theta)$	$P_4^3 = -105 \cos(\theta) \sin^3(\theta)$
$P_1^0 = \cos(\theta)$	$P_3^1 = -3/2 \sin(\theta) [5 \cos^2(\theta) - 1]$	$P_4^2 = 15/2 \sin^2(\theta) [7 \cos^2(\theta) - 1]$
$P_2^{-1} = \sin(\theta)$	$P_3^0 = 1/2 \cos(\theta) [5 \cos^2(\theta) - 3]$	$P_4^1 = -5/2 \sin(\theta) \cos(\theta) [7 \cos^2(\theta) - 3]$
$P_2^2 = 3 \sin^2(\theta)$	$P_3^{-1} = -3/2 \sin(\theta) [5 \cos^2(\theta) - 1]$	$P_4^0 = 1/8 [35 \cos^4(\theta) - 30 \cos^2(\theta) + 3]$
$P_2^1 = -3 \sin(\theta) \cos(\theta)$	$P_3^{-2} = 15 \sin^2(\theta) \cos(\theta)$	$P_4^{-1} = -5/2 \sin(\theta) \cos(\theta) [7 \cos^2(\theta) - 3]$
$P_2^0 = 1/2 [3 \cos^2(\theta) - 1]$	$P_3^{-3} = -15 \sin^3(\theta)$	$P_4^{-2} = 15/2 \sin^2(\theta) [7 \cos^2(\theta) - 1]$
$P_2^{-1} = -3 \sin(\theta) \cos(\theta)$		$P_4^{-3} = -105 \cos(\theta) \sin^3(\theta)$
$P_2^{-2} = 3 \sin^2(\theta)$		$P_4^{-4} = 105 \sin^4(\theta)$

Tableau III.1 – Premières fonctions propres jusqu'au degré 4

2.2 Harmoniques sphériques

Dans la suite de ce chapitre, on travaillera en coordonnées sphériques définies comme dans la figure III.2 :

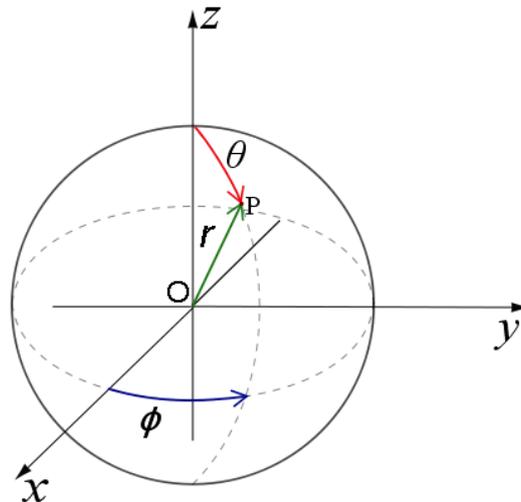


Figure III.2 – Définition des coordonnées sphériques

$$\begin{cases} x = r \sin \theta \cos \phi \\ y = r \sin \theta \sin \phi \\ z = r \cos \theta \end{cases} \quad (\text{III.5})$$

On peut écrire l'équation de Laplace dans ce nouveau système de coordonnées :

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial \Phi}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial \Phi}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 \Phi}{\partial \phi^2} = 0 \quad (\text{III.6})$$

Une solution de cette équation, obtenue par séparation des variables, est exprimée en série dont les termes sont appelés harmoniques sphériques :

$$\Phi(P) = \sum_{n=0}^{+\infty} \sum_{m=-n}^n \left(L_n^m \cdot r^n + \frac{M_n^m}{r^{n+1}} \right) \cdot Y_n^m(\theta, \phi) \quad (\text{III.7})$$

avec, les coefficients Y_n^m définis à l'aide des polynômes de Legendre de la manière suivante :

$$Y_n^m(\theta, \phi) = \sqrt{\frac{(n - |m|)!}{(n + |m|)!}} \cdot P_n^{|m|}(\cos \theta) \cdot e^{im\phi} \quad (\text{III.8})$$

Les termes $Y_n^m \cdot r^n$ et Y_n^m / r^{n+1} sont respectivement appelés harmoniques sphériques de degré n et $-(n + 1)$ et les termes L_n^m et M_n^m , moments de l'expansion multipolaire et locale. Le potentiel peut être approximé en tronquant cette somme infinie à un certain degré. Ainsi une décomposition monopolaire, bipolaire, quadrupolaire, octopolaire, etc. peut être utilisée pour approximer le potentiel.

On peut remarquer que dans le cas d'une expansion multipolaire du potentiel lointain (le point de calcul du potentiel est situé à l'extérieur d'une sphère contenant toutes les sources, ex. Fig. III.3), les coefficients L_n^m doivent être nuls afin de satisfaire la décroissance du potentiel à l'infini. Dans une expansion locale, les coefficients M_n^m doivent être nuls. Ces décompositions multipolaires ou locales sont basées sur des sphères de validité. C'est ce que nous verrons à travers les différents théorèmes suivants.

2.3 Théorèmes sur la décomposition multipolaire du potentiel dû à une distribution de charges ponctuelles

Dans cette partie, on présentera la décomposition du potentiel en un point P quelconque de l'espace, de coordonnées sphériques (r, θ, ϕ) , dû à un ensemble de N charges ponctuelles q_i de coordonnées $Q_i = (\rho_i, \alpha_i, \beta_i)$ (Fig. III.3). Le théorème suivant est issu de la théorie des harmoniques sphériques [47] :

III.2 Théorie sur la décomposition du potentiel et du champ électrique en harmoniques sphériques

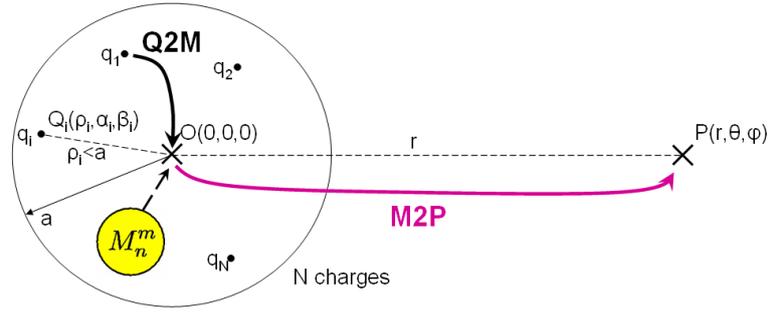


Figure III.3 – Principe de l'expansion multipolaire : chemin d'interaction $Q2M - M2P$

Théorème 2.1 (Expansion multipolaire (Q2M-M2P))

Supposons que N charges, q_1, q_2, \dots, q_N , sont situées aux points $Q_i = (\rho_i, \alpha_i, \beta_i)$ tels que $|\rho_i| < a$ (Fig. III.3). Pour tout point $P = (r, \theta, \phi)$ tel que $r > a$, le potentiel dû à ces charges est donné par :

$$\Phi(P) = \sum_{n=0}^{+\infty} \sum_{m=-n}^n M_n^m \cdot \frac{Y_n^m(\theta, \phi)}{r^{n+1}}, \quad (\text{III.9})$$

où

$$M_n^m = \sum_{i=1}^N q_i \cdot Y_n^{-m}(\alpha_i, \beta_i) \cdot \rho_i^n, \quad (\text{III.10})$$

De plus, pour tout $p \geq 1$,

$$\left| \Phi(P) - \sum_{j=0}^p \sum_{m=-j}^j M_j^m \cdot \frac{Y_j^m(\theta, \phi)}{r^{j+1}} \right| \leq \left(\frac{\sum_{i=1}^N |q_i|}{r-a} \right) \cdot \left(\frac{a}{r} \right)^{p+1}. \quad (\text{III.11})$$

Ce théorème d'expansion multipolaire permet d'approcher le potentiel réel en un point de l'espace avec une précision maîtrisée. On peut aussi remarquer que les coefficients M_n^m sont indépendants du point de calcul du potentiel et peuvent servir à nouveau pour d'autres points de l'espace respectant les hypothèses du théorème, à savoir être situé hors de la sphère qui contient les N charges.

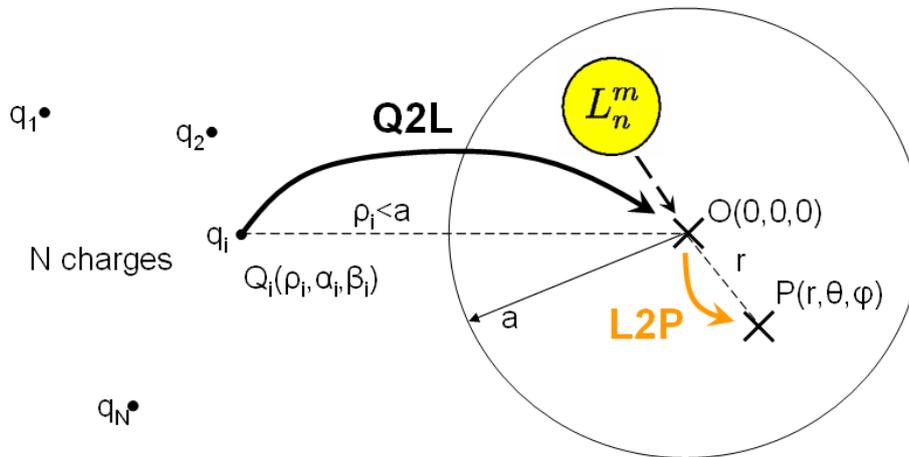


Figure III.4 – Principe de l’expansion locale : chemin d’interaction $Q2L - L2P$

Dans le cas d’une expansion locale, où les charges q_i sont à l’extérieur d’un sphère de centre O dans laquelle on calcule le potentiel (Fig. III.4), on utilise alors le coefficient L_n^m . C’est le cas particulier où M_n^m est nul dans l’équation III.7. Dans la suite de ce manuscrit, on appellera *multipole* et *local* les coefficients M_n^m et L_n^m définis comme suit :

$$\begin{cases} M_n^m = \sum_{i=0}^N q_i \cdot Y_n^{-m}(\alpha_i, \beta_i) \cdot \rho_i^n \\ L_n^m = \sum_{i=0}^N q_i \cdot Y_n^m(\alpha_i, \beta_i) / \rho_i^{n+1} \end{cases} \quad (\text{III.12})$$

L. Greengard et V. Rokhlin ont poussé plus loin cette théorie en établissant d’autres théorèmes sur les propriétés des harmoniques sphériques. Les démonstrations de ces théorèmes, détaillés dans l’article [47], ne seront pas présentés ici. Ces théorèmes fondamentaux sont la base de la FMM.

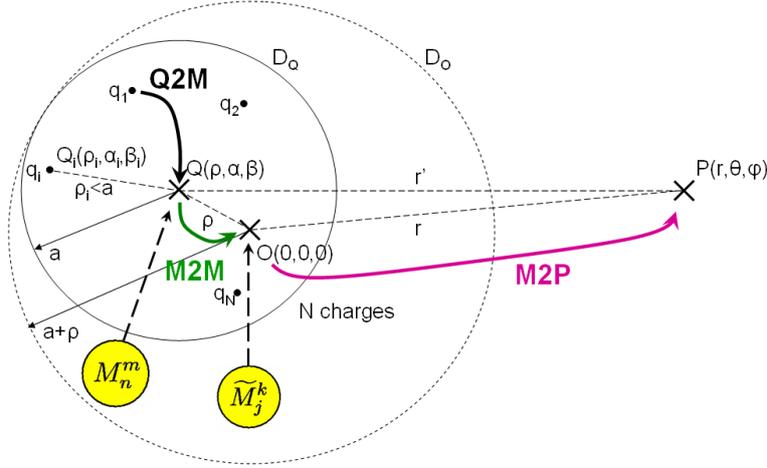


Figure III.5 – Principe de la translation d’une expansion multipolaire : chemin d’interaction $Q2M - M2M - M2P$

Théorème 2.2 (Translation d’une expansion multipolaire (M2M-M2P))

Supposons que N charges, q_1, q_2, \dots, q_N , sont situées à l’intérieur d’une sphère D_Q de rayon a de centre $Q = (\rho, \alpha, \beta)$, et que le point $P = (r, \theta, \phi)$ est situé à l’extérieur de D_Q (Fig. III.5). Le potentiel dû à ces charges est donné par l’expansion multipolaire :

$$\Phi(P) = \sum_{n=0}^{+\infty} \sum_{m=-n}^n M_n^m \cdot \frac{Y_n^m(\theta', \phi')}{r'^{n+1}}, \quad (\text{III.13})$$

où $P-Q = (r', \theta', \phi')$. Alors, pour tout $P = (r, \theta, \phi)$ à l’extérieur d’une sphère D_O de rayon $(a+\rho)$,

$$\Phi(P) = \sum_{j=0}^{+\infty} \sum_{k=-j}^j \tilde{M}_j^k \cdot \frac{Y_j^k(\theta, \phi)}{r^{j+1}}, \quad (\text{III.14})$$

où

$$\tilde{M}_j^k = \sum_{n=0}^j \sum_{\substack{m=-n, \\ |k-m| < j-n}}^n K_{j,n}^{k,m} \cdot M_{k-m}^{j-n} \cdot Y_n^{-m}(\alpha, \beta) \cdot \rho^n, \quad (\text{III.15})$$

$$K_{j,n}^{k,m} = \frac{j^{|k|-|m|-|k-m|} \cdot A_n^m \cdot A_{j-n}^{k-m}}{A_j^k}, \quad (\text{III.16})$$

$$A_n^m = \frac{(-1)^n}{\sqrt{(n-m)!(n+m)!}} \quad (\text{III.17})$$

De plus, pour tout $p \geq 1$,

$$\left| \Phi(P) - \sum_{j=0}^p \sum_{k=-j}^j \tilde{M}_j^k \cdot \frac{Y_j^k(\theta, \phi)}{r^{j+1}} \right| \leq \left(\frac{\sum_{i=1}^N |q_i|}{r - (a + \rho)} \right) \cdot \left(\frac{a + \rho}{r} \right)^{p+1}. \quad (\text{III.18})$$

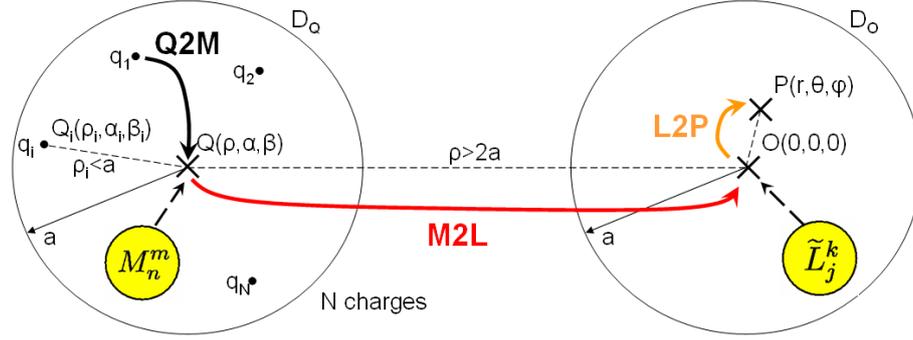


Figure III.6 – Principe de la conversion d'une expansion multipolaire en expansion locale : chemin d'interaction $Q2M - M2L - L2P$

Théorème 2.3 (Conversion d'une expansion multipolaire en expansion locale (M2L-L2P))

Supposons que N charges, q_1, q_2, \dots, q_N , sont situées à l'intérieur d'une sphère D_Q de rayon a de centre $Q = (\rho, \alpha, \beta)$, et que $\rho > a(c + 1)$ avec $c > 1$ (Fig. III.6). Alors, l'expansion multipolaire correspondante converge à l'intérieur d'une sphère D_O de rayon a de centre l'origine. A l'intérieur de D_O , le potentiel dû aux charges q_1, q_2, \dots, q_N , s'exprime grâce à l'expansion locale suivante :

$$\Phi(P) = \sum_{j=0}^{+\infty} \sum_{k=-j}^j \tilde{L}_j^k \cdot Y_j^k(\theta, \phi) \cdot r^j, \quad (\text{III.19})$$

où

$$\tilde{L}_j^k = \sum_{n=0}^{+\infty} \sum_{m=-n}^n K_{j,n}^{k,m} \cdot M_n^m \cdot \frac{Y_{j+n}^{m-k}(\alpha, \beta)}{r^{j+n+1}}, \quad (\text{III.20})$$

$$K_{j,n}^{k,m} = \frac{i^{|k-m|-|k|-|m|} \cdot A_n^m \cdot A_j^k}{(-1)^n \cdot A_{j+n}^{m-k}} \quad (\text{III.21})$$

avec A_n^m défini au 2.2. De plus, pour tout $p \geq 1$,

$$\left| \Phi(P) - \sum_{j=0}^p \sum_{k=-j}^j \tilde{L}_j^k \cdot Y_j^k(\theta, \phi) \cdot r^j \right| \leq \left(\frac{\sum_{i=1}^N |q_i|}{ca - a} \right) \cdot \left(\frac{1}{c} \right)^{p+1}. \quad (\text{III.22})$$

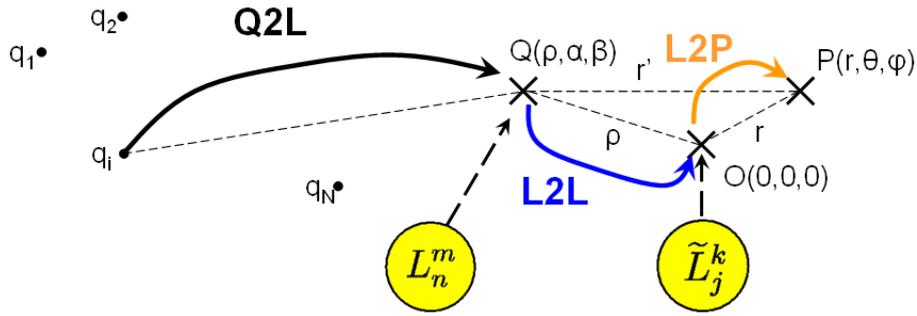


Figure III.7 – Principe de la translation d’une expansion multipolaire : chemin d’interaction $Q2L - L2L - L2P$

Théorème 2.4 (Translation d’une expansion locale (L2L-L2P))

Soit $Q = (\rho, \alpha, \beta)$, l’origine d’une expansion locale (Fig. III.7), on a :

$$\Phi(P) = \sum_{n=0}^{+\infty} \sum_{m=-n}^n L_n^m \cdot Y_n^m(\theta', \phi') \cdot r'^n, \quad (\text{III.23})$$

où $P = (r, \theta, \phi)$ et $P - Q = (r', \theta', \phi')$. Alors,

$$\Phi(P) = \sum_{j=0}^p \sum_{k=-j}^j \tilde{L}_j^k \cdot Y_j^k(\theta, \phi) r^j, \quad (\text{III.24})$$

où

$$\tilde{L}_j^k = \sum_{n=j}^p \sum_{\substack{m=-n, \\ |l-k| < n-j}}^n K_{j,n}^{k,m} \cdot L_n^m \cdot Y_{n-j}^{m-k}(\alpha, \beta) \cdot \rho^{n-j}, \quad (\text{III.25})$$

$$K_{j,n}^{k,m} = \frac{i^{|m|-|m-k|-|k|} \cdot A_{n-j}^{m-k} \cdot A_j^k}{(-1)^{n+j} \cdot A_n^m} \quad (\text{III.26})$$

avec A_n^m défini au 2.2.

2.3.1 Application des théorèmes pour un problème à 1 000 charges ponctuelles

Les différents théorèmes sont appliqués à la configuration de la figure III.8.

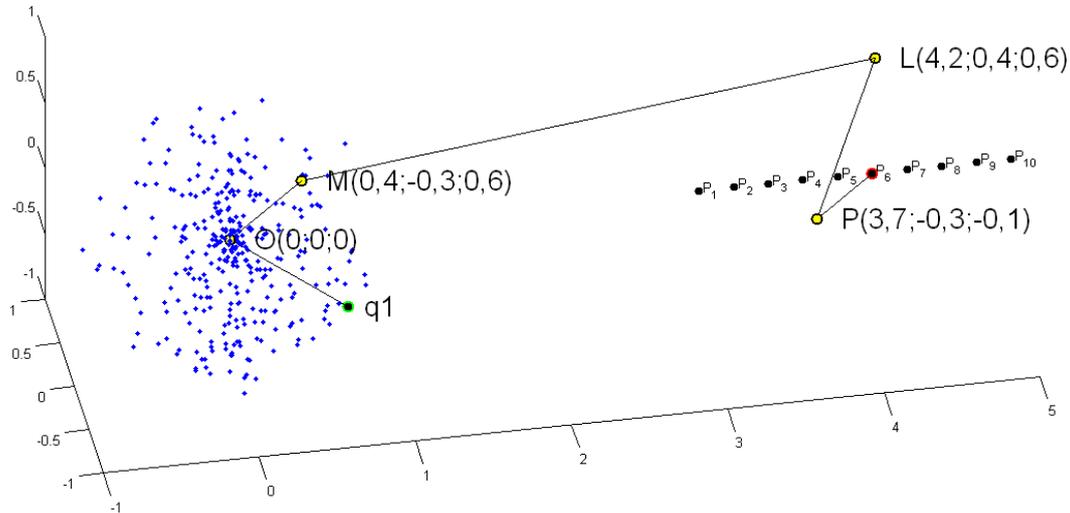


Figure III.8 – Calculs de 10 potentiels P_i créés par une répartition de 1 000 charges ponctuelles - exemple de chemin d’interaction entre la charge q_1 et le point de calcul P_6 : $Q2M - M2M - M2L - L2L - L2P$

Les 1 000 charges sont contenues dans une sphere de centre O et de rayon 1. Les chemins d’interaction testés sont les suivants : $Q2M - M2P$, $Q2M - M2M - M2P$, $Q2M - M2L - L2P$, $Q2M - M2L - L2L - L2P$ et $Q2M - M2M - M2L - L2L - L2P$. On a calculé les erreurs relatives en valeur absolue sur 10 calculs de potentiel entre la méthode en interaction directe (en ponctuel) et la FMM pour différents chemins d’interaction et différents ordre d’intégration. Les points utilisés pour les *multipoles* (M) et les *locals* (L) sont les suivants (dans l’ordre de la figure III.8) : le premier *multipole* (du vecteur $Q2M$) est le point O , le deuxième le point M , le premier *local* L et le deuxième P . Les erreurs pour ces différents chemins sont présentées dans les figures III.9 et III.10.

On peut observer une décroissance rapide des erreurs relatives en fonction du degré pour les différents chemins. Dès le degré 2, les erreurs relatives sont inférieures à 3,5% et inférieures à 0,005% pour le degré 5. On remarque que les erreurs relatives maximales concernent pour les points P_i les plus éloignés du dernier centre de décomposition : un *local* pour le $L2P$ ou un *multipole* pour le $M2P$. Entre les points P_3 et P_7 , les erreurs relatives sont minimales. Les majorants (équation III.11) de l’erreur relative du théorème d’expansion multipolaire ($Q2M - M2P$) sont très supérieurs quelque soit le degré de décomposition. Les majorants (équation III.18) du théorème de translation multipolaire ($M2M - M2P$) n’ont pas été tracé sur la figure III.9 car leur valeur minimale au point P_{10} est à chaque fois supérieure à l’erreur maximale des échelles en ordonnée des différents graphiques. Pour ces derniers majorants, on obtient au point P_{10} dans l’ordre des degrés : 55%, 20%, 7%, 2,5%, 0,9% et 0,32%. Pour les degrés supérieurs à 5 (Fig. III.10), la décroissance des erreurs relatives est plus lente pour les points les plus éloignés des centres L et P (Fig. III.8).

III.2 Théorie sur la décomposition du potentiel et du champ électrique en harmoniques sphériques

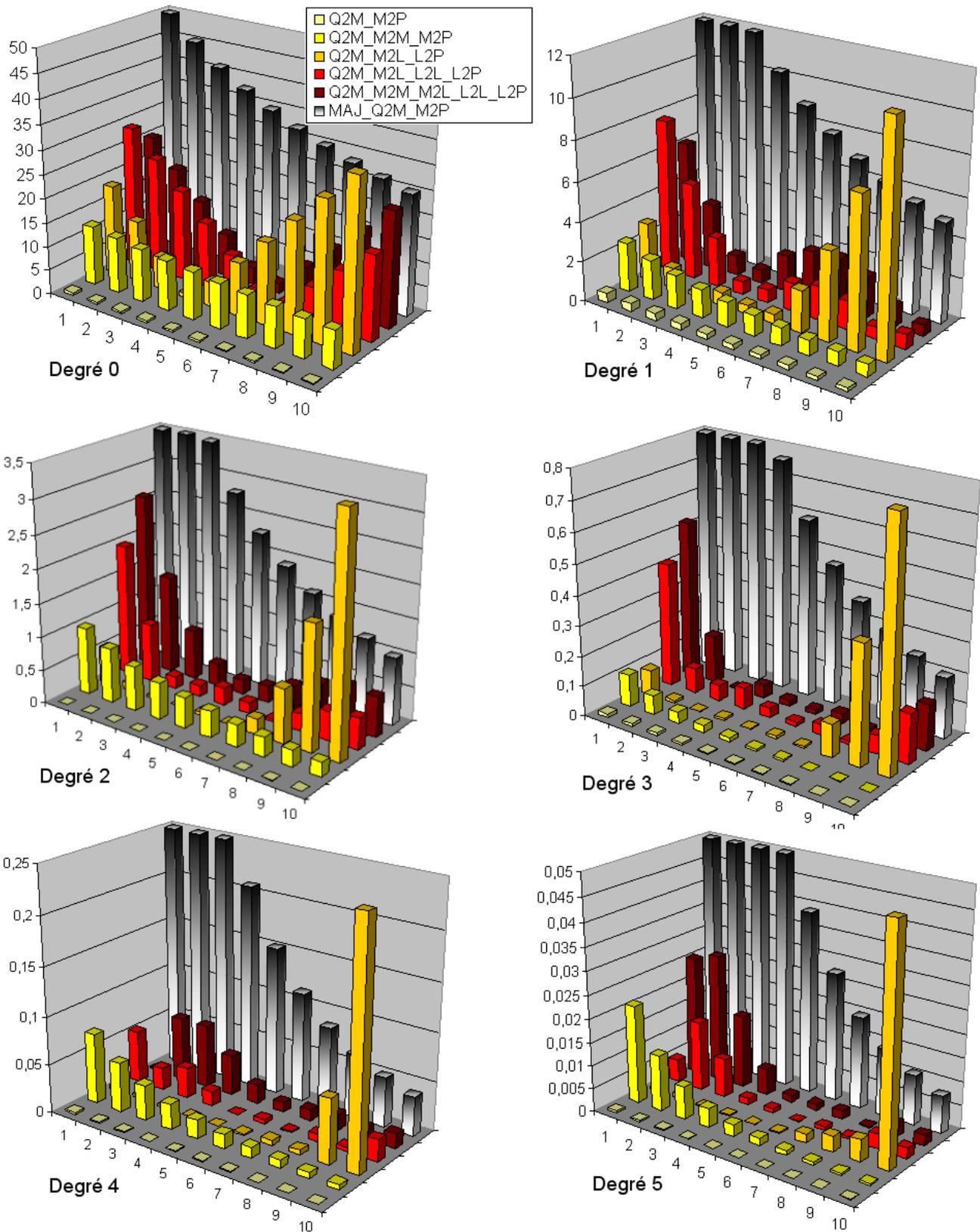


Figure III.9 – Comparaison des erreurs relatives sur 10 calculs de potentiel de la figure III.8 entre des calculs directs en ponctuel et la FMM pour différents chemins d’interaction et différents degrés (de 0 à 5) - en dégradé, les majorants des erreurs du théorème d’expansion multipolaire III.3 (équation III.11)

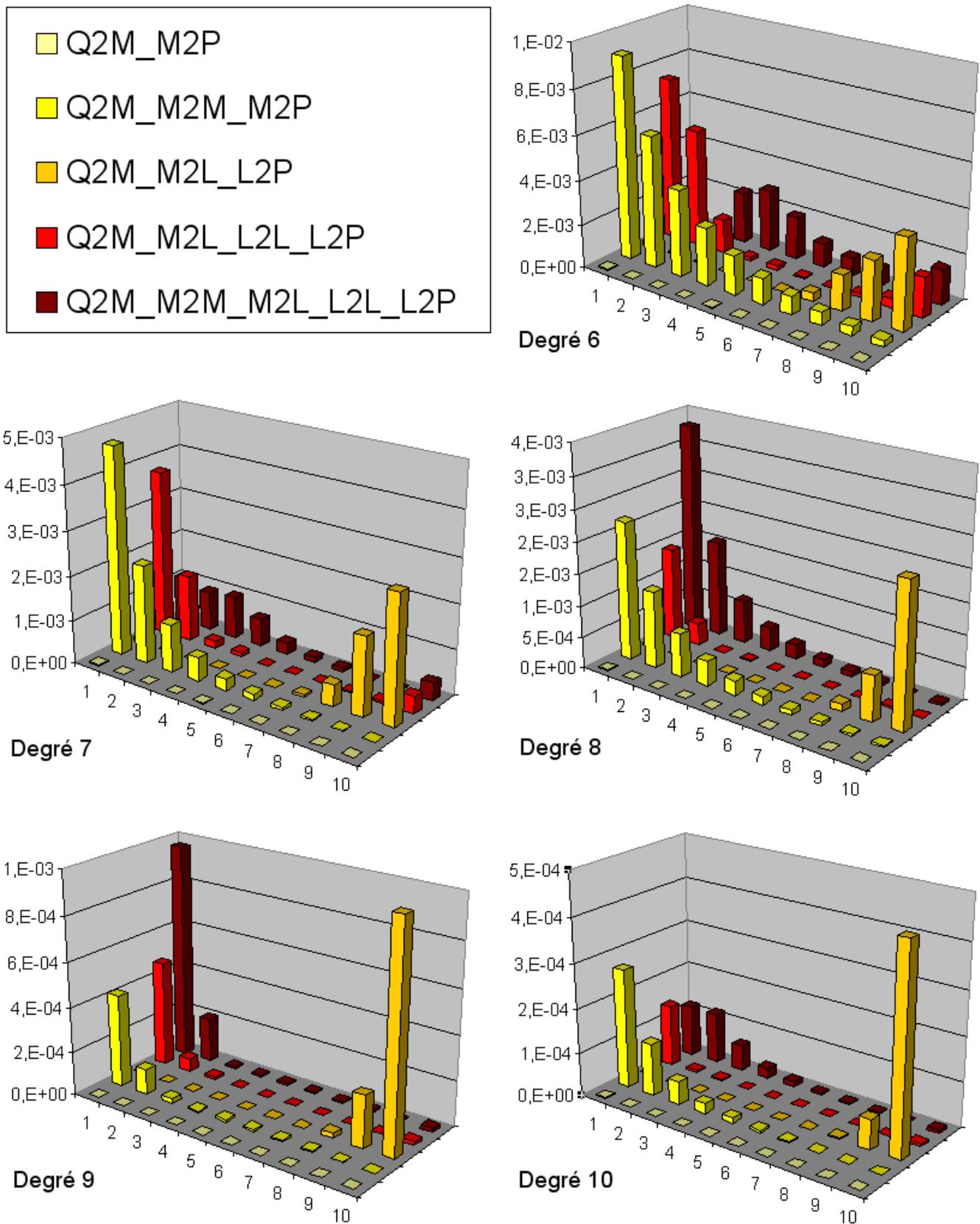


Figure III.10 – Mêmes calculs que dans la figure III.9 pour les degrés 6 à 10

2.4 Applications des harmoniques sphériques à l'électrostatique

Revenons à notre problème d'électrostatique. Pour résumer, la fonction de Green (en régime quasi-statique) peut être décomposée à l'aide de polynômes harmoniques N_n^m et F_n^m en coordonnées sphériques de la façon suivante :

$$\frac{1}{\|\mathbf{X} - \mathbf{X}'\|} = \begin{cases} F_n^m(r, \theta, \phi) \cdot N_n^m(r', \theta', \phi') & \text{si } r < \rho' \\ N_n^m(r', \theta', \phi') \cdot F_n^m(r, \theta, \phi) & \text{si } r' < \rho \end{cases} \quad (\text{III.27})$$

avec,

$$N_n^m(r, \theta, \phi) = Y_n^{-m}(\theta, \phi) \cdot r^n \quad (\text{III.28})$$

$$F_n^m(r, \theta, \phi) = \frac{Y_n^m(\theta, \phi)}{r^{n+1}} \quad (\text{III.29})$$

2.4.1 Approximation du potentiel

On rappelle qu'on considère une distribution surfacique de charges. Ainsi, on ne peut plus maîtriser la deuxième partie du théorème concernant l'erreur de la décomposition du potentiel à un ordre p donné. En reprenant les hypothèses du chapitre précédent II, à l'ordre 0 (surfaces uniformément chargées), on peut réécrire l'équation en potentiel II.1 à l'aide du théorème d'expansion multipolaire 2.1 :

$$V(r, \theta, \phi) \approx \frac{1}{4\pi\epsilon_0} \sum_{n=0}^p \sum_{m=-n}^n \langle N_n^m, Q \rangle \cdot F_n^m(r, \theta, \phi) \quad (\text{III.30})$$

avec,

$$\langle N_n^m, Q \rangle = \sum_{i=0}^N q_i \cdot \rho_i^n \cdot Y_n^{-m}(\alpha_i, \beta_i) \quad (\text{III.31})$$

On remarque qu'avec cette écriture le coefficient M_n^m du théorème 2.1 est équivalent à $\langle N_n^m, Q \rangle$ et les coordonnées sphériques $(\rho_i, \alpha_i, \beta_i)$ sont les coordonnées des barycentres des N éléments surfaciques.

On peut réécrire le potentiel donné par les trois autres théorèmes en remplaçant le *multipole* M_n^m par les termes $\langle N_n^m, Q \rangle$ dans les théorèmes 2.2 et 2.3, ainsi que le *local* L_n^n par $\langle F_n^m, Q \rangle$ dans le théorème 2.4.

2.4.2 Approximation du champ normal

De la même façon que dans le chapitre II.2, on va dériver formellement le potentiel pour calculer le champ. En reprenant l'équation précédente III.30, on peut obtenir le champ en coordonnées

Chapitre III. Extraction des capacités par méthode multipolaire rapide à multi-niveaux adaptatifs

sphériques suivant :

$$\begin{aligned} \mathbf{E}(r, \theta, \phi) &\approx -\nabla \left(\frac{1}{4\pi\epsilon_0} \sum_{n=0}^p \sum_{m=-n}^n \langle N_n^m, Q \rangle F_n^m(r, \theta, \phi) \right) \\ &\approx -\frac{1}{4\pi\epsilon_0} \sum_{n=0}^p \sum_{m=-n}^n \langle N_n^m, Q \rangle \nabla F_n^m(r, \theta, \phi) \end{aligned} \quad (\text{III.32})$$

Or on souhaite obtenir le champ en coordonnées cartésiennes et calculer directement les trois composantes E_x , E_y et E_z . Par exemple, pour le terme $\frac{dF_n^m}{dx}(r, \theta, \phi)$, on a :

$$\frac{dF_n^m}{dx}(r, \theta, \phi) = \left. \frac{\partial F_n^m}{\partial r} \right|_{\theta, \phi} \cdot \frac{dr}{dx} + \left. \frac{\partial F_n^m}{\partial \theta} \right|_{r, \phi} \cdot \frac{d\theta}{dx} + \left. \frac{\partial F_n^m}{\partial \phi} \right|_{r, \theta} \cdot \frac{d\phi}{dx} \quad (\text{III.33})$$

En procédant de même pour les autres variables, on peut écrire :

$$\begin{bmatrix} \frac{dF_n^m}{dx} \\ \frac{dF_n^m}{dy} \\ \frac{dF_n^m}{dz} \end{bmatrix} = \begin{bmatrix} \frac{dF_n^m}{dr} & \frac{dF_n^m}{d\theta} & \frac{dF_n^m}{d\phi} \end{bmatrix} \cdot \begin{bmatrix} \frac{dr}{dx} & \frac{d\theta}{dx} & \frac{d\phi}{dx} \\ \frac{dr}{dy} & \frac{d\theta}{dy} & \frac{d\phi}{dy} \\ \frac{dr}{dz} & \frac{d\theta}{dz} & \frac{d\phi}{dz} \end{bmatrix} \quad (\text{III.34})$$

avec,

$$\begin{bmatrix} \frac{dr}{dx} & \frac{d\theta}{dx} & \frac{d\phi}{dx} \\ \frac{dr}{dy} & \frac{d\theta}{dy} & \frac{d\phi}{dy} \\ \frac{dr}{dz} & \frac{d\theta}{dz} & \frac{d\phi}{dz} \end{bmatrix} = \begin{bmatrix} \sin \theta \cos \phi & \cos \theta \cos \phi & -\frac{\sin \phi}{r \sin \theta} \\ \sin \theta \sin \phi & \frac{\cos \theta \sin \phi}{r} & \frac{\cos \phi}{r \sin \theta} \\ \cos \theta & -\frac{\sin \theta}{r} & 0 \end{bmatrix} \quad (\text{III.35})$$

Si l'on se place désormais dans les hypothèses de l'un des quatre théorèmes vu précédemment, pour obtenir l'expression du champ, on doit dériver un seul coefficient par rapport aux coordonnées sphériques : soit le terme F_n^m (théorèmes 2.1 et 2.2) soit le terme N_n^m (théorèmes 2.3 et 2.4). Il s'agit des derniers vecteurs dans les figures des théorèmes (ceux qui arrivent au niveau des points d'évaluation). Cela est bien pratique car tout *multipole* peut servir au calcul du champ ou du potentiel. La dérivation de ces termes selon r donne :

$$\begin{cases} \frac{dN_n^m}{dr}(r, \theta, \phi) = n.Y_n^{-m}(\theta, \phi).r^{n-1} \\ \frac{dF_n^m}{dr}(r, \theta, \phi) = -(n+1)\frac{Y_n^m(\theta, \phi)}{r^{n+2}} \end{cases} \quad (\text{III.36})$$

On observe que pour l'ordre 0 ($n = 0$), la dérivée du terme N_n^m par rapport à r est nulle. Pour les harmoniques sphériques $Y_n^m(\theta, \phi)$, on dérive formellement les polynômes de Legendre en θ , $P_n^{|m|}(\cos \theta)$, et les exponentielles complexes en ϕ , $e^{im\phi}$:

$$\begin{cases} \frac{dY_n^m}{d\theta}(\theta, \phi) = \frac{P_n^{|m|}(\cos \theta)}{d\theta} \cdot e^{im\phi} \\ \frac{dY_n^m}{d\phi}(\theta, \phi) = im.Y_n^m(\theta, \phi) \cdot e^{im\phi} \end{cases} \quad (\text{III.37})$$

Une fois l'expression du champ électrique obtenue au barycentre d'un élément surfacique en

III.2 Théorie sur la décomposition du potentiel et du champ électrique en harmoniques sphériques

coordonnées cartésiennes, il suffit de la projeter sur la normale sortante de cet élément pour obtenir celle du champ normal. Ainsi, on dispose des formulations en interactions lointaines pour les calculs du potentiel et du champ normal.

2.4.3 Application des théorèmes pour le champ en ponctuel

Le même problème des 1 000 charges de la figure III.8 précédente a été traité en champ électrique. Les figures III.11 et III.12 montre les erreurs relatives en valeur absolue sur les 10 modules du champ.

Les conclusions que l'on peut tirer de ces comparaisons sont les mêmes que pour le potentiel. On remarque une bonne décroissance de l'erreur en fonction du degré pour les différents chemins. Cependant cette décroissance est moins rapide que pour le potentiel. On peut remarquer aussi que l'erreur est plus sensible à la distance entre le point de calcul du champ et le dernier centre de décomposition. Effectivement le champ s'exprime en $1/r^2$ et le potentiel en $1/r$. On remarque que pour le degré 0, l'introduction d'un *local* donne une valeur de champ nulle car dans l'équation III.36, le terme $\frac{dN_n^m}{dr}(r, \theta, \phi)$ est nul car $n = 0$ (d'où l'erreur de 100% dans la figure III.11).

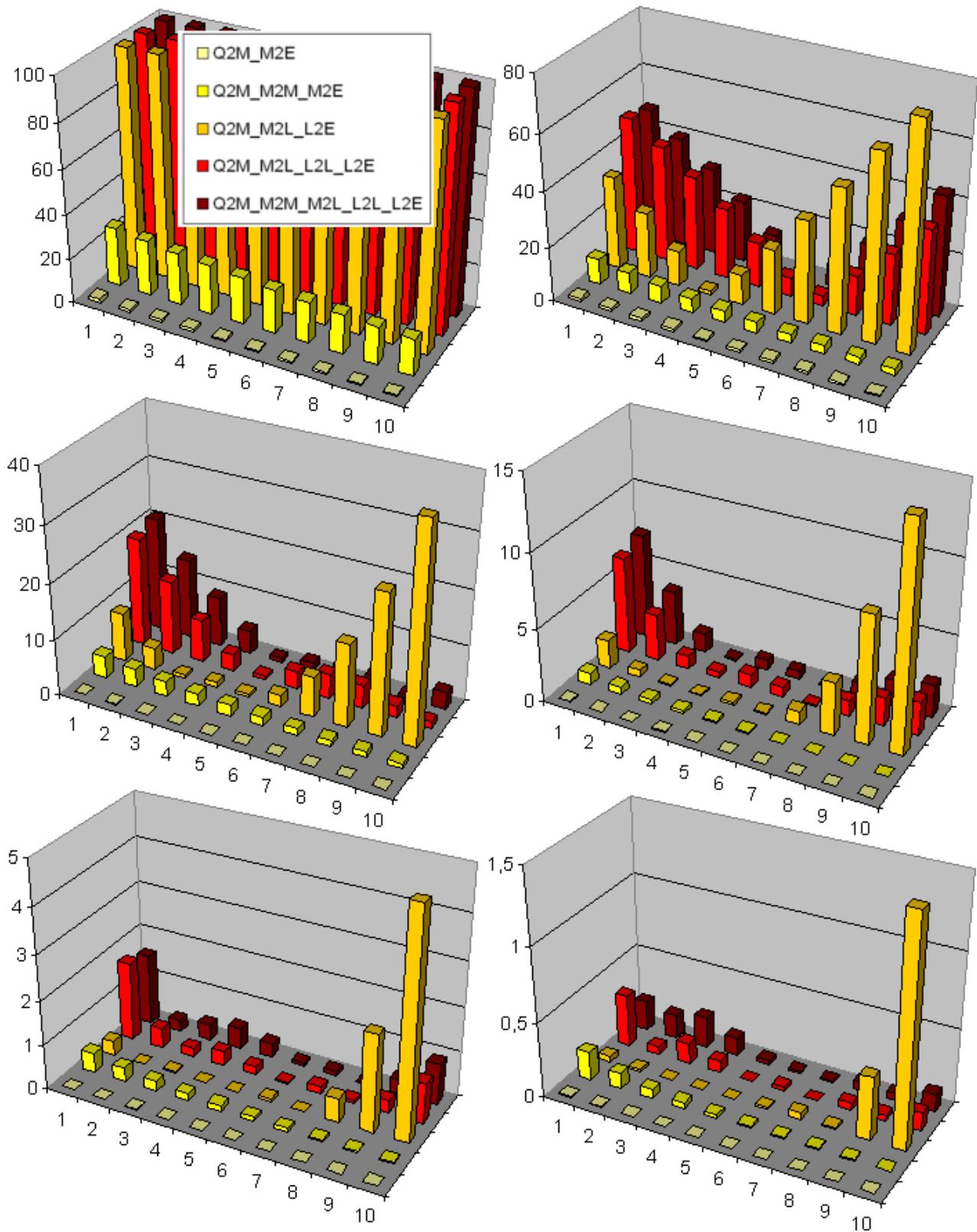


Figure III.11 – Comparaison des erreurs relatives sur 10 calculs de module du champ de la figure III.8 entre des calculs directs en ponctuel et la FMM pour différents chemins d'interaction et différents degrés (de 0 à 5)

III.2 Théorie sur la décomposition du potentiel et du champ électrique en harmoniques sphériques

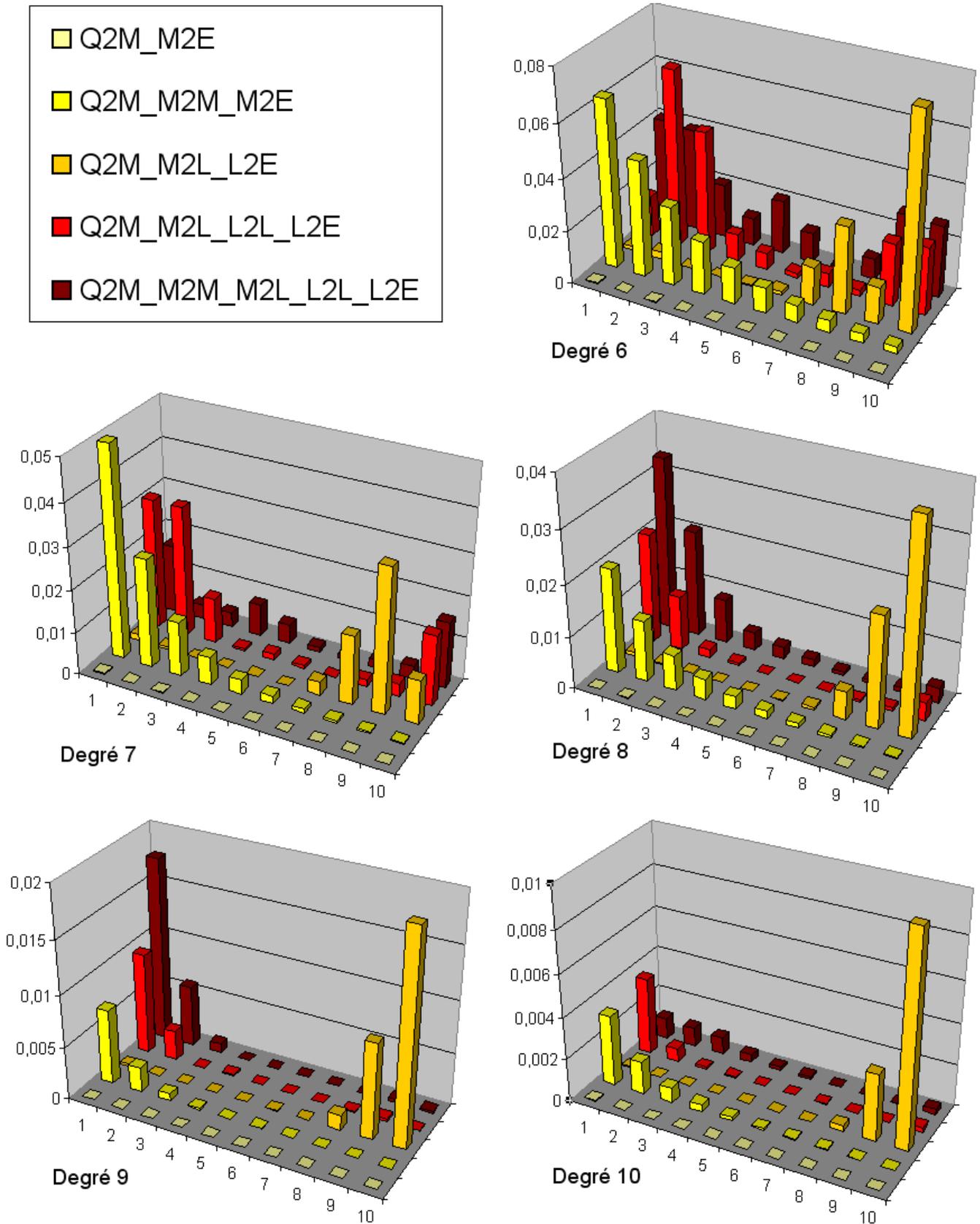


Figure III.12 – Mêmes calculs que dans la figure III.11 pour les degrés 6 à 10

3 Partitionnement : Construction d'un octree adaptatif

Dans cette partie, on présente le partitionnement de la géométrie servant à définir de façon automatique les différents types d'interactions utilisés. L'éloignement entre les différentes zones déterminera les « chemins » d'interactions. En résumé, on utilisera la méthode en interaction totale présentée dans le chapitre II pour calculer les interactions proches et les décompositions multipolaires pour les interactions lointaines. On utilisera pour ce découpage, un algorithme de type octree (structure en arbre 3D). L'objectif étant de répartir les barycentres des éléments surfaciques de façon adaptative dans des cubes de différents niveaux.

3.1 Principe du multi-niveaux à niveau constant

Tout d'abord, on s'intéresse à un découpage en multi-niveaux à niveau constant. Le découpage en octree se fait de la façon suivante : on part d'un cube englobant toutes les charges - cube de niveau 0 appelé « racine » -, on le découpe en 8 cubes égaux - cubes de niveau 1 -, ensuite chacun de ces 8 cubes est découpé en 8 cubes égaux - cubes de niveau 2 -, et ainsi de suite... On forme alors 8^L cubes si L est le nombre de niveaux. Le principe du niveau constant est que l'on définit au départ, L , le nombre de niveaux souhaité (Fig. III.13). On ne découpe que les cubes contenant au moins une charge. Ceux qui n'en contiennent pas seront ignorés dans l'algorithme. Au niveau de la hiérarchie, on utilise un vocabulaire spécifique : tout cube non vide a au plus 8 cubes « fils », tout cube (de niveau supérieur à 0) a obligatoirement un cube « père », enfin, tous les cubes du niveau le plus bas sont appelés cubes « feuilles ».

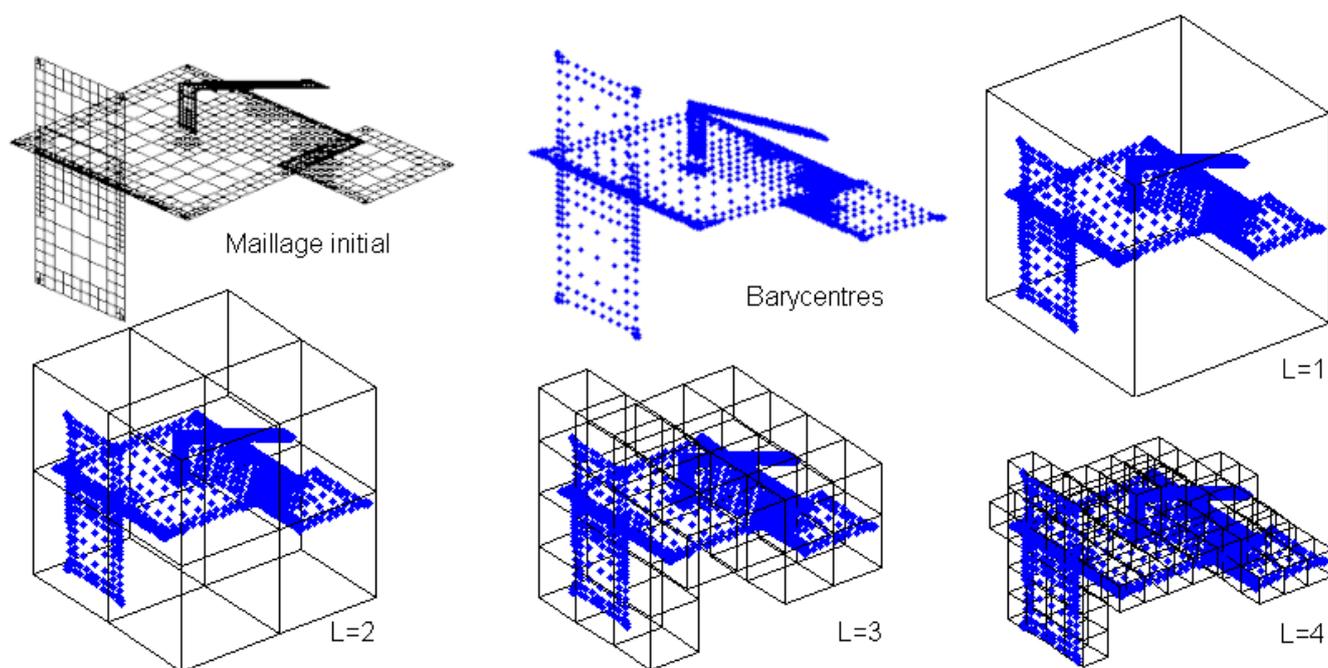


Figure III.13 – Principe du partitionnement sur les barycentres d'une géométrie à un niveau L choisi

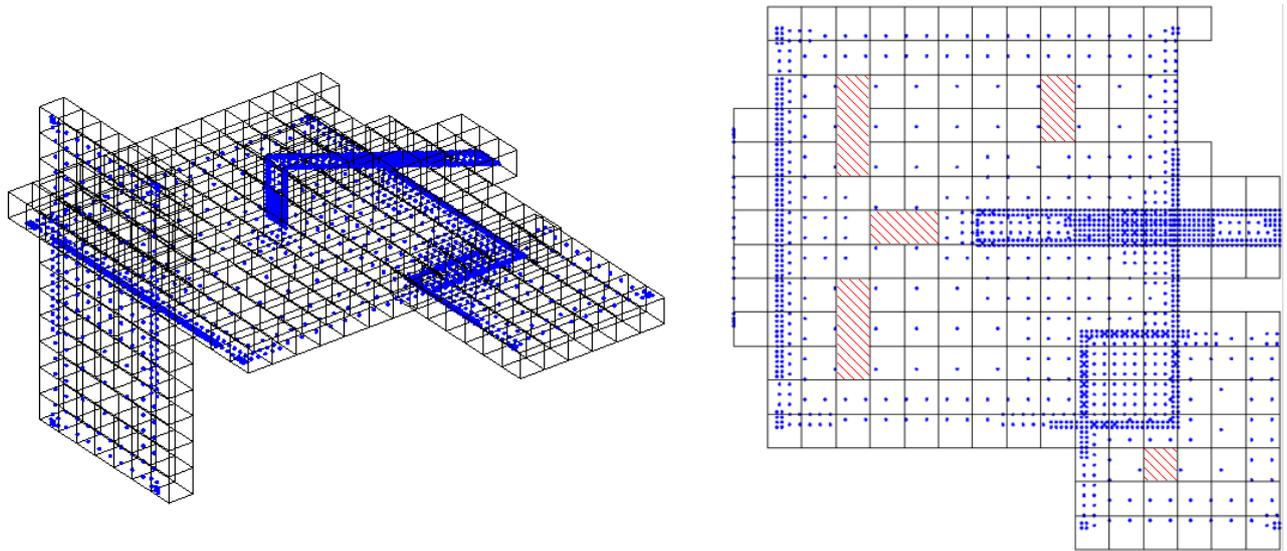


Figure III.14 – Partitionnement au niveau 4 faisant apparaître des trous (zones hachurées) et une répartition non homogène du nombre de charges dans les cubes

Le principal inconvénient de ce partitionnement est qu'on ne maîtrise pas le nombre de charges présents dans les cubes. Pour obtenir une répartition homogène des barycentres, il faut nécessairement avoir une géométrie maillée de façon homogène. Dans le cas de la figure III.13, le maillage est non uniforme (maillage issu du mailleur adaptatif de la section 4.4.3). On peut avoir pour un partitionnement au niveau 6 une à plusieurs charges par cube et des trous peuvent être formés comme le montre la figure III.14.

3.2 Partitionnement multi-niveaux et interactions lointaines

Une fois le partitionnement obtenu, les chemins d'interactions entre les éléments situés dans les cubes de différents niveaux sont directement liés à ce partitionnement. Les chemins des interactions lointaines ne sont pas tous identiques. En fonction du partitionnement, on peut obtenir toutes les combinaisons possibles entre les charges appelées Q , les *multipoles* M , les *locals* L , et les points d'évaluations P ou E qui sont en fait les barycentres des éléments. Ces chemins d'interactions lointaines sont construits comme un jeu de dominos avec pour pions, les vecteurs d'interaction $Q2M$, $M2E$, $M2M$, $M2L$, $L2L$, $L2P$ et $L2E$. Comme on l'a vu précédemment, les interactions lointaines peuvent utiliser des *multipoles* (M) ou des *locals* (L) qui peuvent avoir plusieurs origines ($Q2M$, ou $M2M$ pour les *multipoles* ou $M2L$ ou $L2L$ pour les *locals*). De même pour les points d'évaluation en potentiel et en champ normal.

Pour calculer le potentiel ou le champ normal dans un cube feuille, on calcule les vecteurs $L2P$ (avec L provenant de $L2L$ et/ou de $M2L$). Or pour calculer ces vecteurs on doit calculer le *multipole* et/ou le *local* qui peuvent être la somme de plusieurs *multipoles* ou *locals*. On remonte ainsi tous les chemins d'interaction qui contribue au calcul du potentiel dans chaque cube feuille.

Chapitre III. Extraction des capacités par méthode multipolaire rapide à multi-niveaux adaptatifs

Pour plus de compréhension, ce processus est illustré avec un exemple dans la figure III.15.

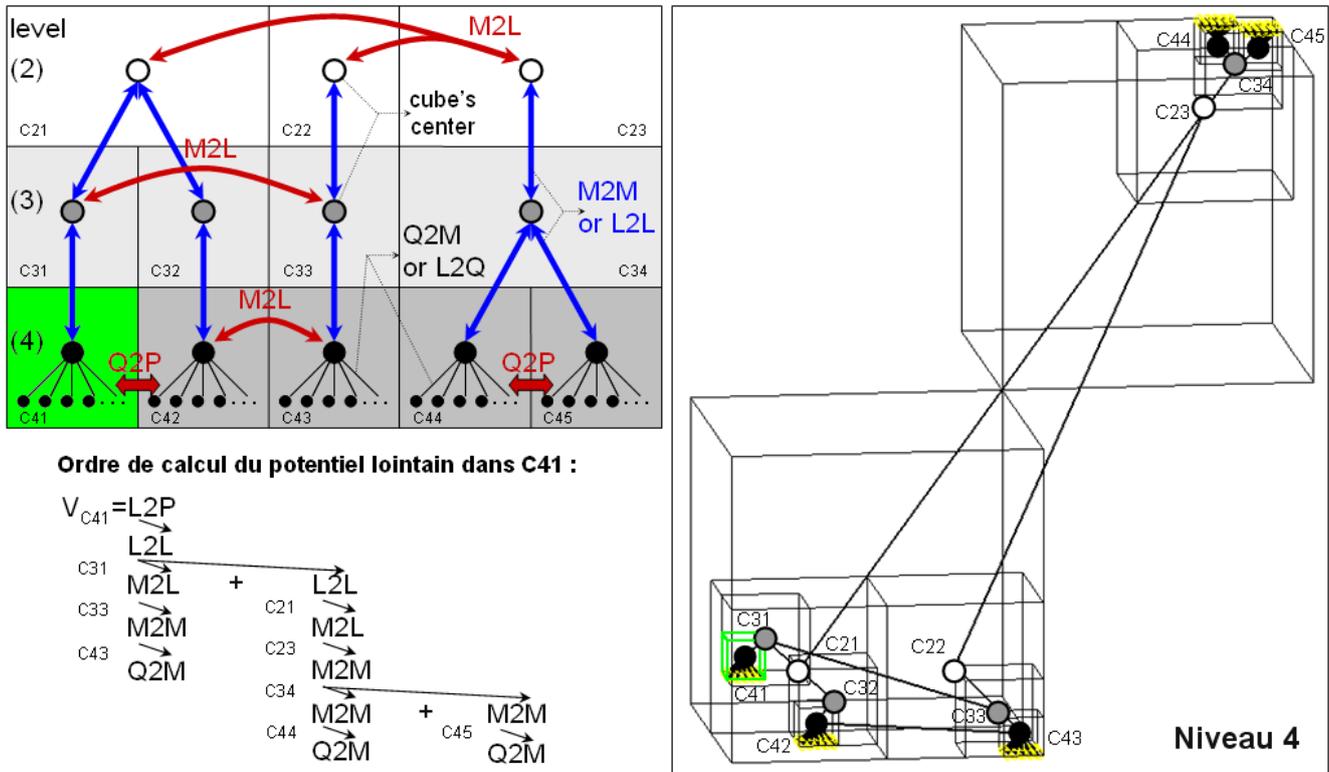


Figure III.15 – Processus de calcul du potentiel dans le cube feuille 41 de centre C41.

L’algorithme FMM qui utilise ce découpage en multi-niveau à niveau constant pour calculer les interactions entre charges ponctuelles d’un nuage de charges donné est décrit par L. Greengard et V. Rokhlin en 1987 [47]. Cependant on ne présentera pas ici en détail cet algorithme initial car ils ont proposé l’année suivante, un partitionnement adaptatif permettant une optimisation du nombre de cubes utiles ainsi qu’une meilleure prise en compte d’une répartition non uniforme de charges ponctuelles donnée [48]. Et pour nos problèmes, les maillages sont adaptés à la répartition des charges et sont donc raffinés sur les bords et plus lâche au centre des conducteurs, par conséquent ces maillages ne sont pas uniformes.

3.3 Principe du multi-niveaux adaptatif

Dans cette nouvelle version de partitionnement, le processus est le même que pour le niveau constant sauf qu’on ne va subdiviser que les cubes contenant un nombre de charges supérieur à un nombre $N_Q > 0$ fixé. Cela permet d’obtenir un nombre de charges par cubes relativement homogène et évite de nombreux trous sur un même conducteur (Fig. III.16). Avec ce nouvel octree, on a besoin d’utiliser d’autres décompositions du potentiel (ou du champ normal). On peut calculer le potentiel sans passer nécessairement par une conversion $M2L$ en faisant $Q2L - L2P$ (de même pour le champ, $Q2L - L2E$). On peut réécrire les équations du potentiel II.1 (ou du champ

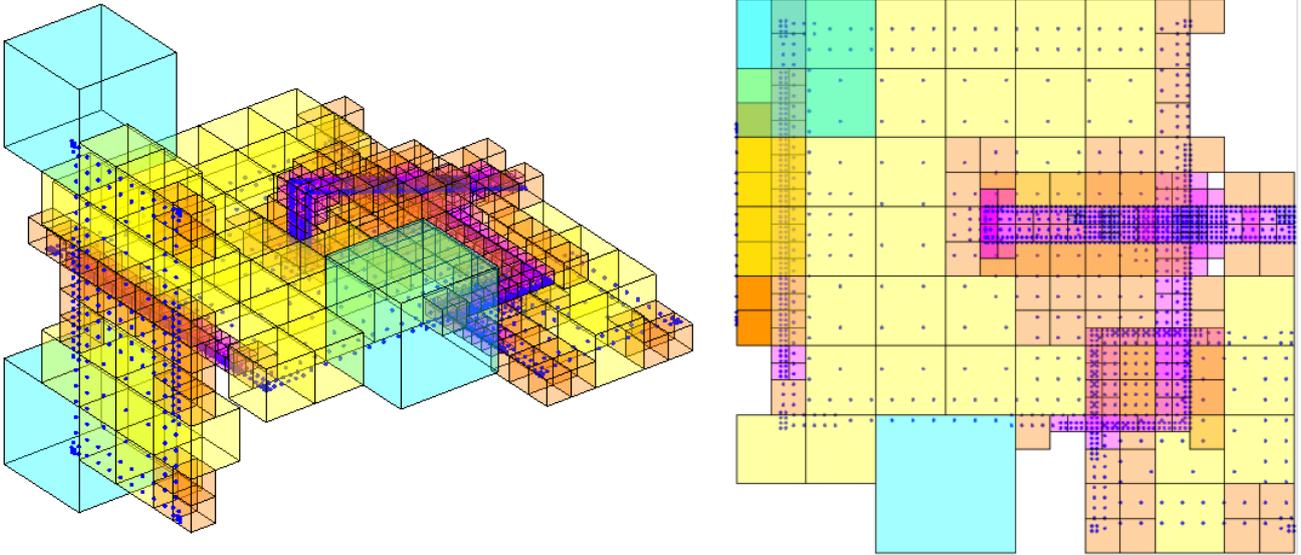


Figure III.16 – Partitionnement adaptatif au niveau 6

II.2) comme dans l'équation III.30 (ou III.32) en combinant les théorèmes 2.1 et 2.3. Une fois la partition de la géométrie effectuée, l'octree pilote les interactions entre les cubes de différents niveaux à partir de différentes listes d'interaction.

3.4 Listes d'interaction avec le cube c : \mathcal{U}_c , \mathcal{V}_c , \mathcal{W}_c et \mathcal{X}_c

Les listes des différents types d'interactions entre les charges et les points d'évaluation du potentiel (ou du champ normal) qui se trouvent dans les cubes du partitionnement adaptatif sont introduites par Greengard et al. dans [48]. Suivant l'éloignement des cubes, leur niveau et leur position dans la partition, quatre listes suffisent pour décrire tous les types d'interactions proches ou lointaines. Elles sont définies pour un cube c de n'importe quel niveau dans lequel on calcule soit le potentiel soit un *local* qui servira au calcul du potentiel. Pour aider à la compréhension, ces listes sont illustrées dans la figure III.17.

Liste \mathcal{U}_c Cette liste est composée de tous les cubes feuilles (qui n'ont pas de fils) qui sont adjacents au cube c . Un cube est adjacent dès qu'il a en commun une face, une arête ou un coin avec un autre cube voisin. Ces cubes ne sont pas nécessairement du même niveau que le cube c . Cette liste est utilisée pour calculer les interactions totales des cubes feuilles, du type $Q2P$ (ou $Q2E$).

Liste \mathcal{V}_c On retrouve dans cette liste tous les cubes (non nécessairement feuilles) de même niveau que le cube c qui sont les cubes fils des cubes adjacents et du même niveau que le cube père du cube c . Les cubes de cette liste sont en interaction $M2L$ avec le cube c . Cette liste s'applique à tous les cubes de tout niveau.

Liste \mathcal{W}_c Cette liste se compose des fils du cube adjacent au père du cube c et qui n'appartiennent pas à la liste \mathcal{U}_c . Les interactions considérées sont du type $M2P$ (ou $M2E$). Ces cubes sont obligatoirement de niveau plus bas que le cube c mais pas nécessairement feuilles. Ces cubes sont

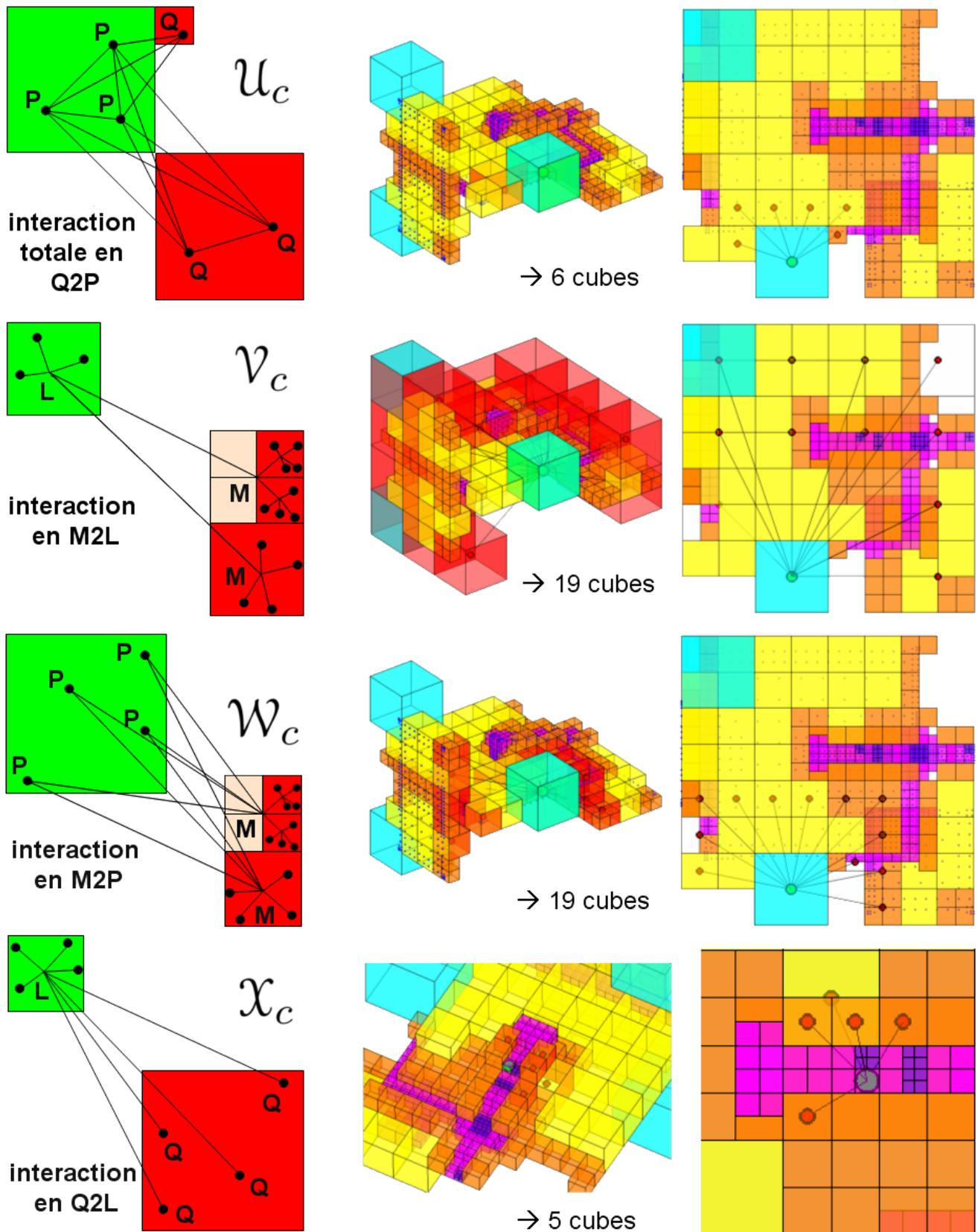


Figure III.17 – Listes d’interaction avec la boîte c : \mathcal{U}_c , \mathcal{V}_c , \mathcal{W}_c et \mathcal{X}_c - illustration sur un exemple

situés entre ceux des listes \mathcal{U}_c et \mathcal{V}_c .

Liste \mathcal{X}_c Enfin, cette dernière liste regroupe tous les cubes i dont le cube c est dans la liste \mathcal{W}_i . Les interactions formées sont du type $L2P$ (ou $L2E$). Ces cubes sont obligatoirement feuilles et de niveau strictement inférieur au cube c .

3.5 Adaptation de l'octree pour considérer la surface des éléments

La version originale de l'octree adaptatif est adaptée pour une répartition non uniforme de charges ponctuelles. Or dans notre problème, on a vu que le maillage n'est pas nécessairement uniforme et que le rapport des surfaces entre les éléments peut être très grand notamment entre ceux situés au centre d'une surface et ceux aux bords. On a adapté le partitionnement et la définition des listes d'interactions en prenant en compte la surface des éléments dans le même esprit que dans [50]. Chaque élément possède deux paramètres généraux : un barycentre et un rayon défini comme le rayon de la sphère englobante de l'élément. A partir de ces deux paramètres les éléments sont rangés dans des cubes de différents niveaux en fonction d'une tolérance qui permet à un élément de sortir légèrement d'un cube. D'ailleurs ce paramétrage est très général, il peut s'appliquer à des éléments ponctuels (de rayon nul), surfaciques et volumiques quelconques. Un élément appartient à un cube de côté c si sa sphère englobante est complètement incluse dans un cube de même centre et de côté $c \times (1 + tol)$, avec la tolérance tol , définie comme un pourcentage III.18.

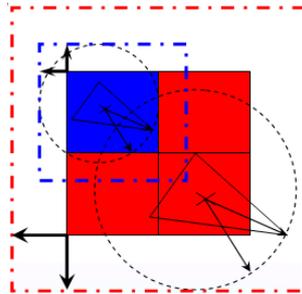


Figure III.18 – Prise en compte de la surface des éléments dans le partitionnement en fonction d'un tolérance définie comme le pourcentage du côté d'un cube

Ce paramètre tol permet à un élément de sortir légèrement d'un cube. Désormais, certains cubes feuilles peuvent se retrouver inclus dans d'autres cubes feuilles plus gros (de niveau supérieur) ou inclure d'autres feuilles plus petits comme le « cube 0 » de la figure III.19. Cela complexifie la définition des listes \mathcal{U}_c , \mathcal{V}_c , \mathcal{W}_c et \mathcal{X}_c présentées précédemment. On ne les redétaillera pas ici mais le principe est le même. Les seules modifications sont la prise en compte des hiérarchies père-fils et de voisinage (au sens des arêtes communes à plusieurs cubes). Une illustration de ces nouvelles listes est présentée en bas de la figure III.19.

Avec la prise en compte des surfaces dans le partitionnement on peut traiter des géométries complexes et non-uniformes en garantissant des notions d'éloignement entre les éléments de toute

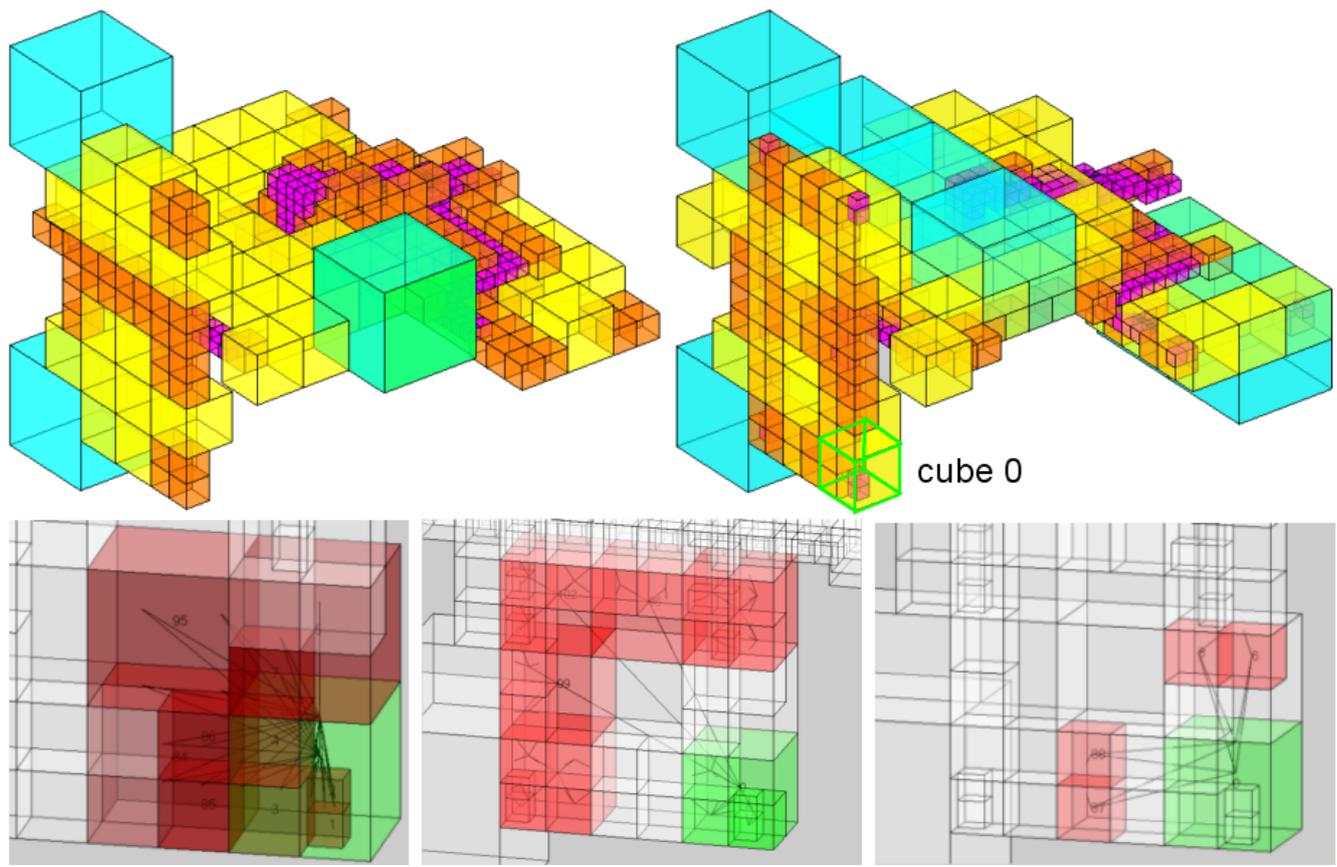


Figure III.19 – Partitionnement adaptatif au niveau 6 avec (à droite) et sans (à gauche) prise en compte des surfaces des éléments; en bas, illustration des nouvelles listes d’interactions \mathcal{U}_0 , \mathcal{V}_0 et \mathcal{W}_0 du cube 0 de niveau 3 (de gauche à droite)

taille. Ainsi, on améliore la précision mais cela dégrade un peu la rapidité des temps de calculs des interactions et augmente légèrement la mémoire requise car cela forme plus d’interactions proches.

Dans la suite, ces listes d’interactions seront utilisées par l’octree pour former la « pseudo matrice » (ensemble de matrices-vecteurs) d’interaction reliant les charges des conducteurs et des diélectriques aux potentiels des conducteurs.

4 Intégration : Construction de la "matrice" d'interaction avec la FMM

On rappelle qu'avec la FMM on ne peut plus construire une vraie matrice d'interaction. Cependant, toutes les matrices d'interaction proche et certains vecteurs d'interaction lointaines de la FMM peuvent être conservés et utilisés à chaque itérations avec un vecteur de charges différents.

4.1 FMM et méthodes de collocation ou de Galerkin à l'ordre 0

On a vu que dans le chapitre précédent qu'on souhaite utiliser une méthode de collocation ou une méthode de Galerkin à l'ordre 0. Avec la FMM, on calcule naturellement les potentiels ou les champs normaux. Ainsi, la méthode de collocation à l'ordre 0 est la méthode la plus simple à mettre en œuvre car les charges des interactions proches et lointaines sont identiques.

En ce qui concerne la méthode de Galerkin, il faut adaptée la FMM, à savoir réintégrer le potentiel et le champ normal sur les fonctions de formes. Dans la section 2.2, on a vu qu'avec une fonction de forme valant 1 sur chaque élément, il suffisait d'intégrer les coefficients P_{ij} ou En_{ij} sur la surface de l'élément. De la même façon que pour les interactions proches on pourrait utiliser l'intégration avec points de Gauss : sur chaque élément, il faudrait calculer avec la FMM le potentiel ou le champ normal à chaque points de Gauss et les multiplier par leur poids respectifs. D'un point de vu pratique, cela ne complexifie pas beaucoup le processus d'intégration car au lieu d'avoir autant de vecteurs $M2P$, $L2P$, $M2E$ ou $L2E$ que d'éléments par cube, il y en aurait N_{pg} fois plus. On a testé cela, et sans surprise, cela n'améliore pas la précision du calcul des charges car il s'agit d'interactions lointaines. Finalement, on suppose que le potentiel et le champ normal sont constants sur l'élément où on les calcule. Et par conséquent, on multiplie le potentiel et le champ normal lointain par la surface de l'élément.

4.2 Calcul des interactions proches

L'intégration proche peut se faire indépendamment de l'intégration lointaine avec les FMM. Dans chaque cube feuille c , on va intégrer¹ une matrice carrée $[P/En]$ pour l'ensemble des éléments chargés du cube c , et des matrices rectangulaires $[P/En]$ composées des interactions entre les éléments du cube c et ceux appartenant aux cubes adjacents du cube c présents dans la liste \mathcal{U}_c . La méthode de calcul des matrices d'interaction est exactement la même que celle présentée dans le chapitre II.

Ces matrices n'étant pas liées à la valeur des charges, une fois intégrées, elles sont stockées pour être utilisées dans le solveur itératif. La taille de ces matrices est égale au nombre d'éléments au carré dans les cubes qui est généralement inférieure à quelques dizaines. Suivant le niveau

1. ces calculs ont été parallélisés, voire l'annexe C, p. 179

de partitionnement et le nombre de charges dans les cubes feuilles, le stockage des ces matrices nécessite très peu de place mémoire (souvent inférieur à quelques pour-cents du stockage d'une matrice d'interaction totale du chapitre précédent).

4.3 Calcul des interactions lointaines

Dans la version originale de l'AMLFMM [48], les vecteurs d'interaction sont calculés dans un certain ordre. On commence par former les vecteurs qui sont liés aux charges ($Q2L$ et $Q2M$) au niveau des cubes feuilles, puis, on translate, converti et/ou additionne éventuellement les *multipoles* ($M2M$ et $M2L$). Enfin, on construit les vecteurs qui rejoignent les points d'évaluations en faisant éventuellement des translation et des additions ($M2P$, $M2E$, $L2L$, $L2P$ et $L2E$). Dans notre code, le calcul des interactions se fait dans le sens inverse : des points de calcul vers les sources comme dans la figure précédente III.15. Dans le calculs de ces interactions lointaines, certains coefficients des théorèmes cités précédemment sont indépendants de la valeur des charges. Ils peuvent être calculés une première fois, puis stockés en mémoire. Il s'agit des coefficients Y_n^m des harmoniques sphériques et des coefficients K_n^m et A_m^m des théorèmes 2.1, 2.2, 2.3 et 2.4. L'intégration² de tous les *multipoles* et *locals* se fait lors du calcul du produit $[P/En].q$.

4.4 Calcul du produit $[P/En].q$

Ce calcul est décomposé en une somme de deux vecteurs : celui du potentiel et du champ normal proche et celui lointain obtenu par la FMM.

$$\begin{pmatrix} V \\ En \end{pmatrix} = \begin{bmatrix} P \\ En \end{bmatrix}_{proche} \cdot \begin{pmatrix} q_c \\ q_d \end{pmatrix} + \begin{pmatrix} V \\ En \end{pmatrix}_{lointain} \quad (\text{III.38})$$

En ce qui concerne le calcul du potentiel et du champ normal proche, il s'agit de simples produit matrice-vecteur. Pour ceux lointains, c'est lors de ce calcul que l'on va additionner, traduire, convertir les *multipoles* et les *locals* en potentiels et champ normal. Grâce à la FMM, ce calcul est relativement rapide par rapport au nombre d'éléments dans la géométrie et consomme très peu de place mémoire. Avec parfois seulement un pour-cent de mémoire d'une matrice en interaction totale, on obtient les mêmes valeurs de potentiel et de champ normal avec une précision relative moyenne de l'ordre de 0,1% (ce que l'on verra plus loin, dans la section 6).

2. ces calculs ont été parallélisés, voire l'annexe C, p. 179

5 Résolution : Calcul des capacités à l'aide de solveur itératif

5.1 Introduction du choix de solveur itératif

On rappelle que contrairement au chapitre II avec la méthode intégrale en interaction totale, on ne dispose plus d'une vraie matrice d'interaction ($[P/En]$). Ainsi, on ne peut pas résoudre directement (avec une décomposition LU par exemple) le problème $[P/En].Q = (V/0)$. C'est pourquoi on utilise une résolution itérative dans laquelle on cherche à minimiser le résidu $r = [P/En].q - (V/0)$ en itérant sur le vecteur de charges q .

De nombreux solveurs itératifs existent, mais cela élimine le choix de certains utilisant des lignes, colonnes ou transposées de la matrice d'interaction puisqu'on n'y a pas accès avec la FMM. Par exemple, les solveurs itératifs comme les doubles gradients conjugués, le Biconjugate Gradient (BCG) ou le Quasi-Minimal Residual (QMR) utilisent la transposée de la matrice d'interaction. Avec la FMM, on ne peut pas y avoir accès ou alors il faudrait faire une seconde intégration en modifiant la construction des interactions.

Dans la littérature, les solveurs itératifs utilisant les sous-espaces vectoriels de Krylov (nom du mathématicien russe travaillant sur des problèmes de valeur propres : décomposition de matrice en polynômes caractéristiques pour des systèmes de grande taille [51]) sont réputés pour être de loin les plus performants. Cette méthode (ainsi que la FMM) développée dans les années 50 par Hestenes, Stiefel, et Lanczos fait partie des dix algorithmes les plus performants du XXème siècle selon SIAM [52, 53]. De plus certains utilisent seulement des produits matrice-vecteur (ici $[P/En].q$) sans avoir besoin d'accéder à la matrice d'interaction. Pour le solveur, la matrice d'interaction est comme une boîte noire. Initialement, cette méthode était conçue pour résoudre de façon itérative un problème linéaire du type $Ax = b$, avec A une matrice carré symétrique et définie positive. Cette résolution utilise une décomposition des résidus dans une base orthogonale formé par les sous-espaces vectoriels de Krylov. Cette méthode a connue de nombreuses évolutions au cours de la deuxième moitié du XXème siècle pour aboutir à des solveurs itératifs performants permettant de résoudre des problèmes non-symétriques : comme BiCGStab (BiConjugate Gradient Stabilized) [54, 55] et GMRES (Generalized Minimum RESidual) [56, 57].

On a choisi dans un premier temps d'utiliser l'algorithme Restart GMRES ou GMRES(m) développé par Saad et Schultz dans les années 90, car celui-ci semble être le plus rapide pour résoudre nos problèmes (nécessitant qu'un produit matrice-vecteur par itération au lieu de 2 pour le BiCGSTAB) [57, 58]. L'algorithme BiCGStab sera également présenté et comparé au GMRES(m). En ce qui concerne le calcul des N^2 capacités parasites à partir de N résolutions, on procède exactement comme au chapitre 3.1.

5.2 Solveur itératif GMRES

Dans cette partie, on va s'intéresser à la résolution itérative du système linéaire $[P/En]Q = (V/0)$ renommé dans cette partie $Ax = b$, avec $A \in \mathbb{R}^{n \times n}$.

5.2.1 Principe des méthodes de Krylov

Le solveur GMRES utilise une méthode de Krylov qui a pour but de minimiser le résidu $r = b - Ax$ de la façon suivante :

$$\min_{x \in x_0 + \mathcal{K}_m(A, r_0)} \|b - Ax\| \quad (\text{III.39})$$

à l'aide d'une décomposition de ce résidu dans une base de l'espace vectoriel de Krylov d'ordre m noté $\mathcal{K}_m(A, r_0)$:

$$\mathcal{K}_m(A, r_0) = \text{vect} \{r_0, Ar_0, A^2r_0, \dots, A^{m-1}r_0\} \quad (\text{III.40})$$

Or en pratique, on ne construit pas directement les vecteurs de cette base de Krylov $\mathcal{K}_m(A, r_0)$ pour cause de dégénérescence numérique : pour des grands systèmes, les vecteurs de cette base peuvent rapidement devenir colinéaire. Ceci implique une stagnation dans la recherche de la solution.

Pour éviter cela, supposons que l'on ait une base orthogonal de $\mathcal{K}_m(A, r_0)$ notée \mathcal{V}_m :

$$\mathcal{V}_m = \{v_1, \dots, v_m\} \quad (\text{III.41})$$

En fait, on cherche à déterminer les itérés x_m qui s'expriment avec les vecteurs de la base orthogonale \mathcal{V}_m tels que :

$$x_m = x_0 + V_m y_m \quad (\text{III.42})$$

avec $V_m \in \mathbb{R}^{n \times m}$, la matrice des m premiers vecteurs de \mathcal{V}_m . On peut réécrire le résidu de [III.39](#) à l'aide de [III.42](#) :

$$\|A(x_0 + V_m y_m) - b\| = \|AV_m y_m - r_0\| \quad (\text{III.43})$$

Le problème revient désormais à rechercher le vecteur y_m qui minimise ce résidu tel que :

$$\min_{y \in \mathbb{R}^m} \|AV_m y_m - r_0\| \quad (\text{III.44})$$

5.2.2 Construction d'une base orthonormée de Krylov : processus d'Arnoldi et Gram-Schmidt modifié

Le processus d'orthonormalisation de Gram-Schmidt permet de construire rapidement (avec peu de produits Ax qui sont coûteux) une base orthonormée notée \mathcal{V}_m de $\mathcal{K}_m(A, r_0)$. Ce processus est aussi appelé processus d'Arnoldi. Le problème se réduit à construire les vecteurs v_i de façon itérative via le processus de Gram-Schmidt et à calculer le vecteur y_m comportant les m compo-

santes définissant la combinaison linéaire des vecteurs de base. Or pendant la construction de la base, à cause d'erreurs numériques, on peut perdre l'orthogonalité des vecteurs. Le processus de Gram-Schmidt modifié permet de garantir cette orthogonalité [59] :

Algorithm 1 Gram-Schmidt modifié

```

1:  $v_1 = r_0 / \|r_0\|$ ;
2: for  $i = 1$  to  $m$  do
3:    $w = Av_i$ ;
4:   for  $k = 1$  to  $i$  do
5:      $h_{k,i} = v_k^T w$ ;
6:      $w = w - h_{k,i} v_k$ ;
7:   end for
8:    $h_{i+1,i} = \|w\|$ ;
9:    $v_{i+1} = w / h_{i+1,i}$ ;
10: end for

```

Dans ce processus, on forme une matrice $H_{m+1,m} \in \mathbb{R}^{m+1 \times m}$ de type Hessenberg :

$$H_{m+1,m} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m} \\ 0 & h_{3,3} & \ddots & h_{3,m} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_{m+1,m} \end{bmatrix} \quad (\text{III.45})$$

D'ailleurs, si la matrice A est symétrique, tous les termes hors diagonale, diagonale inférieure et diagonale supérieure sont nuls. C'est quasiment ce que l'on vérifie lorsqu'il n'y a pas de diélectrique dans la configuration (les coefficients hors diagonales, sont presque nuls car la matrice n'est pas exactement symétrique).

Avec le processus de Gram-Schmidt modifié, en multipliant les lignes 8 et 9 de l'algorithme 1, on peut écrire que, pour tout $i \in (1, 2, \dots, m)$:

$$h_{i+1,i} v_{i+1} = Av_i - \sum_{k=1}^i h_{k,i} v_k \quad (\text{III.46})$$

Et donc,

$$Av_i = \sum_{k=1}^{i+1} h_{k,i} v_k \quad (\text{III.47})$$

Ainsi, on obtient une relation importante entre les vecteurs de la base \mathcal{V}_m , A et $H_{m+1,m}$:

$$AV_m = V_{m+1} H_{m+1,m} \quad (\text{III.48})$$

On peut illustrer cette relation dans la figure III.20 pour mieux se rendre compte des tailles des

matrices en jeu.

$$\begin{array}{ccc}
 \boxed{A} & \boxed{V_m} & = \boxed{V_{m+1}} \boxed{H_{m+1,m}} \\
 \in \mathbb{R}^{n \times n} & \in \mathbb{R}^{n \times m} & \in \mathbb{R}^{n \times m+1} \in \mathbb{R}^{m+1 \times m}
 \end{array}$$

Figure III.20 – Relation $AV_m = V_{m+1}H_{m+1,m}$

Revenons à la minimisation du résidu III.44. Ce dernier peut se réécrire à l'aide de III.48 :

$$\min_{y \in \mathbb{R}^m} \|r_0 - V_{m+1}H_{m+1,m}y_m\| \tag{III.49}$$

Or le premier vecteur de la base orthonormée \mathcal{V}_m vérifie $\|r_0\|v_1 = r_0$. En utilisant $e_1^{(m+1)} = (1, 0, \dots, 0)^T$, le premier vecteur canonique de \mathbb{R}^{m+1} et les propriétés d'orthogonalité des vecteurs v_i , on peut écrire :

$$\|V_{m+1} [\|r_0\|e_1^{(m+1)} - H_{m+1,m}y_m]\| = \|\|r_0\|e_1^{(m+1)} - H_{m+1,m}y_m\| \tag{III.50}$$

Ainsi, en notant $\beta = \|r_0\|$, il reste à trouver la solution y_m du problème linéaire des « moindres carrés » suivant :

$$\min_{y \in \mathbb{R}^m} \|\beta e_1^{(m+1)} - H_{m+1,m}y_m\| \tag{III.51}$$

5.2.3 Factorisation QR avec les rotations de Givens

Pour résoudre ce problème III.51 et vu la faible taille et les propriétés de la matrice $H_{m+1,m}$, on dispose d'une méthode particulièrement pratique : la factorisation QR avec l'utilisation des rotations de Givens. Cela permet de résoudre à moindre coût ce problème par rapport à une factorisation classique [60]. L'objectif est d'obtenir une matrice triangulaire supérieure R_m à partir de $H_{m+1,m}$ telle que :

$$\bar{H}_{m+1,m} = Q_{m+1}H_{m+1,m} = \begin{bmatrix} R_m \\ [0, \dots, 0] \end{bmatrix} \tag{III.52}$$

avec Q_{m+1} la matrice des rotations de Givens.

Une matrice de rotation de Givens en deux dimensions notée $G(\theta)$ avec $\theta \in [-\pi, \pi]$, s'écrit de la façon suivante :

$$G(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \tag{III.53}$$

C'est une matrice orthogonale de déterminant 1. Elle permet la rotation d'un vecteur de la façon suivante :

$$G(\theta) \cdot \begin{pmatrix} \cos \theta \\ -\sin \theta \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (\text{III.54})$$

Une matrice de Hessenberg peut devenir triangulaire supérieure avec l'utilisation des matrices $G_i(c_i, s_i)$:

$$G_i(c_i, s_i) = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \\ & \ddots & c_i & -s_i \\ \vdots & & s_i & c_i & 0 & \vdots \\ & & & 0 & 1 & \ddots \\ & & & & \ddots & \ddots & 0 \\ 0 & & \cdots & & 0 & 1 \end{bmatrix} \leftarrow i\text{\`eme ligne} \quad (\text{III.55})$$

avec i , l'indice de la ligne de la matrice de Givens en deux dimensions. Le processus de factorisation de la matrice $H_{m+1,m}$ utilise $m - 1$ matrices $G_i(c, s)$ pour $i \in [1, \dots, m]$. Ainsi la matrice $Q_{m+1} = G_{m-1} \cdots G_1$ et on peut réécrire le processus de factorisation III.52 de la façon suivante :

$$\overline{H}_{m+1,m} = G_{m-1} \cdots G_1 H_{m+1,m} = \begin{bmatrix} R_m \\ [0, \dots, 0] \end{bmatrix} \quad (\text{III.56})$$

Les coefficients c_i et s_i des matrices $G_i(c_i, s_i)$ sont calculés comme suit :

$$\begin{cases} c_i = & h_{i,i} / \sqrt{h_{i+1,i}^2 + h_{i+1,i}^2} \\ s_i = & -h_{i+1,i} / \sqrt{h_{i+1,i}^2 + h_{i+1,i}^2} \end{cases} \quad (\text{III.57})$$

Pour la dernière colonne, le traitement est légèrement différent mais on applique également une rotation de Givens (détaillé dans [60]) On peut résumer cette section dans l'algorithme 2.

Algorithm 2 Givens Rotations

```

1:   $i - 1^{\text{ème}}$  rotation :
2:  for  $k = 1$  to  $i - 1$  do
3:     $h_{tmp} = c_k \cdot h_{k,i} + s_k \cdot h_{k+1,i}$ ;
4:     $h_{k+1,i} = -s_k \cdot h_{k,i} + c_k \cdot h_{k+1,i}$ ;
5:     $h_{k,i} = h_{tmp}$ ;
6:  end for
7:   $i^{\text{ème}}$  rotation :
8:  if  $h_{i+1,i} = 0$  then
9:     $c_i = 1$ ;  $s_i = 0$ ;
10: else if  $|h_{i,i}| < |h_{i+1,i}|$  then
11:    $h_{tmp} = h_{i,i}/h_{i+1,i}$ ;  $s_i = 1/\sqrt{1+h_{tmp}^2}$ ;  $c_i = h_{tmp} \cdot s_i$ ;
12: else
13:    $tmp = h_{i+1,i}/h_{i,i}$ ;  $c_i = 1/\sqrt{1+tmp^2}$ ;  $s_i = h_{tmp} \cdot c_i$ ;
14: end if
15:  $h_{tmp} = c_i \cdot s_i$ ;
16:  $s_{i+1} = -s_i \cdot h_{tmp}$ ;
17:  $s_i = h_{tmp}$ ;
18:  $h_{i,i} = c_i \cdot h_{i,i} + s_i \cdot h_{i+1,i}$ ;
19:  $h_{i+1,i} = 0$ ;

```

Si on revient à notre résidu, puisque la matrice Q_{m+1} est orthogonale, on peut écrire :

$$\|\beta e_1^{(m+1)} - H_{m+1,m} y_m\| = \|\beta Q_{m+1} e_1^{(m+1)} - \bar{H}_{m+1,m} y_m\| \quad (\text{III.58})$$

Minimiser ce résidu est désormais trivial avec la matrice $\bar{H}_{m+1,m} \in \mathbb{R}^{m+1 \times m}$ dont les m première lignes forme la matrice triangulaire supérieure R_m . Si l'on note $s = (s_1, \dots, s_{m+1}) \in \mathbb{R}^{m+1}$ le vecteur $\|\beta Q_{m+1} e_1^{(m+1)}\|$, il ne reste plus qu'à trouver la solution y_m telle que :

$$\min_{y \in \mathbb{R}^m} \left\| \begin{bmatrix} \begin{pmatrix} s_1 \\ \vdots \\ s_m \end{pmatrix} + R_m y_m \\ s_{m+1} + 0 \end{bmatrix} \right\| \quad (\text{III.59})$$

La solution est triviale, il suffit de résoudre :

$$R_m y_m = S_m \quad (\text{III.60})$$

Avec $S_m = (s_1, \dots, s_m) \in \mathbb{R}^m$. D'ailleurs, on n'a même pas besoin de résoudre ce système trivial III.60 pour connaître la valeur minimale du résidu de III.59. Ce terme obtenu par construction est s_{m+1} . Il représente l'approximation de la valeur du résidu m ième $r_m = b - Ax_m$ avec $x_m = x_0 + V_m y_m$.

5.2.4 Algorithme GMRES

On dispose désormais de toutes les étapes constitutives de l'algorithme GMRES. Celles-ci sont regroupées de façon synthétique et avec les notations utilisées précédemment dans l'algorithme 3.

Algorithm 3 GMRES($A, b, x_0, m, \varepsilon$)

```

1:  $r_0 = b - Ax$ ;  $\beta = \|r_0\|$ ;  $s = \beta e_1^{(m+1)}$ ;
2: for  $i = 1$  to  $m$  do
3:   Gram-Schmidt modifié :
4:   calculer la  $i^{\text{ème}}$  colonne de  $V_m$  et  $H_{m+1,m}$ ;
5:   Rotations de Givens :
6:   appliquer la  $i^{\text{ème}}$  rotation pour former  $\bar{H}_{i+1,i}$ ;
7:   calculer  $S_i = (s_1, \dots, s_i)$  et le résidu  $s_{i+1}$ ;
8:   if  $s_{i+1} < \varepsilon \cdot \|b\|$  then
9:      $y_i = \text{solve}(\bar{H}_{i,i}, S_i)$ ;
10:     $\Rightarrow$  return  $x = x_0 + V_i y_i$ ;
11:   end if
12: end for
13:  $\Rightarrow$  Pas convergé;

```

Si le GMRES n'a pas convergé en m itérations, il faut augmenter la valeur de m . La théorie affirme que le GMRES doit convergé en au moins n itérations si n est la taille de la matrice A . Si le problème est bien conditionné et quasiment symétrique, le GMRES converge souvent en moins de 20 itérations et même pour de grands problèmes (avec des dizaines de milliers d'inconnues). Or, si le conditionnement est mauvais et qu'en plus la matrice n'est pas symétrique (ce qui est le cas avec l'introduction des matériaux diélectrique), le GMRES peut stagner et/ou converger après beaucoup d'itérations (plusieurs centaines).

Pour accélérer la convergence du GMRES, on peut utiliser la version initialement proposée par Saad et Schultz [56] : le restart GMRES ou GMRES(m).

5.3 Solveur itératif GMRES(m)

Si la décomposition du résidu dans le sous-espace de Krylov de taille m ne suffit pas pour annuler le résidu, on incrémente la solution avec la dernière décomposition et on recommence (restart) le GMRES avec un nouveau résidu. Ce processus itératif est décrit dans l'algorithme 4.

5.4 Solveur itératif BiCGSTAB

Le solveur itératif BiCGSTAB est une variante du solveur CGS (Conjugate Gradient Squared [61]). Il est censé améliorer la convergence du solveur et augmenter sa stabilité. L'algorithme CGS est également une adaptation du double gradient conjugué (Biconjugate Gradient - BCG) permettant de ne pas utiliser la matrice transposée A^T . Le BCG, utilise également un processus

Chapitre III. Extraction des capacités par méthode multipolaire rapide à multi-niveaux adaptatifs

Algorithm 4 GMRESm($A, b, x_0, m, \varepsilon, max_{it}$)

```

1:  $r_0 = b - Ax$ ;  $\beta = \|r_0\|$ ;
2: for  $it = 0$  to  $max_{it}$  do
3:    $s = \beta e_1^{(m+1)}$ ;
4:   for  $i = 1$  to  $m$  do
5:     construire la base  $\mathcal{V}_i$  (Gram-Schmidt modifié);
6:     minimiser le résidu et calculer  $s_{i+1}$  (Rotations de Givens);
7:     if  $s_{i+1} < \varepsilon \cdot \|b\|$  then
8:        $y_i = \text{solve}(\overline{H}_{i,i}, S_i)$ ;
9:        $\Rightarrow$  return  $x = x_0 + V_i y_i$ ;
10:    end if
11:  end for
12:   $y_m = \text{solve}(\overline{H}_{m,m}, S_m)$ ;
13:   $x = x_0 + V_m y_m$ ;
14:   $r_{it} = b - Ax$ ;
15:   $s_{i+1} = \|r_{it}\|$ ;
16:  if  $s_{i+1} < \varepsilon \cdot \|b\|$  then
17:     $\Rightarrow$  return  $x$ ;
18:  end if
19: end for
20:  $\Rightarrow$  Pas convergé;

```

de projection du sous-espace de Krylov $\mathcal{K}_m(A, v_0)$ orthogonalement au sous-espace $\mathcal{L}_m(A^T, w_0)$:

$$\mathcal{L}_m = \text{vect} \{w_0, A^T w_0, (A^T)^2 w_0, \dots, (A^T)^{m-1} w_0\} \quad (\text{III.61})$$

avec $v_0 = r_0 / \|r_0\|$ et w_0 est un vecteur quelconque tel que $\langle v_0, w_0 \rangle \neq 0$.

Le principe du CGS est d'évaluer le résidu à chaque itération dans une simple boucle de la façon suivante :

$$r_j = \phi_j(A) r_0 \quad (\text{III.62})$$

avec ϕ_j un polynôme de degré j satisfaisant $\phi_j(0) = 1$. Dans le BiCGSTAB, l'utilisation d'un autre polynôme noté ψ_j permet de stabiliser et lisser la convergence du solveur. Ce dernier polynôme est défini de façon récursive à chaque itération :

$$\psi_{j+1}(A) = (1 - \omega_j A) \psi_j(A) \quad (\text{III.63})$$

Ainsi, à chaque itération, le résidu est calculé de la façon suivante :

$$r'_j = \psi_j(A) \phi_j(A) r_0 \quad (\text{III.64})$$

On ne détaillera pas ici le principe de fonctionnement (bien détaillé dans [57]). L'algorithme utilisé est la version de van der Vorst [54, 55] :

Algorithm 5 BiCGSTAB($A, b, x_0, \varepsilon, max_{it}$)

```

1:  $r = b - Ax_0$ ;  $\omega = 1.0$ ;  $r_{ini} = r$ ;
2: for  $it = 0$  to  $max_{it}$  do
3:    $p = r$ ;  $\rho = r_{ini}^T \cdot r$ ;
4:   if  $iter > 1$  then
5:      $\beta = (\rho / \rho_0) \cdot (\alpha / \omega)$ ;
6:      $p = p + \beta(p - \omega v)$ ;
7:   end if
8:    $v = Ap$ ;
9:    $\alpha = \rho / (r_{ini}^T \cdot v)$ ;
10:   $s = r - \alpha v$ ;
11:  if  $\|s\| < \varepsilon$  then
12:     $\Rightarrow$  return  $x = x + \alpha p$ ;
13:  end if
14:   $t = As$ ;
15:   $\omega = (t^T s) \cdot (t^T t)$ ;
16:   $x = x + \alpha p + \omega s$ ;
17:   $r = s - \omega t$ ;
18:  if  $\|r\| / \|b\| < \varepsilon$  then
19:     $\Rightarrow$  return  $x$ ;
20:  end if
21:   $\rho_0 = \rho$ ;
22: end for
23:  $\Rightarrow$  Pas convergé;

```

5.5 Préconditionnement à gauche

Pour optimiser le temps de résolution de l'algorithme itératif GMRES(m) 4 (ou BiCGSTAB 5), on peut utiliser un preconditionnement. Ce dernier a pour objectif d'améliorer le conditionnement de la matrice A qui vaut :

$$\begin{aligned} \mathcal{C}(A) &= \|A\| \|A^{-1}\| \\ &= \frac{\sqrt{\lambda_{\min} A^T A}}{\sqrt{\lambda_{\max} A^T A}} \end{aligned} \quad (\text{III.65})$$

avec λ_{\min} et λ_{\max} respectivement la plus petite et la plus grande valeur propre de A . L'idéal est d'avoir un conditionnement proche de 1 comme pour la matrice identité I . L'idée est de trouver une matrice de preconditionnement M^{-1} telle que $M^{-1}A \approx I$. On résout ensuite le problème $M^{-1}Ax = M^{-1}b$.

Par exemple on peut utiliser un preconditionnement à gauche : $r = M^{-1} \cdot ([P/En] \cdot q - (V/0))$, soit M^{-1} est l'inverse de la diagonale de la « matrice » $[P/En]$ (méthode de jacobi), soit les inverses de tous les blocs diagonaux (vraies matrices) de la pseudo matrice (méthode bloc-jacobi) [38, 49]. Grâce aux calculs des interactions proches, on a facilement accès à ces petites matrices pleines. En réalité, dans le GMRES(m) on décompose le résidu dans une autre base de Krylov $\mathcal{K}_m(M^{-1}A, r_0)$:

$$\mathcal{K}_m = \text{vect} \left\{ r_0, M^{-1}Ar_0, (M^{-1}A)^2r_0, \dots, (M^{-1}A)^{m-1}r_0 \right\} \quad (\text{III.66})$$

Ce preconditionnement est réalisé³ à chaque calcul de résidu (r_i pour $i \in (0, \dots, m)$) dans les algorithmes GMRES 3 et GMRES(m) et 4) et puis lors de la construction des vecteurs de la base orthonormée d'Arnoldi avec l'algorithme de Gram-Schmidt modifié au niveau du calcul $w = Av_i$ dans 1. De même dans le BiCGSTAB, ce preconditionnement à gauche peut être réalisé sur r avant le calcul de v et sur s avant le calcul de t dans l'algorithme BiCGSTAB 5.

5.6 Résolution à plusieurs seconds membres

Comme dans la section 3.1 du chapitre précédent, on peut résoudre le problème avec N seconds membres pour calculer la matrice des capacités. Cela permet de gagner beaucoup de temps. Dans les deux algorithmes, certaines routines sont facilement factorisables comme la construction de la base d'Arnoldi, les rotations de Givens et les preconditionnements ainsi que le calcul du potentiel et du champ normal proche. Malheureusement, le calcul du champ lointain n'est pas factorisable.

3. ces calculs ont été parallélisés, voire l'annexe C, p. 179

6 Validations et performances

6.1 Intégration avec la FMM

6.1.1 Performances générales

L'algorithme FMM à niveau constant a été implémenté en langage Matlab dans un premier temps, puis la version adaptative en langage java (plus adapté pour la programmation objet).

En ce qui concerne les performances de la FMM, on a repris l'exemple du chapitre 4.1.1 à savoir les deux plaques conductrices parallèles de plus en plus maillées. La figure III.21 présente la comparaison des temps d'intégration (méthode de collocation avec 4 points de Gauss) et d'une résolution de charges avec un GMRES(30) préconditionné par les blocs diagonaux et une tolérance de $\varepsilon = 10^{-7}$.

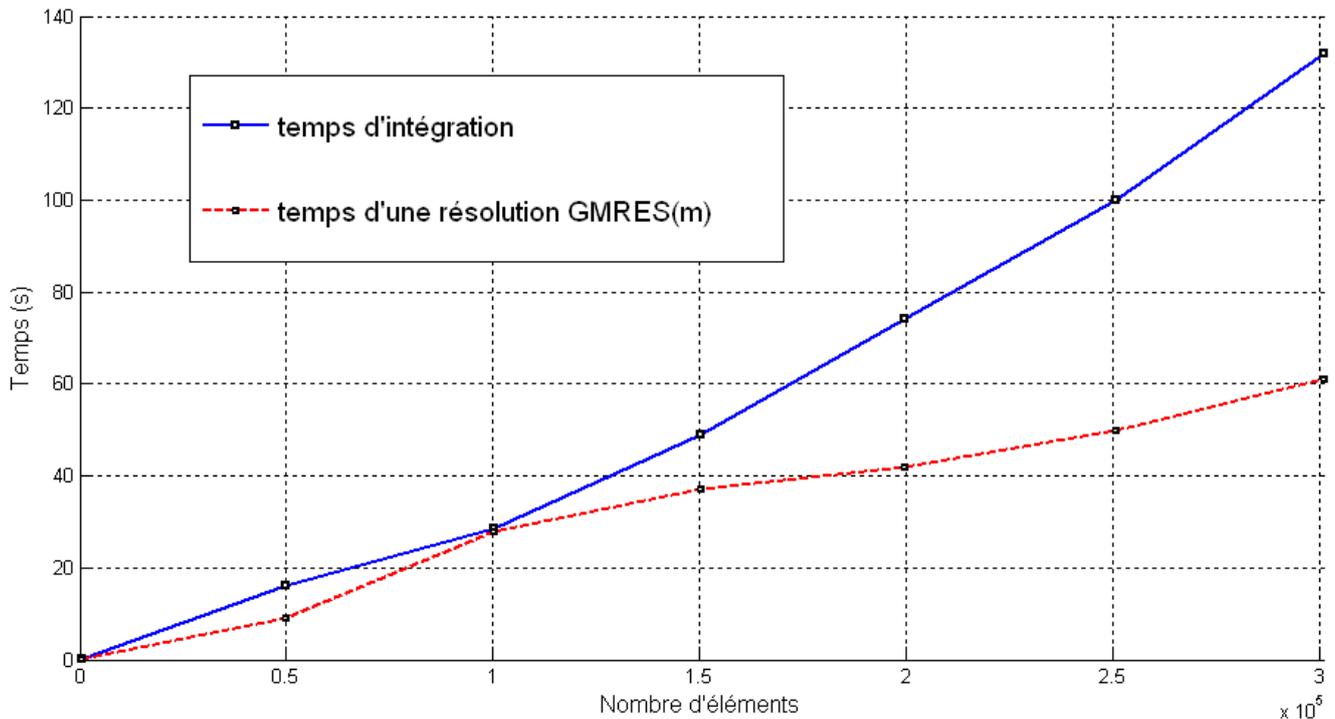


Figure III.21 – Temps d'intégration et de résolution des charges pour deux plaques parallèles de plus en plus maillées avec l'AMLFMM

Dans la figure III.21, on voit bien que la FMM est très rapide et peu gourmande en mémoire. Le temps d'intégration comprend le temps de calculs des petites matrices pleines en interaction totale, le temps de partitionnement et du premier calcul de potentiel lointain avec l'AMLFMM. On a réussi à intégrer et résoudre une géométrie maillée en 300 000 éléments environ avec un ordinateur de 3 Go de mémoire vive en consommant moins de 1,6 Go (taille limite de la machine virtuelle en java).

Le tableau III.2 présente le détail des performances pour deux configurations des 2 plaques

Chapitre III. Extraction des capacités par méthode multipolaire rapide à multi-niveaux adaptatifs

parallèles maillées en 10 000 et 200 000 éléments environ. Le niveau de partitionnement est fixé à 6, l'intégration est réalisée avec la méthode de Galerkin avec 4 points de Gauss. Le solveur utilisé est le GMRES(m) préconditionné par blocs pour $m = 50$, sa tolérance est $\varepsilon = 10^{-10}$. On retrouve

Nb élts	FMM	MoM	% full	Prec	tps(Prec)	GMRES(50)	tps(reso)	Nb it	% err.
10 082	76 Mo	8,3 Mo	1	0,86 Mo	0,047 s	4,6 Mo	8,3 s	52	3.10^{-12}
199 712	470 Mo	350 Mo	0,11	39 Mo	1,2 s	91 Mo	133,6 s	85	1.10^{-12}

Tableau III.2 – Performances pour les 2 plaques parallèles

dans ce tableau l'espace mémoire nécessaire (exprimée en Mo) pour le calcul des interactions lointaines avec la FMM et celui des interactions proches avec la méthode intégrale en interaction totale (ici noté MoM). La colonne notée « % full » représente le pourcentage de mémoire des interactions proches par rapport à celui d'une méthode en interaction totale (sans FMM). Le nombre d'itérations noté « Nb it » représente le nombre de fois où l'on décompose le vecteur $A_v i$ dans la base de Krylov. Enfin, « % err » représente l'erreur relative moyenne en valeur absolue sur le calcul des N charges, définie par :

$$Err_{moy} = \frac{1}{100N} \sum_{i=1}^N \frac{|b_i - Ax_i|}{|b_i|} \quad (\text{III.67})$$

avec dans notre cas $b = V$: le second membre fixé, et $x = q$: le vecteur de charge solution donnée par le GMRES(m).

Ce tableau III.2 montre qu'avec très peu de place mémoire (moins de 1% de mémoire par rapport à un problème en interaction totale) on obtient de très bons résultats sur les valeurs des charges.

6.1.2 Tests sur le degré de la troncature des développements multipolaire

On a validé les calculs de charges et de capacités sur un panel de configurations avec ou sans matériaux diélectriques. Le partitionnement adaptatif en cubes de différents niveaux avec l'octree, permet d'optimiser les chemins d'interaction et donc garantit une meilleure précision sur les calculs de potentiel. Si on peut avoir jusqu'à 1% d'erreur sur le potentiel lointain, cette erreur diminue beaucoup sur le potentiel exacte car c'est la somme de celui lointain et de celui proche. Et comme les interactions sont calculées en $1/r$ (ou $1/r^2$), les interactions proches sont beaucoup plus influentes. Les mêmes études ont été effectuées sur le champ normal lointain et donnent des conclusions similaires.

On essaye également d'obtenir un minimum de 4 éléments par cubes environ. En règle générale, pour obtenir une bonne précision de calcul (inférieure à 0,01% voir 0,001% d'écart par rapport à la méthode en interaction totale) avec une répartition de charges surfaciques, une décomposition multipolaire au degré 2 suffit. Par exemple, la figure III.22 montre sur une configuration maillée

en éléments surfaciques les erreurs sur le potentiel entre la méthode en interaction totale et la FMM.

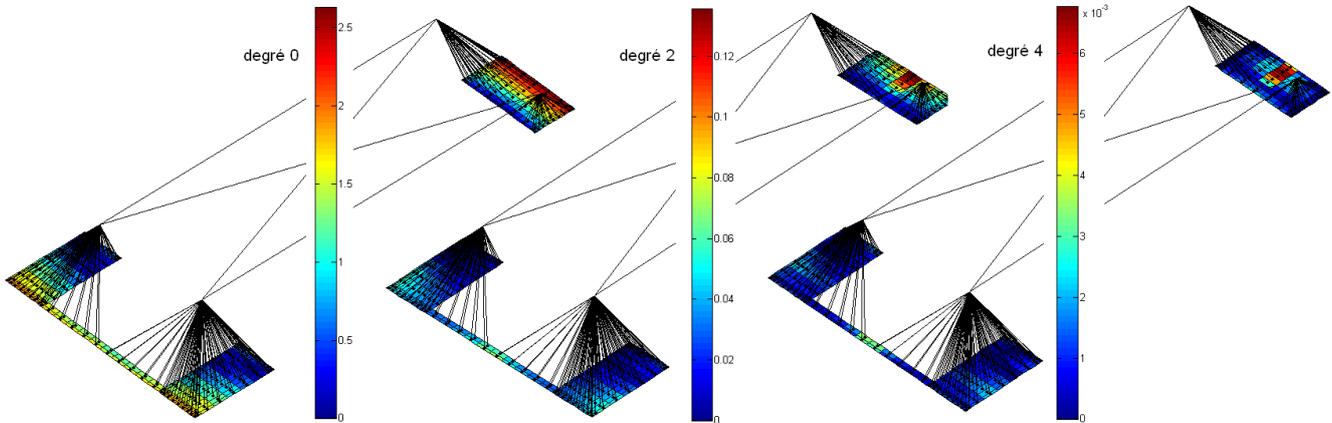


Figure III.22 – Erreurs relatives en % sur le potentiel entre la méthode en interaction totale et un partitionnement à niveau $L = 2$ et aux degrés 0, 2 et 4

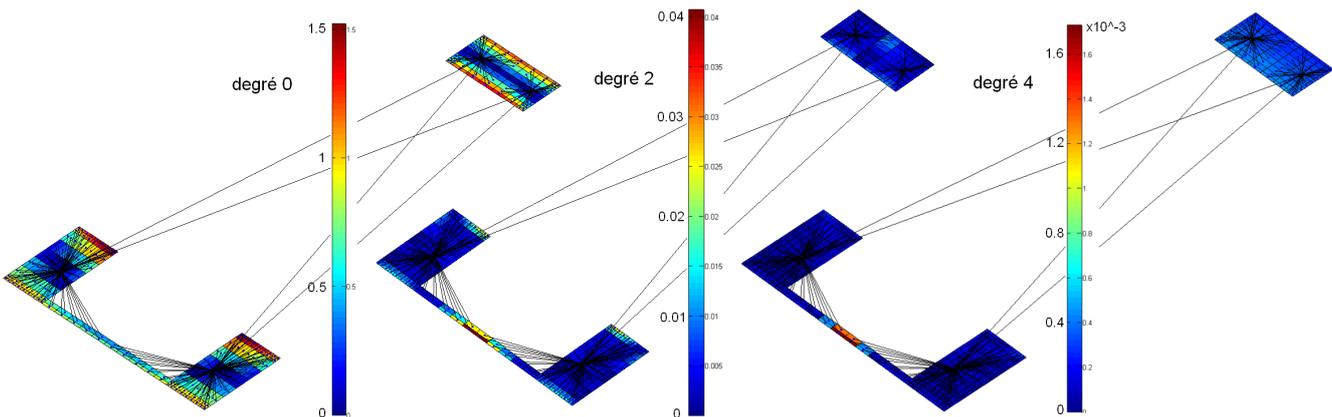


Figure III.23 – Erreurs relatives en % sur le potentiel entre la méthode en interaction totale et l'AMLFMM avec recentrage des centres des cubes de plus niveau à niveau $L = 2$ et aux degrés 0, 2 et 4

On remarque dans la figure III.22 que ces erreurs sont plus importantes dans les coins des cubes feuilles. Ainsi, à précision égale ($< 0.01\%$ sur les valeurs de charges) on obtient les mêmes résultats avec seulement 1 à 10% de mémoire de la matrice en interaction totale.

6.1.3 Vers une décomposition mutlipolaire adaptative

En regardant, la figure III.22, on remarque que les erreurs sont maximales pour les charges (ou les points de potentiels, tous situés au barycentre des éléments à l'ordre 0) situées loin du centre de décomposition mutlipolaire. On s'est intéressé au recentrage des centres des cubes de plus bas niveau à partir des barycentres des éléments (Fig. III.24).

On peut comparer les figures III.22 et III.23. Le tableau III.3 illustre les erreurs relatives moyennes en valeur absolue et maximales de ces deux figures. Pour des partitionnements à niveau

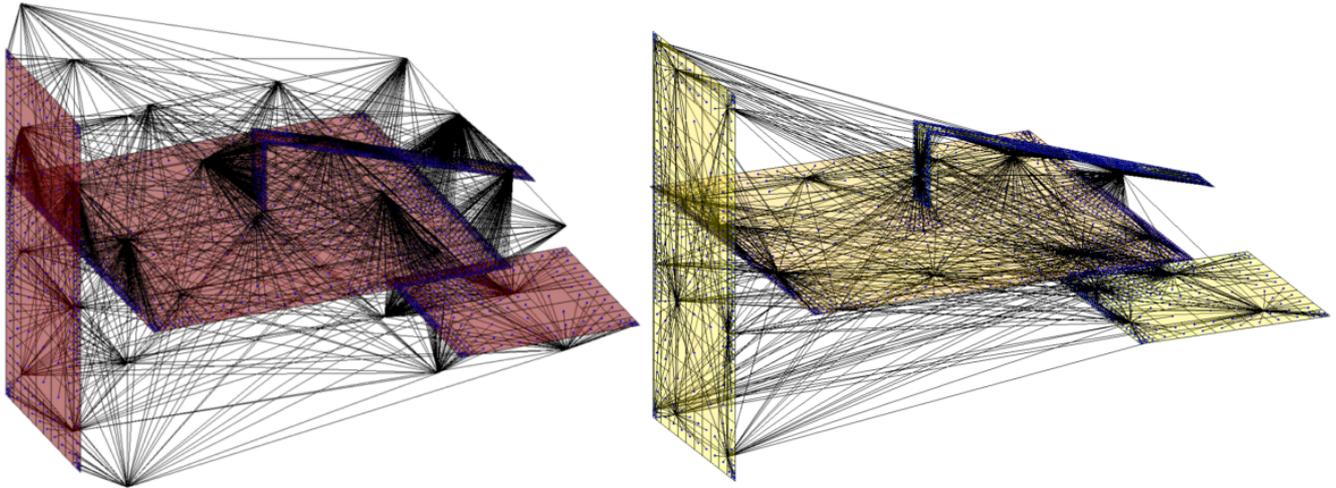


Figure III.24 – Interactions lointaines pour un partitionnement adaptatif au niveau $L = 2$ avec (à droite) ou sans (à gauche) recentrage des centres des cubes de plus bas niveau

degré	ε_{moy}	ε_{moy}^{rec}	$\varepsilon_{moy}/\varepsilon_{moy}^{rec}$	ε_{max}	ε_{max}^{rec}	$\varepsilon_{max}/\varepsilon_{max}^{rec}$
0	1,14	0,72	1,6	2,63	1,52	1,7
2	0,032	$6,6 \cdot 10^{-3}$	4,8	0,14	0,041	3,3
4	$9,9 \cdot 10^{-4}$	$2 \cdot 10^{-4}$	4,95	$6,8 \cdot 10^{-3}$	$1,7 \cdot 10^{-3}$	4

Tableau III.3 – Comparaison des erreurs relatives (exprimées en %) en valeur absolue moyennes et maximales entre la méthode en interaction totale et l'AMLFMM ($L = 2$) avec (ε_{moy}^{rec} ou ε_{max}^{rec}) ou sans (ε_{moy} ou ε_{max}) recentrage des centres de plus bas niveau ; les colonnes 4 et 7 sont les rapports entre ces erreurs - configuration des figures III.22 et III.23

niveau	ε_{moy}	ε_{moy}^{rec}	$\varepsilon_{moy}/\varepsilon_{moy}^{rec}$
$L = 2$	0,53	0,32	1,7
$L = 4$	1,85	1,32	1,4
$L = 6$	2,24	1,97	1,1

Tableau III.4 – Comparaison des erreurs moyennes relatives en valeur absolue entre la méthode en interaction totale et l'AMLFMM pour différents niveaux avec ou sans recentrage - configuration des 4 conducteurs de la figure III.24

bas ($L = 2$ ou 3), le recentrage permet de gagner beaucoup en précision comme le montre le tableau III.3. Cependant pour des géométries plus complexes comme celle de la figure III.24 avec un niveau de partitionnement plus élevé ($L = 4$ à 6), ces gains en précision sont atténués à cause des chemins d'interaction passant par des $M2M$ non recentrés comme le montre le tableau III.4.

Nous devons rester prudent avec ces résultats car la Fast Multipole Method et les théorèmes d'expansion multipolaire sont basés sur des sphères de validité, et déplacer les centres de décomposition revient à déplacer et modifier par conséquent ces sphères de validité. Ce travail demeure à l'état de recherche et doit être plus approfondi.

6.1.4 Comparaison avec le logiciel utilisant la MLFMM - FastCap

De nombreuses configurations ont été modélisées avec un solveur utilisant la Fast Multipole Method. Il s'agit du logiciel FastCap développé par le MIT sur les travaux de Nabors et White dans les années 90 [32, 33, 49, 62]. La méthode Multi-Level Fast Multipole Method est utilisée (premier article de Greengard et al. [47]). Celle-ci a été développée en réel⁴ et codée en langage C dans un logiciel libre disponible sur internet [63].

Les écarts relatifs entre les valeurs des capacités (de Maxwell) obtenues par FastCap et par notre méthode sont quasiment identiques. Considérons par exemple la configuration de 8 bus-barres de l'article [32] présentée dans la figure III.25.

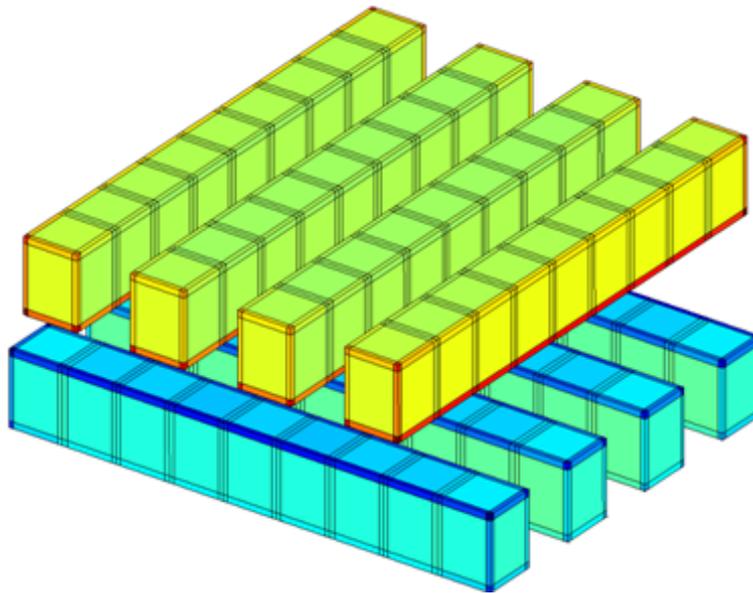


Figure III.25 – Bus-barre de 8 conducteurs et 2 conducteurs dont l'un est entouré de matériaux diélectrique

Les matrices de capacités obtenues ont pour écart relatif en valeur absolue moyen entre 0,5 et 4% selon la qualité d'intégration choisie à niveau de partitionnement égal ($L = 3$) et pour la même tolérance dans le GMRES(m) ($\varepsilon = 10^{-6}$). Dans ces conditions, on a comparé les performances en temps et en précision par rapport à FastCap et à la méthode en interaction totale pour 3 algorithmes de FMM et 3 techniques d'intégration avec une méthode de collocation à l'ordre 0. Ces résultats sont présentés dans le tableau III.5.

Dans ce tableau, on a utilisé la Multi-Level FMM (notée « MLFMM »), l'Adaptive MLFMM (notée « AMLFMM(4,1) » avec si possible entre 4 et 1 charge par cube⁵) et notre méthode adaptée à la surface des éléments avec l'AMLFMM (notée « A2MLFMM(5%) » : on tolère un dépassement de

4. La FMM utilise des grandeurs complexes (harmoniques sphériques et d'autres coefficients) pour calculer des potentiels (ou des champs) qui sont réels

5. On rappelle que le partitionnement adaptatif essaye de contrôler le nombre de charges par cubes en fonction du niveau de partitionnement le plus bas.

Chapitre III. Extraction des capacités par méthode multipolaire rapide à multi-niveaux adaptatifs

	FastCap	MLFMM			AMLFMM(4,1)			A2MLFMM(4,1,5%)		
		col0(64)	col0(an)	col0(4)	col0(64)	col0(an)	col0(4)	col0(64)	col0(an)	col0(4)
ε_{IT}	1,54	1,72	1,72	5,17	1,72	1,72	5,17	1,38	4,83	1,39
ε_{FC}	-	0,97	0,79	4,39	0,97	0,79	4,39	0,53	0,53	0,53
tps	23	20	9,8	5,2	20	10,7	6,2	52	25	11
mém	47		31			31			57	

Tableau III.5 – Comparaison de différentes FMM et techniques d'intégration avec FastCap et la méthode en interaction totale

5% des éléments dans les cubes de différents niveaux). Les techniques d'intégration sont numériques avec 4 ou 64 points de Gauss (col0(4) ou col0(64)) ou analytique (col0(an)). Le GMRES(m) est préconditionné à gauche par les matrices diagonales. Les erreurs notées ε_{IT} et ε_{FC} (exprimées en %) sont respectivement les erreurs relatives moyennes en valeur absolue entre les matrices de capacités calculées avec les différentes FMM et la méthode en interaction totale et FastCap. Les temps du tableau III.5 (notée « tps », exprimés en secondes) sont les temps totaux des problèmes (intégration + préconditionnement + résolution + calcul des capacités). Enfin, la dernière ligne du tableau indique la mémoire nécessaire (exprimée en Mo) des différents algorithmes.

Les conclusions que l'on peut tirer de ces comparaisons et que l'on obtient des valeurs de capacités à peu près similaires (précision inférieure à quelques %) entre FastCap ou la méthode en interaction totale. Les calculs sont plus précis en prenant en compte la taille des éléments, liée directement à leur surface (Fig. III.18). D'ailleurs dans ce problème, les écarts de taille sont vraiment très important. Concernant la consommation de mémoire des différentes FMM du tableau III.5, elles sont à peu près identiques. La prise en compte de la taille des éléments accroît le nombre d'interactions proches, c'est pourquoi cela consomme un peu plus de mémoire. Cependant, pour des gros problèmes de plusieurs dizaines de milliers d'éléments, cet accroissement devient quasiment négligeable. On peut noter aussi que pour cet exemple, les temps d'exécution de notre code sont dans les mêmes ordres de grandeur que FastCap. A ce sujet, nous avons fait une autre série de test pour comparer les temps d'exécution en fonction du nombre d'éléments et du niveau de partitionnement.

Considérons le problème des deux plaques parallèles maillées de plus en plus. La figure III.26 montre les temps d'exécution des différents problèmes (temps d'intégration + préconditionnement + résolution + calcul des capacités). La méthode FMM choisie est l'A2MLFMM(4,1,25%) en collocation à l'ordre 0 avec 4 points de Gauss. Le GMRES(m) est préconditionné à gauche par les matrices diagonales et à pour tolérance $\varepsilon = 10^{-6}$ (la même que FastCap). La figure III.26 montre que notre méthode est au-moins aussi rapide que le logiciel FastCap lorsque le nombre d'éléments par cube est faible (pour un niveau de partitionnement élevé, le pourcentage d'interaction calculées avec la FMM est largement majoritaire). Pour des niveaux de partitionnement plus bas, les interactions proches sont de plus en plus nombreuses et dans ce cas notre méthode est beaucoup plus rapide que FastCap. Par exemple pour la configuration maillée en 25 088 éléments, les nombres moyens d'éléments par cube avec notre méthode pour les niveaux $L = 4, 5$ ou 6 sont

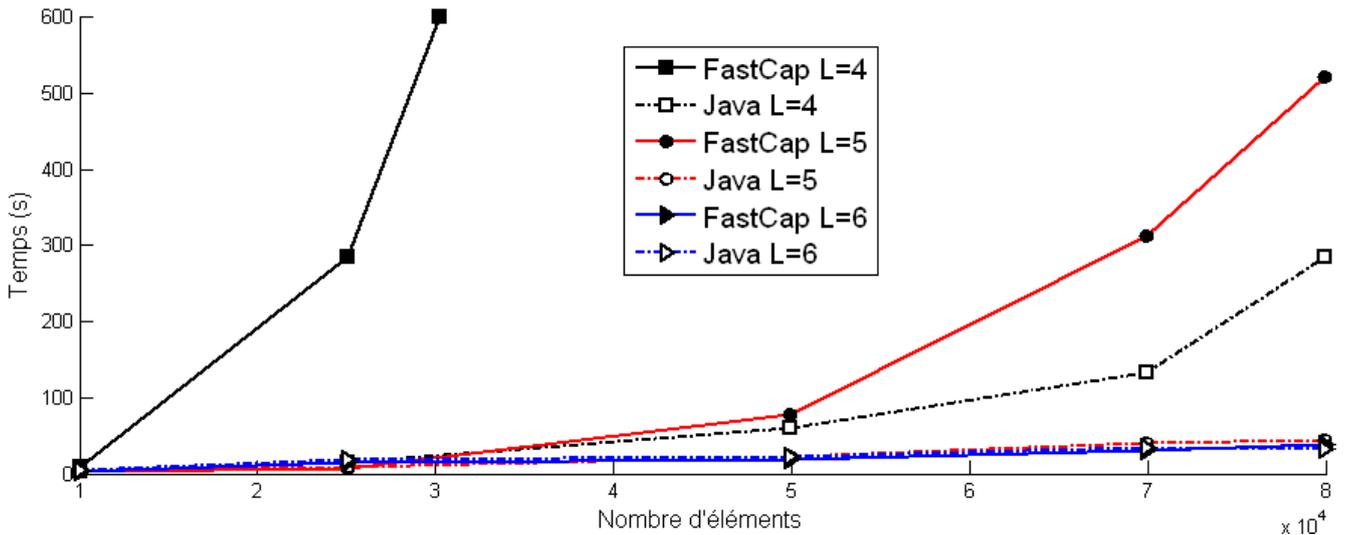


Figure III.26 – Comparaison des temps d’exécution du problème des 2 plaques parallèles différemment maillées entre le logiciel FastCap (MLFMM) et notre code java : l’A2MLFMM(4,1,25%)

respectivement de 5, 18 et 74. Pour la configuration maillée en 80 000 éléments, on obtient pour les mêmes niveaux dans l’ordre, 15, 59 et 235. Concernant la consommation de mémoire, celle-ci est quasiment identique entre les deux approches.

Dans la suite de ce manuscrit, on ne détaillera pas nécessairement le type de FMM utilisée. Dans la plupart des cas, on utilise l’A2MLFMM(4,1,25%) : une méthode qui donne un bon compromis entre la rapidité d’exécution, la précision et la place mémoire.

6.2 Résolution avec solveur itératif

6.2.1 Comparaisons des solveurs pour un problème en matrice pleine

La FMM impose l’utilisation d’un solveur itératif. Or on s’est aperçu que, pour tous nos problèmes intégrés avec la méthode en interaction totale du chapitre II, le GMRES(m) ou le BiCGSTAB sont beaucoup plus rapides « à précision égale »⁶ que le solveur direct LU de Matlab. Ces comparaisons sont illustrées avec l’exemple des 2 plaques parallèles dans la figure III.27 pour un problème en interaction totale intégré par la méthode de collocation à l’ordre 0 avec 4 points de Gauss.

De plus, ces algorithmes consomment vraiment très peu de mémoire. On remarque que le GMRES(m) est plus rapide que le BiCGSTAB. Les seuls vecteurs les plus lourds à stocker sont les m vecteurs $v_i \in \mathbb{R}^n$ de la base orthonormée \mathcal{V}_m plus 8 autres pour le GMRES(m) et seulement 11 pour le BiCGSTAB. En revanche, pour la configuration des deux plaques maillées en 10 000 éléments, la mémoire nécessaire pour le stockage de la matrice d’interaction et pour la résolution LU dépasse les 3 Go. Ce problème a pu être résolu avec le GMRES(m) avec seulement 8 Mo de mémoire !

6. Le solveur LU donne une solution exacte avec la précision de la machine

Chapitre III. Extraction des capacités par méthode multipolaire rapide à multi-niveaux adaptatifs

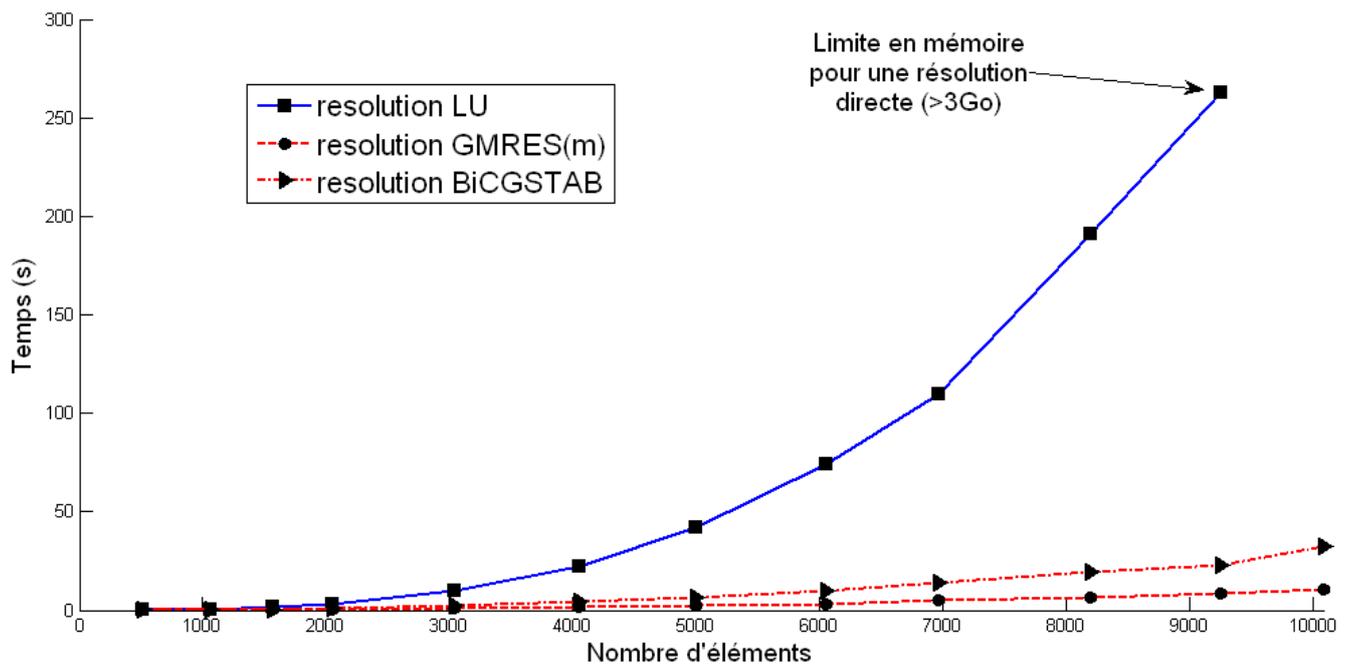


Figure III.27 – Comparaison des temps d’une résolution de charges pour l’exemple des 2 plaques parallèles (+1 V et 0 V) différemment maillées avec le solveur direct LU de Matlab, le GMRES(m) et le BiCGSTAB pour un préconditionnement diagonal et une tolérance de $\varepsilon = 10^{-8}$

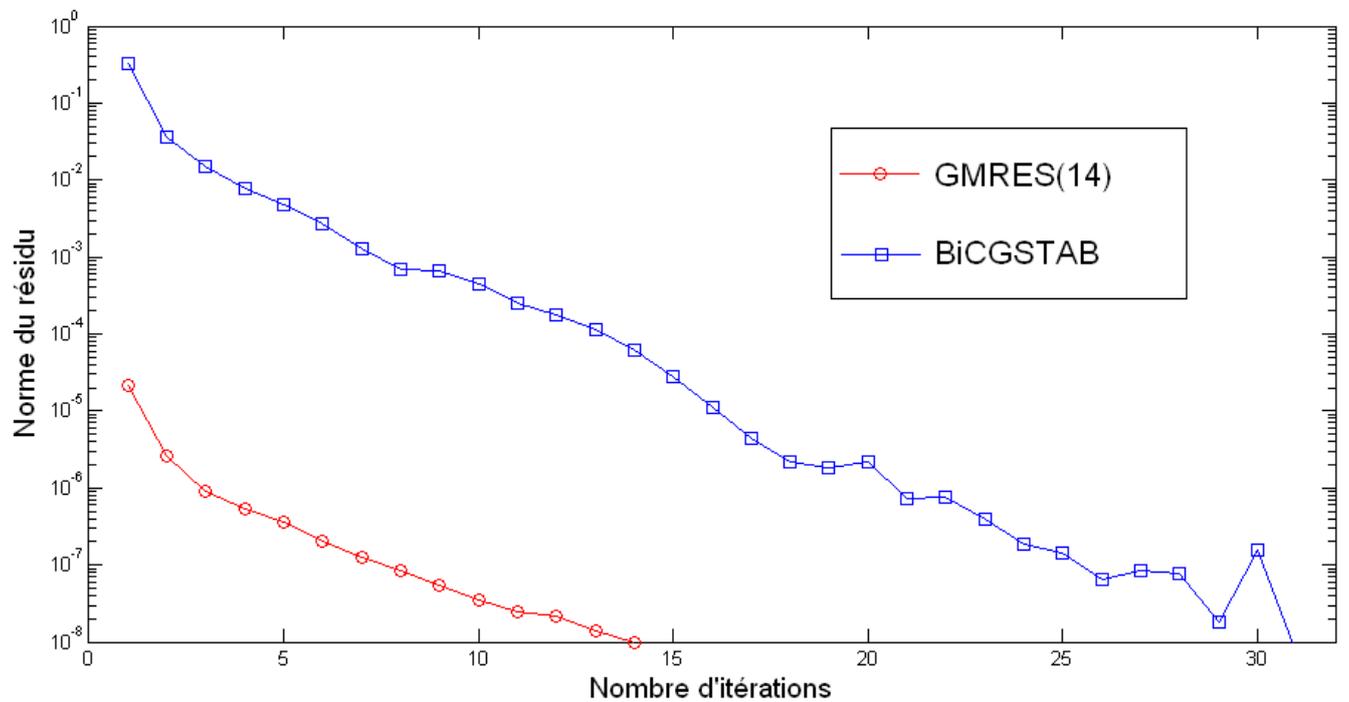


Figure III.28 – Norme du résidu en fonction des itérations du GMRES(m) (ici pour $m = 14$) et du BiCGSTAB pour le problème des 2 plaques maillées en 10 082 éléments et une tolérance de $\varepsilon = 10^{-8}$, les deux solveurs sont préconditionnés par la diagonale des matrices en interaction totale

Le BiCGSTAB étant moins rapide que le GMRES(m) pour toutes les configurations traitées, dans la suite ce manuscrit, celui-ci ne sera plus utilisé dans la suite de ce manuscrit.

6.2.2 Résolution avec le solveur GMRES(m)

L'objectif de cette partie est de se familiariser avec les différents paramètres du GMRES(m). Il s'agit de trouver un compromis entre la consommation de mémoire et le temps de résolution. En règle générale, plus il y a d'inconnues, plus la résolution est longue. Et plus on consomme de mémoire pour résoudre (dans l'algorithme et dans le préconditionnement), plus la résolution est rapide.

On a vu dans lors de la présentation de l'algorithme du GMRES(m) que celui-ci dépend de plusieurs paramètres : le point de départ x_0 , la taille du sous-espace de Krylov m , la tolérance sur la norme du résidu ε et la nature du préconditionnement (diagonale ou matrices des blocs diagonaux).

Paramètre x_0 :

En ce qui concerne le vecteur x_0 , aucun essai n'a permis de trouver un meilleur point de départ que le vecteur nul ! En partant d'un vecteur x_0 proche de la solution, par exemple, en résolvant un GMRES(m) avec les interactions proches seulement (Fig. III.29), le nombre d'itérations ne diminue pas. D'ailleurs dans la littérature, toutes les références utilisent $x_0 = (0, 0, \dots, 0)$.

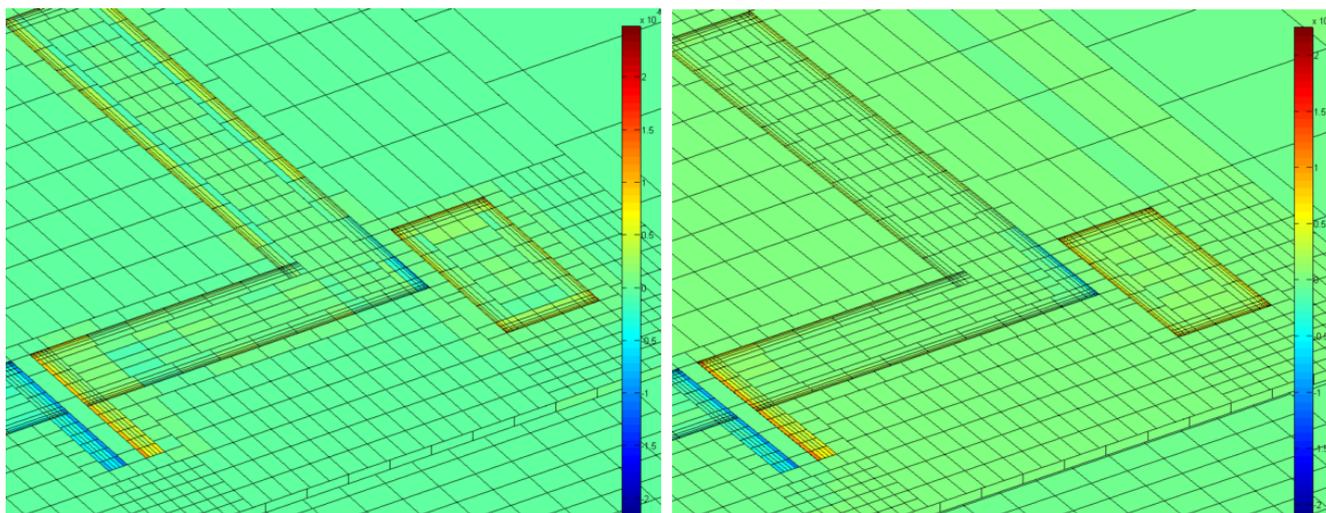


Figure III.29 – A gauche, charges surfaciques (x_0) pour un GMRES(m) en utilisant que les interactions proches (matrices pleines) ; à droite, charges surfacique solutions du GMRES(m) complet (interaction proche + interaction lointaine) - configuration d'un hacheur (Fig. V.4)

Taille m du sous-espace de Krylov :

La taille m du sous-espace de Krylov influence directement la rapidité du solveur. Plus m est grand, plus la taille de la base orthonormée est grande et plus la décomposition du résidu r_0 est riche en information (valeurs propres). Cependant, plus m est grand, plus le solveur utilise de mémoire pour résoudre. On observera cela plus loin, dans la figure III.32.

Influence du préconditionnement :

Le préconditionnement est le paramètre qui influence le plus la rapidité de convergence du solveur. Par exemple pour l'exemple des deux plaques parallèles maillées en 100 000 éléments la convergence

du GMRES(m) pour $m = 30$ est beaucoup plus rapide avec le préconditionnement par bloc que sans préconditionnement (courbe notée « prec=bloc-diag » dans la figure III.30) que sans préconditionnement (« prec = 0 »). Les matrices diagonales ont une taille moyenne de 25×25 (niveau $L = 6$).

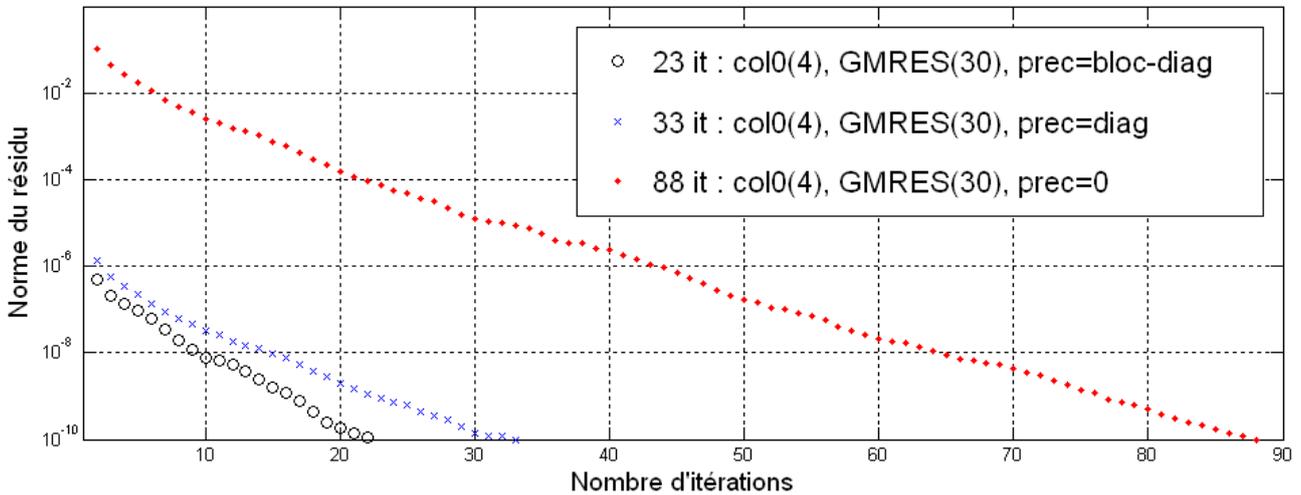


Figure III.30 – Influence du préconditionnement, « prec » (« diag » pour diagonale et « bloc-diag » pour les matrices de la diagonale de la « matrice d’interaction ») du GMRES(m) sur la norme du résidu en fonction du nombre d’itérations ou « it » (nombre de calculs de vecteur de \mathcal{V}_m) - configuration : 2 plaques parallèles maillées en 100 352 éléments, $L = 6$

Dans la figure III.30 (et de même pour les suivantes), on a tracé la norme du résidu correspondant à chaque évaluation de l’erreur de la ligne 8 du GMRES(m) (algo. 4). Le nombre d’itération noté « it », correspond au nombre de fois que l’on décompose le résidu dans la base de Krylov pour le GMRES(m). L’écart entre le préconditionnement avec les diagonales des matrices (« prec=dial ») et le préconditionnement par bloc est peu important car le problème est symétrique et uniforme. Cet écart dépend également du nombre de charges par cubes. Aussi, le temps de calcul des matrices de préconditionnement (inversion des matrices diagonales) ainsi que le temps de préconditionnement dans le GMRES(m) n’est pas beaucoup plus long que pour le préconditionnement par la diagonale. On s’est intéressé au conditionnement $\mathcal{C}([P/En])$ des matrices des configurations des deux plaques en fonction du nombre d’éléments de maillage III.31.

Plus le nombre d’élément est élevé, plus le conditionnement est élevé. Cette configuration est également très symétrique. Cela se vérifie dans le GMRES(m) au niveau de la matrice de Hessenberg $H_{m+1,m}$ dont seules les trois diagonales principales sont non nulles.

Considérons une configuration plus complexe, celle d’un hacheur (modélisé dans la suite, Fig. V.4). Le maillage n’est pas uniforme et il y a un matériau diélectrique (plus de symétrie dans la matrice d’interaction). Ce problème est donc nettement moins bien conditionné que l’exemple des deux plaques parallèles ($\mathcal{C}[P/En] = 1,88.10^6$). Les paramètres utilisés pour l’intégration avec l’AMLFMM sont les suivants : $L = 6$, intégration avec collocation ou Galerkin à l’ordre 0 avec 4 points de Gauss (col0(4) ou gal0(4)).

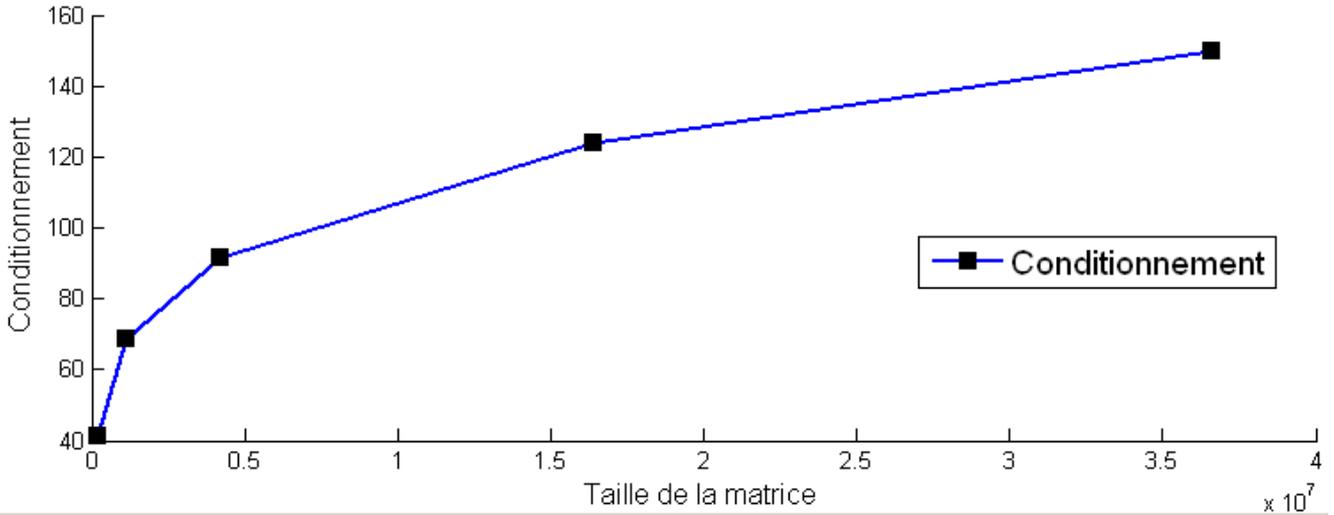


Figure III.31 – Conditionnement de la matrice en interaction totale, $\mathcal{C}([P/En])$, en fonction de la taille de la matrice (nombre d'éléments au carré) - configuration des deux plaques parallèles

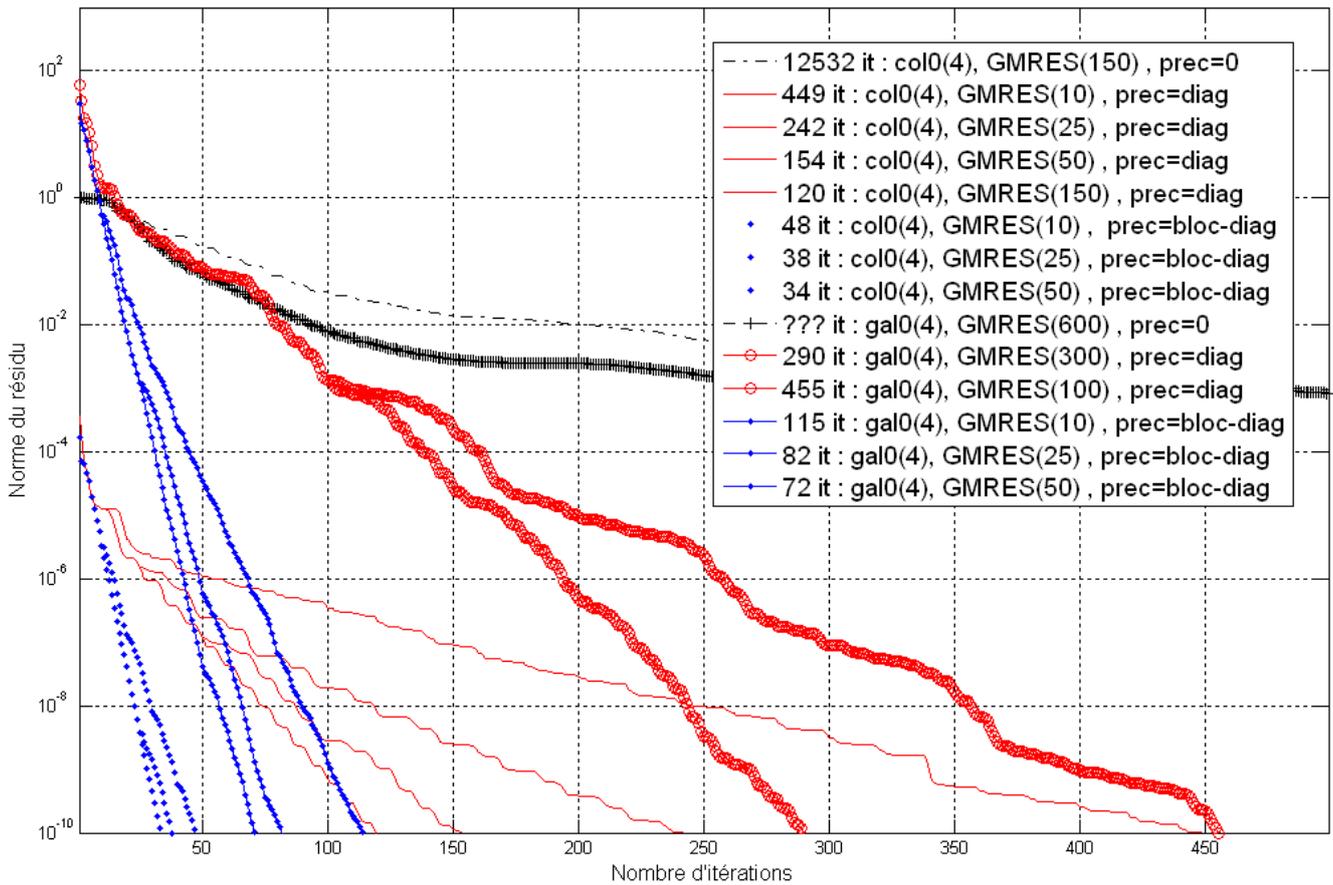


Figure III.32 – Influence du préconditionnement pour le GMRES(m) (pour différentes valeurs de m) sur la norme du résidu en fonction du nombre d'itérations ou « it » (nombre de calculs de vecteur de \mathcal{V}_m) - configuration d'un hacheur maillé en 7067 éléments avec un partitionnement adaptatif de $L = 6$ (Fig. V.4)

Chapitre III. Extraction des capacités par méthode multipolaire rapide à multi-niveaux adaptatifs

La nature de ce préconditionnement influence beaucoup plus la convergence du GMRES(m) pour une intégration avec une méthode de collocation qu'avec une méthode de Galerkin. Pour des petites valeurs de m avec la méthode de Galerkin avec préconditionnement diagonal, le GMRES(m) peut stagner comme les courbes « 12 352 it » et « ??? it » (dont on a pas attendu la convergence).

Pour certains gros problèmes (plusieurs dizaines de milliers d'éléments) très mal maillés (existence de triangles très pointus) en multi-couches, supposés par conséquent très mal conditionnés, on s'est aperçu que le GMRES(m) pouvait stagner et prendre énormément de temps (comme pour la configuration du variateur de vitesse maillé en 50 000 mailles environ présentée au chapitre suivant à la figure V.15). Dans ce cas, comme dans la figure III.32, le préconditionnement influence beaucoup, car en préconditionnant seulement avec la diagonale on obtient une stagnation du GMRES(m) alors qu'avec les matrices diagonales, on arrive à converger sans stagnation.

Lors de tous ces tests, on a remarqué que la qualité de l'intégration avait très peu d'influence sur les temps de convergence du solveur. On obtient généralement les mêmes courbes avec 4 ou 144 points de Gauss par éléments. Par contre, si on utilise un préconditionnement par bloc, plus le niveau de partitionnement est bas (et par conséquent plus la mémoire utilisée par le préconditionnement est grande) plus la convergence est rapide. Pour ce variateur de vitesse, pour le niveau de partitionnement vaut $L = 4$, le GMRES(50) converge en 100 itérations et pour $L = 6$, en 600 itérations.

Tolérance ε sur la norme du résidu :

Il reste un dernier paramètre non traité : la tolérance du GMRES(m) ε . La tolérance sur la norme du résidu ε ne garantit pas que les charges sont calculées avec une précision de ε . La courbe « 154 it » de la figure III.32 donne une erreur relative de $5,8 \cdot 10^{-3}\%$ contre $5,1 \cdot 10^{-10}\%$ pour la courbe « 120 it ». Les courbes utilisant la méthode de Galerkin ont en général une erreur relative moyenne inférieure à celle de la méthode de collocation. Cette erreur est souvent légèrement supérieure (en échelle logarithmique) à la tolérance ε fixé. Pour la figure III.32, $\varepsilon = 10^{-10}$ et l'erreur relative de ces courbes vaut entre $10^{-9}\%$ et $10^{-10}\%$.

Pour une configuration très symétrique comme celle des 2 plaques parallèles ou celle de la figure III.33, ce paramètre influence peu la valeur des capacités.

Pour des tolérances de $\varepsilon = 10^{-7}$ ou 10^{-16} , on obtient des matrices dont l'écart moyen relatif entre les capacités est de 1,45%, et 0,0015% entre 10^{-10} et 10^{-16} . Le pourcentage de non-symétrie de la matrice des capacités vaut 1% pour $\varepsilon = 10^{-7}$, 0,0021% pour $\varepsilon = 10^{-10}$ et 0,0016% pour $\varepsilon = 10^{-16}$. On obtient les mêmes comportements pour des problèmes moins bien conditionnés comme le hacheur. Pour $\varepsilon = 10^{-7}$ et 10^{-16} , l'écart relatif entre les matrices est de $1,86 \cdot 10^{-3}\%$. En fait, ces matrices de capacités dépendent plus du type d'intégration (paramètre non influent du GMRES(m)) et du niveau de partitionnement comme on l'a vu dans le chapitre précédent 4. Par exemple, pour le hacheur entre $L = 3$ ou 5, l'écart relatif entre les matrices de capacités est de 0,7%.

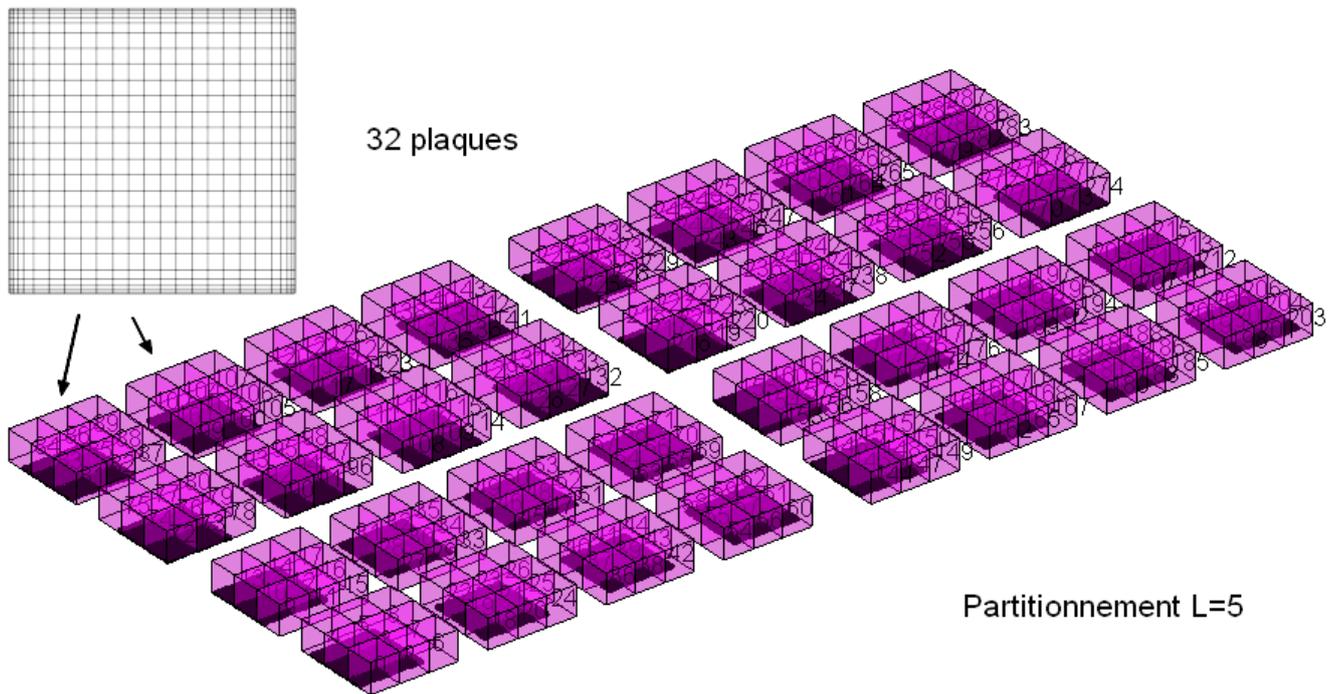


Figure III.33 – Partitionnement au niveau 5 de 32 plaques maillées en 20 000 éléments

6.2.3 Erreurs numériques et/ou mauvaises convergences

Il arrive parfois que dans la matrice des capacités, certaines capacités mutuelles C_{ij} soit positives hors de la diagonales. On suppose que cela se produit lorsqu'il y a beaucoup de régions capacités et que certaines sont très petites par rapport à d'autres, ou que les maillages ne sont pas très bien réalisés, ou encore qu'il s'agit d'un problème de résolution (du au mauvais conditionnement). Il se peut également que le solveur itératif converge vers une mauvaise solution. Cela peut se vérifier en utilisant un mauvais préconditionnement.

7 Calcul de la matrice de capacités avec la FMM du capteur à courant de Foucault

Le capteur de position à courant de Foucault a été maillé grâce au mailleur automatique de InCA3D (détaillé dans le chapitre précédent à la section 4.4.2, p. 65) :

Le maillage obtenu comporte 135 072 éléments. On a supposé que ces spires sont dans un milieu diélectrique homogène de permittivité $\epsilon_r = 4,8$ (normalement constitué de colle et de FR04 de même permittivité). La taille de la matrice des capacités calculée dépend du nombre des régions de conducteurs considérés. Pour tester différentes modélisations capacitives, de la plus simpliste à la plus fine, on a choisi de modéliser une capacité entre plusieurs couches de spires ou plusieurs par couche (on ne donnera pas le nombre car le nombre de couches du capteur est confidentiel).

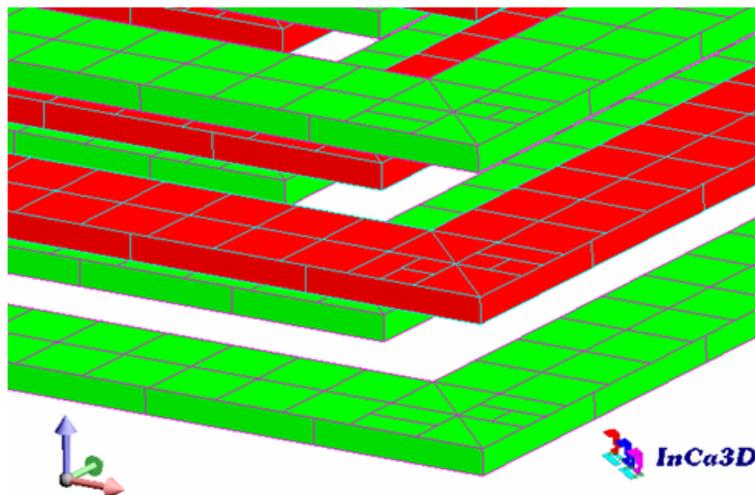


Figure III.34 – Maillage capacitif du capteur en 135 072 éléments surfaciques

Les matrices de capacités sont extraites via la FMM et le GMRES(m). Le calcul seul de la matrice des capacités correspondant à la modélisation la plus fine permet d'obtenir les matrices des autres configurations plus simples. En fait il suffit de « compacter » la matrice en fonction des régions capacitives considérées (avec la méthode précédemment présentée à la section 4.3.5, p. 63).

De plus, on peut se poser la question où est comment comment placer ces capacités dans le schéma PEEC. Aussi pour la modélisation capacitive la plus discrétisée cela aboutit à une matrice de capacité de « grande taille ». Par conséquent la connexion de toutes ces capacités $(n \times (n+1)/2)$ avec le macro-bloc inductif RLM dans un solveur circuit prendrait plusieurs heures. C'est ce que nous verrons dans le prochain chapitre.

8 Conclusion du chapitre

On a présenté dans ce chapitre, une autre méthode numérique générale d'extraction des capacités parasites : l'AMLFMM que l'on a adaptée à notre problème en prenant en compte la taille des éléments dans le partitionnement.

Elle permet, sans perdre beaucoup de précision, d'obtenir un gain en temps d'intégration et de résolution énorme grâce au solveur itératif GMRES(m) avec une consommation de mémoire extrêmement faible.

Ainsi, il est possible d'extraire les capacités de gros problèmes de complexité industrielle avec des maillages pouvant comporter des dizaines, voir des centaines de milliers d'éléments.

On va s'intéresser dans le chapitre suivant à la construction du schéma équivalent PEEC composé d'une part des éléments parasites inductifs R-L-M et d'autre part des capacités parasites C.