

Stratégie d'hybridation de méthodes de simulation électromagnétique FDTD/MTL - application à l'étude de grands systèmes complexes -

Nathanaël Muot

▶ **To cite this version:**

Nathanaël Muot. Stratégie d'hybridation de méthodes de simulation électromagnétique FDTD/MTL - application à l'étude de grands systèmes complexes -. Physique mathématique [math-ph]. ISAE-ENSMA Ecole Nationale Supérieure de Mécanique et d'Aérotechique - Poitiers, 2013. Français. NNT : . tel-00841708

HAL Id: tel-00841708

https://theses.hal.science/tel-00841708

Submitted on 22 Jul 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut Supérieur de l'Aéronautique et de l'Espace (ISAE)

Présentée et soutenue par :
Nathanaël MUOT

le jeudi 20 juin 2013

Titre :

Stratégies d'hybridation de méthodes de simulation électromagnétique FDTD/MTL
- Application à l'étude de grands systèmes complexes -

École doctorale et discipline ou spécialité :
ED AA : Physique, Mathématiques appliquées

Unité de recherche :
ISAE - ONERA MOIS

Directeur(s) de Thèse :

M. FERRIERES Xavier (directeur de thèse)
Mme. BACHELIER Elodie (co-directrice de thèse)

Jury :

M. REINEIX Alain, Directeur de Recherche CNRS (XLim Limoges) - Rapporteur
M. BONNET Pierre, Maître de Conférence HDR (IFMA Clermond-Ferrand) - Rapporteur
M. PASCAL Olivier, Professeur Université Paul Sabatier (LAPLACE Toulouse) - Examinateur
M. PICHON Lionel, Directeur de Recherche CNRS (LGEP/SUPELEC Gif-Sur-Yvette)-Examinateur
M. VOLLAIRE Christian, Professeur des Universités (ECL Lyon)- Examinateur
M. FERRIERES Xavier, Maître de Recherche HDR (ONERA/DEMR Toulouse)- Directeur de thèse
Mme. BACHELIER Elodie, Ingénieur de Recherche (ONERA/DEMR) - Co-directrice de thèse
M. GIRARD Christophe, Ingénieur de Recherche (AxesSim Strasbourg) - Encadrant AxesSim

A ma femme
Mon fils
Ma famille
Et mes amis

« Some people dream of success . . .
while others wake up and work hard at it »
Winston Churchill

Remerciements

Tout d'abord, je remercie Monsieur Xavier Ferrieres, Maître de Recherche de l'ONERA, d'avoir accepté d'être mon directeur de thèse. Je remercie également Élodie Bachelier, ingénieur de recherche ONERA, d'avoir codirigé cette thèse. Je les remercie tous les deux pour leurs conseils avisés et leur aide apportés durant ces travaux de thèse.

Je remercie également Christophe Girard, Directeur Général d'AxesSim, de m'avoir donné l'opportunité de réaliser ces travaux de recherche et également de me permettre aujourd'hui de prolonger cette collaboration.

Je tiens également à remercier tous les membres du Jury qui ont accepté de juger ce travail de recherche. Je remercie tout particulièrement Messieurs Alain Reineix et Pierre Bonnet d'en avoir été les rapporteurs.

Mes remerciements vont également à toute l'équipe du DEMR de l'ONERA ainsi que celle d'AxesSim, qui m'ont reçu et accompagné durant ces trois dernières années. Elles m'ont apporté, sans retenue, tant sur le plan humain que technique et scientifique.

Je ne remercierai jamais assez ma femme Céline et mon fils Esteban de leur soutien et de leur aide durant ces trois dernières années. Ils ont plus que participé indirectement à la réalisation de ce travail.

J'en profite pour remercier tous ceux qui ont attiré mon attention sur mes limites : cela m'a donné avec le soutien de ma famille, la volonté de surmonter tant de difficultés pour réaliser ce grand projet.

Sans pouvoir tous les citer, je remercie tous ceux qui ont relu, lu et qui liront ce mémoire, espérant qu'ils trouveront un intérêt et peut être une réponse voire une idée :

Dr. Xavier Ferrieres, Dr. Élodie Bachelier, Dr. Christophe Girard, Dr. Alain Reineix, Dr. Pierre Bonnet, Dr. Olivier Pascal, Dr. Lionel Pichon, Dr. Christian Vollaire, Dr. Christian Carel, Dr. Cyril Giraudon, Dr. Didier Roisse, M. Xavier Romeuf, M. Francois Koechlin, Mlle Odile Durst, Mme. Céline Muot, tous ceux que j'ai oublié et vous qui que vous soyez ...

Table des matières

Introduction **13**

I Stratégies d'hybridation de méthodes FDTD/MTL **16**

1 Position du problème **17**
1.1 Introduction . . . 18
1.2 La problématique visée . . . 20
1.2.1 Étude des grands systèmes complexes . . . 20
1.2.2 Un cas d'étude concret . . . 22
1.3 Méthodes de résolution actuelles . . . 23
1.3.1 Parallélisation des méthodes 3D . . . 24
1.3.2 Approche d'hybridation multi-domaine . . . 24
1.3.3 Le formalisme de fils mince et multi-conducteur . . . 27
1.3.4 Les limites des approches existantes . . . 28
1.4 La stratégie proposée . . . 29
1.4.1 Hybridation de domaines avec prise en compte des lignes de transmission 29
1.4.2 Hybridation d'échelle, extension du fil mince . . . 29
1.4.3 Prise en compte d'un sol de conductivité finie (dépendance en fréquence) 29
1.4.4 Mise en place d'une stratégie informatique efficace pour l'hybridation forte . . . 30

2 Modélisation des lignes de transmission dans le domaine temporel avec prise en compte d'un sol non parfaitement conducteur **31**
2.1 Introduction . . . 32
2.2 Généralités sur les lignes de transmission : modèles et paramètres . . . 33
2.2.1 L'équation de ligne de transmission et les paramètres liés à celle-ci . . . 33
2.2.2 Étude du cas d'une ligne de transmission au-dessus d'un sol . . . 39
2.2.3 Modèle de couplage - Source distribuée . . . 46
2.2.4 Multi conducteurs - problématique du faisceau de câbles et des domaines internes . . . 50
2.3 Résolution numérique des équations de ligne dans le domaine temporel . . . 57

2.3.1 Approximation numérique - méthode MTL . . . 57
2.3.2 Conditions aux limites et sources localisées . . . 62
2.3.3 Résultat numérique - le problème de Franck Arreghini . . . 68
2.4 Prise en compte de la dépendance en fréquence des paramètres de ligne - Cas d'une ligne au-dessus d'un sol . . . 71
2.4.1 Approximation par une série de Debye . . . 72
2.4.2 Validation numérique . . . 81
2.4.3 Modèle de Debye dans le formalisme des lignes de transmission - Mise en œuvre et validation . . . 88
2.4.4 Validation et étude de la factorisation en série de Debye . . . 90
2.5 Conclusion . . . 98

3 Hybridation champ/câbles - Conducteurs de géométrie arbitraire 99
3.1 Introduction . . . 100
3.2 Le formalisme de fil mince en FDTD . . . 101
3.2.1 Le fil mince de R. Holland . . . 101
3.2.2 Le modèle de fils minces de Berenger . . . 103
3.3 Nouvelle approche : le formalisme pour des sections à géométrie arbitraire . . . 105
3.3.1 Détermination des équations de ligne de transmission dans le cas d'une géométrie de conducteurs quelconques . . . 105
3.3.2 L'équation de Laplace pour calculer les paramètres de ligne de transmission . . . 107
3.4 Calcul des paramètres de la ligne de transmission . . . 109
3.4.1 Paramètres dans la cellule FDTD (In-Cell) . . . 109
3.4.2 Paramètres en espace ouvert . . . 109
3.4.3 Paramètres avec une référence dans la cellule FDTD . . . 110
3.5 Couplage avec le domaine 3D - Approximation du mode commun . . . 111
3.6 Validation numérique . . . 114
3.6.1 Le fil simple . . . 114
3.6.2 La bifilaire . . . 115
3.6.3 Multi-filaire . . . 116
3.6.4 Ligne de transmission . . . 117
3.6.5 Nœud de courant . . . 119
3.6.6 Blindage et reprise de coupure . . . 121
3.7 Conclusion . . . 124

4 Hybridation forte entre lignes de transmission et modèle 3D 125
4.1 Introduction . . . 126
4.2 Stratégie d'échange de données dans une méthode FDTD - Parallélisation de code MTL . . . 127
4.2.1 Principe général de la parallélisation . . . 127
4.2.2 Hybridation coopérante . . . 128
4.2.3 Hybridation collaborative . . . 129
4.2.4 Validation numérique . . . 130
4.3 Hybridation de domaines . . . 134

4.3.1 Extrapolation de l'hybridation 1D/1D . . . 134
4.3.2 Continuité du champ électromagnétique transversal . . . 136
4.3.3 Calcul du champ dans le plan d'hybridation . . . 137
4.4 Validation numérique et étude paramétrique . . . 141
4.4.1 Influence de la longueur d'intégration . . . 141
4.4.2 Limites fréquentielles de l'approche . . . 142
4.4.3 Hybridation et sol de conductivité finie . . . 145
4.4.4 Prise en compte d'une source rayonnante . . . 147
4.5 Conclusion . . . 152

5 Mise en œuvre de la stratégie d'hybridation sur une maquette réaliste d'un site de lancement 153
5.1 Introduction . . . 154
5.2 Présentation du cas étudié – Positionnement par rapport au problème initial . 155
5.2.1 Positionnement par rapport au problème initial . . . 155
5.2.2 La maquette numérique . . . 157
5.3 Études numériques . . . 160
5.3.1 Source haute fréquence de type onde plane avec un sol parfaitement conducteur . . . 160
5.3.2 Source basse fréquence de type onde plane avec un sol parfaitment conducteur . . . 166
5.3.3 Source basse fréquence avec une injection à la base d'un mât avec un sol réaliste . . . 167
5.3.4 Comportement d'un réseau multifilaire avec une injection de courant à la base d'un mât . . . 170
5.4 Conclusion . . . 173

II Re-visite des solutions informatiques pour l'hybridation et le calcul parallèle 174

6 L'informatique, outil pour la science 175
6.1 Introduction . . . 176
6.2 Bonne pratique de la programmation . . . 177
6.2.1 Lisibilité du code - le codage . . . 177
6.2.2 Pérenniser son travail - Les tests unitaires . . . 177
6.2.3 Architecture robuste - Programmation orientée objet (POO) . . . 178
6.2.4 Conclusion . . . 179
6.3 Les nouvelles orientations de l'informatique . . . 181
6.3.1 Vitesse de calcul ou vitesse de mise en œuvre . . . 181
6.3.2 Mélanger les styles pour allier performance et confort . . . 183
6.3.3 Mise en œuvre d'une programmation parallèle . . . 184
6.4 Conclusion . . . 191

7 Du calcul parallèle vers le calcul hybride **192**
7.1 Introduction . . . 193
7.2 Du besoin à la spécification . . . 194
7.2.1 L'identification d'un besoin . . . 194
7.2.2 Les solutions technologiques pour le calcul parallèle . . . 194
7.2.3 Présentation de la méthode . . . 196
7.3 Mise en œuvre d'un prototype . . . 199
7.3.1 L'interface de communication . . . 199
7.3.2 Mise en œuvre d'un worker . . . 201
7.3.3 Définition de l'interface du worker . . . 206
7.3.4 Mise en œuvre d'une stratégie d'hybridation . . . 208
7.4 Exemples d'utilisation du prototype . . . 211
7.4.1 Mise en œuvre du manager . . . 211
7.4.2 Exemples de drivers . . . 222
7.5 Conclusion . . . 230

Conclusion générale et perspectives **231**

A Équations de Maxwell **234**
A.1 Formulation différentielle des équations de Maxwell . . . 234
A.2 Milieux diélectriques et magnétiques . . . 234
A.3 Milieux conducteurs . . . 235

B Méthode FDTD **236**
B.1 Principe de la méthode FDTD . . . 236
B.2 Les fils minces . . . 238

C Méthode FVTD **240**
C.1 Principe de la méthode FVTD . . . 240
C.2 Les fils minces . . . 241

D Principe du champ total / champ diffracté appliqué aux lignes de transmission **243**

E Modélisation de structures filaires par la méthode des équations intégrales en temporel **246**

F Représentation BLT des équations de lignes de transmission **248**

G Comparaisons entre différents langages de programmation **250**
G.1 La liste chaînée . . . 250
G.2 Les tableaux . . . 255

H La programmation orientée objet, application au calcul hybride **259**

Bibliographie **263**

Notations **270**

Avant propos

Ce travail de thèse a largement été supporté par AxesSim et subventionné par l'ANRT (Association Nationale Recherche Technologie) par le biais d'une bourse CIFRE (Conventions Industrielles de Formation par la REcherche). Cela apporte à cette thèse un caractère industriel indéniable.

Introduction

A l'heure actuelle, les infrastructures terrestres (bâtiments [1, 2], immeubles, réseaux de distribution d'énergie [3, 4, 5], réseaux de transports ferrovières [6, 7], ...) doivent être protégées contre la foudre et ses effets. Pour cela, il existe un certain nombre de normes et d'usages qui s'avèrent suffisants dans la majorité des cas. Pour des structures sensibles (sites classés SEVESO par exemple) ou très spécifiques (très hautes tours, bâtiments militaires, ...), un dossier technique très détaillé doit être fourni aux organismes de certification. Ce dossier doit justifier que les risques liés à un foudroiement sont maîtrisés et que des dispositifs techniques sont mis en œuvre pour en réduire au maximum les effets.

Lors de la construction de nouvelles infrastructures terrestres ou lors de l'expertise d'infrastructures existantes, en cas d'évolution comme par exemple l'ajout d'un bâtiment ou un agrandissement, une étude technique doit être menée pour évaluer la vulnérabilité du nouveau système dans sa globalité. En complément des expertises « classiques » basées sur des données papiers (plans, caractéristiques techniques des dispositifs, statistiques, ...), depuis quelques années, des études et des simulations numériques permettent de fournir des éléments tout à fait intéressants pour évaluer, par exemple plusieurs solutions techniques en phase d'avant-projet, ou pour quantifier des niveaux induits sur des sous-systèmes en cas de foudroiement et ainsi dimensionner les dispositifs à mettre en place.

Bien que l'étude menée dans cette thèse soit générale et vise à présenter des outils et une méthodologie qui puissent s'appliquer à n'importe quel type d'infrastructure terrestre ou à d'autres configurations, nous avons choisi d'illustrer ce travail par une problématique assez spécifique qui est celle des sites de lancement spatial. Ce choix a été motivé par deux études menées récemment par l'ONERA/DEMR pour le compte du CNES et de Vitrociset pour l'ESA, sur les sites récents de SOYOUZ et VEGA, situés sur le Centre Spatial Guyanais de Kourou [1, 2]. Ce travail de thèse s'insert dans une démarche globale d'amélioration des outils de simulation numérique pour l'étude de ces grandes infrastructures complexes et sensibles. Le mémoire est divisé en deux grandes parties.

La première partie traite des problématiques physique et mathématique liées à l'étude de grandes scènes de calcul où il existe d'une part plusieurs bâtiments interconnectés par un réseau de câbles et d'autre part, des faisceaux de câbles internes aux bâtiments. Le principe

de la stratégie proposée pour traiter ce type de configuration est d'utiliser conjointement ou, plus précisément, d'hybrider une méthode de ligne de transmission avec une méthode 3D de résolution des équations de Maxwell.

Dans cette partie, nous présenterons tout d'abord, dans un premier chapitre, la problématique générale de la thèse, en illustrant les difficultés associées à l'étude des grands systèmes actuels et en expliquant également en quoi l'étude de ces systèmes est encore aujourd'hui difficile. A partir des difficultés mises en avant dans les méthodes actuelles, nous préciserons les objectifs de la thèse et nous présenterons la stratégie que l'on propose pour y répondre.

Le deuxième chapitre de cette partie est consacré à la modélisation des lignes de transmission dans le domaine temporel. Nous détaillerons le modèle général des équations de ligne de transmission et nous préciserons également quelques cas canoniques, où il existe des formules et des approximations analytiques pour évaluer les paramètres de ligne. Nous aborderons ensuite, la modélisation numérique des lignes de transmission (MTL Multi-conductor Transmission Lines) par une approche différences finies dans le domaine temporel (FDTD Finite-Difference Time-Domain method) utilisant un schéma « leap-frog » en espace et en temps. Un critère de stabilité sera donné et une étude de l'erreur de dispersion sera menée pour le schéma 1D obtenu. Enfin, nous introduirons dans le modèle numérique développé, la dépendance en fréquence des paramètres de la ligne pour tenir compte de lignes au dessus du sol. Pour cela, nous préciserons deux approches basées respectivement sur l'utilisation des résidus et sur une optimisation globale. Nous comparerons ces deux approches et montrerons leurs avantages et limitations.

Dans le troisième chapitre, nous présenterons enfin la stratégie d'hybridation 1D/3D entre une méthode différences finies 3D et une méthode différences finies 1D pour résoudre les équations de ligne de transmission. Cette première hybridation sera effectuée dans le but de pouvoir prendre en compte de manière précise les faisceaux de câbles se trouvant à l'intérieur d'un domaine 3D. On parlera ici d'une hybridation d'échelle où la géométrie fine du faisceau sera traitée par une équation de ligne interagissant avec le domaine 3D via un couplage par les champs évalués dans le domaine 3D et les courants évalués sur la ligne. On montrera notamment que ce couplage peut être réalisé par l'utilisation d'un mode commun, permettant ainsi de pouvoir prendre en compte facilement des faisceaux obliques dans un domaine 3D. Cette approche sera illustrée et validée sur plusieurs exemples numériques plus ou moins complexes montrant les avantages de la méthode.

Dans le quatrième chapitre de cette première partie, nous aborderons la problématique de l'hybridation entre le domaine 3D et un modèle de ligne de transmission dans un cadre multi-domaine ; c'est-à-dire dans le cas où nous avons un sous-domaine 1D non inclus dans les sous-domaines 3D. Nous présenterons le principe des échanges liés à l'hybridation ou à la parallélisation dans le cas des équations de ligne de transmission, avant d'appliquer un principe similaire dans les cas 1D/3D. Pour cela nous considérerons que la ligne de transmission doit être résolue en partie à la fois dans le domaine 3D et dans le domaine 1D. Nous montrerons en particulier que cette hybridation nécessite un recouvrement des courants calculés sur la ligne en 1D, dans le domaine 3D pour tenir des champs rayonnés dont les composantes tangentielles au plan d'interface des deux méthodes, doivent être continues. Enfin, nous donnerons un ensemble de configurations pour montrer la validité de la méthode,

ainsi que ses limitations.

Enfin, un dernier chapitre présentera l'application de la stratégie d'hybridation 1D/3D que nous avons étudiée, pour traiter un problème de foudroiement sur le site de lancement de Soyouz. Nous donnerons les avantages à utiliser cette approche plutôt qu'une approche globale et nous préciserons les difficultés rencontrées.

La deuxième partie est consacrée à la problématique informatique liée à l'hybridation entre deux méthodes temporelles qui doivent avoir des interactions « fortes » entre-elles. On parle alors d'hybridation forte entre les méthodes.

Dans cette deuxième partie, nous présenterons d'abord les « bonnes pratiques » de l'informatique qui permettent de faire de l'informatique un outil pérenne pour les sciences et dans un deuxième temps, nous donnerons un état de l'art des nouvelles avancées en informatique en rapport avec notre problème. Nous nous intéresserons principalement à la question du compromis entre vitesse de réalisation et vitesse d'exécution.

Enfin, dans le dernier chapitre, nous présenterons la solution proposée pour traiter l'hybridation forte entre différentes méthodes numériques. Ce chapitre se découpe en deux points : la présentation du besoin et la mise en place d'un prototype.

Première partie

Stratégies d'hybridation de méthodes FDTD/MTL

Chapitre 1

Position du problème

Sommaire

1.1 Introduction 18
1.2 La problématique visée 20
1.2.1 Étude des grands systèmes complexes 20
1.2.2 Un cas d'étude concret 22
1.3 Méthodes de résolution actuelles 23
1.3.1 Parallélisation des méthodes 3D 24
1.3.2 Approche d'hybridation multi-domaine 24
1.3.3 Le formalisme de fils mince et multi-conducteur 27
1.3.4 Les limites des approches existantes 28
1.4 La stratégie proposée 29
1.4.1 Hybridation de domaines avec prise en compte des lignes de transmission 29
1.4.2 Hybridation d'échelle, extension du fil mince 29
1.4.3 Prise en compte d'un sol de conductivité finie (dépendance en fréquence) 29
1.4.4 Mise en place d'une stratégie informatique efficace pour l'hybridation forte 30

1.1 Introduction

La modélisation électromagnétique des grands systèmes devient aujourd'hui un enjeu majeur dans l'évaluation de leur vulnérabilité aux environnements électromagnétiques. L'estimation de cette vulnérabilité nécessite, dans un premier temps, de pouvoir évaluer les contraintes rayonnées et/ou conduites appliquées aux systèmes. Dans un second temps, ces contraintes sont comparées aux niveaux de susceptibilité des systèmes pour évaluer leur vulnérabilité au contexte opérationnel étudié.

La phase d'évaluation des contraintes, dans le contexte d'un grand système, est une opération complexe à plusieurs titres à cause de :

- la variété des géométries présentes dans la scène, pouvant inclure de grands objets courbes, de très petits éléments comme les structures filaires ainsi que des objets de taille intermédiaire qui peuvent être à la fois denses et complexes comme des antennes ou des équipements ;
- la présence de matériaux complexes pouvant être volumiques ou surfaciques. Certains de ces matériaux possèdent parfois des propriétés d'inhomogénéité ou de rugosité de surface comme dans le cas des sols. D'autres, composites voire anisotropes, doivent traduire la pénétration et la réflexion des ondes sur les structures ;
- la taille des problèmes par rapport à la longueur d'onde qui a des conséquences sur le temps de calcul et la volumétrie du problème mais également sur la qualité du résultat final.

Plus précisément, les sites terrestres qui nous intéressent, ont une taille pouvant s'étendre sur plusieurs kilomètres, possèdent des interconnexions multiples entre différents bâtiments et reposent sur un milieu diélectrique de faible conductivité représentatif du sol. Ainsi, les études menées actuellement sur ce type de site se concentrent généralement sur un bâtiment unique. Cependant, l'étude d'un site doit tenir compte de celui-ci dans sa globalité pour ne pas négliger, en particulier, des courants induits qui pourraient se propager entre les bâtiments via les réseaux de terre ou les interconnexions.

Dans le cadre de cette thèse, nous apportons une contribution afin de mieux prendre en compte les spécificités de ce type de grands systèmes terrestres.

Ce premier chapitre illustre les difficultés associées à l'étude des grands systèmes en donnant un ensemble de configurations réelles qui entrent dans les objectifs finaux de cette thèse. Nous n'allons pas bien évidemment traiter l'ensemble de ces configurations, mais nous focaliser à celles où nous avons plusieurs bâtiments interconnectés par un réseau de câbles sur un sol de conductivité finie. Pour cela, nous nous intéresserons plus particulièrement au site de lancement du lanceur Soyouz qui nous servira de support à la validation numérique finale de nos travaux.

Avant de proposer une stratégie pour simuler efficacement ce type de grand système, nous ferons une présentation des principales méthodes numériques actuelles utilisées pour étudier ceux-ci. En particulier, nous présenterons deux méthodes d'hybridations qui permettent d'ores et déjà de traiter de grandes scènes. Nous analyserons également les modèles de fils minces actuels, proposés pour modéliser les structures filaires. Nous expliquerons alors, en quoi ces différentes méthodes ne permettent pas de répondre de manière satisfaisante à la problématique visée.

Nous terminerons ensuite ce chapitre en présentant la stratégie proposée pour étudier numériquement un grand système composé de plusieurs bâtiments interconnectés par un réseau de câbles sur un sol de conductivité finie. Cette stratégie est découpée en trois problèmes qui sont :

- La modélisation de grands systèmes où certaines zones peuvent être modélisées par la théorie des lignes de transmission,
- La prise en compte de faisceaux de câbles,
- La prise en compte d'un sol de conductivité finie.

1.2 La problématique visée

1.2.1 Étude des grands systèmes complexes

Aujourd'hui, la demande d'expertiser de grands systèmes ou des sites devient de plus en plus pressante. La place de la simulation électromagnétique est devenue importante dans ces expertises. Il faut donc disposer d'outils et de méthodes de simulations efficaces pour décrire toute la complexité des scènes étudiées. Parmi les problèmes qui nous intéressent, on peut citer plus particulièrement :

– **Protection de grands sites terrestres à un rayonnement électromagnétique.** Dans ce type de configuration, nous cherchons à étudier les perturbations induites par une agression électromagnétique sur un réseau de câbles ou de fils électriques présents dans un ensemble de bâtiments. La difficulté réside dans la taille du problème, la présence d'un sol et la complexité du réseau de conducteurs (Fig. 1.1).

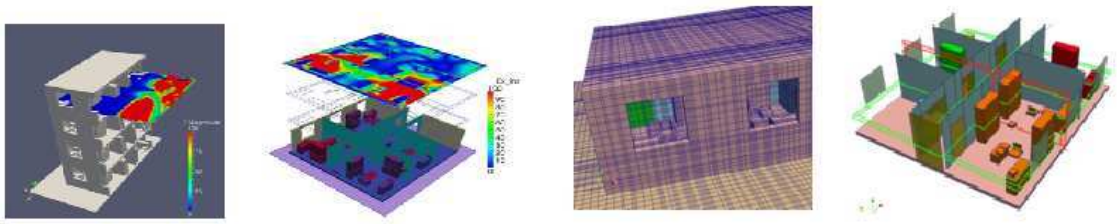


FIGURE 1.1 – Exemple de réseau de câbles et de champ diffracté dans un bâtiment (images CEA).

Dans le cas de la protection foudre des sites terrestres, aux problématiques précédentes s'ajoute la prise en compte de la nature basse fréquence de la source. Cette caractéristique entraîne des contraintes supplémentaires sur le temps d'observation (durée des signaux) et sur certains modèles numériques (conditions aux limites, ...). De plus, les interactions entre bâtiments ainsi que celles avec le sol sont plus importantes.



FIGURE 1.2 – Photo de l'ensemble de lancement Soyouz en Guyane, proposé par ENJOY SPACE. [8]

– **Problématique de CEM interne.**
Dans une étude de CEM, il est important de tenir compte précisément de la géométrie des câbles (Fig. 1.3). En particulier, il est nécessaire de pouvoir introduire les blindages et les éléments diélectriques dans cette géométrie. Généralement les méthodes actuelles ne permettent de prendre en compte que des réseaux de fils simples, mais il est souvent intéressant de connaître la contrainte sur l'ensemble des conducteurs d'un faisceau de câbles.

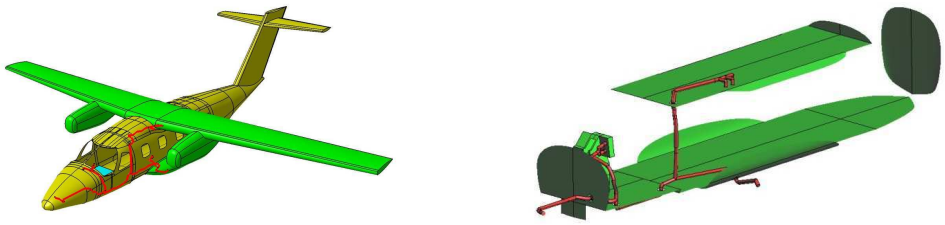


FIGURE 1.3 – Exemple de problématique interne sur un avion traité dans le cadre du projet HIRF-SE [9]

– **Rayonnement d'antenne sur des structures.**
Dans ce type de configuration, nous cherchons à étudier les perturbations induites sur un système par le rayonnement d'une antenne. Ce rayonnement ne peut pas être considéré comme une onde plane car l'antenne est située trop proche de la structure, mais suffisamment loin pour qu'il y ait une zone de vide importante entre la structure et celle-ci. Nous pouvons aussi envisager la présence d'un plan de masse ou d'un sol sur lequel repose la structure (Fig. 1.4). Dans le cas d'un sol réel, le problème devient très compliqué.

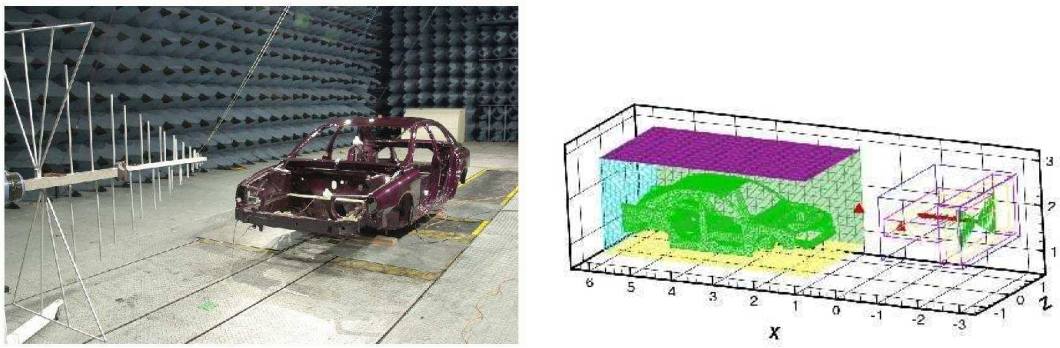
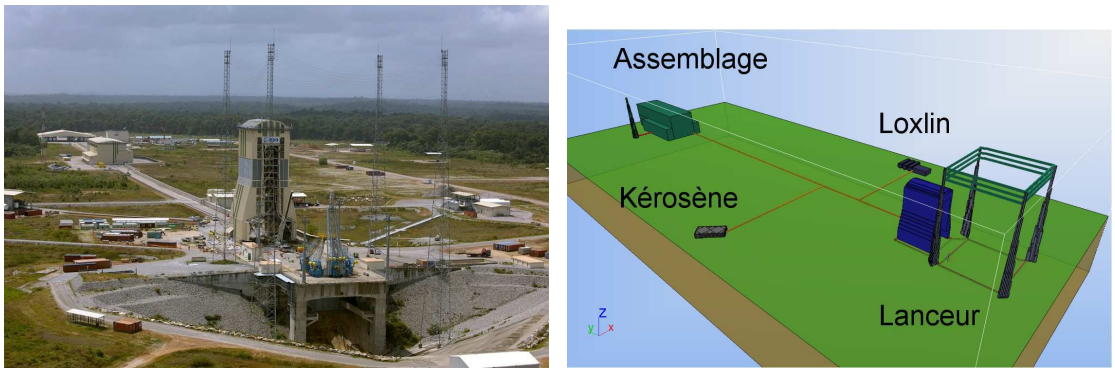


FIGURE 1.4 – Application de la stratégie d'hybridation par volumes disjoints au cas GEM-CAR.

Dans l'ensemble de ces configurations, nous nous sommes plus particulièrement intéressés à l'étude du site de lancement de Soyouz sur la base de Kourou en Guyane.

1.2.2 Un cas d'étude concret

Pour illustrer la problématique visée, nous considérons le problème suivant (Fig. 1.5) où nous avons plusieurs bâtiments interconnectés par des conducteurs, constitués par des rails, des canalisations de fluides ou des réseaux électriques de communications, placés au-dessus du sol. L'ensemble peut alors être illuminé par différents types d'agressions comme, par exemple, une onde plane électromagnétique ou bien un foudroiement.



(a) (b)

FIGURE 1.5 – Photo (a) de l'ensemble de lancement Soyouz en Guyane, proposé par ENJOY SPACE. [8] et vue d'ensemble de la maquette numérique (b)

Le choix du site de lancement Soyouz a été motivé par des études menées, récemment par l'ONERA/DEMR pour le compte du CNES et de Vitrociset pour l'ESA, sur celui-ci. En particulier, ces différentes études nous ont permis d'avoir un recul suffisant pour définir un modèle simplifié mais réaliste du site, pour les effets liés à la foudre. Ce modèle se compose de 5 bâtiments (Fig. 1.5(b)), définit comme suit :

- la zone de lancement,
- le bâtiment d'assemblage,
- le centre de contrôle lancement et de stockage,
- la zone de stockage de fluide « loxlin »
- et une zone technique « kérosène ».

Au cours de la première étude numérique, seule la zone lanceur avait été prise en compte dans une modélisation globale 3D. Dans cette thèse nous nous sommes focalisés plus précisément sur la modélisation des réseaux de câbles suivant deux objectifs :

- pouvoir modéliser l'influence d'un réseau d'interconnexion entre les bâtiments sur l'évacuation des courants,
- pouvoir modéliser toute la complexité des faisceaux de câbles de manière à dimensionner les systèmes de protection sur les équipements.

1.3 Méthodes de résolution actuelles

Il existe de nombreuses méthodes numériques pour traiter les équations de Maxwell. Les principales méthodes sont données dans la figure 1.6. A titre indicatif, nous pouvons classer ces méthodes en fonction du rapport de la longueur d'onde λ avec la longueur caractéristique de la structure étudiée L.

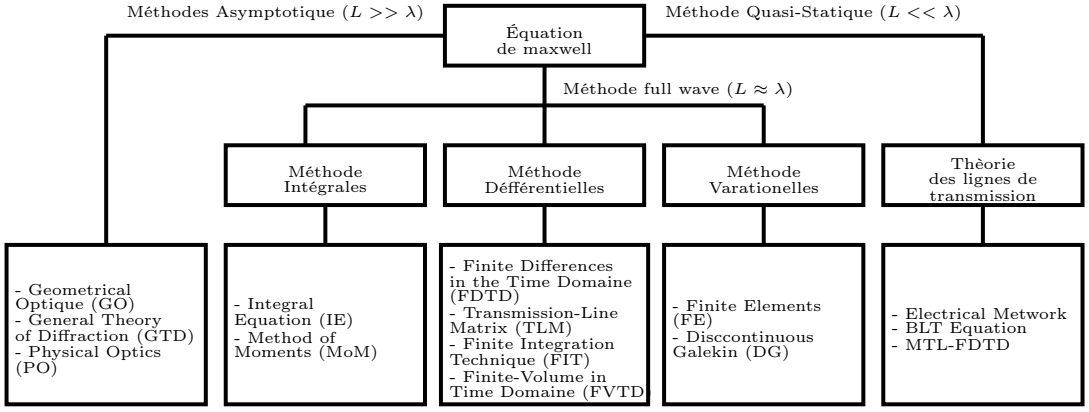


FIGURE 1.6 – Organigramme des méthodes numériques

Les méthodes asymptotiques sont des méthodes « hautes fréquences » qui introduisent la notion de rayon modélisant la propagation des champs par des trajectoires rectilignes (lancer de rayon).

Les méthodes quasi-statiques sont des méthodes basses fréquences, particulièrement bien adaptées à l'étude des effets de foudre. Par exemple des approches par circuit équivalent permettent de modéliser le comportement global d'un système en utilisant un modèle équivalent simplifié de type (R,L,C). Ces approches sont très intéressantes car les temps de calcul sont souvent extrêmement courts. La grande difficulté de ces approches est de définir les modèles équivalents pour tous les composants de notre système (mât, fond de fouille, bâtiment, etc). Dans la plupart des cas, la définition de ces modèles nécessite un effort considérable qui rend ces approches inappropriées à l'étude d'une infrastructure unique. En revanche, elles sont particulièrement bien adaptées pour étudier des systèmes composés de plusieurs structures identiques (éoliennes [10], réseau de distribution de l'énergie [11], ...), ou faire des études paramétriques.

Parmi les autres méthodes, plus adaptées au calcul large bande, nous nous intéressons plus particulièrement aux méthodes différentielles :

- L'approche MTL-FDTD (Chapitre 2 page 31) est un méthode différentielle de type différences finies dans le domaine temporel à une dimension appliquée aux équations de lignes de transmission multifilaire.
- La méthode FDTD-3D (Annexe B page 236), basée sur le schéma de Yee, souvent décriée pour sa non conformité géométrique, a l'avantage d'être efficace, robuste, bien maîtrisée et de disposer de nombreux modèles physiques.
- Les méthodes conformes (volumes finis (Annexe C page 240), Galerkin discontinu (GD)) ont le grand avantage de mieux maîtriser la géométrie mais sont plus complexes

à mettre en œuvre. La phase de génération des maillages reste un point délicat pour ce type de méthodes. La méthode GD donne en plus l'avantage d'être d'ordre élevé mais reste encore une méthode peu utilisée dans le contexte industriel.

Au vue de ces différents points, nous avons principalement basé notre étude sur la méthode des différences finies dans le domaine temporel. Cependant, nous avons pris soin, dans la mesure du possible, de faire des choix qui pourront être utilisés dans d'autres méthodes temporelles (FVTD, GD, ...).

Nous allons maintenant détailler un certain nombre de techniques et méthodes permettant d'étendre les méthodes numériques actuelles afin de pallier les difficultés liées à l'étude des grands systèmes.

1.3.1 Parallélisation des méthodes 3D

Nous plaçons notre discours dans le cadre de la méthode différences finies basée sur le schéma de Yee [12] qui est actuellement la méthode la plus utilisée pour résoudre les problèmes électromagnétiques instationnaires. Pour traiter des objets de grandes dimensions, une des premières améliorations des performances de calcul a consisté à porter la méthode sur machine parallèle à mémoire distribuée. Ceci correspond à la configuration actuelle de la plupart des machines parallèles ou clusters de PC. La stratégie utilisée [13] consiste en un découpage de la taille $n_x \times n_y \times n_z$ du volume de calcul en $m_y \times m_z$ pavés de dimension $n_x \times \frac{n_y}{m_y} \times \frac{n_z}{m_z}$. Pour des raisons de vectorisation, nous ne découpons pas suivant la direction n_x. Nous définissons ensuite une grille cartésienne de processus à laquelle nous associons le découpage du domaine de calcul, et où chaque processus possède au plus 4 voisins. A chaque itération en temps, chaque processus P_c envoie et reçoit les champs sur la frontière de son domaine de calcul. En ce qui concerne les structures filaires, une procédure analogue est mise en place pour transférer l'information en limite du domaine de calcul propre à chaque processus pour les courants et les charges attribuées à chaque segment.

L'utilisation de machines parallèles permet d'augmenter la taille du problème physique à traiter. Elle permet également d'avoir un gain de temps non négligeable comme montré sur la figure 1.7. Néanmoins, cette solution nécessite des moyens informatiques importants (supercalculateur). De plus, d'un point de vue numérique l'utilisation d'un nombre important de mailles dans le domaine de calcul, lié généralement à un temps d'observation long peut induire des erreurs de dispersion non négligeables à cause de la nature du schéma de Yee. Enfin, cette méthode n'est pas optimale pour traiter certaines classes de problème où une grande zone de l'espace pourrait être traitée analytiquement (propagation en espace libre).

1.3.2 Approche d'hybridation multi-domaine

La résolution efficace d'un problème électromagnétique de grandes dimensions ne se limite pas uniquement à la recherche d'une solution parallèle pour un schéma numérique donné. Il existe aussi des stratégies qui consistent à découper le domaine de calcul en plusieurs sous-domaines traités individuellement par la méthode numérique la plus adaptée. Nous présentons ici deux techniques d'hybridation qui entrent dans cette démarche :

– une première approche (domaines disjoints) très générale où nous décomposons le domaine de calcul en plusieurs sous-domaines en se préoccupant plus ou moins de la

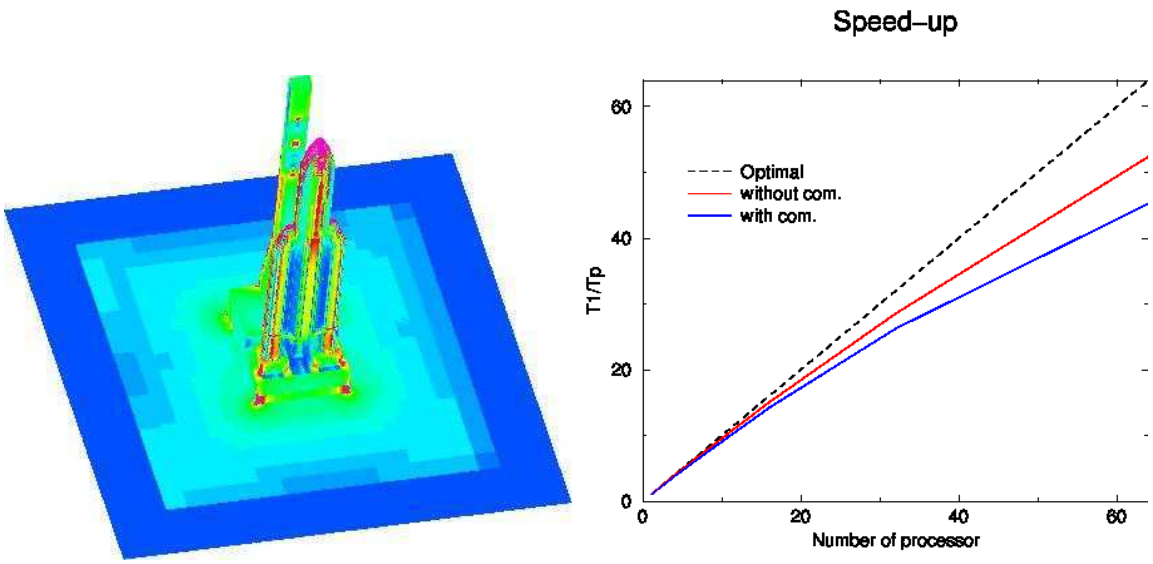


FIGURE 1.7 – Apport du calcul parallèle sur le site de lancement ELA3.

géométrie des objets ;
- une deuxième approche particulière (méthode de caractérisation de domaines) à un type de problème, où nous décomposons le domaine de calcul en plusieurs sous-domaines suivant les objets présents dans la scène de calcul.

a. Méthode de caractérisation de domaines par une matrice admittance

La première approche d'hybridation de méthodes consiste à découper l'ensemble du domaine de calcul en un ensemble de sous-domaines et à caractériser chacun d'eux par une matrice d'admittance ou de scattering. Nous réduisons ainsi la taille globale du problème par la résolution d'une ensemble de problèmes locaux de plus petite dimension. Les matrices d'admittance ou de scattering définissent une relation au niveau de l'interface de chaque sous-domaine, entre une source excitatrice et le champ diffracté par le sous-domaine. Par exemple, dans le cas de la caractérisation d'un sous-domaine par une matrice d'admittance, nous imposerons une source de champ électrique E sur l'interface et nous évaluerons sur celle-ci le champ magnétique H créé par cette source. L'interface d'un sous-domaine est définie par les zones de communications naturelles de celui-ci avec l'extérieur, mais aussi par les points de mesure que nous désirons connaître. L'évaluation d'une matrice d'admittance consiste alors, tout d'abord, à définir un espace d'approximation pour les champs au niveau de l'interface et un ensemble de n fonctions de base $\varphi_i(x),\ i = 1, n$ pour cet espace. Nous imposons ensuite sur chaque fonction de base i une source $\boldsymbol{E}(t, x)$ et nous relevons $\forall j = 1, n$, $\boldsymbol{H}(t, x).\varphi_j(x)$. Nous déterminons ainsi une matrice $\boldsymbol{Y}$ qui vérifie dans le domaine fréquentiel $E_p(f, x) = Y(f)H_p(f, x)$ où E_p et H_p sont les composantes des champs dans la base de l'interface. Nous pouvons ensuite décrire le problème initial sous forme d'un ensemble de conditions de continuité des traces de champs tangentiels entre sous-domaines au niveau

de chaque interface et donc sous forme d'un système linéaire dans le domaine fréquentiel [14]. Cette approche ne va s'avérer vraiment intéressante que si dans le problème étudié, les différents sous-domaines ne communiquent que par de petites ouvertures. En effet, dans ce cas, l'évaluation des matrices caractérisant chaque sous-domaine ne nécessite que très peu de degrés de liberté dans l'interface. Une réduction du nombre de degrés de liberté dans les interfaces diminue aussi la taille du système linéaire correspondant au problème initial.

Dans le domaine temporel [15], nous pouvons appliquer cette méthode en utilisant une méthode temporelle pour évaluer les matrices admittances caractérisant chaque sous-domaine. Cela est relativement coûteux en raison de l'impossibilité de traiter plusieurs sources à la fois comme dans le cas fréquentiel et en raison d'un stockage important de données fréquentielles pour évaluer le problème global (résolution d'un système linéaire par fréquence). En dépit de ces désavantages, cette stratégie peut trouver un intérêt sur des applications bien particulières. Par exemple, lorsque nous avons le choix entre plusieurs configurations d'un élément dans un système et que nous cherchons celui qui est optimal suivant un critère, il peut être alors intéressant de travailler avec cette approche. En effet, suivant le nombre de configurations possibles, le coût engendré par les calculs successifs sur la structure complète en changeant uniquement l'élément peut être prohibitif. A titre d'exemple [16] sur la figure 1.8, nous montrons un cas académique du choix d'un compartiment d'un avion entre deux configurations possibles pour minimiser les courants induits par rayonnement sur un câble positionné à l'intérieur de ce même compartiment. Dans ce cas, l'utilisation de la stratégie de décomposition par sous-domaines décrite précédemment, permet d'obtenir un gain en temps de calcul de 1.5 par rapport à l'étude séparée des deux configurations complètes. Ceci n'est bien évidemment pas énorme (le nombre de configurations était petit), mais montre que cette approche plutôt fréquentielle, peut aussi être intéressante en temporel.

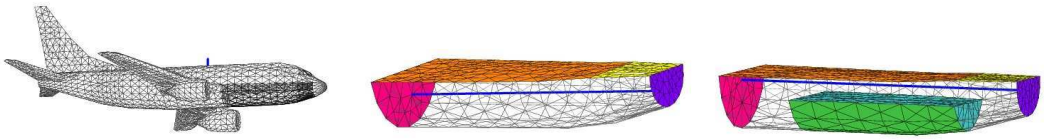


FIGURE 1.8 – Configuration étudiée pour l'approche sous-domaine.

b. Domaines disjoints

Il existe aussi des problèmes dans lesquels les objets sont disjoints les uns des autres. Dans ce cas, nous pouvons envisager de traiter le problème par des sous-domaines de calculs localisés autour de chaque objet et donc diminuer la taille du problème global à traiter. L'interaction entre sous-domaines est alors effectué par l'introduction de surfaces de rayonnement et de surfaces de Huygens dans chaque sous-domaine pour tenir compte du couplage entre zones. En pratique, sur chaque sous-domaine, nous évaluons le rayonnement provenant des autres sous-domaines comme un terme source sur la surface de Huygens et nous faisons rayonner vers les autres sous-domaines les courants évalués sur la surface de rayonnement

du sous-domaine considéré. La figure 1.9 montre les différentes étapes du processus.

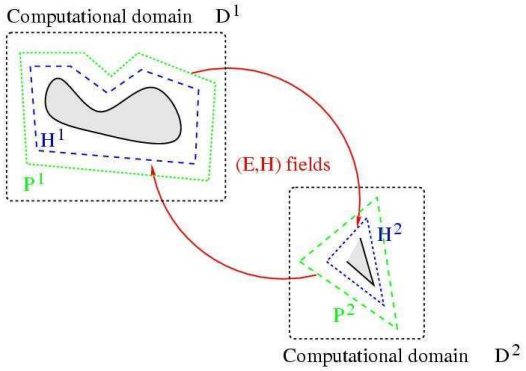


FIGURE 1.9 – Étapes du processus d'hybridation dans le cadre d'objets disjoints. Nous évaluons les champs sur chaque domaine à partir de sources provenant des champs rayonnés par les autres objets, et introduites sous forme de surface de Huygens, puis nous faisons rayonner pour l'étape suivante les champs vers les autres domaines pour définir les nouvelles sources.

Pour le rayonnement des courants d'un sous-domaine sur un autre, les champs sont calculés par une formule intégrale approchée [17]. Plus particulièrement, dans cette approche les méthodes FDTD et FVTD ont été couplées dans le but de pouvoir pour un domaine donné, utiliser la méthode la plus appropriée, comme le montre la figure 1.4 dans le cadre de l'exemple du projet GEMCAR. Dans cet exemple, nous découpons le domaine total en deux sous-domaines où la zone attribuée à l'antenne est considérée en différences finies à cause d'une géométrie très cartésienne et la zone attribuée à la voiture est considérée en volumes finis à cause de la géométrie courbe de la voiture. Cette approche naturellement parallèle, permet à précision équivalente de réaliser des gains mémoire et temps de calcul importants par rapport à une approche mono-domaine [18].

1.3.3 Le formalisme de fils mince et multi-conducteur

L'étude numérique d'un système face à une agression électromagnétique ne peut être complète sans prendre en compte la notion de câbles car ils jouent un rôle important dans l'étude de la compatibilité électromagnétique du fait qu'ils sont les voies de couplage privilégiées vers les interfaces des équipements.

La bonne modélisation des réseaux de câbles et de leur complexité est devenue un enjeu industriel majeur. L'introduction de modèles de fils minces, que ce soit en FDTD, (le fil mince de R. Holland [19] d'une part et le contour path de Taflove [20] d'autre part), ou dans d'autres méthodes [21, 22], ont permis de résoudre de nombreux problèmes industriels. Toutefois, ces modèles sont toujours en évolution pour traiter la problématique multi-conducteur dans la cellule [23] ou encore la localisation des câbles dans le maillage 3D (fils obliques) [24, 25, 26, 22].

A l'exception du contour path qui est une ré-interprétation de l'équation de Maxwell-Faraday, ces modèles sont construits en considérant le fil comme un modèle local dans la cellule ou sur quelques cellules autour de celui-ci. Les équations internes qui définissent l'évolution du courant et de la charge, sont déterminées à partir de l'équation de Maxwell dans la zone locale sous une hypothèse quasi-statique. Les paramètres de la ligne de transmission apparaissent naturellement et se présentent comme l'intégration géométrique de la distribution du champ dans la zone local au fil. Enfin, à ces premières équations, est associée l'équation de couplage avec le domaine 3D par la densité de courant $\boldsymbol{J}$.

1.3.4 Les limites des approches existantes

L'approche « full wave », qui consiste à modéliser l'ensemble de la scéne en 3D et à résoudre les équations de Maxwell par une discrétisation des équations sur ces domaines 3D, se veut générale de par sa nature, mais se heurte à certaines difficultés. D'une part, nous devons avoir à disposition une machine de puissance et d'autre part nous devons nous accommoder d'une dispersion numérique parfois importante liée à la taille et au temps d'observation du problème.

Les approches hybrides en découpage de domaines précédemment citées, ne répondent pas à l'ensemble des problématiques visées. L'hybridation par paramètre S est très spécifique à certaines applications (connexion de sous domaines par petites ouvertures). L'hybridation en domaines disjoints, quant à elle, se limite à des domaines séparés et ne prend pas en compte les connexions filaires entre ceux-ci.

En ce qui concerne la modélisation des faisceaux de câbles, les modèles de fils usuelle ne permettent pas de traiter toute la complexité de ceux-ci. La principale difficulté des approches actuelles est qu'à chaque configuration de faisceaux correspond un modèle (fil simple, fil oblique, multi-conducteur simple, coaxial, etc). Le fil oblique de C. Guiffaut est aujourd'hui le modèle le plus performant, mais certains points restent encore difficiles. Nous pouvons citer par exemple :

- la gestion des nœuds de courant quand plusieurs ensembles de câbles non connectés suivent une même arête du maillage 3D, puis se séparent ;
- la gestion des routages de câbles complexes composés de conducteurs, de gaines isolantes, et de blindages à plusieurs niveaux ;
- la modélisation des conditions aux limites, qui est un problème relativement simple dans le cas d'une bifilaire, mais qui devient nettement plus compliqué dans le cas d'un faisceau pouvant atteindre la centaine de câbles.

1.4 La stratégie proposée

La méthode que nous proposons d'étudier dans le cadre de cette thèse, pour pallier les déficiences des modèles actuels, est une approche hybride 1D/3D de méthodes temporelles. Plus précisément, il s'agit d'hybrider un schéma différences finies 3D pour le calcul des champs électromagnétiques avec un schéma différences finies 1D résolvant les équations de ligne de transmission. Dans cette approche nous focaliserons nos efforts sur l'hybridation 1D/3D pour prendre en compte des faisceaux de câbles complexes dans une zone de calcul 3D, mais aussi pour éviter le maillage de câbles d'interconnexion entre bâtiments en 3D dans une approche multi-domaine. Concernant le modèle d'équations de ligne de transmission, nous tiendrons compte de lignes au-dessus d'un sol réel en introduisant des paramètres de ligne dépendant de la fréquence. Une solution par décomposition en série de Debye sera alors proposée pour introduire ceux-ci dans notre schéma temporel différences finies 1D.

1.4.1 Hybridation de domaines avec prise en compte des lignes de transmission

L'objectif de cette hybridation est de modéliser les effets d'un réseau de câbles sur plusieurs systèmes en évitant l'utilisation d'une méthode 3D sur le domaine global. Pour cela, nous proposons une approche similaire à l'hybridation en domaines disjoints qui consiste à découper la zone globale en plusieurs domaines de calcul, chaque sous-domaine étant traité soit par une méthode FDTD 3D, soit par une méthode FDTD 1D, selon la précision recherchée et les phénomènes physiques mis en jeu. Ce découpage en sous-domaines est intéressant car il permet de réduire la volumétrie du problème, en particulier, les problèmes où une grande partie de l'espace peut raisonnablement être modélisée par une ligne de transmission au-dessus d'un plan métallique ou d'un sol. Nous pourrons utiliser dans ces zones ce formalisme qui est plus compact (nombre d'inconnues réduit, ...), tout en conservant la souplesse de la modélisation 3D là où cela est nécessaire. Ce choix permet de combiner performance, précision et souplesse de mise en œuvre.

1.4.2 Hybridation d'échelle, extension du fil mince

L'objectif de cette hybridation est de tenir compte de la complexité des faisceaux de câbles dans le problème 3D et d'être capable de décliner la contrainte sur chacun des câbles à partir d'un champ électromagnétique. En effet, la prise en compte d'un ensemble de câbles avec des blindages (pouvant avoir plusieurs niveaux) et des gaines diélectriques est important. Pour cela, nous proposons d'utiliser un modèle de ligne de transmission pour tenir compte des faisceaux dont l'encombrement est petit par rapport au maillage. Les paramètres de la ligne sont obtenus en calculant la solution d'un problème de Laplace. Enfin, le couplage avec le domaine 3D est réalisé par le courant de mode commun.

1.4.3 Prise en compte d'un sol de conductivité finie (dépendance en fréquence)

L'utilisation d'une stratégie hybridation par découpage de domaine permet d'assouplir l'étude de grands systèmes terrestres. Cependant les paramètres d'une ligne de transmission

au-dessus d'un sol de conductivité finie dépendent de la fréquence, ce qui est une difficulté pour une méthode temporelle. Nous présenterons donc un ensemble de modèles existants permettant de prendre en compte les lignes de transmission au-dessus d'un sol et nous proposerons une stratégie qui permette de décomposer ces paramètres dépendant de la fréquence en une série de Debye pour les introduire dans notre schéma temporel, tout en assurant la stabilité de celui-ci.

1.4.4 Mise en place d'une stratégie informatique efficace pour l'hybridation forte

La place des solutions hybrides dans les méthodes numériques est de plus en plus importante. Cependant, ces méthodes restent délicates à mettre en œuvre d'un point de vue informatique. Nous proposons dans le cadre de cette thèse de formaliser la problématique de l'hybridation forte dans le but de définir une approche non intrusive. Cette étude donnera lieu à la mise en place d'un prototype satisfaisant au besoin de l'hybridation.

La problématique de la thèse devient alors celle de l'utilisation conjointe de deux approches : l'une 3D, basée sur le calcul des champs E et H par les équations de Maxwell discrétisées, et l'autre 1D, basée sur le calcul de i et v le long d'une ligne de transmission.

Chapitre 2

Modélisation des lignes de transmission dans le domaine temporel avec prise en compte d'un sol non parfaitement conducteur

Sommaire

2.1 Introduction 32
2.2 Généralités sur les lignes de transmission : modèles et paramètres 33
2.2.1 L'équation de ligne de transmission et les paramètres liés à celle-ci 33
2.2.2 Étude du cas d'une ligne de transmission au-dessus d'un sol 39
2.2.3 Modèle de couplage - Source distribuée 46
2.2.4 Multi conducteurs - problématique du faisceau de câbles et des domaines internes 50
2.3 Résolution numérique des équations de ligne dans le domaine temporel 57
2.3.1 Approximation numérique - méthode MTL 57
2.3.2 Conditions aux limites et sources localisées 62
2.3.3 Résultat numérique - le problème de Franck Arreghini 68
2.4 Prise en compte de la dépendance en fréquence des paramètres de ligne - Cas d'une ligne au-dessus d'un sol 71
2.4.1 Approximation par une série de Debye 72
2.4.2 Validation numérique 81
2.4.3 Modèle de Debye dans le formalisme des lignes de transmission - Mise en œuvre et validation 88
2.4.4 Validation et étude de la factorisation en série de Debye 90
2.5 Conclusion 98

2.1 Introduction

Nous venons de voir dans le chapitre précédent les limitations des codes 3D pour traiter de grandes scènes de calcul. La présence de câbles ou de conducteurs entre les bâtiments complexifie grandement le problème. En effet, du fait des grandes distances entre les bâtiments, il est difficile de considérer la globalité de la scène dans un seul domaine 3D. Aussi, comme nous l'avons expliqué dans le premier chapitre (Sec. 1.2.1 page 20), une solution consiste à scinder la géométrie en plusieurs sous-domaines, avec des parties 3D et des parties ne contenant que les liaisons et/ou interconnexions, qui peuvent être modélisées en 1D par une équation de ligne.

Ce chapitre présente une méthode de résolution des équations de ligne qui a été développée et mise en œuvre dans le cadre de cette thèse pour répondre en particulier à la problématique des lignes au-dessus d'un sol. Cette méthode est basée sur un schéma temporel, qui doit d'une part être suffisamment général pour répondre de manière autonome, à des problématiques « classiques » de couplage sur câbles, mais qui doit aussi, d'autre part, pouvoir être facilement adaptable à une hybridation avec des méthodes de résolution des équations de Maxwell telles que les méthodes différences finies (FDTD) ou volumes finis (FVTD).

Après avoir rappelé dans une première section, le modèle mathématique des équations de ligne de transmission et les grandeurs physiques associées à celles-ci, nous nous penchons particulièrement sur la ligne de transmission au-dessus d'un sol, qui est une configuration importante pour nos applications. Nous nous intéressons ensuite, à la modélisation du couplage du champ électromagnétique sur une ligne de transmission et nous terminons la section par l'introduction de la notion de ligne multifilaire.

La deuxième section présente le schéma différences finies utilisé pour approximer le modèle d'équations de ligne de transmission. Le choix de la méthode numérique a été effectué pour permettre une hybridation facile avec une méthode FDTD-3D. Nous avons notamment étudié les propriétés numériques de ce schéma pour vérifier sa convergence et sa stabilité. Par la suite, nous introduisons dans celui-ci un certain nombre de conditions aux limites, dont en particulier les nœuds de courants qui permettent de connecter différentes lignes entre-elles. Enfin, nous validons notre méthode numérique sur un exemple canonique.

Dans la dernière section de ce chapitre, nous montrons comment prendre en compte un modèle de ligne au-dessus d'un sol de conductivité finie dans notre méthode numérique. Cette configuration est difficile à mettre en œuvre dans une méthode temporelle car les paramètres de la ligne dépendent de la fréquence. Pour introduire ces paramètres, nous donnons deux approches qui consistent à décomposer ceux-ci en série de Debye, soit en résolvant un problème de moindres carrés, soit en utilisant un algorithme d'optimisation globale qui permet de garantir la stabilité de notre méthode numérique.

2.2 Généralités sur les lignes de transmission : modèles et paramètres

Nous présentons dans cette section le modèle physique des équations de ligne de transmission et les paramètres généraux liés à celui-ci, dans le cas d'une ligne au-dessus d'un sol. Pour le cas d'un sol non parfaitement conducteur, le modèle classique de ligne de transmission nécessite certaines adaptations comme notamment la prise en compte de paramètres dépendant de la fréquence que nous allons détailler.

2.2.1 L'équation de ligne de transmission et les paramètres liés à celle-ci

La théorie des lignes de transmission repose sur un système d'équations linéaires qui découle des équations de Maxwell par intégration du champ électromagnétique et en faisant une hypothèse de quasi-stationnarité. Les équations différentielles obtenues décrivent l'évolution d'un courant et d'une tension sur la ligne en fonction du champ électromagnétique présent dans le domaine.

L'utilisation d'une équation de ligne de transmission nécessite la connaissance d'une référence électrique. D'une manière générale, cette référence est donnée soit par le blindage du câble, soit par un élément conducteur extérieur, comme par exemple un plan métallique ou un sol conducteur. Pour cela, nous considérons toujours qu'une ligne de transmission est définie par au moins deux conducteurs qui représentent l'âme et la référence de la ligne. Par la suite, nous supposerons que l'âme est traversée par un courant $i(t, l)$ qui dépend de la position l sur la ligne et de l'instant d'observation t et que la différence de potentiel entre celle-ci et le conducteur de référence est donnée par $v(t, l)$.

Le principe du modèle, pour prendre en compte une ligne de transmission, consiste en un découpage de celle-ci en plusieurs segments. Nous définissons sur chaque segment une impédance Z et une admittance Y par unité de longueur, comme représenté sur la figure 2.1 pour un segment de longueur dl.

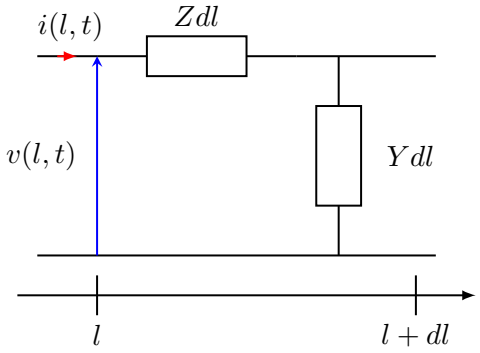


FIGURE 2.1 – Schéma équivalent ZY d'une ligne de transmission.

Sur chaque segment, nous pouvons écrire les équations différentielles suivantes dans le domaine fréquentiel :

$$\partial_l \widetilde{v} = -Z\widetilde{i} \tag{2.1a}$$
$$\partial_l \widetilde{i} = -Y\widetilde{v} \tag{2.1b}$$

Avec le symbole $\widetilde{.}$ désignant la forme harmonique de la variable. Nous définissons usuellement l'impédance caractéristique de la ligne de transmission par $Z_c = \sqrt{\frac{Z}{Y}}$. Nous pouvons aussi définir l'impédance et l'admittance sous la forme (Fig. 2.2) :

$$Z = R + j\omega L \tag{2.2a}$$
$$Y = G + j\omega C \tag{2.2b}$$

où R, L, G et C sont respectivement la résistivité (Ω/m), l'inductance (H/m), la capacité (F/m), et la conductance (S/m) linéiques de la ligne de transmission.

En introduisant ces formules dans les équations (2.1), nous obtenons alors le modèle de Heaviside suivant :

$$\partial_l \widetilde{v} = -(R + j\omega L)\widetilde{i} \tag{2.3a}$$
$$\partial_l \widetilde{i} = -(G + j\omega C)\widetilde{v} \tag{2.3b}$$

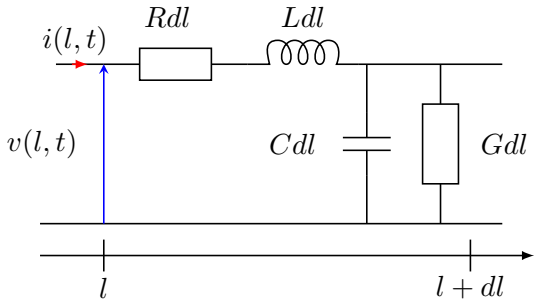


FIGURE 2.2 – Paramètres distribués R, L, C et G , schéma équivalent de Heaviside.

Ce modèle reste valable lorsque nous avons n conducteurs dans la ligne. Dans ce cas, la différence de potentiel $\boldsymbol{v}_k(l,t)$ entre le conducteur k et le conducteur de référence ainsi que le courant $\boldsymbol{i}_k(l,t)$ traversant chaque conducteur k, sont des vecteurs. Le courant i_0 dans le conducteur de référence est obtenu par $i_0 = -\sum_{k=1}^{n} i_k(l,t)$, où :

$$\boldsymbol{i}(l,t) = \big(i_1(l,t), \cdots, i_n(l,t)\big)^T \tag{2.4a}$$
$$\boldsymbol{v}(l,t) = \big(v_1(l,t), \cdots, v_n(l,t)\big)^T \tag{2.4b}$$

Les paramètres de la ligne sont alors décrits par des matrices carrées de dimension n qui sont toutes symétriques et réelles.

$$\boldsymbol{R} = Diag\left[R_1, \cdots, R_n\right] \tag{2.5}$$

$$\boldsymbol{L} = \begin{bmatrix} L_1 & M_{1,2} & \cdots & M_{1,n} \\ M_{2,1} & L_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & M_{n-1,n} \\ M_{n,1} & \cdots & M_{n,n-1} & L_n \end{bmatrix} \tag{2.6}$$

$$\boldsymbol{C} = \begin{bmatrix} \sum_{k\neq 1} C_{1,k} & -C_{1,2} & \cdots & -C_{1,n} \\ -C_{2,1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -C_{n-1,n} \\ -C_{n,1} & \cdots & -C_{n,n-1} & \sum_{k\neq n} C_{n,k} \end{bmatrix} \tag{2.7}$$

$$\boldsymbol{G} = \begin{bmatrix} 0 & G_{1,2} & \cdots & G_{1,n} \\ G_{2,1} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & G_{n-1,n} \\ G_{n,1} & \cdots & G_{n,n-1} & 0 \end{bmatrix} \tag{2.8}$$

Les matrices, R_i, L_i, $M_{i,j}$, $C_{i,j}$ et $G_{i,j}$ définissent respectivement la résistance linéique du conducteur i, son inductance propre, l'inductance mutuelle, la capacité et la conductance entre les conducteurs i et j. Ces paramètres dépendent de la géométrie considérée et des propriétés des matériaux.

Dans le modèle que l'on vient de présenter, R et G désignent les termes de perte dans la ligne. Or, ces termes de perte sont la conséquence de plusieurs phénomènes :

- Les pertes dans les conducteurs dues à la conductivité finie des matériaux utilisés pour ces conducteurs.
- L'effet de peau, qui limite la section utile d'un conducteur, et accroît les pertes suivant la fréquence.
- Les courants de fuite dans la capacité introduite par une conductance G en parallèle avec la capacité, souvent caractérisée par un angle de perte $\tan\phi = \frac{G}{C\omega}$ avec ϕ, de l'ordre de 10^{-4} pour les meilleurs conducteurs.

Pour rendre compte correctement de cette réalité physique, nous devons prendre en compte d'une part les pertes dans le conducteur principal modélisées par Z_c, et d'autre part celles du conducteur de référence modélisées par Z_r. De même, nous devons également prendre en compte les courants de fuites introduits par le milieu entre les conducteurs (représentés par le paramètre G', nul en espace libre), et par les conducteurs eux-mêmes que l'on représente sous forme d'admittances Y_c et Y_r. Nous aboutissons ainsi au modèle étendu de ligne, en remplaçant Z et Y par :

$$Z = jL\omega + Z_c + Z_r \tag{2.9a}$$

$$Y = jC\omega + G' + Y_c + Y_r \tag{2.9b}$$

où les termes Z_c, Z_r et Y_c, Y_r désignent des termes complexes. L'introduction de ces termes dans le modèle de ligne impliquera une modification des quantités R, L, G, C du modèle de Heaviside telles que nous les avions définis.

Si nous conservons le modèle de Heaviside, R et L sont plutôt définis comme les parties réelle et imaginaire de Z, alors que G et C sont celles de Y. Dans le cas de configurations géométriques canoniques, certains paramètres de ligne peuvent s'exprimer à l'aide de formules analytiques. A titre d'exemple, le tableau 2.1 fournit, respectivement, les paramètres (L, C) pour différentes configurations de ligne.

De même, concernant les impédances (Z_c, Z_r, Y_c, Y_r), il existe pour certaines configurations canoniques des valeurs analytiques (Tab. 2.2) qui ont été formalisées par Edward F. Vance [27]. Edward F. Vance définit en particulier, deux approximations asymptotiques dont l'une est pour les basses fréquences (BF) et l'autre s'applique pour les hautes fréquences (HF). L'introduction de ces approximations est motivée par l'extrême difficulté de travailler directement avec des fonctions de Bessel pour tenir compte de ces paramètres fréquentiels.

Pour un conducteur, ces approximations dépendent de l'ordre de grandeur de l'épaisseur de peau δ par rapport au diamètre du conducteur a. Dans le cas où $\delta \gg a$ alors la fréquence f de calcul doit vérifier $f \ll \frac{1}{d^2 \pi \mu \sigma}$ [1] et on a donc un comportement dans les basses fréquences. A l'inverse, si $\delta \ll a$, on obtient un comportement dans les hautes fréquences.

Concernant l'admittance complexe, qui représente les courants de fuite, celle-ci est négligeable dans un grand nombre de cas. Néanmoins, si cette admittance devient significative elle peut être approchée par :

$$Y_i = \frac{\gamma_i^2}{Z_i} \tag{2.10}$$

avec :

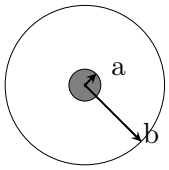
$$\gamma_i = \sqrt{j\omega\mu_i(\sigma_i + j\omega\varepsilon_i)} \tag{2.11}$$

où ε_i, σ_i et μ_i sont les paramètres diélectriques du milieu.

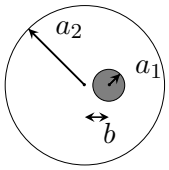
Le modèle et les paramètres associés étant définis dans le cas général d'une ligne de transmission, nous allons maintenant nous intéresser aux cas particuliers suivants :

- les fils au-dessus d'un plan métallique ;
- les fils au-dessus d'un sol réel.

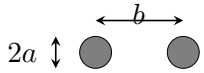
1. Cette fréquence remarquable est obtenue en développant la contrainte sur l'épaisseur de peau δ



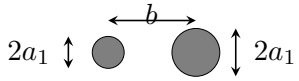
$$Z_0 = \frac{\eta}{2\pi} \log \frac{b}{a}$$



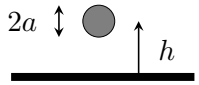
$$Z_0 = \frac{\eta}{2\pi} \cosh^{-1}\left(\frac{a_1^2+a_2^2-b^2}{2a_1a_2}\right)$$



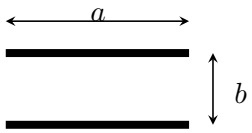
$$Z_0 = \frac{\eta}{\pi} \cosh^{-1}\left(\frac{b}{2a}\right) \approx \frac{\eta}{\pi} \log\left(\frac{b}{a}\right), \quad \left(\frac{b}{a} \gg 1\right)$$



$$Z_0 = \frac{\eta}{2\pi} \cosh^{-1}\left(\frac{b^2-a_1^2-a_2^2}{2a_1a_2}\right)$$



$$Z_0 = \frac{\eta}{2\pi} \cosh^{-1}\left(\frac{h}{a}\right) \approx \frac{\eta}{2\pi} \log\left(\frac{2h}{a}\right), \quad \left(\frac{b}{a} \gg 1\right)$$



$$Z_0 = \eta \frac{b}{a}, \quad (a \gg b)$$

avec :

$\eta =$	$\sqrt{\frac{\mu}{\varepsilon}}$	Impédance du milieu [Ω/m]
$\nu =$	$\frac{1}{\sqrt{\mu\varepsilon}}$	Vitesse de propagation dans le milieu [m/s]
$Z_0 =$	$\sqrt{\frac{L}{C}}$	Impédance caractéristique de la ligne en ohm [Ω]
$L =$	$\frac{Z_0}{\nu}$	Inductance propre distribuée induite en Henry [H]
$C =$	$\frac{1}{\nu Z_0}$	Capacité distribuée induite en Farad [F]

TABLE 2.1 – Impédances caractéristiques de quelques configurations canoniques de ligne de transmission.

Description	Section	Impédance exacte	Impédance approchée
Impédance interne pour retour externe		$Z_i = -\frac{j\gamma}{2\pi a\sigma}\frac{J_0(j\gamma a)}{J_1(j\gamma a)}$	$Z_i^{BF} = \frac{1}{\pi a^2\sigma}, \quad (\delta \gg a)$ $Z_i^{HF} = \frac{1+j}{2\pi a\sigma\delta}, \quad (\delta \ll a)$
Impédance interne pour retour interne		$Z_i = -\frac{j\omega}{4\pi h\sigma}\frac{H_0^{(1)}(j\gamma_s 2h)}{H_1^{(1)}(j\gamma_s 2h)}$	$Z_i^{BF} = \frac{\omega\mu_s}{8} + j\omega\frac{\mu}{2\pi}\log\left(\frac{\sqrt{2}\delta}{\gamma_0 a}\right), \quad (\delta \gg a)$ $Z_i^{HF} = \frac{1+j}{2\pi a\sigma\delta}, \quad (\delta \ll a)$
Impédance interne d'un demi espace par retour cylindrique		$Z_i = -\frac{j\omega}{4\pi h\sigma}\frac{H_0^{(1)}(j\gamma 2h)}{H_1^{(1)}(j\gamma_s 2h)}$	$Z_i^{BF} = \frac{\omega\mu}{8} + j\omega\frac{\mu}{2\pi}\log\left(\frac{\delta}{\sqrt{2}\gamma_0 h}\right), \quad (\delta \gg 2h)$ $Z_i^{HF} = \frac{1+j}{4\pi h\sigma\delta}, \quad \begin{bmatrix}\delta \ll 2h \\ \sigma \gg \omega\varepsilon\end{bmatrix}$

Avec :

$\gamma = \sqrt{j\omega\mu(\sigma + j\omega\varepsilon)}$ $\quad \delta = \frac{1}{\sqrt{\pi f\mu\sigma}}$ $\quad \omega = 2\pi f$ $\quad \gamma_0 = 1.7811...$

ε permittivité [F/m] $\quad \mu$ perméabilité [H/m] $\quad \sigma$ conductivité [Ω/m] $\quad f$ fréquence [Hz]

TABLE 2.2 – Impédance interne caractéristique de quelques configurations canoniques de conducteurs.

2.2.2 Étude du cas d'une ligne de transmission au-dessus d'un sol

Dans cette section, nous allons nous pencher sur la détermination des paramètres électriques (R, L, C et G) d'une ligne de transmission au-dessus d'un sol représentée par la figure 2.3.

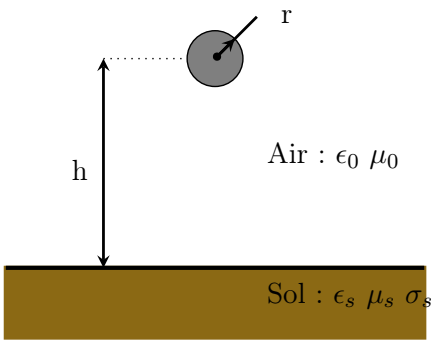


FIGURE 2.3 – Ligne de transmission au-dessus d'un sol plan à une hauteur h. La ligne de transmission est constituée d'un conducteur cylindrique de rayon r, parallèle au sol dont les paramètres physiques sont : ε_s, μ_s et σ_s

a. Cas d'un sol et d'une ligne parfaitement conducteurs

Si nous considérons une ligne parfaitement conductrice au-dessus d'un sol parfaitement conducteur, la ligne de transmission est sans perte ($Z = j\omega L$ et $Y = j\omega C$) et ses paramètres sont définis par (Tab. 2.1) :

$$L \approx \frac{\mu_0}{2\pi} \log(\frac{2h}{r}) \quad \text{lorsque } (h \gg r) \tag{2.12a}$$

$$C \approx \frac{2\pi\varepsilon_0}{\log(\frac{2h}{r})} \quad \text{lorsque } (h \gg r) \tag{2.12b}$$

Toutefois, il est à noter que ces expressions montrent des limites de validité dans les hautes fréquences où l'effet de peau n'est plus négligeable et entraîne des pertes.

b. Cas d'un sol et d'une ligne non parfaitement conducteurs

Dans cette section, nous allons introduire les impédances internes de chaque conducteur ; c'est-à-dire du sol et de la ligne, puis nous étudierons leur influence sur l'impédance totale de la ligne de transmission.

Soient les hypothèses suivantes :

- Les paramètres physiques du sol (ε_s, σ_s et μ_s) sont supposés homogènes et indépendants de la fréquence.
- La ligne est supposé de conductivité σ_c et homogène sur toute sa longueur.

L'impédance Z de la ligne de transmission peut se décomposer en trois termes :

- un premier terme qui définit l'inductance mutuelle L entre le conducteur de la ligne et le sol. Celle-ci est égale à l'inductance obtenue dans le cas d'un sol parfaitement conducteur ;
- un deuxième terme qui définit l'impédance Z_c propre au conducteur ;
- un dernier terme Z_s qui définit l'impédance propre au sol.

Nous obtinons donc :

$$Z = j\omega L + Z_c + Z_s \tag{2.13}$$

De manière générale, nous pouvons aussi écrire l'admittance de la ligne sous la forme de trois termes qui définissent la capacité mutuelle C entre le conducteur et le sol, l'admittance (Y_g) propre au sol et celle (Y_c) propre au conducteur. Nous avons donc l'admittance suivante :

$$\frac{1}{Y} = \frac{1}{j\omega C} + \frac{1}{Y_c} + \frac{1}{Y_g} \tag{2.14}$$

Nous allons maintenant nous intéresser dans l'impédance et l'admittance de la ligne aux termes liés au conducteur de celle-ci et à ceux liés au sol, afin de quantifier leur importance.

Termes propres aux conducteurs

L'impédance Z_c est le premier paramètre du conducteur que nous allons considérer. Cette impédance est définie comme suit (Tab. 2.2) :

$$Z_c^{BF} = \frac{1}{\pi r^2 \sigma_c}, \quad (\delta_c \gg r) \tag{2.15a}$$

$$Z_c^{HF} = \frac{1+j}{2\pi r \sigma_c \delta_c}, \quad (\delta_c \ll r) \tag{2.15b}$$

avec :

δ_c	$= \frac{1}{\sqrt{\pi f \mu_c \sigma_c}}$	l'épaisseur de peau dans le conducteur,
ε_c		la permittivité du conducteur [F/m],
μ_c		la perméabilité du conducteur [H/m]
et σ_c		la conductivité du conducteur [S/m].

En basse fréquence (Eq. 2.15a), l'impédance est réelle et est donnée par la résistivité linéique d'un fil $r = \frac{1}{\sigma S}$ avec S la surface de sa section. En haute fréquence (Eq. 2.15b) la section utile (dans laquelle s'écoule les électrons) est restreinte et matérialisée par l'épaisseur de peau δ qui introduit un terme résistif et inductif, comme schématiquement représenté en figure 2.4. Clayton R. Paul dans son ouvrage [28] indique que les domaines $\delta_c > 2r$ pour les basses fréquences et $\delta_c < 2r$ pour les hautes fréquences sont suffisants.

Nous avons voulu évaluer quelle était la contribution de la nature du conducteur sur l'impédance linéique du câble au-dessus d'un sol. Pour réaliser cette comparaison, nous avons tracé d'une part les pertes linéiques $\Re\{Z_c\}$ ainsi que la contribution relative sur la partie

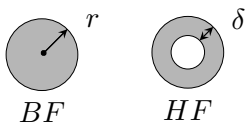


FIGURE 2.4 – Illustration de l'effet de peau dans un conducteur cylindrique

imaginaire par rapport à l'impédance de la ligne de transmission donnée par $\Im\{Z_c\}/\omega L$. Nous avons fait cette comparaison pour deux métaux, l'un très conducteur le cuivre $\sigma_{Cu} = 58.5$ [MS/m] et l'autre peu conducteur l'acier inoxydable 310 $\sigma_{inox310} = 0.14$ [MS/m] [29]. Nous avons également choisi de prendre deux rayons différents pour voir l'influence de ce paramètre. La hauteur du fil est fixée à 1 mètre. Les fréquences de transition du domaine basse fréquence au domaine haute fréquence sont données dans le tableau 2.3.

	Cuivre	Inox 310
rayon r	f_c	f_c
1cm	10.82Hz	4.523kHz
1mm	1082.Hz	452.3kHz

TABLE 2.3 – Fréquence de transition dans le conducteur entre le domaine basse fréquence et haute fréquence définie par : $\delta = 2r$.

La figure 2.5 nous donne l'évolution des paramètres pour deux configurations. Nous constatons que le terme inductif introduit par le conducteur est négligeable devant l'inductance propre de la ligne. En revanche les pertes dans le conducteur augmentent avec la fréquence. Dans le cas le plus défavorable (petit rayon, faible conductivité), ce terme devient important et devra être pris en compte, alors que dans la majorité des cas, l'impédance du conducteur reste négligeable.

L'admittance Y_c peut être approximée par (Eq. 2.10). Comme pour l'impédance nous avons cherché à évaluer sa contribution sur l'admittance linéique. Nous observons d'une part les pertes $\Re\{Y_c\}$ ainsi que l'erreur relative sur la capacité. Rappelons que les admittances sont en série (Eq. 2.14) et que donc l'admittance totale est donnée par $Y_t = jC\omega Y_c/(jC\omega + Y_c)$. Dans le cas où les pertes sont faibles, cette admittance peut être approchée par : $Y_t \approx G_c C/(C_c + C) + j\omega C C_c/(C_c + C)$ ou $G_c = \Re\{Y_c\}$ et $\omega C_c = \Im\{Y_c\}$, où le rapport $1 - C_c/(C_c + C)$ représente l'influence relative du sol sur la capacité et le terme G_c les pertes dans le sol.

La figure 2.6 trace l'évolution de ces deux paramètres. Nous constatons d'une part que les pertes sont effectivement négligeables et inférieures à 1nS/m, ce qui justifie l'approximation « faibles pertes ». Enfin, nous constatons que l'admittance du conducteur a une contribution relative de l'ordre de 10^{-9}, donc négligeable par rapport à l'admittance totale.

En conclusion nous pouvons dire que le conducteur a une influence négligeable sur le comportement de la ligne. En conséquence, nous ne tiendrons pas compte du paramètre Y_c dans le calcul de l'admittance de la ligne.

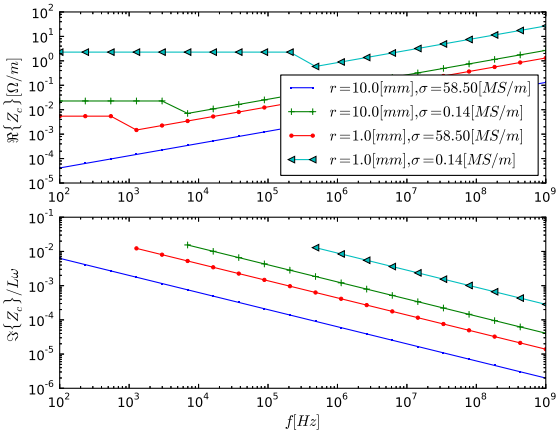


FIGURE 2.5 – Contribution du conducteur cylindrique par rapport à l'impédance linéique de la ligne.

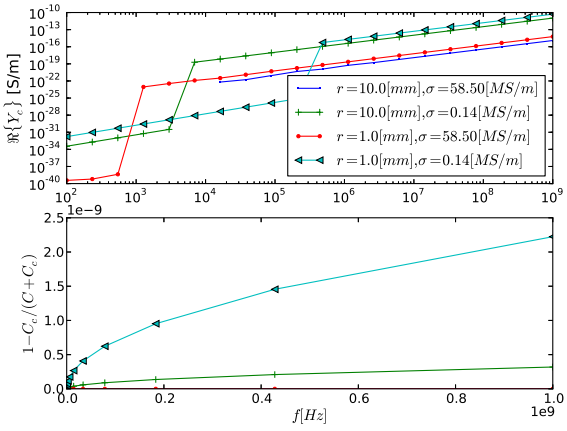


FIGURE 2.6 – Contribution du conducteur cylindrique par rapport à l'admittance linéique de la ligne.

Termes propres aux sols

L'impédance Z_s est le premier paramètre du sol que nous allons considérer. Cette impédance est définie comme suit (Tab. 2.2) :

$$Z_s^{BF} = \frac{\omega\mu_s}{8} + j\omega\frac{\mu_s}{2\pi}\log\left(\frac{\delta_s}{\sqrt{2}\gamma_0 h}\right), \quad (\delta_s \gg 2h) \tag{2.16a}$$

$$Z_s^{HF} = \frac{1+j}{4\pi h\sigma_s\delta_s}, \quad \begin{bmatrix}\delta_s \ll 2h \\ \sigma_s \gg \omega\varepsilon_s\end{bmatrix} \tag{2.16b}$$

avec

δ_s	$= \frac{1}{\sqrt{\pi f\mu_s\sigma_s}}$	l'épaisseur de peau dans le sol,
ε_s		la permittivité du sol [F/m],
μ_s		la perméabilité du sol [H/m]
et σ_s		la conductivité du sol [S/m].

À basse fréquence, nous retrouvons un terme en $j\omega k\log(x)$ de forme similaire à l'inductance linéique de la ligne (Eq. 2.12a) qui est équivalent à l'image de la ligne de transmission dans un miroir imparfait. À haute fréquence, l'impédance est similaire à un terme multiplicatif près, à l'impédance d'un conducteur cylindrique (Eq. 2.15b) car la longueur d'onde devient suffisamment petite pour que le conducteur cylindrique puisse s'apparenter à un plan infini.

Comme pour le cas du conducteur, nous avons tracé les deux mêmes indicateurs pour des configurations représentatives de la disparité du sol que nous pouvons trouver. Nous avons choisi deux types de sol : un sol très sec et un sol très humide dont les paramètres électriques sont donnés dans le tableau 2.4. La hauteur entre la ligne et le sol varie de 50cm à 5m et le rayon du conducteur est fixé à 1cm. Les fréquences de transition du domaine basse fréquence au domaine haute fréquence sont données dans le tableau 2.5.

Nature du sol	ε_s	σ à 1MHz	σ à 10MHz	σ à 100MHz	σ à 1GHz
Sol très sec	3	0.1	0.1	0.1	0.2
Glace d'eau douce à $-12°$C	3	0.01	0.03	0.08	0.4
Sol de conductivité moyenne	15	1	1	2	40
Terrain très humide	30	10	10	20	200
Eau douce à 20°C	80	3	3	5	200
Eau de mer à 20°C	70	5000	5000	5000	5000

TABLE 2.4 – Conductivité du sol (terre immergée ou océan) σ [mS/m] à diverses fréquences et permittivité ε_s [F/m] (constante sur la bande de fréquence).

La figure 2.7 donne l'évolution des indicateurs permettant d'apprécier l'influence du sol sur les pertes et l'inductance de la ligne. Nous constatons que le sol a une influence non négligeable (les pertes peuvent être de l'ordre de l'ohm) et qu'il contribue pour 1/10 ème à l'admittance totale.

	Sol très sec	Sol très humide
hauteur h	f_c	f_c
50cm	2.53GHz	25.3MHz
5m	25.3MHz	253kHz

TABLE 2.5 – Fréquence de transmission dans le sol entre le domaine basse fréquence et haute fréquence définie par : $\delta = 2h$.

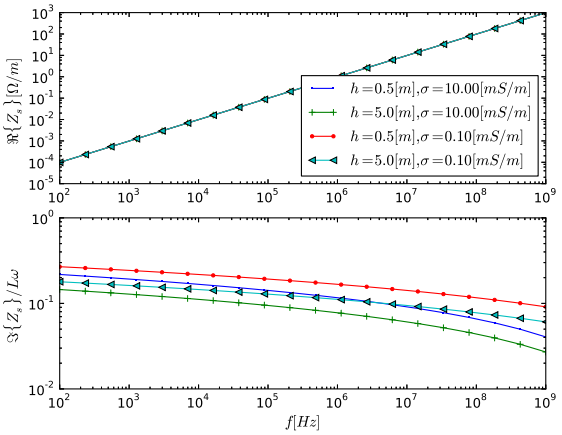


FIGURE 2.7 – Contribution du sol par rapport à l'impédance linéique de la ligne.

L'admittance Y_s peut être approximée par la relation $Y_i Z_i = \gamma_i^2$ donnée par l'équation (2.10). Nous faisons la même analyse que précédemment et nous constatons que l'admittance propre au sol peut être négligée dans le calcul de l'admittance de la ligne (Fig. 2.8).

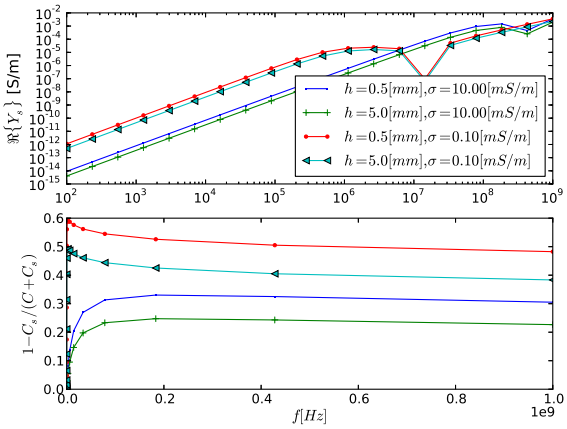


FIGURE 2.8 – Contribution du sol par rapport à l'admittance linéique de la ligne.

Conclusion De l'étude menée, nous pouvons conclure que pour modéliser correctement une ligne au-dessus d'un sol non parfaitement conducteur, il faut prendre en compte l'impédance propre du sol dans le calcul des paramètres du modèle de ligne.

2.2.3 Modèle de couplage - Source distribuée

Il existe différentes approches pour représenter l'action d'un champ électromagnétique sur une ligne de transmission (Fig. 2.9). Ces différents modèles sont parfaitement équivalents. Le modèle référent est celui de Taylor, Satterwhile et Harrison, développé en 1965, qui représente l'action conjointe du champ électrique et magnétique sur une longueur élémentaire de la ligne de transmission. Ainsi, le modèle de couplage distribué de Taylor définit une source de courant et une source de tension distribuées le long de la ligne. La source de courant représente l'excitation de la ligne par le champ électrique entre les deux conducteurs. La source de tension quant à elle, représente l'induction du champ magnétique enlacé par les conducteurs. Nous pouvons alors réécrire les équations de ligne de transmission dans le domaine fréquentiel par :

$$\partial_x \widetilde{v} + j\omega L \widetilde{i} = \widetilde{v}_s \tag{2.17a}$$

$$\partial_x \widetilde{i} + j\omega C \widetilde{v} = \widetilde{i}_s \tag{2.17b}$$

où $\widetilde{v}_s$ est la source de tension distribuée et $\widetilde{i}_s$ la source de courant distribuée définies par :

$$\widetilde{v}_s = -j\omega\mu_0 \int_\Gamma \widetilde{H}_y^i dz \tag{2.18a}$$

$$\widetilde{i}_s = -j\omega C \int_\Gamma \widetilde{E}_z^i dz \tag{2.18b}$$

avec les notations données sur la figure 2.9.

A partir de ce modèle, d'autres modèles ont été dérivés, notamment ceux de Agrawal (1980) et de Rachidi (1993). Dans ces modèles, l'action d'un champ électromagnétique externe sur une ligne de transmission est caractérisée par les composantes du champ électrique longitudinal au fil pour le modèle de Agrawal et par le champ magnétique tangent à la surface entre les deux conducteurs pour le modèle de Rachidi. Des conditions aux limites ont ensuite été ajoutées à ces modèles pour refermer la ligne de transmission qui peut être ainsi vue comme une boucle de courant.

Dans le cadre de cette thèse, nous nous sommes essentiellement intéressés au modèle de Agrawal, Price et Gurbaxani qui est particulièrement bien adapté pour le couplage d'une ligne de transmission avec une méthode 3D. En effet, pour réaliser ce couplage, nous avons juste à relever le champ électrique incident E^i longitudinal au fil. Certaines précautions devront tout de même être prises dans le cas où la ligne n'est pas en court-circuit ou en circuit ouvert de manière à bien avoir la tension totale en bout de ligne.

La première équation de Agrawal s'obtient à partir de la première équation de Taylor (Eq. 2.17a). Nous décomposons la tension totale $\widetilde{v}$ en deux tensions, la tension incidente $\widetilde{v}^i$ et la tension propagée $\widetilde{v}^s$.

$$\widetilde{v} = \widetilde{v}^s + \widetilde{v}^i \tag{2.19}$$

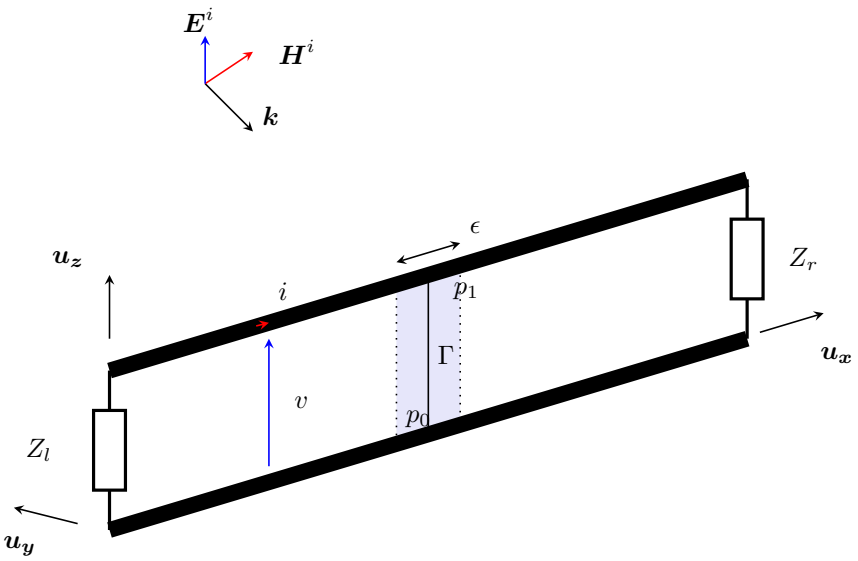


FIGURE 2.9 – Représentation du couplage d'une onde électromagnétique sur une ligne de transmission

avec :

$$\widetilde{v}^s = -\int_\Gamma \widetilde{E}_z^s dz \tag{2.20a}$$

$$\widetilde{v}^i = -\int_\Gamma \widetilde{E}_z^i dz \tag{2.20b}$$

Nous pouvons introduire la grandeur $\widetilde{v}$ dans la première équation de Taylor.

$$\partial_x \widetilde{v}^s + j\omega L\widetilde{I} = \widetilde{v}_s + \int_\Gamma \partial_x \widetilde{E}_z^i dz = \widetilde{v}_{s2} \tag{2.21}$$

Le second membre de cette équation peut être écrit comme la limite suivante où $\widetilde{v}_s$ est remplacée par sa valeur (Eq. 2.18a) :

$$\widetilde{v}_{s2} = \lim_{\epsilon \to 0} \int_{x-\epsilon/2}^{l+\epsilon/2} \left(-j\omega\mu_0 \int_\Gamma \widetilde{H}_y^i dz + \int_\Gamma \partial_x \widetilde{E}_z^i dz \right) d\tau \tag{2.22}$$

Nous utilisons le développement du rotationnel du champ électrique projeté sur le vecteur normal à la surface $\Gamma \times \epsilon$ noté $\boldsymbol{u_y}$

$$\partial_x E_z^i = \left(\boldsymbol{\nabla} \wedge \boldsymbol{E}^i \right) \cdot \boldsymbol{u_y} + \partial_z E_x^i \tag{2.23}$$

que nous introduisons dans le calcul du terme source $\widetilde{v}_{s2}$.

$$\begin{aligned}\widetilde{v}_{s2} &= \lim_{\epsilon\to 0}\int_{x-\epsilon/2}^{x+\epsilon/2}\int_{\Gamma}\left(-j\omega\mu_0\widetilde{H}_y^i dz + \left(\boldsymbol{\nabla}\wedge\boldsymbol{E}^i\right)\cdot\boldsymbol{u_y}dz\right)dx \\ &\quad + \lim_{\epsilon\to 0}\int_{x-\epsilon/2}^{x+\epsilon/2}\int_{\Gamma}\partial_z\widetilde{E}_x^i dzdx \end{aligned} \tag{2.24}$$

Le premier terme est nul parce qu'il doit satisfaire l'équation de Faraday. Le champ magnétique est à chaque instant conservatif ($\oint_{\mathcal{S}} Hds = 0$).

La tension $\widetilde{v}_{s2}$ devient donc :

$$\begin{aligned}\widetilde{v}_{s2} &= \lim_{\epsilon\to 0}\int_{x-\epsilon/2}^{x+\epsilon/2}\int_{\Gamma}\partial_z\widetilde{E}_x^i dzdx \\ &= \lim_{\epsilon\to 0}\int_{x-\epsilon/2}^{x+\epsilon/2}\widetilde{E}_x^i(p_1) - \widetilde{E}_x^i(p_0)dx \\ &= \widetilde{E}_x^i(p_1) - \widetilde{E}_x^i(p_0)\end{aligned} \tag{2.25}$$

La seconde équation de Agrawal dérive également de la seconde équation de Taylor (Eq. 2.17b) dans laquelle nous introduisons les tensions incidente et propagée (Eq. 2.19).

$$\partial_x\widetilde{i} + j\omega C\widetilde{v}^s = -j\omega C\widetilde{v}^i + \widetilde{i}_s \tag{2.26}$$

En exploitant la définition de la tension incidente (Eq. 2.19) et de la source de courant distribuée de Taylor (Eq. 2.18b), nous avons :

$$\partial_x\widetilde{i} + j\omega C\widetilde{v}^s = 0 \tag{2.27}$$

Le modèle de Taylor traite donc uniquement de la tension diffractée $\widetilde{v}^s$. Cependant, il est possible de déterminer la tension totale en ajoutant la tension incidente donnée par la relation (Eq. 2.19). Dans tous les cas, si la ligne de transmission est refermée à ses extrémités par un circuit, nous devons considérer cette tension incidente comme une source de tension additive qui définit les conditions aux limites du modèle de Taylor. Il est important de se souvenir que, dans le modèle de Taylor, la tension en bout de ligne doit être la tension totale qui peut être la tension diffractée dans certains cas de figure. Cette condition permet de fermer la boucle de courant formée par la ligne, ce qui permet de prendre implicitement en compte le flux du champ magnétique à travers la surface, définie par cette boucle. Cette propriété donne la consistance du modèle de Agrawal par rapport au modèle de Taylor. En définitive, nous pouvons donc synthétiser le modèle de Taylor comme suit :

– Équations sur la ligne :

$$\partial_x\widetilde{v}^s + j\omega L\widetilde{i} = \widetilde{E}_x^i(p_1) - \widetilde{E}_x^i(p_0) \tag{2.28a}$$

$$\partial_x\widetilde{i} + j\omega C\widetilde{v}^s = 0 \tag{2.28b}$$

– Condition aux limites dans le cas où la ligne est chargée par un circuit :

$$\widetilde{v}(0) = Z_l i(0) + \int_\Gamma \widetilde{E}_z^s(0) dz \tag{2.29a}$$

$$\widetilde{V}(L) = Z_r i(L) + \int_\Gamma \widetilde{E}_z^s(L) dz \tag{2.29b}$$

– Calcul de la tension totale :

$$\widetilde{v}(x) = \widetilde{v}^s(x) - \int_\Gamma \widetilde{E}_z^i(x) dz \tag{2.30}$$

2.2.4 Multi conducteurs - problématique du faisceau de câbles et des domaines internes

Au début de ce chapitre, nous avons présenté des notions générales sur les lignes de transmission. Dans la suite, nous avons étudié le cas particulier des lignes de transmission au-dessus d'un sol de conductivité finie. Dans cette section, nous allons nous intéresser aux cas particuliers des lignes multi-conducteurs et des blindages. Ces configurations représentent la plupart des lignes de transmission présentes dans les systèmes réels (exemple figure 2.10).



(a) (b) (c)

FIGURE 2.10 – Exemple de sections transversales, câble ethernet (a), avec blindage (b) ou domestique (c).

Le faisceau de câbles est composé d'un ensemble de conducteurs regroupés, qui est souvent référencé par rapport à une masse. Cette masse peut-être définie par un plan de masse ou un blindage.

Mode commun - mode différentiel : Les faisceaux de câbles ne présentent pas de problème numérique particulier. En définitive, un faisceau à n conducteurs en regard d'une référence, plan de masse par exemple n'est qu'une ligne de transmission à $n+1$ conducteurs. De manière pratique, pour implémenter une ligne de transmission à plusieurs conducteurs nous utilisons des termes matriciels au lieu de termes scalaires.

Le comportement des faisceaux de câbles est décrit par deux notions qui sont d'une part, le mode commun et d'autre part, le mode différentiel. Le mode commun est souvent appréhendé par un courant i_c, dit de mode commun, défini par le courant de retour qui circule dans le conducteur de référence ($i_c = \sum_{j=1}^{n}(i_j)$). A l'inverse, le mode différentiel est souvent représenté par la différence de potentiel qui existe entre deux conducteurs. En définitive, nous pouvons voir le mode commun comme le comportement global d'un faisceau de câbles par rapport à une référence, alors que le mode différentiel représente le comportement d'un conducteur par rapport à un autre.

En général, le couplage d'un faisceau de câbles avec un champ électromagnétique incident se fait sur le mode commun. Le mode différentiel quant à lui est une quantité interne au faisceau. En première approximation le mode différentiel ne se couple pas avec le champ électromagnétique extérieur car les effets des deux conducteurs se compensent (exemple d'une bifilaire).

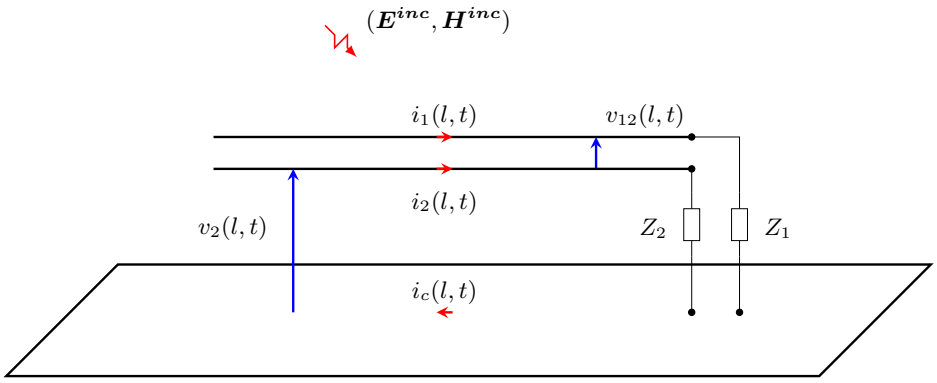


FIGURE 2.11 – Exemple d'une bifilaire au-dessus d'un plan de masse.

De manière générale dans l'industrie, les caractéristiques et la géométrie des faisceaux de câbles sont relativement bien connues. De plus et dans la grande majorité des cas, les conducteurs sont utilisés en mode différentiel (coaxial de communication, paire torsadée, ou liaison énergie, ...). L'utilisation du mode différentiel, comme nous l'avons déjà dit, permet de s'affranchir des problèmes de couplage avec le champ électromagnétique. En revanche, le mode commun d'un faisceau de câbles peut être excité par un champ externe.

On peut cependant noter que dans une paire fonctionnelle, les deux conducteurs ont des rôles différents. Par exemple pour une liaison énergie, nous avons un conducteur d'alimentation Vcc et un conducteur GND de référence. A leurs extrémités, ces deux conducteurs voient une impédance différente par rapport à la référence globale. Pour l'alimentation (Vcc), nous avons par exemple $Z_{Vcc} = 50\Omega$ alors que pour la masse cette impédance est très petite $Z_{GND} \approx 0$. Nous avons donc un déséquilibre de charge par rapport à la référence globale (et donc au mode commun) entre les deux conducteurs. Ce déséquilibre va faire remonter une partie du mode commun dans le mode différentiel. Ce comportement est appelé la réjection de mode commun. Ces courants sont très importants car ils vont directement interagir avec les équipements terminaux. Cette perturbation peut entraîner une défaillance voire une destruction des équipements.

L'étude du mode commun et du mode différentiel se révèle un enjeu important pour étudier les perturbations induites sur un faisceau de câbles. L'étude du mode commun permet d'étudier le couplage d'un faisceau avec le champ électromagnétique environnant. L'étude du mode différentiel permet entre autre d'étudier le comportement de la réjection du mode commun.

Nous verrons dans la suite de ce document (Chapitre 3 page 99) une stratégie d'hybridation qui permet d'étudier le comportement des faisceaux de câbles du mode commun au mode différentiel.

L'impédance et l'admittance de transfert [30, 31] : Nous venons de voir dans le paragraphe précédent que le mode différentiel dépendait peu du champ électromagnétique environnant. Si dans la majorité des cas, cette approximation est raisonnable, elle n'en reste pas moins une approximation. Dans cette section, nous allons notamment présenter

l'impédance et l'admittance de transfert qui sont deux quantités permettant de modéliser l'influence du champ électromagnétique sur le mode différentiel [30]. En général, ces quantités se définissent par rapport à une référence locale liée aux différents niveaux de blindages existants dans la ligne. Pour simplifier l'écriture finale du problème tenant compte de l'impédance et de l'admittance de transfert, nous introduirons un changement de variables qui permet de référencer tous les conducteurs par rapport à la référence globale [31]. Ce choix permet en particulier de simplifier la prise en compte des impédances terminales.

Avant de présenter l'impédance et l'admittance de transfert, nous allons définir deux autres notions importantes qui sont, la section transversale et le blindage.

La section transversale d'une ligne de transmission est la géométrie en deux dimensions de la ligne de transmission dans le plan orthogonal au vecteur de propagation (Fig. 2.12). La section transversale est très importante car elle définit les paramètres électriques de la ligne de transmission (R, L, C et G). Dans le cas de matériaux sans perte, il est relativement simple [32, 33, 34] de déterminer ces paramètres.

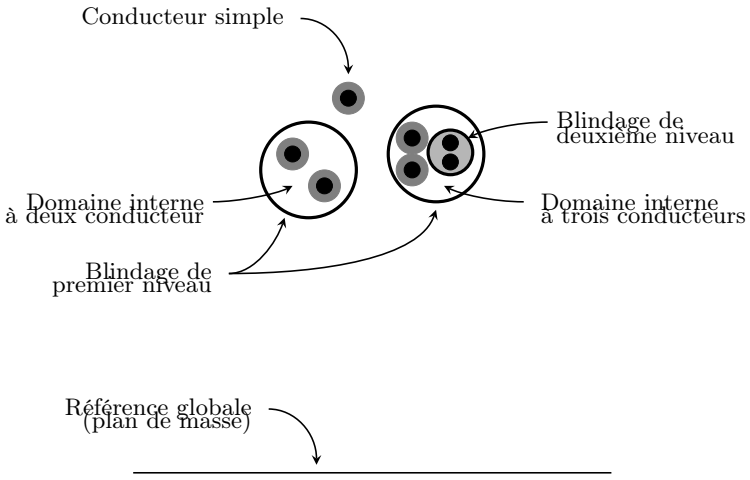


FIGURE 2.12 – Représentation d'un faisceau de câbles à plusieurs niveaux de blindage.

Nous pouvons distinguer deux types de section transversale. Les sections transversales avec ou sans blindage. Par définition, un blindage est un conducteur creux qui tente d'isoler les conducteurs d'un domaine interne de l'extérieur (domaine externe). Le blindage définit donc la limite entre le domaine interne et le domaine externe. Dans le domaine externe le blindage peut être modélisé comme un simple conducteur, alors que dans le domaine interne, celui-ci est vu comme le conducteur de référence. Ces deux domaines sont liés par l'impédance et l'admittance de transfert.

Notons que la notion de blindage peut être à plusieurs niveaux, c'est-à-dire que l'un des conducteurs à l'intérieur d'un blindage peut être le blindage d'un deuxième niveau de blindage et ainsi de suite. Dans ce cas figure on procède par récurrence pour calculer les paramètres de la ligne de transmission. Pour cela, nous calculons tout d'abord, les paramètres de la ligne externe et de toutes les lignes internes indépendamment les uns des autres. Puis nous construisons les paramètres de toute la ligne en liant par l'impédance et l'admittance de transfert les domaines interne et externe de chaque ligne comme indiqué sur la figure 2.13.

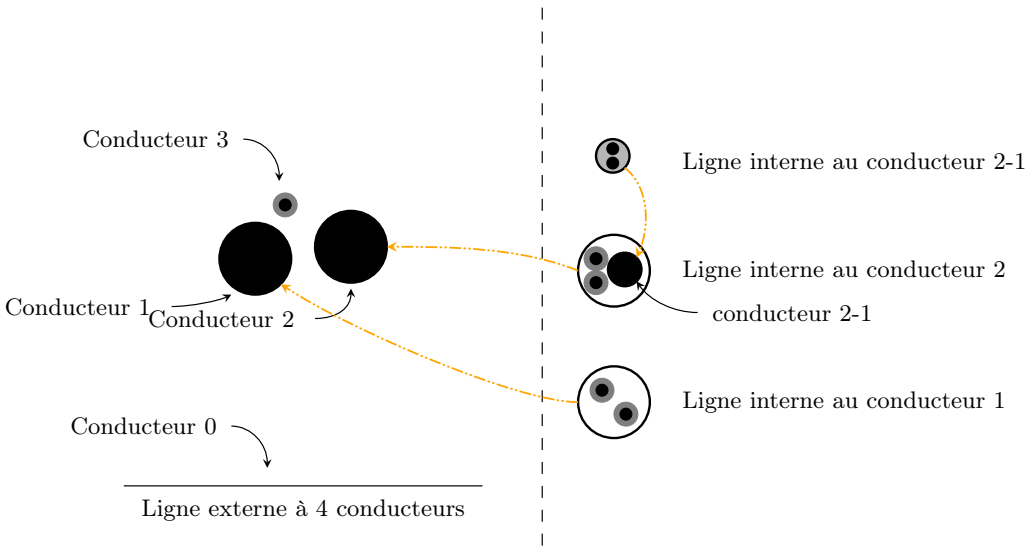


FIGURE 2.13 – Décomposition de la ligne de transmission par niveaux de blindage

Note : Dans le paragraphe précédent, nous avons vu que le mode différentiel était peu perturbé par le champ électromagnétique extérieur car les deux conducteurs se compensent. En revanche, le mode commun peut être excité par le champ environnant. Ainsi, ajouter un blindage revient à créer un mode différentiel particulier entre le faisceau de câbles et le blindage. Le faisceau de câble a la même interaction avec le champ électromagnétique mais dans cette configuration cette interaction est compensée par le courant dans le blindage.

Par ailleurs, le couplage entre le domaine interne et externe traduit la pénétration des champs électrique et magnétique à travers le blindage, qui ne peut être techniquement parfait. Ces perturbations peuvent être traduites en une perturbation sur le courant et la tension.

L'impédance de transfert Z_t (Eq. 2.31a) traduit la perturbation sur la tension différentielle v_{int} induite par le courant de blindage i_{ext} (courant de la ligne de transmission externe). En revanche, l'admittance de transfert Y_t (Eq. 2.31b) traduit la perturbation, introduite par la pénétration du champ électrique. Elle se matérialise par un courant de blindage i_{int} (courant de retour de la ligne de transmission interne), quand la tension totale v_t est excitée. Ces perturbations peuvent être vues comme des termes de sources distribuées (v_{int}^{inc} et i_{int}^{inc}) sur les conducteurs internes. Nous les définissons comme suit :

$$\partial_l \widetilde{v}_{int}^{inc} = Z_t \widetilde{i}_t \tag{2.31a}$$

$$\partial_l \widetilde{i}_{int}^{inc} = -Y_t \widetilde{v}_t \tag{2.31b}$$

Les équations de ligne de transmission (Eq. 2.1) associées aux équations de couplage (Eq. 2.31) nous permettent d'écrire sous la forme matricielle l'équation de ligne de transmission :

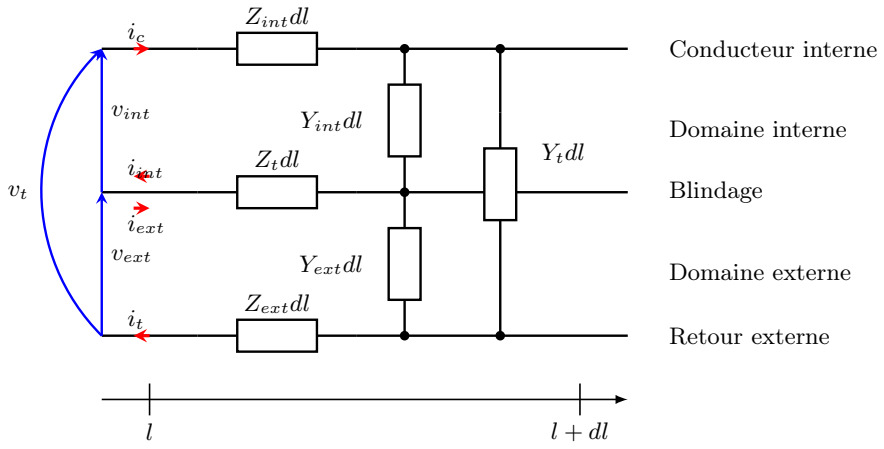


FIGURE 2.14 – Représentation sous la forme d'un schéma électrique du couplage entre les domaines internes et externes d'une ligne de transmission blindée

$$\underbrace{\partial_t \begin{pmatrix} \widetilde{\boldsymbol{v}}_{ext} \\ \widetilde{\boldsymbol{v}}^1_{int} \\ \vdots \\ \widetilde{\boldsymbol{v}}^n_{int} \end{pmatrix}}_{\widetilde{\boldsymbol{v}}_n} = \underbrace{\begin{bmatrix} \boldsymbol{Z}_{ext} & -(\boldsymbol{Z}^1_t)^T & \cdots & -(\boldsymbol{Z}^n_t)^T \\ -(\boldsymbol{Z}^1_t) & \boldsymbol{Z}^1_{int} & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \boldsymbol{0} & \ddots & \boldsymbol{0} \\ -(\boldsymbol{Z}^n_t) & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Z}^n_{int} \end{bmatrix}}_{\boldsymbol{Z}_n} \cdot \underbrace{\begin{pmatrix} \widetilde{\boldsymbol{i}}_{ext} \\ \widetilde{\boldsymbol{i}}^1_{int} \\ \vdots \\ \widetilde{\boldsymbol{i}}^n_{int} \end{pmatrix}}_{\widetilde{\boldsymbol{i}}_n} \tag{2.32}$$

où $\boldsymbol{i}^k_{int}$ correspond aux courants des conducteurs à l'intérieur du blindage k en considérant celui-ci comme référence. La notation i^k_{ext} représente, quant à elle, le courant d'un conducteur (ou blindage) par rapport à la ligne de transmission externe. Cette équation (2.32) peut être mise sous une forme condensée $\widetilde{\boldsymbol{v}}_n = \widetilde{\boldsymbol{Z}}_n \widetilde{\boldsymbol{i}}_n$ où $\widetilde{\boldsymbol{v}}_n$ et $\widetilde{\boldsymbol{i}}_n$ sont les tensions et courants dans le cas où chaque blindage est la référence de son domaine interne et $\boldsymbol{Z}_n$ l'impédance de la ligne du modèle de Heaviside. L'indice $_n$ dans cette notation signifie : « à n référence ».

Avec les mêmes notations, nous avons la deuxième équation de ligne :

$$\underbrace{\partial_t \begin{pmatrix} \widetilde{\boldsymbol{i}}_{ext} \\ \widetilde{\boldsymbol{i}}^1_{int} \\ \vdots \\ \widetilde{\boldsymbol{i}}^n_{int} \end{pmatrix}}_{\widetilde{\boldsymbol{i}}_n} = \underbrace{\begin{bmatrix} \boldsymbol{Y}_{ext} & (\boldsymbol{Y}^1_t)^T & \cdots & (\boldsymbol{Y}^n_t)^T \\ (\boldsymbol{Y}^1_t) & \boldsymbol{Y}^1_{int} & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \boldsymbol{0} & \ddots & \boldsymbol{0} \\ (\boldsymbol{Y}^n_t) & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Y}^n_{int} \end{bmatrix}}_{\boldsymbol{Y}_n} \cdot \underbrace{\begin{pmatrix} \widetilde{\boldsymbol{v}}_{ext} \\ \widetilde{\boldsymbol{v}}^1_{int} \\ \vdots \\ \widetilde{\boldsymbol{v}}^n_{int} \end{pmatrix}}_{\widetilde{\boldsymbol{v}}_n} \tag{2.33}$$

Dans le cas où il y a plusieurs niveaux de blindage, nous construisons le système de manière récursive en partant du domaine le plus interne.

Pour simplifier l'écrire de notre modèle nous pouvons utiliser ces relations de couplage entre le domaine interne et externe pour construire une autre équation de ligne de transmission, où tous les conducteurs sont référencés à la même référence. Ainsi, les paramètres

de ligne avec une seule référence sont construits à partir du changement de variables sur les courants et les tensions. Pour le cas du courant, ce changement de variables est donné par les relations suivantes :

$$\begin{cases} \widetilde{i}_b^k & = & \widetilde{i}_{ext}^k - \sum_{m=1}^{M_k} \widetilde{i}_{int}^{k(m)} \\ \widetilde{i}_c^{m(k)} & = & \widetilde{i}_{int}^{m(k)} \end{cases} \tag{2.34}$$

Où $i_{int}^{k(m)}$ est le courant du conducteur m à l'intérieur du blindage k, i_b représente le courant sur les conducteurs (ou blindages) du domaine externe dans le formalisme à une référence et i_c le courant sur les conducteurs du domaine interne dans ce même formalisme.

Pour la différence de potentiel, nous avons les relations suivantes :

$$\begin{cases} \widetilde{v}_b^k & = & \widetilde{v}_{ext}^k \\ \widetilde{v}_c^{m(k)} & = & \widetilde{v}_{ext}^k + \widetilde{v}_{int}^{m(k)} \end{cases} \tag{2.35}$$

Nous obtenons donc la forme matricielle suivante :

$$\widetilde{\boldsymbol{i}}_1 = \boldsymbol{P_{int}}\widetilde{\boldsymbol{i}}_n \tag{2.36a}$$

$$\widetilde{\boldsymbol{v}}_1 = \boldsymbol{P_v}\widetilde{\boldsymbol{v}}_n \tag{2.36b}$$

Avec :

$\boldsymbol{x}_n = (x_{ext}, x_{int})^T$, le vecteur d'inconnue x dans le contexte à n références, $\boldsymbol{x}_1 = (x_b, x_c)^T$ est le vecteur d'inconnue x dans le contexte à 1 référence, $\boldsymbol{P_i}$ et $\boldsymbol{P_v}$ sont les matrices de passage définies comme suit :

$$\boldsymbol{P_i} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{\delta} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \quad \boldsymbol{P_v} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{\delta}^T & \boldsymbol{I} \end{bmatrix} \tag{2.37}$$

Où $\delta(m,k)$ vaut 1 si le conducteur k est un conducteur interne au blindage m, sinon 0. Nous pouvons remarquer la relation suivante qui simplifiera le calcul par la suite : $\boldsymbol{P_i}^{-1} = \boldsymbol{P_v}^T$ et $\boldsymbol{P_v}^{-1} = \boldsymbol{P_i}^T$.

Nous allons maintenant déterminer la matrice impédance de ligne de transmission à une référence en exploitant l'équation de lignes $\widetilde{\boldsymbol{v}}_n = \boldsymbol{Z}_n\widetilde{\boldsymbol{i}}_n$ et les relations de passage vues précédemment. Ce qui nous permet d'écrire très simplement :

$$\widetilde{\boldsymbol{v}}_1 = \boldsymbol{P_v}\widetilde{\boldsymbol{v}}_n = \boldsymbol{P_v}\boldsymbol{Z}_n\widetilde{\boldsymbol{i}}_n = \boldsymbol{P_v}\boldsymbol{Z}_n\boldsymbol{P_i}^{-1}\boldsymbol{P_i}\widetilde{\boldsymbol{i}}_n = \boldsymbol{P_v}\boldsymbol{Z}_n\boldsymbol{P_i}^{-1}\widetilde{\boldsymbol{i}}_1 = \boldsymbol{P_v}\boldsymbol{Z}_n\boldsymbol{P_v}^T\widetilde{\boldsymbol{i}}_1 = \boldsymbol{Z}_1\widetilde{\boldsymbol{i}}_1 \tag{2.38}$$

En faisant de même pour la matrice admittance, nous pouvons écrire la relation qui lie le formalisme à n références à celui ramené à une référence.

$$\boldsymbol{Z}_1 = \boldsymbol{P_v}\boldsymbol{Z}_n\boldsymbol{P_v}^T \tag{2.39a}$$

$$\boldsymbol{Y}_1 = \boldsymbol{P_i}\boldsymbol{Y}_n\boldsymbol{P_i}^T \tag{2.39b}$$

Au final, nous remarquons que quelque soit la complexité de la ligne considérée, nous pouvons toujours nous ramener au modèle de ligne de transmission suivant :

$$\partial_l \widetilde{\boldsymbol{v}} = -\boldsymbol{Z}\widetilde{\boldsymbol{i}} \tag{2.40a}$$

$$\partial_l \widetilde{\boldsymbol{i}} = -\boldsymbol{Y}\widetilde{\boldsymbol{v}} \tag{2.40b}$$

Dans la suite du mémoire nous traiterons donc indifféremment le cas monofilaire et multifilaire.

Note : Nature fréquentielle des matrices $\boldsymbol{Z}_1$ et $\boldsymbol{Y}_1$.

Nous pourrions être tentés de croire que le blindage introduit une dépendance en fréquence sur les paramètres de la ligne de transmission, ce qui pourrait être difficile à considérer en temporel. Cependant, si nous faisons l'hypothèse du modèle de Heaviside ($a = a_r + j a_l \omega$) pour les paramètres de la ligne et pour l'impédance de transfert, alors le formalisme à une référence peut également se mettre sous la forme du modèle de Heaviside. Nous pouvons montrer cette propriété dans le cas de l'impédance par exemple.

Supposons que $\boldsymbol{Z}_n = \boldsymbol{R}_n + j\omega \boldsymbol{L}_n$ avec $\boldsymbol{R}_n$ et $\boldsymbol{L}_n$ réels et constants. Dans ce cas, nous pouvons écrire :

$$\begin{aligned} \boldsymbol{Z}_1 &= \boldsymbol{P_v} \boldsymbol{Z}_n \boldsymbol{P_v}^T \\ &= \boldsymbol{P_v} \boldsymbol{R}_n \boldsymbol{P_v}^T + j\omega \boldsymbol{P_v} \boldsymbol{L}_n \boldsymbol{P_v}^T \\ &= \boldsymbol{R}_1 + j\omega \boldsymbol{L}_1 \end{aligned} \tag{2.41}$$

Comme $\boldsymbol{P_v}$, $\boldsymbol{R}_n$ et $\boldsymbol{L}_n$ sont réels, alors $\boldsymbol{R}_1$ et $\boldsymbol{L}_1$ le sont aussi.

Tous les blindages ne peuvent pas être mis sous une forme simple du type $Z_t = R_t + j L_t \omega$. Cependant, un grand nombre de problèmes peuvent être traités ou approchés de manière satisfaisante par cette approximation.

2.3 Résolution numérique des équations de ligne dans le domaine temporel

Dans cette section, nous allons détailler le modèle numérique d'approximation des équations de ligne de transmission que nous allons utiliser. Les modèles et techniques implémentés dans celui-ci ont été choisis de façon à pouvoir répondre par la suite, au besoin d'hybridation avec un outil FDTD-3D. C'est pourquoi, nous avons choisi une résolution des équations de ligne dans le domaine temporel, et nous proposons d'utiliser un schéma explicite en temps et en espace de type « leap-frog ».

Nous décrivons tout d'abord le modèle de ligne choisi, ainsi que le modèle numérique. Nous étudions ensuite les propriétés du schéma numérique. Enfin nous montrons un résultat de validation de notre modèle numérique sur un exemple générique.

2.3.1 Approximation numérique - méthode MTL

a. Discrétisation des équations de ligne de transmission - Formulation temporelle

Le modèle de Heaviside, présenté précédemment, s'appuie sur les équations de Maxwell et permet de considérer le champ électromagnétique guidé le long d'une ligne sous la forme d'une onde électrique (tension v et courant i).

Rappelons que Heaviside (Sec. 2.2 page 33) propose de décomposer une ligne en segments élémentaires (Fig. 2.15), sur lesquels les paramètres R, L, G et C représentent les paramètres linéiques de la ligne de transmission.

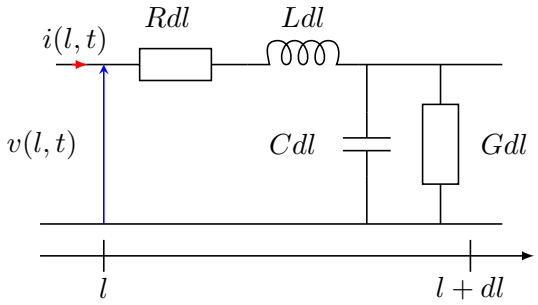


FIGURE 2.15 – Schéma équivalent de Heaviside pour un segment de ligne.

Les grandeurs physiques du modèle sont les suivantes :

L'intérêt de ce modèle est de pouvoir directement utiliser la théorie des circuits pour formaliser l'équation d'onde, ce qui nous permet d'écrire très simplement le long de la ligne :

$$\partial_l \widetilde{v} = -(R + jL\omega) \cdot \widetilde{i}(l)dl \tag{2.42a}$$

$$\partial_l \widetilde{i} = -(G + jC\omega) \cdot \widetilde{v}(l)dl \tag{2.42b}$$

R	Résistance par unité de longueur (Ω/m)
L	Inductance par unité de longueur (H/m)
G	Admittance par unité de longueur (S/m)
C	Capacité par unité de longueur (F/m)
dt	Pas de discrétisation temporelle (s)
dl	Pas de discrétisation spatiale (m)

TABLE 2.6 – Grandeurs physiques du modèle de Heaviside.

Nous pouvons aussi écrire cette relation dans le domaine temporel par la transformée inverse de Fourier.

$$-\partial_l v = Ri + L\partial_t i \tag{2.43a}$$
$$-\partial_l i = Gv + C\partial_t v \tag{2.43b}$$

Par la suite, sans ambiguïté nous n'écrirons plus la dépendance en espace l et en temps t des grandeurs physiques v et i pour alléger les notations.

Nous avons choisi de traiter le problème en utilisant l'approximation des différences finies dans le domaine temporel et spatial. La discrétisation des inconnues se fait suivant le schéma « leap-frog » (Fig. 2.16) en plaçant le courant au milieu des segments et les tensions aux extrémités :

$$V|_p^n = v(p \cdot \Delta l, n \cdot \Delta t) \tag{2.44a}$$
$$I|_{p+1/2}^{n+1/2} = i\left((p+1/2) \cdot \Delta l, (n+1/2) \cdot \Delta t\right) \tag{2.44b}$$

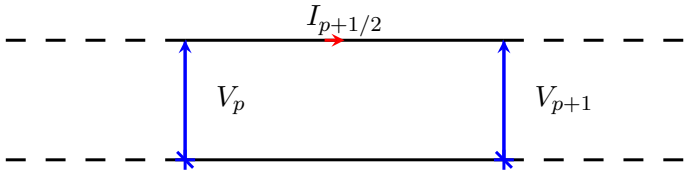


FIGURE 2.16 – Discrétisation spatiale des équations de ligne de transmission.

Si nous considérons que les paramètres de la ligne de transmission R, L, C et G sont constants tout le long de la ligne, nous avons alors sur chaque segment :

$$I|_{p+1/2}^{n+1/2} = -Q_i \cdot\ I|_{p+1/2}^{n-1/2} + Q_v \cdot \left(V|_{p+1}^n - V|_p^n\right) \tag{2.45a}$$
$$V|_p^{n+1} = -K_v \cdot\ V|_p^n + K_i \cdot \left(I|_{p+1/2}^{n+1/2} - I|_{p-1/2}^{n+1/2}\right) \tag{2.45b}$$

avec :

$$Q_i = \left(\frac{R\Delta t - 2L}{R\Delta t + 2L}\right),\ Q_v = \frac{2}{R\Delta t + 2L}\frac{\Delta t}{\Delta l}$$
$$K_v = \left(\frac{G\Delta t - 2C}{G\Delta t + 2C}\right),\ K_i = \frac{2}{G\Delta t + 2C}\frac{\Delta t}{\Delta l}$$

Dans le cas multi-filaire nous obtenons les mêmes équations où R, L, C et G sont des matrices et i, v des vecteurs.

$$\boldsymbol{I}\big|_{p+1/2}^{n+1/2}(\Delta t\boldsymbol{R} + 2\boldsymbol{L}) = -\boldsymbol{I}\big|_{p+1/2}^{n-1/2}(\Delta t\boldsymbol{R} - 2\boldsymbol{L}) + \left(\boldsymbol{V}\big|_{p+1}^{n} - \boldsymbol{V}\big|_{p}^{n}\right)\frac{2\Delta t}{\Delta l} \tag{2.46a}$$
$$\boldsymbol{V}\big|_{p}^{n+1}(\Delta t\boldsymbol{G} + 2\boldsymbol{C}) = -\boldsymbol{V}\big|_{p}^{n}(\Delta t\boldsymbol{G} - 2\boldsymbol{C}) + \left(\boldsymbol{I}\big|_{p+1/2}^{n+1/2} - \boldsymbol{I}\big|_{p-1/2}^{n+1/2}\right)\frac{2\Delta t}{\Delta l} \tag{2.46b}$$

b. Propriétés du schéma numérique

Dans cette section, nous allons nous pencher sur les propriétés du schéma numérique. Elle a en particulier pour but de donner des éléments justifiant du domaine de validité et d'applicabilité de la méthode. Cette analyse est similaire à celle faite pour les FDTD dans le cas 3D dans [35, 36] ou encore [37, Chapitre17].

Etude de la stabilité. Nous avons fait cette analyse en utilisant le critère de Von Neumann [38] sur une ligne de transmission sans perte. La première étape pour établir le critère de Von Neumann est de formaliser le système linéaire explicite. Pour écrire ce système, nous repartons des équations de lignes discrétisées que nous simplifions dans le cas où nous avons des lignes sans perte ($R = 0$ et $G = 0$).

$$I\big|_{p+1/2}^{n+1/2} = I\big|_{p+1/2}^{n-1/2} - \frac{\Delta t}{L\Delta l}\left(V\big|_{p+1}^{n} - V\big|_{p}^{n}\right) \tag{2.47a}$$
$$V\big|_{p}^{n+1} = V\big|_{p}^{n} - \frac{\Delta t}{C\Delta l}\left(I\big|_{p+1/2}^{n+1/2} - I\big|_{p-1/2}^{n+1/2}\right) \tag{2.47b}$$

En introduisant (2.47a) dans (2.47b), nous obtenons le système suivant :

$$I\big|_{p+1/2}^{n+1/2} = I\big|_{p+1/2}^{n-1/2} - \frac{\Delta t}{L\Delta l}\left(V\big|_{p+1}^{n} - V\big|_{p}^{n}\right) \tag{2.48a}$$
$$V\big|_{p}^{n+1} = V\big|_{p}^{n} + \left(\frac{\nu\Delta t}{\Delta l}\right)^2\left(V\big|_{p+1}^{n} - 2\,V\big|_{p}^{n} + V\big|_{p-1}^{n}\right) - \frac{\Delta t}{C\Delta l}\left(I\big|_{p+1/2}^{n-1/2} - I\big|_{p-1/2}^{n-1/2}\right) \tag{2.48b}$$

Nous avons maintenant un système linéaire, où $\nu = 1/\sqrt{LC}$ définit la vitesse de propagation de l'onde sur la ligne sans perte. En utilisant le critère de Von Neumann, nous introduisons dans le système (Eq. 2.48) une solution modale définie par $\boldsymbol{u}^n = (\widetilde{V}\big|^{n} e^{ikl}, \widetilde{I}\big|^{n-1/2} e^{ik(l+\Delta l/2)})^T$.

Nous pouvons alors écrire la relation de récurrence suivante $\boldsymbol{u}^{n+1} = \boldsymbol{G} \cdot \boldsymbol{u}^n$. Avec $\boldsymbol{G}$ la matrice d'amplification donnée par :

$$\boldsymbol{G} = \begin{bmatrix} 1 - \left(\frac{\nu\Delta t}{\Delta l}\right)^2 \left(e^{jk\Delta l} - 2 + e^{-jk\Delta l}\right) & -\frac{\Delta t}{C\Delta l}\left(1 - e^{-jk\Delta l}\right) \\ -\frac{\Delta t}{L\Delta l}\left(e^{jk\Delta l} - 1\right) & 1 \end{bmatrix} \tag{2.49}$$

Pour s'assurer que le schéma est stable il suffit de garantir que $\|\boldsymbol{G}\| \leq 1$. Pour cela nous devons calculer les valeurs propres de la matrice et vérifier que leur module est inférieur à 1. Ces valeurs propres sont déterminées par les racines du polynôme caractéristique $P(g) = \|G - gI\|$ de $\boldsymbol{G}$, où I définit la matrice identité. Posons $\eta = \frac{\nu\Delta t}{\Delta l}$, nous cherchons alors g tel que :

$$\begin{aligned} & P(g) = 0 \\ \Rightarrow \quad & (1-g)\left(1 - \eta^2\left(e^{jk\Delta l} - 2 + e^{-jk\Delta l}\right) - g\right) + \eta^2\left(1 - e^{-jk\Delta l}\right)\left(e^{jk\Delta l} - 1\right) = 0 \\ \Rightarrow \quad & g^2 - 2g\left(1 + \eta^2\left(\cos(k\Delta l) - 1\right)\right) + 1 = 0 \end{aligned} \tag{2.50}$$

Il en découle deux valeurs propres définies par :

$$g_\pm = \left(1 + \eta^2\left(\cos(k\Delta l) - 1\right)\right) \pm \sqrt{\left(1 + \eta^2\left(\cos(k\Delta l) - 1\right)\right)^2 - 1} \tag{2.51}$$

Nous pouvons maintenant étudier la stabilité du schéma. Pour faire cette étude, nous allons considérer les cas de figure suivants :

- Cas où $\eta > 1$:
  Nous avons pour $k\Delta l = \pi$, $g_\pm = 1 - 2\eta^2 \pm \sqrt{\left(1 - 2\eta^2\right)^2 - 1}$ que l'on peut aussi écrire $g_\pm = 1 - 2\eta^2\left(1 \mp \sqrt{1 - \frac{1}{\eta^2}}\right)$. Comme $\eta > 1 \Rightarrow \frac{1}{\eta^2} < 1 \Rightarrow 1 + \sqrt{1 - \frac{1}{\eta^2}} > 1 \Rightarrow g_+ < 1 - 2\eta^2 \Rightarrow \|g_+\| > 1$. Il existe donc une valeur propre de module strictement supérieure à 1 et donc le schéma est instable dans ces conditions.
- Cas où $\eta \leq 1$:
  En introduisant $x = 1 + \eta^2\left(\cos(k\Delta l) - 1\right)$, nous avons $-1 < x < 1$ et $-1 < x^2 - 1 < 0$. Ce qui nous permet d'écrire $g_\pm = x \pm j\sqrt{1 - x^2}$, d'où $\|g_\pm\| = 1$. Ce qui implique que le schéma est stable.

En conclusion, le schéma FDTD proposé pour résoudre les équations de ligne de transmission est stable sous la condition $\eta \leq 1$, soit $\Delta t \leq \Delta l/\nu$.

Le critère de stabilité pour notre schéma FDTD appliqué aux équations de ligne de transmission est identique à une CFL près à celui du schéma FDTD-3D. Cette propriété est intéressante dans l'objectif d'hybrider ces deux méthodes entre elles. Nous pourrons ainsi avoir des discrétisations temporelle et spatiale similaires dans les deux méthodes.

Etude de la dispersion du schéma. La dispersion est une propriété importante car elle permet de quantifier la capacité de la méthode à correctement représenter la propagation de chaque fréquence. Nous pouvons définir la dispersion comme la différence entre la fréquence

théorique et la fréquence numérique. Pour étudier la dispersion, nous repartons des mêmes conditions que pour établir la stabilité ; c'est-à-dire que nous utilisons les équations de lignes discrétisées (Eq. 2.45) simplifiées en considérant le cas sans perte.

La première étape consiste là encore à construire un système linéaire explicite mais cette fois « indépendant » en espace et en temps. Ce système est obtenu à partir de la discrétisation des EDP de la ligne de transmission sans perte.

$$I|_{p+1/2}^{n+1/2} = I|_{p+1/2}^{n-1/2} - \frac{\Delta t}{L\Delta l}\left(V|_{p+1}^{n} - V|_{p}^{n}\right) \tag{2.52a}$$

$$V|_{p}^{n+1} = V|_{p}^{n} - \frac{\Delta t}{C\Delta l}\left(I|_{p+1/2}^{n+1/2} - I|_{p-1/2}^{n+1/2}\right) \tag{2.52b}$$

Pour rendre ce système « indépendant » en espace et en temps, nous introduisons un changement de variable faisant intervenir une solution modale en espace et en temps. Ce changement de variable fait ressortir la pulsation ω et le nombre d'onde k.

$$u_p^n = \widetilde{u}e^{j(\omega t+\lambda l)} = \widetilde{u}e^{j(\omega n\Delta t+kp\Delta l)} \tag{2.53}$$

En introduisant ce changement de variables dans le système d'équation (2.52a) et (2.52b) et après simplification, nous obtenons le système linéaire suivant :

$$\begin{bmatrix} W & \Lambda/L \\ \Lambda/C & W \end{bmatrix} \cdot \begin{pmatrix} \widetilde{I} \\ \widetilde{V} \end{pmatrix} = 0 \tag{2.54}$$

avec $W = \frac{2}{\Delta t}\sin\left(\frac{\omega\Delta t}{2}\right)$,et $\Lambda = \frac{2}{\Delta l}\sin\left(\frac{k\Delta l}{2}\right)$.

Nous pouvons désormais étudier la dispersion du schéma en introduisant la notion de pulsation numérique $\breve{\omega}$ associée à une pulsation de référence ω (théorique). La dispersion est la différence entre ces deux pulsations.

Cette dispersion est étudiée à partir de la solution du système linéaire explicite (2.54) dont la solution est donnée par :

$$W^2 - \nu^2\Lambda^2 = 0 \tag{2.55}$$

avec $\nu = \sqrt{LC}$

Dans le cas où nous prenons la limite quand le pas spatial et le pas temporel tendent vers zéro, nous voyons que nous avons une dispersion nulle.

$$\lim_{\Delta l\to 0,\Delta t\to 0} \breve{W}^2 - \nu^2\Lambda^2 = \breve{\omega}^2 - \nu^2k^2 = \breve{\omega}^2 - \omega^2 = 0 \tag{2.56}$$

Dans le cas général, la pulsation numérique s'exprime comme suit, en introduisant le facteur de stabilité $\eta = \frac{\nu\Delta t}{\Delta l}$.

$$\breve{\omega} = \frac{2\nu}{\eta\Delta l}\arcsin\left|\eta\sin\left(\frac{\omega\Delta l}{2\nu}\right)\right| \tag{2.57}$$

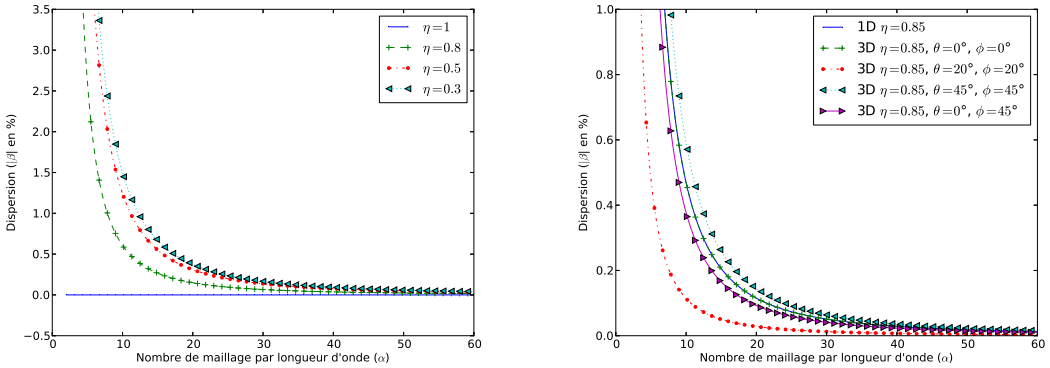
Il est relativement difficile d'interpréter cette relation. Pour simplifier celle-ci, nous introduisons le changement de variable suivant : $\Delta l = \alpha\lambda$. En général, nous choisissons α entre

1/10 et 1/100. De plus, nous avons par définition $2\pi\nu = \omega\lambda$. En introduisant ces changements de variables dans la relation précédente (Eq. 2.57), nous pouvons écrire la dispersion sous la forme d'un facteur de compression $\beta = (\breve{\omega} - \omega)/\omega$, en fonction du facteur de stabilité η et du cœfficient de discrétisation de la longueur d'onde α.

$$\beta = \frac{1}{\eta\alpha\pi} \arcsin |\eta \sin(\alpha\pi)| - 1 \tag{2.58}$$

Dans le cas particulier où nous avons $\eta = 1$ (limite de stabilité) nous n'avons pas de dispersion ($\beta = 0$).

La figure 2.17(a) nous donne la dispersion numérique ($\beta = (\breve{\omega} - \omega)/\omega$) en fonction du nombre de mailles par longueur d'onde ($1/\alpha = \lambda/\Delta l$). Nous constatons que plus nous discrétisons finement le maillage plus la dispersion diminue pour tendre vers 0.



(a) (b)

FIGURE 2.17 – Représentation de la dispersion et comparaisons au cas 3D.

La figure 2.17(b) compare la dispersion du schéma 1D de la ligne de transmission avec la dispersion de la FDTD-3D liée aux calculs des champs pour différents angles d'incidence. Cette comparaison est faite dans le cas d'une cellule cubique d'arête Δl et où $\eta = 0.85$. La dispersion 1D est la limite de la dispersion 3D quand le vecteur d'onde devient parallèle à un axe de la grille. Nous vérifions ainsi que la dispersion 3D peu est supérieure à la dispersion 1D. Outre le gain en coût calcul, l'utilisation d'une méthode de ligne de transmission va s'avérer plus précise pour la propagation des courants le long d'une ligne entre deux domaines.

$$\left(\frac{\sin(\breve{\omega}\Delta t/2)}{\Delta t}\right)^2 = c_0^2 \left[\left(\frac{\sin(k_x\Delta_x/2)}{\Delta_x}\right)^2 + \left(\frac{\sin(k_y\Delta_y/2)}{\Delta_y}\right)^2 + \left(\frac{\sin(k_z\Delta_z/2)}{\Delta_z}\right)^2\right] \tag{2.59}$$

2.3.2 Conditions aux limites et sources localisées

Nous avons vu dans la partie précédente une méthode permettant de traiter les lignes de transmission. Nous avons également donné les propriétés du schéma numérique. Dans

cette section, nous allons nous intéresser à la modélisation des conditions aux limites. Tout d'abord nous montrons comment tenir compte d'une condition de circuit ouvert, de court-circuit ou d'adaptation. Ensuite nous étudions comment prendre en compte une condition de type de nœud de courant qui permet la jonction de plusieurs lignes. Finalement, nous introduisons les conditions aux limites de type Thévenin ou Norton sur un nœud.

Avant de décrire les différentes conditions aux limites, nous faisons l'hypothèse que celles-ci sont appliquées à l'extrémité finale (droite) de la ligne. L'écriture de celles-ci sur l'extrémité initiale (gauche) de la ligne, ne pose pas plus de difficulté.

a. La condition aux limites de type court-circuit (CC)

Cette condition consiste à imposer une tension nulle en bout de ligne $V_p = 0$. En introduisant cette relation, directement dans l'équation de mise à jour du courant (Eq. 2.45a) nous obtenons :

$$I|_{p-1/2}^{n+1/2} = -Q_i \cdot\ I|_{p-1/2}^{n-1/2} + Q_v \cdot\ V|_{p-1}^{n} \tag{2.60}$$

où Q_i et Q_v sont les cœfficients de mise à jour du courant (Eq. 2.45).

b. La condition aux limites de type circuit ouvert (CO)

Cette condition consiste à imposer un courant nul en bout de ligne $I_p = 0$ (extrémité droite). De la même manière, nous pouvons introduire cette relation dans l'équation en tension (Eq. 2.66), ce qui conduit à :

$$V|_{p}^{n+1} = -K_v \cdot\ V|_{p}^{n} - 2K_i \cdot\ I|_{p-1/2}^{n+1/2} \tag{2.61}$$

où K_i et K_v sont les cœfficients de mise à jour de la tension (Eq. 2.45).

c. La condition aux limites de type adaptation

Cette condition s'écrit très simplement avec la notion d'onde. En effet, c'est une condition aux limites pour laquelle l'onde réfléchie est nulle. Imposer cette condition est par définition équivalent à imposer la relation suivante :

$$v - Z_c i = 0 \tag{2.62}$$

avec $Z_c = \sqrt{\frac{R+jL\omega}{G+jC\omega}}$

En discrétisant cette équation et en faisant l'hypothèse que $I|_{p+1/2}^{n-1/2} = I|_{p}^{n}$, nous avons :

$$V|_{p}^{n} = Z_c\ I|_{p+1/2}^{n-1/2} \tag{2.63}$$

Dans cette condition aux limites, nous avons fait l'hypothèse que la ligne était sans perte $Z_c = \sqrt{L/C}$.

d. Gestion des nœuds de courant

Nous allons maintenant nous intéresser aux conditions aux limites qui permettent de lier plusieurs lignes de transmission entre elles. Pour cela nous définissons la notion de nœud de courant et nous montrons comment ce formalisme permet de définir des sources localisées de type Thévenin ou Norton. Ce formalisme permet également d'introduire naturellement la variation spatiale des paramètres R, L, C et G le long de la ligne.

Un nœud est un embranchement entre plusieurs branches (de ligne). Ces branches peuvent avoir des paramètres physiques différents. De manière arbitraire et en renumérotant, nous supposons que les nœuds sont localisés à un pas entier de l'espace sur l'extrémité droite de la ligne, c'est-à-dire que toutes les branches partagent l'inconnue $V|_p$.

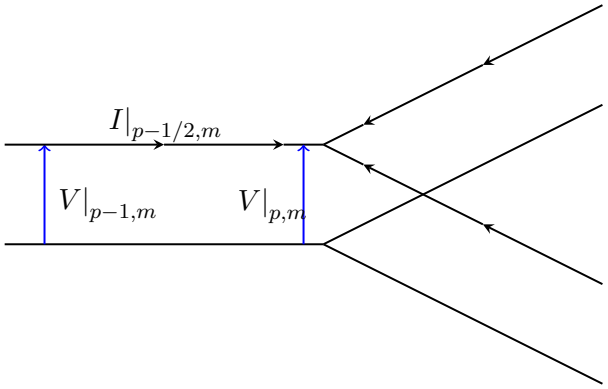


FIGURE 2.18 – Schéma équivalent pour un nœud de courant à trois branches.

Aux nœuds de courant, nous pouvons écrire les lois de Kirchhoff (loi des nœuds, loi des mailles) données par :

$$\sum_{m=1}^{M} i_m(p) = 0 \tag{2.64a}$$

$$V_m(p) = V_{m'}(p) \tag{2.64b}$$

avec M le nombre de lignes arrivant au nœud et m, m' deux de ces lignes.

Sur un noeud, nous ne pouvons pas faire une discrétisation spatiale centrée de l'équation en tension (Eq. 2.43b). Pour pallier ce problème, nous introduisons la demie discrétisation décentrée à gauche ou à droite en espace. La demie discrétisation à gauche se formalise comme suit :

$$\frac{\partial f}{\partial u}(u) = \frac{\partial f}{\partial u}(u-) \approx \frac{f(u) - f(u-\Delta/2)}{\Delta/2} + \mathcal{O}\big(\Delta^2/4\big) \tag{2.65}$$

En l'appliquant à l'équation en courant (Eq. 2.43b), on extrait le courant $I_m|_p$ au nœud p pour chaque branche m :

$$- I_m|_p^{n+1/2} = V|_p^{n+1} \cdot \left(\frac{G_m\Delta t + 2C_m}{4}\right)\frac{\Delta l_m}{\Delta t} + V|_p^n \cdot \left(\frac{G_m\Delta t - 2C_m}{4}\right)\frac{\Delta l_m}{\Delta t} - I_m|_{p-1/2}^{n+1/2} \tag{2.66}$$

Notons que la tension au point p n'est pas indicée du numéro de la ligne car toutes ces tensions sont égales (loi des mailles (Eq. 2.64b)). En utilisant le fait que la somme des $I_m|_p$ est nulle par la loi des nœuds (Eq. 2.64a) et en factorisant V_p, nous obtenons l'équation de mise à jour de la tension aux nœuds :

$$V|_p^{n+1}\sum_{m=1}^{M} A_m = - V|_p^n \sum_{m=1}^{M} B_m + \sum_{m=1}^{M} I_m|_{p-1/2}^{n+1/2} \tag{2.67}$$

avec pour chaque branche :

$$A_m = \left(\frac{G_m\Delta t + 2C_m}{4}\right)\frac{\Delta l_m}{\Delta t},\ B_m = \left(\frac{G_m\Delta t - 2C_m}{4}\right)\frac{\Delta l_m}{\Delta t}$$

Cette formulation peut s'écrire dans le cas de deux segments et permet de faire varier la valeur des paramètres d'un segment à l'autre.

Le formalisme du nœud de courant permet également de définir très simplement des sources ponctuelles (Thévenin et Norton). L'introduction de ces sources est très simple car il suffit d'introduire le courant lié aux sources dans l'équation de Kirchhoff.

La condition aux limites de type Thévenin se formalise à partir de la notion de nœud de courant. En particulier nous réécrivons l'équation de Kirchhoff en courant (Eq. 2.64a) avec les conventions définies à la figure 2.19. Nous avons donc :

$$\sum_{m=1}^{M} i_m(p) = -i_g \tag{2.68}$$

où Z_g, V_g sont les paramètres du générateur de Thévenin.

De plus, la théorie des circuits dans le générateur nous donne l'équation de Thevenin que nous pouvons discrétiser en temps :

$$\begin{aligned} v(l) &= v_g - Z_g i_g \\ V|_p^{n+1/2} &= V_g|^{n+1/2} - Z_g\ I_g|^{n+1/2} \\ \frac{V|_p^{n+1} + V|_p^n}{2} &= V_g|^{n+1/2} - Z_g\ I_g|^{n+1/2} \end{aligned} \tag{2.69}$$

En utilisant cette dernière relation, comme pour obtenir l'équation de mise à jour de la tension à un nœud de courant (Eq. 2.67), nous avons pour le modèle de Thévenin la relation suivante :

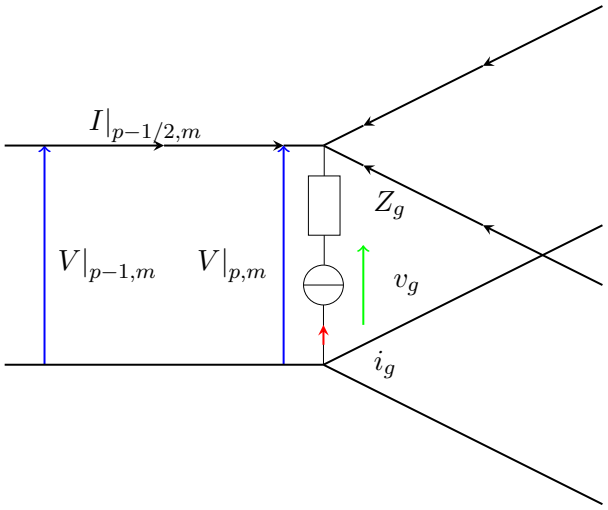


FIGURE 2.19 – Schéma équivalent pour un nœud avec un générateur de Thévenin.

$$V|_p^{n+1}\left(Z_g\sum_{m=1}^{M}A_m+1/2\right)=-V|_p^n\left(Z_g\sum_{m=1}^{M}B_m+1/2\right)+Z_g\sum_{m=1}^{M}I_m|_{p-1/2}^{n+1/2}+V_g|^{n+1/2} \tag{2.70}$$

Ce modèle peut être utilisé pour introduire un générateur de tension dans un embranchement. De plus, nous pouvons tout à fait utiliser ce modèle dans le cas où la tension V_g est nulle, nous avons donc également un modèle pour des conditions aux limites de type charge résistive.

La condition aux limites de type Norton est obtenue par le même développement. La différence réside dans l'équation de la source et de sa discrétisation temporelle. Dans un premier temps, nous réécrivons l'équation de Kirchhoff en courant (Eq. 2.64a), conformément à la figure 2.20.

$$\sum_{m=1}^{M}i_m(p)=-i \tag{2.71}$$

Nous écrivons ensuite la relation qui lie le courant du générateur au courant injecté dans le nœud. Enfin, nous discrétisons l'équation de Norton par :

$$\begin{aligned} i_t(l) &= i_g-Y_g v(l)\\ I_t^{n+1/2} &= I_g^{n+1/2}-Y_g\,V|_p^{n+1/2}\\ I_t^{n+1/2} &= I_g^{n+1/2}-Y_g\frac{V|_p^{n+1}+V|_p^n}{2} \end{aligned} \tag{2.72}$$

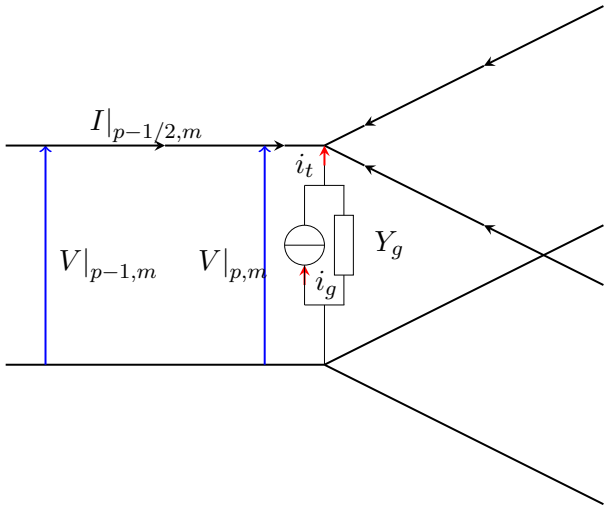


FIGURE 2.20 – Schéma équivalent pour un nœud avec un générateur de Norton.

Nous introduisons l'équation du générateur de Norton (2.72) dans le formalisme du nœud de courant (Eq. 2.67), ce qui donne après simplification :

$$V|_p^{n+1}\left(\sum_{m=1}^{M} A_m + \frac{Y_g}{2}\right) = -\,V|_p^{n}\left(\sum_{m=1}^{M} B_m + \frac{Y_g}{2}\right) + \sum_{m=1}^{M} I_m|_{p-1/2}^{n+1/2} + I_g|^{n+1/2} \qquad (2.73)$$

Ce modèle est parfaitement équivalent au modèle de Thévenin. Nous aurions pu d'ailleurs l'obtenir en utilisant les relations de passage du modèle de Thévenin au modèle de Norton. Nous déduisons de ces relations que la source équivalente de Norton I_g est le courant de court circuit du modèle de Thévenin, soit $I_g = V_g/R_g$. De plus, l'admittance de ce modèle est celle vue aux bornes de la charge quand la source est passive, soit $Y_g = 1/R_g$. Nous pouvons passer de la condition aux limites de Thévenin à celle de Norton en utilisant ces relations.

Conclusion : la formulation du nœud de courant s'avère très pratique pour définir des conditions aux limites sans faire d'approximation sur le courant en bout de ligne $I|_{p+\delta/2}^{n-1/2} \approx I|_p^n$. De plus, elle offre un cadre formel rigoureux pour introduire une variation des paramètres R, L, C, G et Δl le long de la ligne de transmission.

Par ailleurs, il existe aussi de nombreuses autres méthodes permettant de définir des conditions aux limites complexes que nous n'avons pas étudiées. Par exemple, l'utilisation de variables d'états [28, 490-496] ou la notion d'impédance généralisée [39]. Nous aurions également pu examiner un couplage avec un outil circuit type SPICE pour prendre en compte les charges complexes dynamiques et/ou non linéaires [40].

2.3.3 Résultat numérique - le problème de Franck Arreghini

Nous terminerons cette partie consacrée à la résolution des équations de lignes en temporel par une étude numérique permettant de valider notre schéma. Nous nous proposons d'étudier le problème de Franck Arreghini, problème générique pour lequel il existe une solution analytique. Le problème consiste à calculer la réponse d'un réseau formé de trois branches au-dessus d'un plan parfaitement conducteur.

a. Définition du problème

Le problème est défini par trois branches de paramètres physiques identiques, et de longueurs différentes ($AB = 5$m, $BC = 6$m et $BD = 7$m). Chaque extrémité est chargée par une résistance $R = 10\Omega$ identique pour toutes les branches. Les branches sont disposées parallèlement à un plan de masse supposé infini à une hauteur $h = 1$m pour un rayon $a = 5$mm (Fig. 2.21). Le système est agressé par une source de tension $V_s(t)$ en A, définie par le premier lobe d'une sinusoïde ($f = 100$MHz). Dans l'étude de ce problème, nous faisons l'hypothèse qu'il y a l'établissement d'un mode conduit TEM. De plus, nous pouvons considérer que le rayon des fils est très inférieur à la hauteur du fil $h \gg r$. Nous pouvons donc étudier analytiquement et numériquement ce problème par la théorie des lignes de transmission.

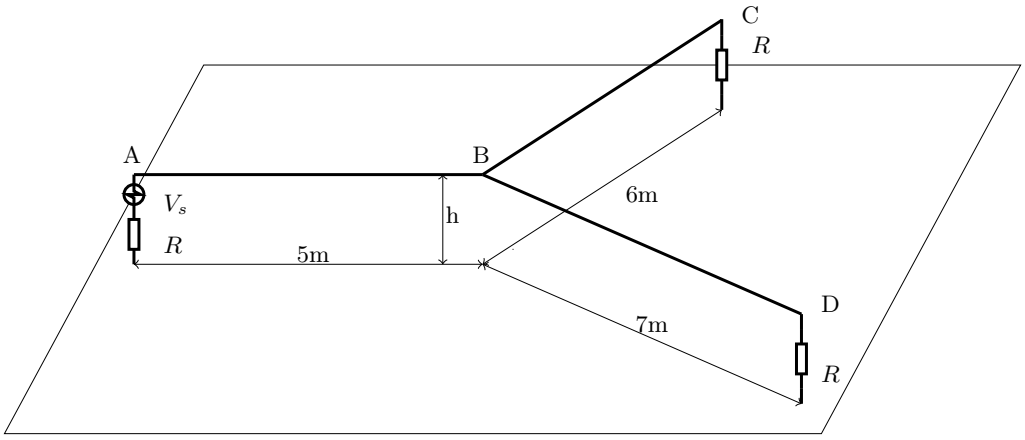


FIGURE 2.21 – Schéma du problème d'Arreghini.

b. Résultat de référence, approche réseau

Nous allons utiliser la théorie des lignes de transmission pour donner une solution analytique à ce problème. Nous avons choisi une approche réseau type BLT en temporel pour analyser ce réseau (Annexe F page 248).

Les cœfficients de réflexion et de transmission aux différentes terminaisons et jonctions peuvent être déterminés par une analyse géométrique et électrique du problème. Soit Z_c l'impédance caractéristique de la ligne définie par $Z_c \approx \frac{\mu_0}{\pi} \log\left(\frac{2h}{a}\right) \approx 360\Omega$.

Aux terminaisons (points A, C et D), le paramètre S est donné par le cœfficient de réflexion :

$$\boldsymbol{S}_A = \boldsymbol{S}_C = \boldsymbol{S}_D = \left[S_e\right] = \left[\frac{R-Z_c}{R+Z_c}\right] \tag{2.74}$$

De plus au point A nous avons une source qui induit une onde incidente

$$\boldsymbol{W}^{inc} = \left(2\frac{Z_c}{Z_c+R}V_s\right)^T \tag{2.75}$$

Au point B nous avons une jonction entre trois branches. A cette jonction, nous avons le paramètre S suivant :

$$\boldsymbol{S}_B = \begin{bmatrix} S_r & S_t & S_t \\ S_t & S_r & S_t \\ S_t & S_t & S_r \end{bmatrix} \tag{2.76}$$

Où le cœfficient de transmission $S_t = 2/3$ et le cœfficient de réflexion $S_r = -1/3$.

Nous pouvons maintenant calculer la tension en D en propageant le front d'onde dans toutes les branches. Nous n'allons pas détailler tout le calcul, mais seulement la première itération qui concerne la propagation de l'onde injectée en A et qui se propage vers B, le reste du calcul étant identique.

L'onde incidente d'amplitude $\boldsymbol{W}^{inc}$ se propage le long de la ligne et arrive en B avec un retard $\Delta t_1 = \|AB\|/c$ et donne naissance à trois ondes se propageant de B vers A, C, et D. L'onde réfléchie se propage de B vers A et a pour amplitude $\boldsymbol{W}_1 = S_r\boldsymbol{W}^{inc}$. Les deux ondes transmises se propagent de B vers C et D et ont toutes deux une amplitude de $\boldsymbol{W}_2 = S_t\boldsymbol{W}^{inc}_A$. Nous répétons ce calcul (calcul du retard et des amplitudes des ondes réfléchies/transmises) jusqu'à couvrir le temps d'observation.

Le tableau suivant nous donne le retard et l'amplitude de chaque front d'onde arrivant en D.

Les paramètres ne sont pas factorisés et conservent l'ordre (temporel) dans lequel ils ont été vus par l'onde.

#	$c\Delta t$	$\boldsymbol{W}/\boldsymbol{W}^{inc}$
1	$\|AB\| + \|BD\|$	S_t
2	$3\|AB\| + \|BD\|$	$S_rS_eS_t$
3	$\|AB\| + 2\|BC\| + \|BD\|$	$S_tS_eS_t$
4	$\|AB\| + 3\|BD\|$	$S_tS_eS_r$
5	$5\|AB\| + \|BD\|$	$S_tS_eS_rS_eS_r$
6	$\|AB\| + 2\|BC\| + 2\|AB\| + \|BD\|$	$(S_tS_eS_tS_eS_t + S_rS_eS_tS_eS_t)$
7	$\|AB\| + 2\|BD\| + 2\|AB\| + \|BD\|$	$(S_tS_eS_tS_eS_t + S_tS_eS_rS_eS_t + S_rS_eS_tS_eS_r)$
9	$\|AB\| + 2\|BC\| + 3\|BD\|$	$(S_tS_eS_tS_eS_r + S_tS_eS_tS_eS_t)$
10	$\|AB\| + 5\|BD\|$	$S_tS_eS_rS_eS_r$

c. Résultat numérique

La figure 2.22 superpose la tension calculée en D par la méthode différence finie précédemment décrite, avec la tension de référence obtenue analytiquement. Le résultat est très satisfaisant et permet de montrer la validité de la méthode.

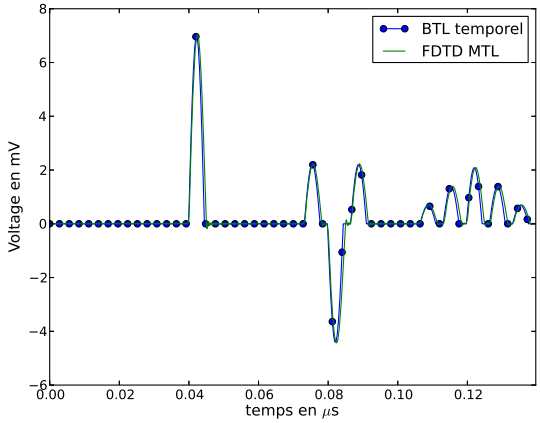


FIGURE 2.22 – Résultat numérique au problème d'Arreghini.

2.4 Prise en compte de la dépendance en fréquence des paramètres de ligne - Cas d'une ligne au-dessus d'un sol

Comme nous venons de le voir, la mise en œuvre d'un code de ligne en temporel pour une ligne au-dessus d'un plan parfaitement conducteur ne présente aucune difficulté particulière. Nous rappelons que notre objectif est de pouvoir modéliser la propagation dans des lignes au-dessus du sol, pour des câbles reliant plusieurs bâtiments entre eux. Or, dans le cas d'un plan de référence non parfaitement conducteur, les paramètres de la ligne vont dépendre de la fréquence. Dans la section précédente (Sec. 2.2 page 33), nous avons déjà donner l'expression des paramètres électriques d'une ligne au-dessus d'un sol de conductivité finie en fonction de la fréquence. Pour notre approche temporelle, la dépendance en fréquence de ces paramètres est problématique car elle nous impose de faire une convolution de tout le signal temporel avec la réponse impulsionnelle du paramètre linéique de la ligne de transmission. Nous avons donc, dans un premier temps, cherché une approximation raisonnable des paramètres d'une ligne de transmission permettant l'utilisation de techniques de convolution récursives [28, 410-443]. Nous trouvons notamment dans la littérature deux méthodes :

Méthode de Carson / Pollaczek Dans les années 1930, Carson et Pollaczek furent les initiateurs de l'étude des lignes de transmission intégrant le sol comme conducteur de retour [41]. Leurs travaux furent majoritairement utilisés dans le monde du transport et de la distribution de l'énergie électrique.

Dans leur modèle, Carson et Pollaczek introduisent un terme correctif à l'impédance linéique de la ligne de transmission. Ce terme est défini par :

$$Z_{carson} = \frac{\mu_s}{\pi} \int_0^{+\infty} \frac{\exp(-2hx)}{x + \sqrt{x^2 j\omega\mu_s\sigma_s}} dx \tag{2.77}$$

Le formalisme ne présente pas de propriété particulière permettant son utilisation dans un code temporel ; c'est pourquoi nous n'avons pas retenu ce modèle.

Méthode de Gary / Deri - profondeur de pénétration complexe. Dans les années 1980, Gary propose une nouvelle approche [3] dans laquelle il introduit la « profondeur de pénétration complexe » p. Cette grandeur complexe définit la profondeur de pénétration des courants dans le sol en prenant en compte la résistivité du milieu et la fréquence des courants d'excitation. Cette grandeur s'ajoute à la distance séparant les conducteurs de la terre. Le sol est alors modélisé via l'éloignement des conducteurs. Cette approche fut confortée par Deri quelques années plus tard [6].

Dans sa méthode, Gary définit une profondeur complexe de pénétration qui vient s'ajouter à la distance sol - câble, donnée par :

$$p = \sqrt{\frac{\sigma_s}{j\omega\mu_s}} \tag{2.78}$$

Ce qui conduit à l'inductance suivante :

$$L = \frac{\mu_0}{2\pi} \cosh^{-1}\left(\frac{h+p}{r}\right) \tag{2.79}$$

Cette alternative ne donne pas lieu non plus à un formalisme adapté à une convolution récursive

En conclusion, les modèles de Carson et de Gary sont plus complexes à déterminer et moins précis qu'un modèle proposé par E. F. Vance. Nous avons donc choisi de travailler avec le modèle de E. F. Vance tel que défini dans la section (Sec. 2.2 page 33) et d'étudier une méthode permettant d'introduire celui-ci dans notre méthode temporelle.

2.4.1 Approximation par une série de Debye

Nous pouvons voir dans la littérature que les séries de Debye présentent un avantage certain pour modéliser des matériaux qui dépendent de la fréquence dans une méthode temporelle. De plus, l'implémentation d'une série de Debye dans une méthode temporelle est relativement simple, car elle ne fait intervenir qu'une somme de termes auxiliaires associés à chaque pôle de la série de Debye. Ces termes auxiliaires sont indépendants les uns des autres et peuvent être calculés en utilisant un schéma semi-implicite en temps. La convolution temporelle est alors remplacée par une somme de problèmes auxiliaires ne nécessitant pas la connaissance des inconnues à tous les pas de temps.

Pour tenir compte des paramètres de la ligne dépendants de la fréquence, nous avons donc cherché à factoriser les expressions des paramètres données par E. F. Vance sous la forme d'une série de Debye (Eq. 2.80) :

$$\hat{h}(\omega) = h_\infty + (h_s - h_\infty) \sum_{n=1}^{N} \frac{a_n}{1 + j\omega\tau_n} \tag{2.80}$$

avec :

$\hat{h}(\omega)$ série de Debye associée à la fonction à approximer h,

h_∞ $= \lim_{\omega\to\infty} h(\omega)$ la limite à l'infini de h,

h_s la limite statique de h définie par : $(h(0) - h_\infty) = (h_s - h_\infty) \sum_{n=1}^{N} a_n$,

a_n l'amplitude du pôle n

et τ_n le point de relaxation temporelle du pôle n.

Le problème que nous devons maintenant résoudre est la détermination de la série de Debye (c'est-à-dire les paramètres N, h_∞, h_s, a et τ) pour que celle-ci approche correctement l'impédance de la ligne de transmission. Nous proposons dans ce paragraphe deux approches différentes pour traiter ce problème. La première approche étudiée est basée sur l'utilisation des résidus et la seconde sur une optimisation globale.

a. Décomposition par résidu

Nous utilisons une décomposition en pôles et résidus [42] pour reformuler la série de Debye sous la forme d'une fraction rationnelle de $\mathbb{C}(\omega)$, où le degré du dénominateur est directement supérieur à celui du numérateur (degré du dénominateur = degré du numérateur + 1). Nous avons :

$$(h_s - h_\infty) \sum_{n=1}^{N} \frac{a_n}{1 + j\omega\tau_n} = \frac{\sum_{n=0}^{N-1} N_n \omega^n}{\sum_{n=0}^{N} D_n \omega^n} = \frac{N(\omega)}{D(\omega)} \tag{2.81}$$

Cette décomposition en une fraction rationnelle peut être obtenue par le calcul de la réponse fréquentielle d'un filtre numérique. Les cœfficients de N et D sont alors donnés en minimisant $\sum \|N - HD\|^2$ [43] [44], où H est la réponse impulsionnelle numérique de h.

Ensuite, nous obtenons les paramètres de la série de Debye h_s, a_n et τ_n par conversion en pôle résidu de N/D [45].

$$\frac{N(\omega)}{D(\omega)} = \sum_{n=1}^{N} \frac{r_n}{\omega - p_n} = (h_s - h_\infty) \sum_{n=1}^{N} \frac{a_n}{1 + j\omega\tau_n} \tag{2.82}$$

Cette approche donne de très bons résultats en terme d'approximation, mais elle ne permet pas de garantir la stabilité du schéma numérique ($a_n > 0$) [36]. Nous avons donc exploré une autre solution permettant de contrôler la positivité des termes a_n et ainsi d'assurer la stabilité du schéma.

b. Factorisation par heuristique d'optimisation globale

Les techniques d'optimisation globale représentent la classe d'algorithme numérique qui cherche à donner une solution globale en un temps raisonnable à des problèmes d'optimisation réputés très difficiles, comme par exemple, le très connu problème du voyageur de commerce ou encore le problème du sac à dos. Le problème du voyageur de commerce consiste à trouver le plus court chemin qui relie toutes les villes. Le deuxième problème est un problème d'optimisation combinatoire où l'objectif est de remplir au mieux un sac à partir d'un ensemble donné d'objets sans dépasser la capacité maximale du sac. Ces deux problèmes font parties des 21 problèmes NP-complets [2] de Karp [46].

Pour traiter ce type de problème, les algorithmes d'optimisation globale ont souvent une nature stochastique qui leur permet de faire une recherche « aléatoire » conditionnée par le problème traité. Sans entrer dans le détail, il existe plusieurs classes d'algorithmes qui ont tous en commun d'imiter un mécanisme physique (recuit simulé), comportementaliste (colonie de fourmis) ou évolutionniste (algorithme génétique), en y introduisant un caractère aléatoire. Une des particularités de ce type d'algorithme est qu'il ne trouve pas la solution, mais une solution satisfaisante en un temps « raisonnable ».

2. Un problème NP-complet est un problème algorithmique dont il est (algorithmiquement) simple de vérifier que la solution est correcte. C'est-à-dire en un temps polynomial $t <= kN^\alpha$ où N est la taille du problème avec k et α deux constantes. En revanche trouver la solution à ce problème nécessite un temps non polynomial (par exemple une complexité exponentielle $t > \alpha^N$).

En ce qui concerne notre problème, D.F. Kelly [47, 48], dans ses travaux, propose de mixer une approche d'optimisation globale avec la méthode des moindres carrés pour décomposer un matériau en un polynôme de Debye tout en imposant une contrainte sur le signe des amplitudes de chaque pôle pour garantir la stabilité du schéma.

Dans cette section, nous allons présenter cette approche et les adaptations faites pour pouvoir l'utiliser dans notre problème.

Le problème de factorisation peut se mettre sous la forme d'un problème d'optimisation de la forme suivante :

$$\underset{\boldsymbol{\Theta}\in\mathfrak{D}, f(\boldsymbol{\Theta})>0}{\operatorname{argmin}} \left\langle \hat{h}_{\boldsymbol{\Theta}} | h \right\rangle \tag{2.83}$$

où :

h		est la fonction objective que l'on cherche à optimiser,
$\hat{h}_{\boldsymbol{\Theta}}$		la série de Debye paramétrés par le vecteur $\boldsymbol{\Theta}$,
$\boldsymbol{\Theta}$	$=$	$(N, h_s, h_\infty, a_1, ..., a_n, \tau_1, ..., \tau_n)$ le vecteur de paramètres,
$\mathfrak{D}$		le domaine de recherche,
$\langle \vert \rangle$		une mesure de l'erreur d'approximation
et f		la fonction contrainte sur le vecteur $\boldsymbol{\Theta}$.

L'objectif de cette optimisation est de trouver le vecteur Θ qui minimise une certaine fonction test ou objective tout en respectant des contraintes représentées par la fonction f. Les principales étapes de l'algorithme proposé sont succinctement décrites ici et seront plus largement détaillées dans les sections suivantes.

1. Définition du domaine de validité ω_{min}, ω_{max} sur lequel notre approximation sera considérée et la définition d'un échantillonnage Ω discrétisant notre problème.
2. Détermination et approximation des paramètres N, h_s et h_∞.
3. Utilisation d'un heuristique d'optimisation globale pour explorer le domaine des points de relaxation du polynôme de Debye $\boldsymbol{\tau}$ couplé avec un estimateur des moindres carrés pour obtenir les amplitudes $\boldsymbol{a}$ optimales. Une contrainte a été ajoutée de manière à ne conserver que les solutions garantissant la stabilité du schéma numérique.
4. Enfin, introduction d'un biais Δ permettant de corriger à posteriori les paramètres h_s et h_∞.

Nous allons voir plus en détail chacune de ces étapes.

La définition du domaine de validité et d'un échantillonnage est une étape importante car elle permet de simplifier le calcul de l'erreur d'approximation. Rappelons que l'objectif est de trouver une série de Debye $\hat{h}$ qui approche au mieux la fonction h. Pour le trouver, nous minimisons une fonction d'erreur donnée par exemple par $\left\| \hat{h} - h \right\|$. Ainsi, le

fait de passer à un espace discret borné permet de remplacer symboliquement l'opérateur intégral de la norme par une somme finie de termes ($\int_0^\infty \rightarrow \sum_{\omega_i \in \Omega}$).

Le domaine de validité $[\omega_{min}, \omega_{max}]$ est également important car, en dehors de ce domaine, nous ne maîtrisons pas la série de Debye. Celle-ci pourrait osciller, décroître fortement, entraînant des erreurs sur la solution finale.

Il est donc important de choisir le domaine de validité en fonction du problème que nous cherchons à étudier. Nous pourrions être tenté d'utiliser la bande de fréquence étudiée comme domaine de validité. Cependant, il semble plus pertinent de toujours se référer aux paramètres du schéma numérique (pas spatial et temporel). En effet, la discrétisation temporelle n'est pas toujours choisie en fonction de la gamme de fréquences que nous voulons étudier. Par exemple, dans l'étude d'un problème où la géométrie est relativement fine par rapport à la longueur d'onde (le problème de foudroiement par exemple), la discrétisation est conditionnée par la géométrie et non par le spectre de la source d'excitation. Nous devons donc toujours revenir à la discrétisation numérique car elle détermine théoriquement la gamme des fréquences qui peut exister durant la simulation. Le pas de temps Δt définit par Shannon la fréquence maximale que nous pouvons rencontrer $\omega_{max} = 4\pi/\Delta t$ et la durée totale T définit la fréquence minimale $\omega_{min} = 4\pi/T$. Dans ces conditions, nous sommes sûrs d'utiliser une série de Debye dans un domaine fréquentiel où nous maîtrisons sa variation.

Le choix de l'échantillonnage Ω est aussi important car il conditionne la qualité du résultat. Il est évident que si nous échantillonnons « lentement » un phénomène « rapide », nous ne pourrons pas le représenter correctement. De même, si nous sur-échantillonnons celui-ci, nous allons perdre du temps en calcul pour un gain sur l'erreur totale qui peut être négatif (phénomène de Runge). L'expérience montre qu'un choix raisonnable d'échantillons est donné : $\text{card}(\Omega) = 4N$ avec N définissant le nombre de pôles de la série de Debye. De plus, il s'avère qu'un échantillonnage logarithmique est plus performant car il traduit une meilleure décorrélation des fonctions de projection utilisées dans l'estimateur des moindres carrés.

Estimateur à priori : Ne disposant pas de valeur expérimentale comme pour les matériaux dispersifs pour déterminer les paramètres N, h_s et h_∞, nous devons donner une approximation à ces paramètres que nous corrigerons à posteriori.

En ce qui concerne le nombre de pôles N, il n'y a pas de technique particulière pour le déterminer, ce qui est la principale lacune de cette approche. Nous devons alors faire plusieurs tentatives jusqu'à trouver une solution satisfaisante.

Pour les paramètres h_s et h_∞, la première solution est d'utiliser les limites du domaine de validité pour estimer ces paramètres $h_s = \Re[h(f_{min})]$ et $h_\infty = \Re[h(f_{max})]$, mais il semble plus pertinent d'étendre le domaine pour le problème de limite. En effet, la limite à l'infini des impédances vue en section (Sec. b. page 39), tend vers l'infini, alors que pour la série de Debye cette limite est finie et vaut h_∞. L'utilisation d'une fréquence en dehors du domaine de validité permet d'éviter ce problème et de l'enlever du domaine de validité. Nous pouvons par exemple choisir $h'_\infty = \Re[h(k * f_{max})]$ ou $h''_\infty = \Re[h(f^k_{max})]$, avec k une constante positive. De plus, nous avons vu que h_s est fortement lié à la valeur de h en 0, il semble donc judicieux d'utiliser $h'_s = \Re[h(0)]$.

Heuristique d'optimisation de type « essaims particulaires » : Les heuristiques dans les problèmes d'optimisation combinatoire définissent un ensemble d'algorithmes qui fournissent en un temps raisonnable (en un temps polynomial) une solution satisfaisante qui n'est pas forcément la solution optimale. Parmi ceux-ci, l'optimisation par essaims particulaires est un algorithme inspiré du vivant, s'appuyant sur le concept d'auto-organisation. Cette idée veut qu'un groupe d'individus peu intelligents puisse posséder une organisation globale complexe. L'algorithme (2.1) donne les principales étapes de cette heuristique.

L'idée générale de cette algorithme est d'imiter le comportement d'un essaim dans sa recherche de nourriture. Supposons un ensemble d'individus ou particules k à la position X_k animé d'une vitesse V_k. Quand un individu trouve un lieu attractif (point de nourriture), tous les autres individus convergent vers ce point. Ce comportement permet d'explorer localement une zone afin d'y trouver le minimum local. Puis, plus le temps passe, plus les individus s'éloignent du minimum local, ils s'éparpillent dans l'espace de recherche. Ce qui donne la capacité à l'algorithme de trouver d'autres points plus attractifs.

Algorithm 2.1 L'algorithme essaims particulaires initial

$P_g \leftarrow 0$
for $k = 1 \rightarrow K$ **do** ▷ Initialisation des particules
  $X_k \leftarrow random()$
  $V_k \leftarrow 0, P_k \leftarrow X_k$
  if $\phi(X_k) <= \phi(P_g)$ **then** ▷ Initialisation de la meilleure solution
    $P_g \leftarrow X_k$
  end if
end for
for $n = 1 \rightarrow N$ **do**
  for $k = 1 \rightarrow K$ **do**
    $V_k \leftarrow mV_k + b_1() * (P_k - X_k) + b_2() * (P_g - X_k)$
    $X_k \leftarrow X_k + V_k$ ▷ Mise à jour de la particule
    if $\phi(X_k) <= \phi(P_k)$ **then** ▷ Mise à jour des meilleures solutions
      $P_k \leftarrow X_k$
      if $\phi(X_k) <= \phi(P_g)$ **then**
        $P_g \leftarrow X_k$
      end if
    end if
  end for
end for

où

X	est une particule numérotée par k sur le nombre total de particules K.
P_k	est la meilleure solution atteinte par la particule k à l'itération considérée.
P_g	est ma meilleure solution trouvée par essaim à l'itération considérée.
m	est l'inertie des particules
$b_1()$ et $b_2()$	sont deux tirs aléatoires
ϕ	est la fonction objective que nous cherchons à minimiser

La capacité de prospection de cet algorithme réside dans des tirs aléatoires $b_1()$ et $b_2()$,

mais la prospection est cependant orientée de manière à graviter autour de points d'attractions qui sont les meilleures solutions atteintes par les particules. Notons deux cas limites à la prospection dont le premier confère un caractère « individualiste » à notre recherche en imposant $b_1() = 1$ et $b_2() = 0$, et l'autre donne un caractère fortement collectif (le tous pour un) en imposant $b_1() = 0$ et $b_2() = 1$.

Cet algorithme a été modifié de manière à exploiter pleinement un estimateur des moindres carrés et introduire la fonction test f permettant de contraindre la recherche.

Estimateur des moindres carrés : Dans la stratégie proposée, l'optimisation globale a pour rôle d'explorer le domaine de validité $[\omega_{min}, \omega_{max}]$. Cependant, pour avoir le série de Debye optimale, nous devons également trouver les amplitudes a_n les plus adaptées pour chaque point de relaxation τ_n. C'est le rôle de l'estimateur des moindres carrés. Dans notre cas, cet estimateur se base sur le lien entre la partie imaginaire et la partie réelle de l'impédance cherchée.

Ce lien est donné par les relations de Kramers-Kronig (Eq. 2.85) qui lient les parties réelle et imaginaire de certaines fonctions complexes par la transformée de Hilbert. Ces relations peuvent être appliquées à une impédance linéique d'une ligne de transmission qui peut être interprétée comme la transformée de Fourier de la réponse temporelle d'une ligne de transmission linéaire et causale à une excitation infiniment brève, c'est-à-dire sa réponse impulsionnelle. Ainsi approcher la partie imaginaire de l'impédance permet aussi d'approcher sa partie réelle. Cette propriété est centrale, car elle nous permet de ramener le problème d'un espace complexe dans un espace réel.

Lemme 2.4.1. *Les relations de Kramers-Kronig décrivent la relation qui existe entre la partie réelle et la partie imaginaire de certaines fonctions complexes. Plus spécifiquement, elles s'appliquent aux fonctions qui sont analytiques sur le demi-plan supérieur de la variable complexe. Nous pouvons en effet montrer qu'une telle fonction $f(\omega)$ représente la transformée de Fourier d'un processus physique linéaire et causal.*

Si nous écrivons :

$$f(\omega) = f_1(\omega) + j f_2(\omega), \tag{2.84}$$

avec f_1 et f_2 des fonctions réelles causales et analytiques sur le demi-plan supérieur de la variable complexe ω, alors les relations de Kramers-Kronig donne :

$$f_1(\omega) = \frac{2}{\pi} \int_{-\infty}^{0} \frac{s f_2(\omega)}{s^2 - \omega^2} ds \tag{2.85a}$$

$$f_2(\omega) = -\frac{2\omega}{\pi} \int_{0}^{\infty} \frac{f_1(\omega)}{s^2 - \omega^2} ds \tag{2.85b}$$

Cette relation nous permet de définir un estimateur des moindres carrés uniquement sur la partie imaginaire de l'impédance h, ce qui nous permet d'avoir une solution $\boldsymbol{a}$ réelle. Il s'avère utile de normaliser la partie imaginaire de l'impédance par $h_s - h_\infty$, ce qui donne

l'équation (Eq. 2.86). Nous obtenons un modèle linéaire en $\boldsymbol{a}$ que nous pouvons utiliser dans notre estimateur qui doit minimiser la fonction χ^2 (Eq. 2.87).

$$h_1(\omega) = -\frac{\Im\left[h(\omega)\right]}{h_s - h_\infty} = \sum_{n=1}^{N} a_n \frac{\omega\tau_n}{1+(\omega\tau_n)^2} \tag{2.86}$$

$$\chi^2(a) = \sum_{\omega_i \in \Omega} \left(\sum_{n=1}^{N} a_n \varphi_n(\omega_1) - h_1(\omega_i) \right)^2 \tag{2.87}$$

avec $\varphi_n(\omega) = \frac{\omega\tau_n}{1+(\omega\tau_n)^2}$ la base fonctionnelle de décomposition et Ω l'ensemble des fréquences d'échantillonnage.

La solution de ce problème de minimisation est donnée par :

$$\boldsymbol{a} = (\boldsymbol{D}^T\boldsymbol{D})^{-1}\boldsymbol{D}^T\boldsymbol{h} \tag{2.88}$$

où $\boldsymbol{D} = (D_{in})_{i,n}$ et $\boldsymbol{h} = (h_i)_i$ avec :

$$\begin{aligned} D_{in} &= \varphi_n(\omega_i) = \frac{\omega_i\tau_n}{1+(\omega_i\tau_n)^2} \\ h_i &= h_1(\omega_i) = \frac{\Im\left[h(\omega_i)\right]}{h_s - h_\infty} \end{aligned}$$

Dans certaines situations, l'estimateur des moindres carrés (Eq. 2.88) produit des solutions presque toujours négatives, ce qui est problématique pour garantir la stabilité du schéma FDTD associé à notre équation de ligne [36]. En effet, dans cette situation, la décomposition en série de Debye risque d'introduire une instabilité numérique. Ce problème se produit quand deux amplitudes a_n de fréquences adjacentes oscillent entre une valeur positive et négative. Ce problème peut être supprimé en forçant aussi la somme du module des amplitudes $\boldsymbol{a}\cdot\boldsymbol{a} = |\boldsymbol{a}|^2$.

La solution de ce problème de minimisation contraint est donnée par [49, Sec. 6.3] :

$$\boldsymbol{a} = (\boldsymbol{D}^T\boldsymbol{D} + \gamma\boldsymbol{I})^{-1}\boldsymbol{D}^T\boldsymbol{h} \tag{2.89}$$

où $\boldsymbol{I}$ est la matrice identité et γ une constante arbitraire.

Notons que cette procédure revient à renforcer les termes diagonaux et que γ doit être choisi convenablement sous peine de tronquer la solution. Le choix de γ dépend du problème. Il est difficile de donnée un critère absolu pour le choix de cette valeur.

La fonction fitness : Nous avons vu, dans les parties précédentes, les deux grandes étapes (optimisation globale et estimateur des moindres carrés) de notre algorithme de factorisation d'une impédance arbitraire en une série de de Debye. Cependant, l'algorithme d'optimisation globale nécessite aussi de définir une fonction dite fitness. La fonction fitness est centrale pour ce type d'algorithme car elle conditionne la solution finale. Dans notre cas, deux fonctions donnent de très bons résultats. Ces fonctions fitness sont construites à partir de la fonction que nous cherchons à factoriser et d'une mesure d'erreur.

$$\phi(\mathbf{\Theta}) = \left\langle \hat{h_{\mathbf{\Theta}}} | h \right\rangle \tag{2.90}$$

Dans notre étude, deux mesures d'erreurs ont montré leur performance. D'une part, l'erreur moyenne sur la partie imaginaire (Eq. 2.91), initialement proposée par David F. KELLEY. D'autre part, l'erreur moyenne sur l'inductance (Eq. 2.92), plus généralement la partie imaginaire pondérée par la pulsation ω. Ces deux mesures sont très proches mais la deuxième semble plus pertinente car le terme d'induction rentre directement en compte dans l'implémentation temporelle (Sec. 2.4.3 page 88).

$$\phi_i(\mathbf{\Theta}) = \frac{1}{\text{card}\,(\Omega)} \sqrt{\sum_{\omega_i \in \Omega} \left(\Im\{h(\omega_i) - \hat{h_{\mathbf{\Theta}}}\} \right)^2} \tag{2.91}$$

$$\phi_L(\mathbf{\Theta}) = \frac{1}{\text{card}\,(\Omega)} \sqrt{\sum_{\omega_i \in \Omega} \left(\frac{\Im\{h(\omega_i) - \hat{h_{\mathbf{\Theta}}}\}}{\omega_i} \right)^2} \tag{2.92}$$

Nous pouvons également voir la fonction fitness sur la partie imaginaire comme une fonction fitness qui donne plus d'importance aux hautes fréquences, en opposition à l'autre qui pondère l'impédance par la fréquence. De plus, comme nous l'avons déjà évoqué, la fonction fitness normalisée par la fréquence considère l'erreur sur l'inductance, alors que l'autre peut être vu comme l'erreur sur les pertes (R), ce qui se justifie par la relation de Kramers-Kronig.

La fonction contrainte : Jusqu'à présent, nous n'avons pas évoqué la problématique du contrôle de la stabilité du schéma numérique associé à l'équation des lignes. Ce contrôle est garanti par la fonction contrainte. Il n'existe pas de démonstration claire permettant de définir un critère de stabilité pour une série de Debye. Usuellement, nous étendons le critère de stabilité du modèle de Debye à un terme [36] au cas à plusieurs pôles. C'est-à-dire que tous les pôles doivent avoir une amplitude strictement positive. Le paramètre $\mathbf{\Theta}$ est dit acceptable si $f(\Theta) > 0$. Les fonctions tests suivantes $f(\Theta) = min(a)$ où $f'(\Theta) = -\,\text{card}(a < 0)$ sont possibles. La valeur numérique retournée par la fonction contrainte n'est pas exploitée pour rendre la solution acceptable. Le choix entre les deux fonctions test précédentes n'influe en rien sur la solution.

Correcteur à posteriori : Le correcteur à posteriori finalise la procédure de factorisation. L'utilisation de la relation de Kramers-Kronig nous a permis de ramener le problème d'optimisation uniquement sur la partie imaginaire. Ainsi, si nous avons une bonne concordance sur la partie imaginaire, nous avons une bonne concordance sur la partie réelle mais à une constante près. Pour corriger cette erreur, nous introduisons un biais Δ à posteriori sur les paramètres h_s et h_∞, en écrivant :

$$\begin{aligned}\hat{h}'(\omega) &= \Delta + \hat{h}(\omega) \\ &= \Delta + h_\infty + (h_s - h_\infty)\sum_{n=1}^{N}\frac{a_n}{1+j\omega\tau_n} \\ &= h'_\infty + \left(h'_s - h'_\infty\right)\sum_{n=1}^{N}\frac{a_n}{1+j\omega\tau_n}\end{aligned}$$

où $h'_s = h_s + \Delta$ et $h'_\infty = h_\infty + \Delta$.

En ce qui concerne ce biais, D.F. KELLEY propose une valeur qui minimise l'erreur moyenne sur la partie réelle.

$$\Delta_{mean} = \frac{1}{\text{card}\,(\Omega)}\sum_{\omega_i\in\Omega}\Re\left[h(\omega_i) - \hat{h}(\omega_i)\right] \tag{2.93}$$

avec Ω l'ensemble des fréquences d'échantillonnage.

Nous avons également définir un biais Δ_0 qui permet de minimiser l'erreur dans les basses fréquences (domaine de plus faible perte) :

$$\Delta_0 = \Re\left[h(\omega_0) - \hat{h}(\omega_0)\right] \tag{2.94}$$

Pour un problème basse fréquence, la deuxième proposition est plus pertinente car elle permet d'avoir une meilleure représentation pour les fréquences qui subissent le moins ou pas de perte et donc qui se propageront potentiellement plus loin.

c. Algorithme de factorisation en polynôme de Debye

Nous avons vu dans la section précédente les différents composants de l'algorithme de factorisation. Nous allons maintenant présenter l'algorithme final et étudier son comportement en terme de convergence pour deux cas ; une fonction « Cole-Cole » qui a été utilisée par D.F. KELLEY, puis un cas de sol réel.

L'algorithme final est proposé en 2.2. À la ligne 2, nous calculons un échantillonnage Ω optimal en fonction du domaine défini par ω_{min}, ω_{max}. Puis en ligne 3, nous utilisons notre estimateur à priori pour déterminer les paramètres h_s et h_∞. Notons que ces paramètres sont identiques pour toutes les particules que nous allons utiliser. Nous terminons la phase d'initialisation par la définition de toutes les particules. Toutes les particules ont le même nombre de pôles P. Les particules sont scindées en deux parties : les amplitudes a_k et les points de relaxation temporelle τ_k. Les points de relaxation temporelle τ_k sont initialisés de manière aléatoire sur le domaine Ω, car il semble raisonnable de penser que les pôles vont être dans le domaine ou proche de celui-ci. Ce choix permet de gagner un peu sur la vitesse de convergence. En effet, si la valeur initiale est plus proche de la valeur finale, la distance à parcourir est plus petite.

Puis, nous itérons sur N étapes ou époques pour trouver la meilleure solution P_g. La fonction contrainte (ligne 11) permet d'appliquer le bon traitement à chaque particule. Si la particule a des pôles négatifs, nous ajustons le paramètre γ (lignes 12-16) de manière à

contrôler l'estimateur des moindres carrés en renforçant les termes diagonaux. Cependant, ce renforcement doit être limité, choisir $\gamma_{max} = 0.1$ et $\gamma_{stap} = 0.01$ semble raisonnable. Dans le cas où la particule n'a pas de pôle négatif nous appliquons le traitement propre à l'algorithme « essaims particulaires » (lignes 18-26)

Algorithm 2.2 Algorithme de factorisation en polynôme de Debye

1: **function** FACTODEBYE($h, \omega_{min}, \omega_{max}, P, K, N$)
2: $\Omega \leftarrow DOMAIN(\omega_{min}, \omega_{max}, P)$
3: $h_s, h_\infty \leftarrow APRIORI_ESTIMATOR(h, \Omega)$
4: **for** $k = 1 \rightarrow K$ **do** ▷ K particules à P pôles.
5: $\gamma_k \leftarrow 0$
6: $\tau_k \leftarrow RANDOMIZE_RELAX_TIMES(\Omega, P)$
7: $a_k \leftarrow ESTIMATE_WEIGHTS(h, \Omega, \tau_k, \gamma_k)$
8: **end for**
9: **for** $n = 1 \rightarrow N$ **do**
10: **for** $k = 1 \rightarrow K$ **do**
11: **if** card$(a_k \leq 0) > 0$ **then** ▷ Si des amplitudes négatives
12: $\gamma_k \leftarrow \gamma_k + \gamma_{step}$
13: **if** $\gamma_k > \gamma_{max}$ **then** ▷ Si trop instable
14: $\gamma_k \leftarrow 0$
15: $\tau_k \leftarrow RANDOMIZE_RELAX_TIMES(\Omega, P)$ ▷ Randomize particule
16: **end if**
17: **else** ▷ Mise à jour de la particule si elle est valide
18: $V_k \leftarrow mV_k + b_1() * (P_k - \tau_k) + b_2() * (P_g - \tau_k)$
19: $\tau_k \leftarrow \tau_k + V_k$
20: $a_k \leftarrow ESTIMATE_WEIGHTS(h, \Omega, \tau_k, \gamma_k)$
21: **if** $\phi(X_k) <= \phi(P_k)$ **then** ▷ Mise à jour des meilleurs solutions
22: $P_k \leftarrow X_k$ ▷ Note : $X_k = (h_s, h_\infty, \tau_k, a_k)$
23: **if** $\phi(X_k) <= \phi(P_g)$ **then**
24: $P_g \leftarrow X_k$
25: **end if**
26: **end if**
27: **end if**
28: **end for**
29: **end for**
30: **return** $APOSTERIORI_CORRECTOR(P_g, h)$ ▷ Calcul du biais
31: **end function**

2.4.2 Validation numérique

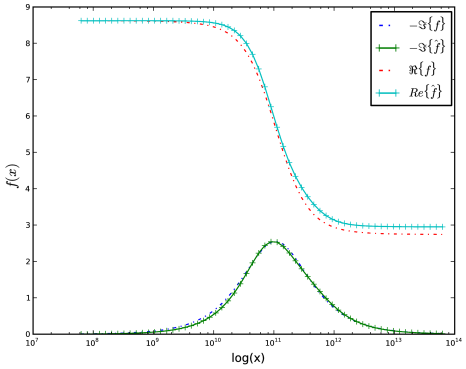
Un exemple classique : La fonction Cole-Cole Dans la littérature, de nombreuses méthodes pour déterminer les paramètres diélectriques d'un matériau sont validées sur la fonction « Cole-Cole » définie comme suit :

$$f(\omega) = f_\infty + \frac{f_s - f_\infty}{1 + (j\omega\tau_0)^\alpha} \tag{2.95}$$

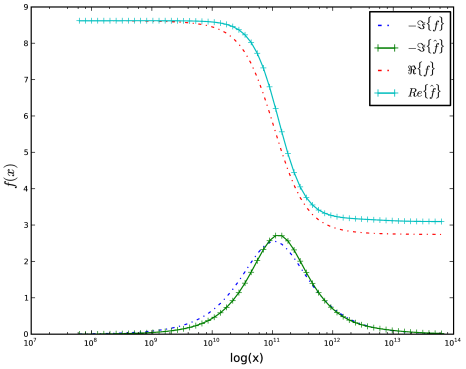
avec $0 \leq \alpha \leq 1$

Nos méthodes précédemment décrites, seront donc, dans un premier temps, testées sur cette fonction, puis, dans un deuxième temps, appliquées sur des modèles d'impédances représentatives d'un fil au-dessus d'un sol.

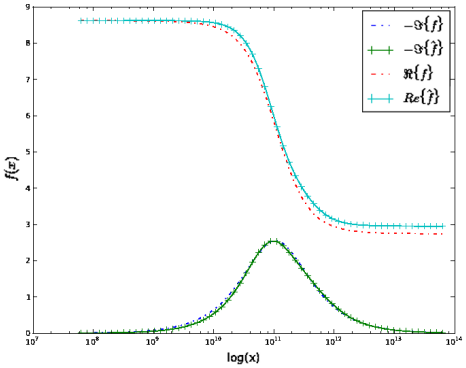
Notre première méthode utilise un algorithme d'optimisation globale itérative de recherche d'optimum. Dans le cas de la fonction Cole-Cole, cet algorithme converge relativement rapidement (< 1000 itérations). La figure 2.23 donne les 1ère, 6ème, 10ème et dernière meilleures solutions trouvées en moins de 1000 itérations.



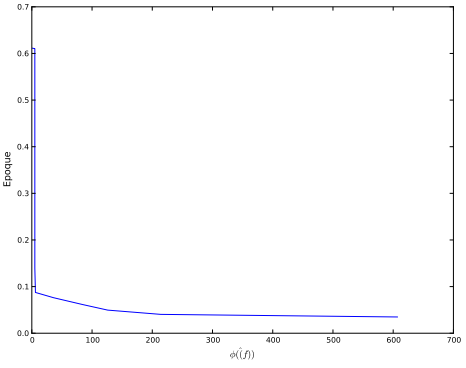
(a) La meilleure solution trouvée après la phase d'initialisation.



(b) La 6ème meilleure solution trouvée après 36 itérations.



(c) La 10ème meilleure solution trouvée en moins de 1000 itérations. Cette solution a été trouvé à 607 itérations.



(d) Évolution de la fonction fitness pour les meilleures solutions trouvées en fonction de l'itération.

FIGURE 2.23 – Hybrid Particle Swarm-Least Quares optimization sur la fonction Cole-Cole pour trois pôles. L'interpolation est faite avec 15 points répartis sur une échelle logarithmique. L'optimisation a été faite avec 100 particules sur 1000 itérations.

Pour bien visualiser comment l'algorithme converge, nous visualisons l'état à plusieurs itérations (Fig. 2.23). Pour cela, nous superposons la partie réelle et imaginaire de la fonction recherchée (Cole-Cole) en trait discontinu et de la série de Debye à trois itérations différentes de l'algorithme. A la première itération (Fig. 2.23(a)), nous avons une très nette différence, puis plus nous itérons, plus la série de Debye se rapproche de la fonction désirée.

Pour avoir une vision d'ensemble, nous avons également tracé (Fig. 2.23(d)) la valeur de la fonction fitness en fonction du nombre d'itérations. Nous constatons que l'algorithme converge relativement rapidement, en 36 itérations, pour un gain d'un facteur 6 sur la fonction fitness. Ensuite, nous ne gagnons plus qu'un facteur 2 sur les 950 itérations suivantes. Le résultat est relativement proche de la fonction désirée. L'avantage de cette approche est qu'elle ne contient pas de pôle négatif qui pourrait rendre instable le schéma FDTD pour la résolution des équations de lignes.

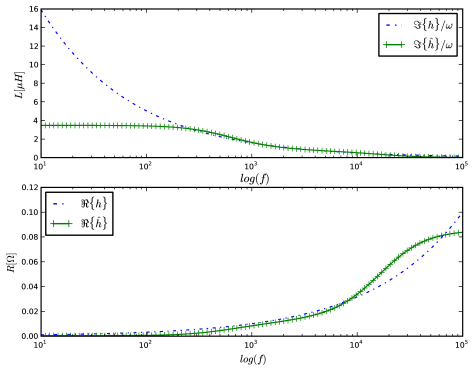
Approximation de l'impédance Z_s du sol Nous allons maintenant tester notre algorithme sur l'impédance Z_s du sol dans le cas d'une ligne de transmission définie par $r = 1\text{cm}$, $h = 5\text{m}$, $\sigma_s = 0.1\text{mS/m}$. Nous ne traitons pas directement l'impédance linéique de la ligne de transmission, mais nous utilisons la décomposition naturelle de l'impédance totale $Z = jL\omega + Z_s$. Dans cette expression, Z_s peut être vu comme l'impédance R du modèle de Heaviside qui dans ce cas est complexe. Nous cherchons donc à factoriser Z_s en une série de Debye $\hat{Z}_s$.

Pour factoriser une impédance, l'utilisation d'une fonction objectif basée sur l'erreur commise sur l'inductance $\Im\{Z_s\}/\omega$, semble plus pertinente que la seule mesure de l'erreur sur la partie imaginaire, car l'inductance L entre directement en compte dans la résolution numérique.

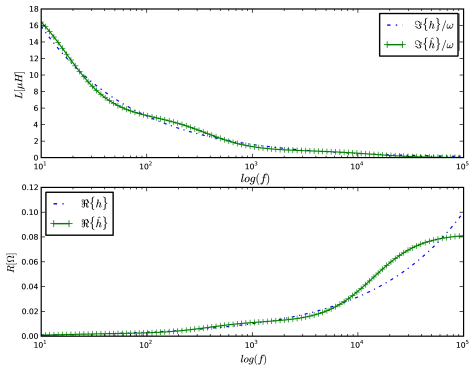
Comme pour la fonction Cole-Cole, nous avons tracé (Fig. 2.24) l'évolution de la solution en fonction du nombre d'itérations. De manière générale, nous constatons que l'algorithme converge rapidement vers une solution acceptable en moins de 300 itérations. Au-delà, la solution ne varie pas beaucoup.

À première vue, le résultat final donne une relative concordance entre les deux fonctions. Les grandes tendances sont représentées. Le défaut, qui visuellement est le plus marqué, est dans les hautes fréquences et sur la partie réelle. Toutefois, ce défaut ne remet pas en cause la validité de notre approche car dans cette gamme de fréquences les pertes sont importantes.

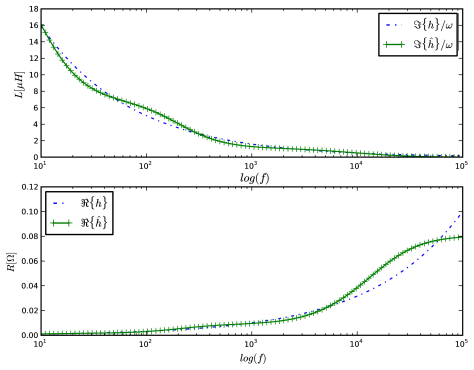
Ainsi, ce défaut ne concerne pas un domaine faiblement énergétique qui se propage peu. L'erreur sur le résultat final ne devrait pas être significatif.



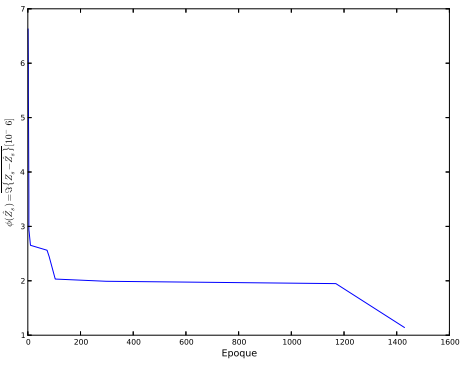
(a) La meilleure solution trouvée après la phase d'initialisation.



(b) La 12ème meilleure solution trouvée après 297 itérations.



(c) La 14ème meilleure solution trouvée en moins de 1000 itérations, cette solution a été trouvé à 1429 itérations.

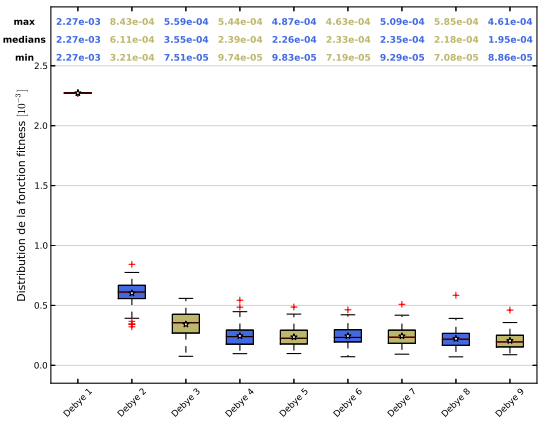


(d) évolution de la fonction fitness en pourcentage d'erreur par rapport à l'inductance propre L de la ligne pour les meilleures solutions trouvées en fonction de l'itération.

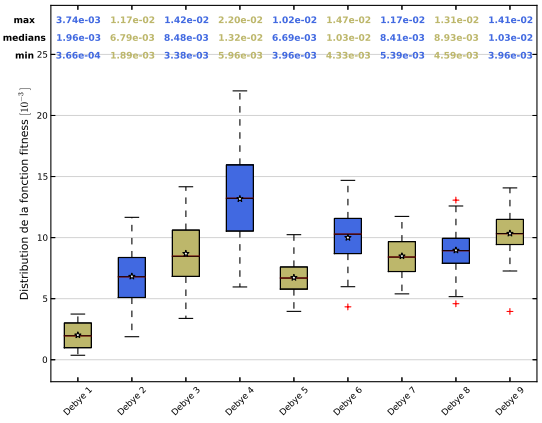
FIGURE 2.24 – Hybrid Particle Swarm-Least Quares optimization d'une l'impédance linéique d'une ligne au-dessus de sol pour 5 pôles. L'interpolation est faite avec 25 points répartis sur une échelle logarithme. L'optimisation a été faite avec 100 particules sur 5000 itérations.

a. Analyse statistique de la convergence

Pour terminer cette section, nous allons observer la convergence vers la solution finale par une approche statistique. D'une part, nous avons voulu quantifier la « qualité » de notre outil de factorisation, mais nous nous sommes confrontés à une difficulté majeure. En effet, l'étude théorique de ce type d'outils est très difficile et dans les quelques cas où elle est envisageable, elle est faite en termes de probabilités. Nous nous sommes tournés vers une étude statistique de la convergence de notre méthode ce qui nous donnera une idée de sa « qualité ».



(a) Distribution des résultats pour 100 factorisations par famille de polynômes de Debye allant de 1 à 9 pôles



(b) Distribution des résultats pour 100 tirages aléatoires par famille de polynômes de Debye allant de 1 à 9 pôles

FIGURE 2.25 – Comparaison de la distribution des résultats pour 100 résultats par famille de polynômes de Debye allant de 1 à 9 pôles

L'objectif de cette étude est double, d'une part nous voulons voir quelle est la disparité de N solutions à un même problème. Ainsi, plus les résultats obtenus pour un même problème sont proches, plus notre outil est performant. Cette performance décrit la capacité de l'algorithme à toujours trouver les meilleures solutions qui doivent être comparables. D'autre part le deuxième objectif est de comparer les résultats de notre algorithme par rapport à un tirage aléatoire. Cette comparaison nous permet de quantifier le gain apporté sur la solution finale par rapport à un choix aléatoire de polynôme de Debye. Le tirage aléatoire nous donne également des ordres de grandeur de la fonction objectif dans un cas proche, ce qui nous permet d'avoir une référence. La figure 2.25 nous donne la répartition de 100 résultats pour des polynômes allant de 1 à 9 pôles.

Revenons sur la notion de tirage presque aléatoire. Dans notre problème, nous cherchons à trouver le polynôme de Debye le plus proche d'une fonction donnée sous la contrainte que toutes les amplitudes soient positives. Pour faire ce tirage, nous avons choisi d'utiliser les résultats de la factorisation pour déterminer les paramètres h_s, h_∞, a, τ puis un tirage uniforme est fait pour déterminer les valeurs de a et τ. Ainsi, notre tirage aléatoire est déjà largement conditionné par le problème initial, ce qui évite d'avoir des résultats complètement divergents.

Les résultats obtenus par tirage aléatoire sont répartis sur un intervalle relativement grand (max-min $\approx 10^{-2}$). En revanche, en utilisant l'algorithme de factorisation, nous obtenons un intervalle de $3 \cdot 10^{-4}$ (en omettant le cas où il y a un seul pôle). La stratégie proposée converge donc bien vers des solutions comparables. De plus, nous remarquons que la solution de factorisation donne des solutions de fitness 10 fois inférieures au tirage aléatoire ; nous avons donc également la propriété de recherche d'un minimum.

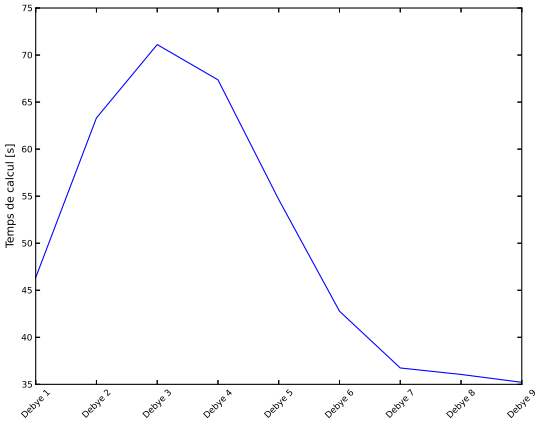


FIGURE 2.26 – Temps moyen sur 100 factorisations pour obtenir la solution finale en fonction du nombre de pôles .

Ces résultats ne sont pas une démonstration de convergence mais permettent d'observer les propriétés indispensables que doivent avoir ce type d'outil : la convergence vers un ensemble de solutions comparables minimisant la fonction fitness en un temps raisonnable.

Nous avons évalué les deux premiers points, maintenant nous allons observer (Fig. 2.26) le temps moyen pour obtenir la solution finale.

Nous constatons que le temps moyen augmente pour atteindre son maximum pour des polynômes de Debye à 3 pôles, puis décroit rapidement jusqu'aux polynômes à 7 pôles où il semble que la tendance se stabilise.

Cette évolution est tout à fait logique pour les polynômes de Debye avec un petit nombre de pôles (de 1 à 3). L'algorithme dispose d'un petit espace de recherche, il est donc fortement contraint et a du mal à trouver des solutions. Maintenant si nous augmentons légèrement le nombre de pôles (plus 1 ou 2 par exemple) nous libérons des contraintes qui simplifient la recherche sans alourdir l'algorithme. Par contre, si nous augmentons davantage (trop) le nombre de pôle, la complexité induite par le nombre de pôles devient supérieure au gain produit par la libération des contraintes apportée par un polynôme plus grand. La stabilisation obtenue à partir de 7 pôles traduit un équilibre.

2.4.3 Modèle de Debye dans le formalisme des lignes de transmission - Mise en œuvre et validation

Dans cette section, nous allons aborder l'introduction d'un modèle de Debye dans le formalisme FDTD pour les lignes de transmission et nous donnons une validation de ce modèle par comparaison avec l'approche fréquentielle BLT (Annexe F page 248).

Soient les équations électriques en fréquentiel le long de la ligne de transmission :

$$\partial_l \widetilde{v} = -(R + jL\omega) \cdot \widetilde{i} \tag{2.96a}$$

$$\partial_l \widetilde{i} = -(G + jC\omega) \cdot \widetilde{v} \tag{2.96b}$$

Dans un premier temps, nous considérons que seulement L dépend de la fréquence, que l'on va noter $\hat{L}(\omega)$. Nous utilisons le modèle de Debye multi-polaire pour décrire $\hat{L}(\omega)$ qui est une grandeur complexe. Le développement fait ici pour L peut également être utilisé pour introduire une dépendance en fréquence sur le terme capacitif C.

$$\hat{L}(\omega) = L_\infty + (L_s - L_\infty) \sum_{i=1}^{N} \frac{a_i}{1 + j\omega\tau_i} = L_\infty + \sum_{i=1}^{N} \frac{L_i}{1 + j\omega\tau_i} \tag{2.97}$$

avec $L_i = a_i \, (L_s - L_\infty)$. Enfin, τ_i désigne le point de relaxation temporelle.
En introduisant (Eq. 2.97) dans (Eq. 2.96) :

$$-\partial_l \widetilde{v} = (R + jL_\infty\omega) \cdot \widetilde{i} + \sum_{i=1}^{N} \frac{j\omega L_i}{1 + j\omega\tau_i} \cdot \widetilde{i} \tag{2.98}$$

$$= (R + jL_\infty\omega) \cdot \widetilde{i} + \sum_{i=1}^{N} \widetilde{J}_i \tag{2.99}$$

avec :

$$\widetilde{J}_i = \frac{j\omega L_i}{1 + j\omega\tau_i} \cdot \widetilde{i} \tag{2.100}$$

Notons que (Eq. 2.99) peut se résoudre sans difficulté après un relèvement en temps, avec la méthode FDTD 1D présentée en (Sec. a. page 57). La deuxième équation (Eq. 2.100) multipliée par $1 + j\omega\tau_i$ donne :

$$\widetilde{J}_i + j\omega\tau_i \widetilde{J}_i = j\omega L_i \widetilde{i} \tag{2.101}$$

qui, par transformation de Fourier inverse, nous donne :

$$J_i + \tau_i \partial_t J_i = L_i \partial_t i \tag{2.102}$$

Cette dernière équation est l'EDO d'un problème auxiliaire que l'on va résoudre avec le schéma semi-implicite en temps suivant :

$$J_i|^{n+1/2} = \alpha_i \; J_i|^{n-1/2} + \xi_i \Big(I|^{n+1/2} - I|^{n-1/2} \Big) \tag{2.103}$$

avec :

$$\alpha_i = \frac{2\tau_i - \Delta t}{2\tau_i + \Delta t} \tag{2.104a}$$

$$\xi_i = \frac{2L_i}{2\tau_i + \Delta t} \tag{2.104b}$$

Dans notre choix de discrétisation temporelle, rappelons que l'équation (2.99) est évaluée aux instants $t = n\Delta t$. Nous devons donc évaluer J_i en n.

$$J_i|^{n} = \frac{1}{2}\left(J_i|^{n+1/2} + J_i|^{n-1/2} \right) = \frac{1}{2}\left[(1+\alpha_i) \; J_i|^{n-1/2} + \frac{\xi_i}{\Delta t}\left(I|^{n+1/2} - I|^{n-1/2} \right) \right] \tag{2.105}$$

Ce qui nous permet d'écrire pour chaque segment p l'équation de mise à jour de la tension avec les variables auxiliaires :

$$\begin{aligned} -\frac{1}{\Delta l}\left(V|_{p+1}^{n} - V|_{p}^{n} \right) &= \frac{R}{2} \cdot \left(I|_{p+1/2}^{n+1/2} + I|_{p+1/2}^{n-1/2} \right) + \frac{L_\infty}{\Delta t} \cdot \left(I|_{p+1/2}^{n+1/2} - I|_{p+1/2}^{n-1/2} \right) \\ &+ \frac{1}{2}\sum_{i=1}^{N} \left[(1+\alpha_i) \; J_i|_{p+1/2}^{n-1/2} + \frac{\xi_i}{\Delta t}\left(I|_{p+1/2}^{n+1/2} - I|_{p+1/2}^{n-1/2} \right) \right] \\ &= \frac{R}{2} \cdot \left(I|_{p+1/2}^{n+1/2} + I|_{p+1/2}^{n-1/2} \right) + \frac{1}{2}\sum_{i=1}^{N} \left[(1+\alpha_i) \; J_i|_{p+1/2}^{n-1/2} \right] \\ &+ \left(\frac{L_\infty}{\Delta t} + \frac{1}{2}\sum_{i=1}^{N} \frac{\xi_i}{\Delta t} \right) \cdot \left(I|_{p+1/2}^{n+1/2} - I|_{p+1/2}^{n-1/2} \right) \end{aligned} \tag{2.106}$$

Nous considérons maintenant le cas où seulement R dépend de la fréquence que l'on note $\hat{R}(\omega)$. Nous pouvons de la même manière, introduire la dépendance en fréquence sur G par analogie avec R.

Comme pour L nous re-écrivons l'EDP (Eq. 2.96b) pour faire sortir les variables auxiliaires K_j que nous discrétisons avec un schéma semi-implicite.

$$K_j|^{n+1} = \beta_j \; K_j|^{n} + \zeta_j \; I|^{n+1/2} \tag{2.107}$$

avec :

$$\beta_j = \frac{2\tau_j - \Delta t}{2\tau_j + \Delta t} \tag{2.108a}$$

$$\zeta_j = \frac{\Delta t R_j}{2\tau_j + \Delta t} \tag{2.108b}$$

Ce qui nous permet d'écrire pour chaque segment p de la ligne :

$$-\frac{1}{\Delta l}\left(V|_{p+1}^{n}-V|_{p}^{n}\right) = \frac{R_\infty}{2}\cdot\left(I|_{p+1/2}^{n+1/2}+I|_{p+1/2}^{n-1/2}\right)+\frac{L}{\Delta t}\cdot\left(I|_{p+1/2}^{n+1/2}-I|_{p+1/2}^{n-1/2}\right) + \sum_{j=1}^{M}\left[\beta_j\ K_j|_{p+1/2}^{n-1}+\zeta_j\ I|_{p+1/2}^{n-1/2}\right] \tag{2.109}$$

Nous pouvons tout à fait considérer le cas où R et L dépendent de la fréquence. Dans ce cas de figure, les équations de mise à jour de la tension et des deux variables auxiliaires deviennent :

$$I|_{p+1/2}^{n+1/2}\cdot\left(2L_\infty+R_\infty\Delta t+\sum_{i=1}^{N}\xi_i\right) = I|_{p+1/2}^{n-1/2}\cdot\left(2L_\infty+R_\infty\Delta t+\sum_{i=1}^{N}\xi_i-2\Delta t\sum_{j=1}^{M}\zeta_j\right) -\Delta t\sum_{i=1}^{N}\left[(1+\alpha_i)\ J_i|_{p+1/2}^{n-1/2}\right]-2\Delta t\sum_{j=1}^{M}\left[\beta_j\ K_j|_{p+1/2}^{n-1}\right] -\frac{2\Delta t}{\Delta l}\left(V|_{p+1}^{n}-V|_{p}^{n}\right) \tag{2.110a}$$

$$J_i|^{n+1/2} = \alpha_i\ J_i|^{n-1/2}+\xi_i\Big(I|^{n+1/2}-I|^{n-1/2}\Big) \tag{2.110b}$$

$$K_j|^{n-1} = \beta_j\ K_j|^{n-2}+\zeta_j\ I|^{n-3/2} \tag{2.110c}$$

avec :

$$\alpha_i=\frac{2\tau_i-\Delta t}{2\tau_i+\Delta t} \qquad \beta_j=\frac{2\tau_j-\Delta t}{2\tau_j+\Delta t}$$
$$\xi_i=\frac{2L_i}{2\tau_i+\Delta t} \qquad \zeta_j=\frac{\Delta t R_j}{2\tau_j+\Delta t}$$

2.4.4 Validation et étude de la factorisation en série de Debye

Dans cette section, nous allons valider l'intégration du modèle de Debye dans une méthode FDTD temporelle pour résoudre les équations de ligne de transmission. Nous ferons cette validation en nous comparant à une méthode fréquentielle basée sur l'équation BLT. Nous regarderons également comment se comporte la factorisation en série de Debye face au schéma numérique temporel. Cette dernière étude sera faite en comparant par rapport à la décomposition en pôles résidus.

a. Cas tests et méthodologie

Tout d'abord, pour valider notre méthode de prise en compte de paramètres dépendants de la fréquence dans une ligne de transmission, nous allons présenter le cas test que nous considérons et la méthodologie utilisée pour comparer les approches fréquentielle et temporelle.

Le cas test proposé se compose d'une ligne bifilaire de longueur l et est définie par des paramètres électriques $RLGC$ qui peuvent dépendre de la fréquence. Cette ligne de transmission est excitée par un générateur de Thevenin E_{th} et R_{th}, et est chargée par une résistance R_s. Le tableau 2.4.1 donne les principales grandeurs numériques utilisées pour ce cas test.

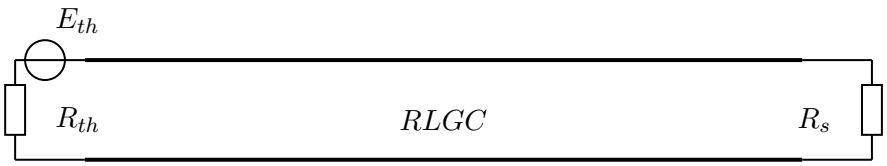


FIGURE 2.27 – Cas test de validation du modèle de Debye

Paramètre	Valeur
l	3000m
L	1.0596µH
C	9.2853pF
R_s	$\sqrt{L/C}$
R_{th}	$\sqrt{L/C}$

TABLE 2.7 – Valeurs utilisées pour valider le modèle de Debye dans le modèle des lignes de transmission

Remarque 2.4.1. *Nous avons choisi de charger la ligne de transmission avec une résistance R_s qui est l'impédance caractéristique de la ligne pour les valeurs fixées dans le tableau .*

Dans cette étude, nous nous sommes limités à l'étude où seule l'impédance linéique $Z = R + jL\omega$ peut avoir une dépendance en fréquence. Cette dépendance en fréquence est introduite en remplaçant R par une série de Debye. Les valeurs numériques utilisées pour la série de Debye dépendront du cas considéré. Dans ce cas de figure, l'impédance linéique s'écrit :

$$\hat{Z}(\omega) = \hat{R}(\omega) + jL\omega = R'(\omega) + jL'(\omega)\omega \tag{2.111}$$

Avec $\hat{R}(\omega)$ une série de Debye définie comme suit :

$$\hat{R}(\omega) = R_\infty + (R_0 - R_\infty)\sum_{p=1}^{N} \frac{a_p}{1 + j\tau_p\omega} \tag{2.112}$$

$$= R_\infty + \sum_{p=1}^{N} \frac{\Delta_p}{1 + j\tau_p\omega} \tag{2.113}$$

Ce qui nous donne :

$$R'(\omega) = R_\infty + \sum_{p=1}^{N} \frac{\Delta_p}{1+(\tau_p\omega)^2} \tag{2.114}$$

$$L'(\omega) = L - \sum_{p=1}^{N} \frac{\Delta_p\tau_p}{1+(\tau_p\omega)^2} \tag{2.115}$$

La comparaison temps-fréquence se fait sur un spectre normalisé. C'est à dire que la source fréquentielle vaut 1 sur la bande de fréquence étudiée et 0 ailleurs. Le résultat temporel devra donc être post-traité pour effectuer des comparaisons temps/fréquence. Ainsi, nous normaliserons le spectre de la solution (obtenue par transformée de Fourier) par le spectre de la source choisie.

$$\breve{I}(\omega) = \frac{TF\{i(t)\}}{\widetilde{I_s}(\omega)} \tag{2.116}$$

Avec $\breve{I}(\omega)$ le spectre reconstruit, $i(t)$ le courant mesuré et $\tilde{I_s}(\omega)$ le spectre de la source que l'on peut avoir analytiquement ou par transformée de Fourier.

Nous pouvons utiliser plusieurs types de fonctions temporelles (gaussienne, dérivée de gaussienne, sinus cardinal) pour calculer la réponse normalisée. Il faut juste s'assurer que le spectre de la source soit convenable par rapport à la bande de fréquences que nous voulons étudier ; c'est-à-dire que le spectre d'étude désiré soit entièrement couvert avec un niveau suffisant par la source temporelle choisie. Nous donnons trois choix possibles de sources avec les paramètres adéquats pour chacune d'entre-elles. Ces différentes sources sont définies comme suit :

– Gaussienne :

$$E_{gauss}(t) = \exp(-(\alpha * (t-\tau))^2) \tag{2.117}$$

Avec $\alpha = \frac{\pi * f_{max}}{\sqrt{\log(10)}}$ et $\tau = \frac{\sqrt{\log(10^{-5})}}{\alpha}$ ce qui nous permet d'avoir une amplitude à f_{max} de 10% de la valeur maximale et $E_g(0) = 10^{-5}$

– Dérivée de gaussienne

$$E_{dgauss}(t) = -\sqrt{2.0}\alpha(t-\tau)\exp(0.5)\exp(-\alpha(t-\tau)^2) \tag{2.118}$$

Avec $\alpha = \sqrt{2.0}\pi f_c$ et $\tau = 3.3565/(\sqrt{2.0}\pi f_c)$ et $f_c = 0.5(f_{max} + f_{min})$ la fréquence centrale de la source.

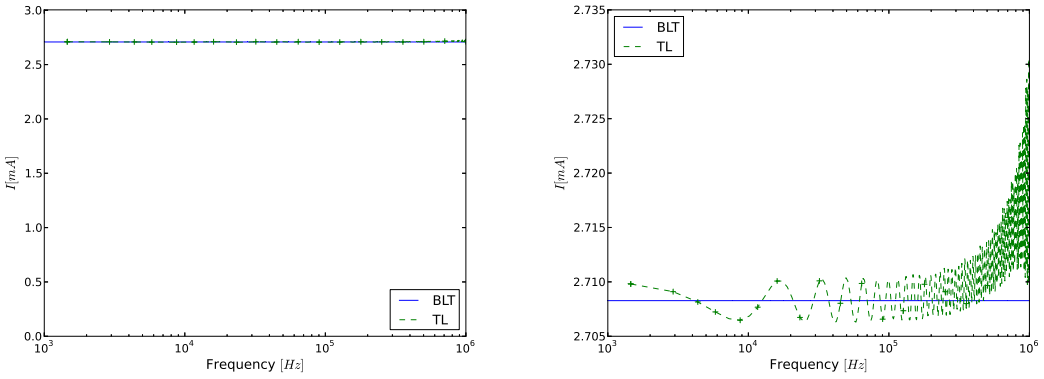
– Sinus cardinal

$$E_{sinc}(t) = \frac{\sin(\pi\alpha(t-\tau))}{\pi\alpha(t-\tau)} \tag{2.119}$$

Avec $\alpha = 2.0(f_{max} + f_{min})$ et τ sont choisis de manière à avoir les bonnes conditions initiales.

L'utilisation de la fonction sinus cardinal est attractive car elle représente la réponse temporelle de la source d'amplitude constante dans la bande $[f_{min}, f_{max}]$. En revanche, elle impose un temps de simulation prohibitif dû au temps nécessaire pour revenir à zéro. La dérivée de gaussienne a l'avantage d'avoir une valeur moyenne nulle, c'est pourquoi nous l'avons préférée.

Une première comparaison entre notre approche FDTD temporelle et l'équation BLT fréquentielle a été réalisée dans le cas sans perte et sans dépendance en fréquence des paramètres L, C. Dans cette configuration, la ligne de transmission est adaptée car elle est chargée par l'impédance caractéristique $R_s = \sqrt{Z/Y} = Z_c$. De plus, comme il n'y a pas de perte dans la ligne, le courant mesuré en bout de ligne a la même amplitude que le courant injecté qui est : $\widetilde{I}_s = \frac{\widetilde{E}_{th}}{R_{th}+Z_c} \approx 2.71\text{mA}$.



(a) (b)

FIGURE 2.28 – Courants évalués sur l'extrémité droite de la ligne du cas test considérée sans dépendance en fréquence, par notre approche FDTD temporelle et l'équation BLT.

Le résultat fréquentiel est tout à fait correct, en revanche le résultat temporel corrigé (TF puis normalisation) présente des oscillations d'une amplitude de 3.65µA, soit une erreur relative de 1% sur la solution. Nous obtenons bien des solutions quasi-similaires. Nous constatons aussi que l'erreur croit avec la fréquence. Cette erreur est certainement due à l'approximation faite dans la condition aux limites de type adaptation (Sec. 2.3.2 page 62) qui fait l'hypothèse que $V|_p^n \approx Z_c\, I|_{p+1/2}^{n-1/2}$.

b. Validation de l'intégration de la dépendance en fréquence

Nous avons choisi d'étudier la dépendance en fréquence pour trois termes de Debye avec 1, 2 et 5 pôles. Le tableau 2.8 nous donne les paramètres de ces trois termes et la figure 2.29 nous donne l'évolution de l'impédance. Nous avons également étudié ce problème avec des termes de Debye qui ont une amplitude négative.

Remarque 2.4.2. *Nous avons choisi de ne pas utiliser R_s mais R_0 dans la définition du terme de Debye. R_0 est la limite de $\hat{R}$ en $\omega \to 0$. Nous pouvons passer de R_0 à R_s par la*

relation $R_s = R_\infty + (R_0 - R_\infty)/\sum a_p$.

Nous avons choisi ce paramètre car il nous permet d'avoir la même limite en 0 *pour toutes nos séries de Debye.*

Terme	$R_0[\Omega]$	$R_\infty[\Omega]$	$\tau[\mu s]$	a
$\hat{R}_1$	0.	0.1	3.183	1
$\hat{R}_2$	0.	0.1	$3.183, 0.318$	$1.0, 0.5$
$\hat{R}_5$	0.	0.1	$3.183, 1.790, 1.007, 0.566, 0.318$	$0.102, 0.416, 0.418, 0.022, 0.786$
$\hat{R}_2^*$	0.	0.1	$3.183, 0.318$	$1.0, -0.5$
$\hat{R}_5^*$	0.	0.1	$3.183, 1.790, 1.007, 0.566, 0.318$	$0.102, 0.416, -0.418, 0.022, 0.786$

TABLE 2.8 – Paramètres des termes de Debye.

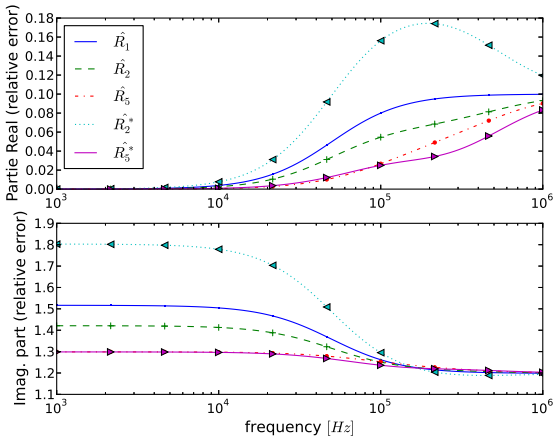
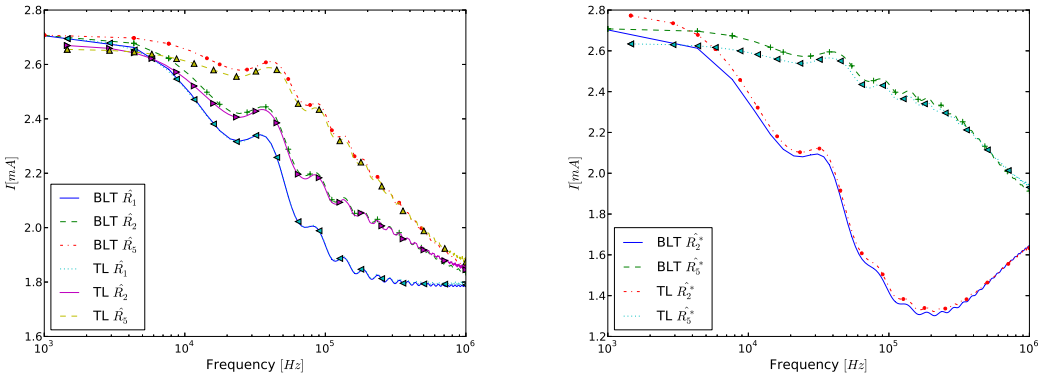


FIGURE 2.29 – Évolution de l'impédance $Z(\omega)$.

En reprenant le cas test défini au début de cette section, nous mesurons le courant en bout de ligne pour les différentes configurations de série de Debye. Le premier résultat que nous allons considérer est le cas où nous avons que des termes de Debye avec des amplitudes positives (Fig. 2.30(a)). Nous constatons une très bonne concordance dans le cas où nous avons un seul pôle. Dans le cas où nous avons plusieurs pôles, il semble que les performances de la série de Debye (somme des variables auxiliaires) trouvent leurs limites en particulier dans les basses fréquences. Ce résultat traduit le fait que l'implémentation des termes de Debye n'utilise que l'instant précédent pour faire la « convolution ». Les phénomènes lents qui se produisent sur un grand nombre d'itérations temporelles vont être difficiles à considérer.

La figure suivante 2.30(b) montre deux autres série de Debye avec un pôle négatif. Il est difficile d'établir un critère de stabilité numérique dans un schéma aux différences finies pour une série de Debye. Nous utilisons couramment comme critère $\forall p, a_p > 0$, qui est l'extension du cas où nous avons un seul pôle de Debye [50]. Ce critère permet de conserver la stabilité mais c'est un critère un peu restrictif. Nous voyons que dans les deux cas choisis rapidement (en inversant le signe d'un pôle), nous avons un résultat qui semble tout à fait correct.



(a) (b)

FIGURE 2.30 – Validation du modèle de Debye par comparaisons avec la solution fréquentielle dans le cas sans pôle négatif (a) et avec un pôle négatif (b)

c. Comparaison de la procédure de Factorisation et de la décomposition par Résidu

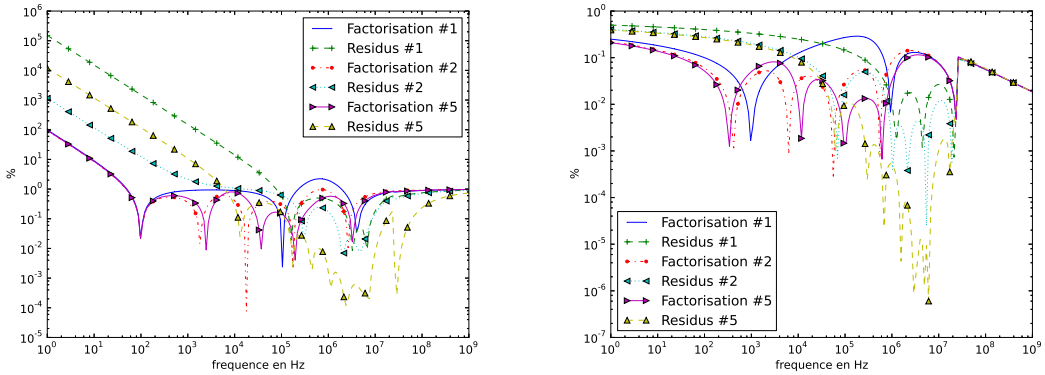
Dans ce dernier cas test, nous comparons la méthode de factorisation en série de Debye avec la technique qui utilise la décomposition en pôles et résidus (Sec. a. page 73) pour le calcul des paramètres fréquentiels de la ligne. Nous avons écarté la dernière approche car elle ne permet pas de contrôler le signe de l'« amplitude » de chaque pôle de la factorisation en série de Debye. Néanmoins, nous avons déjà évoqué le fait que d'avoir des pôles négatifs n'implique pas forcément l'instabilité. Nous allons donc tenter d'évaluer quelle est la méthode la plus pertinente pour tenir compte de la dépendance en fréquence des paramètres dans une méthode FDTD temporelle résolvant les équations de lignes de transmission.

Pour faire cette étude nous allons considérer le cas test présenté au début de cette section, en supposant que la ligne est au-dessus d'un sol de conductivité finie $\sigma = 100\text{S/m}$, à une hauteur $h = 1\text{m}$. Le rayon de la ligne est donné par $a = 1\text{cm}$. L'impédance de la ligne est donnée par l'approximation (Sec. 2.2.2 page 39) de E. F. Vance.

Le premier résultat à considérer est l'erreur directe sur l'approximation de l'impédance par les deux méthodes précédemment vues. Dans la figure (Fig. 2.31), nous donnons l'erreur relative pour R et L commise par les deux approches et pour différents nombres de pôles.

Le premier constat que nous pouvons faire est que pour les deux méthodes, le terme de perte (R) est mal approché. Ce résultat est à relativiser dans ce cas de figure, l'impédance est très petite et tend vers zéro quand la fréquence tend également vers zéro. En revanche, pour l'inductance, l'erreur est au moins quatre fois plus petite que la valeur recherchée. Notons également que la factorisation donne un résultat très satisfaisant relativement aux autres et cela même pour un nombre de pôles faible. C'est une propriété intéressante, car chaque pôle a un coût de calcul mais surtout l'utilisation abusive de pôle nuit aux résultats.

La figure 2.32(a) donne une comparaison entre les courants en bout de ligne pour une décomposition par l'approche des résidus et par factorisation pour deux pôles de décompo-



(a) (b)

FIGURE 2.31 – Erreur relative sur R (a) et L (b).

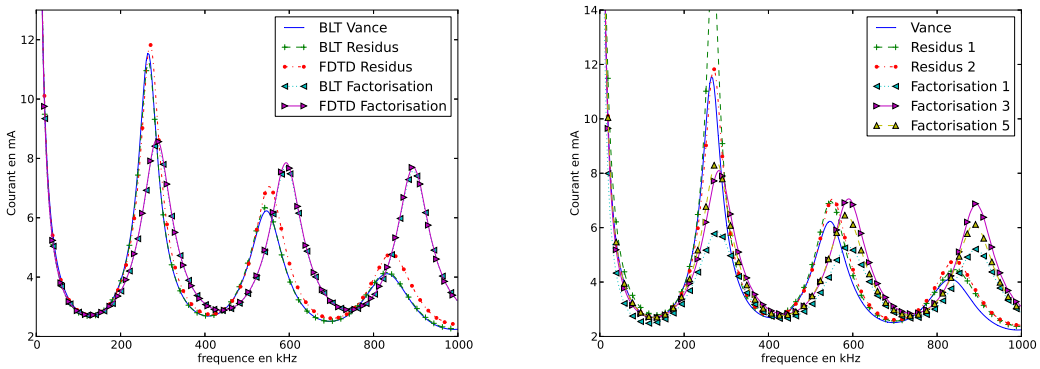
sition. Le résultat obtenu par le code de ligne est normalisé par le spectre de la source et comparé avec l'approche BLT.

Le résultat obtenu en utilisant la décomposition par résidus est très clairement plus proche du résultat obtenu par le modèle de E. F. Vance en fréquentiel et nous notons que la décomposition obtenue par la méthode des résidus ne donne pas le même résultat en fréquentiel et en temporel. Nous remarquons aussi un petit défaut en amplitude et en fréquence qui est plus marqué dans les hautes fréquences. C'est un résultat très intéressant, car il montre qu'une très bonne décomposition peut mal se prêter à une utilisation en temporel. En revanche, l'utilisation d'une décomposition en série de Debye obtenue par factorisation donne un résultat moins précis mais identique quand nous passons dans le domaine temporel. La factorisation se prête donc mieux à une utilisation en FDTD.

Remarquons également (Fig. 2.32(b)) que l'utilisation d'une décomposition avec un grand nombre de pôles ne garantit pas une meilleure solution. De plus, en utilisant les séries obtenues par la procédure des résidus, la solution a divergé pour la décomposition avec 3, 4 et 5 pôles. La factorisation nous permet bien de conserver la stabilité au détriment d'une imprécision qui reste maîtrisée (minimisation) ; cependant, il y a des cas où ce minimum peut être très éloigné de la solution recherchée.

d. Conclusion

La stratégie de factorisation proposée apporte un gain certain. Elle permet de trouver une décomposition satisfaisante d'une impédance complexe sous la forme d'une série de Debye. En revanche, cette approche souffre d'une lacune. En effet, elle ne permet pas d'optimiser le nombre de pôles nécessaires pour la factorisation. Ce paramètre n'est pas intuitif à estimer. De plus, il est intéressant (Sec. 2.4.3 page 88) d'avoir le nombre minimum de pôles pour avoir une plus grande précision numérique. L'heuristique d'optimisation par « essaims particulaires » impose que tous les individus aient le même nombre de pôles pour pouvoir calculer le déplacement des individus. En revanche, l'utilisation d'un « algorithme génétique » par



(a) Comparaison des courants en bout de ligne obtenus par deux méthodes (FDTD et BLT) pour des séries de Debye à deux pôles obtenus par la méthode de résidus et la factorisation.

(b) Comparaison des courants en bout de ligne obtenus en FDTD pour différents séries de Debye (1, 2, 3 et 5 pôles) obtenus par la méthode de résidus et la factorisation.

FIGURE 2.32 – Comparaison du courant terminal pour une ligne dispersive obtenu par différentes approches.

exemple, permettrait d'introduire le nombre de pôles dans l'optimisation, dans le but de le minimiser.

2.5 Conclusion

Dans ce chapitre, nous avons présenté la mise en œuvre du code de ligne temporel développé dans le cadre de cette thèse. Au travers de quelques cas tests, nous avons validé notre approche, d'abord dans le cas d'un plan de masse parfaitement métallique, puis dans le cas d'une ligne au-dessus d'un sol, avec prise en compte de la dépendance en fréquence des paramètres de ligne. La technique proposée consiste à faire une approximation des formulations de E. F. Vance par une série de Debye. Pour cela, un algorithme original de calcul des paramètres d'une série a été proposé et validé pour le cas d'approximation d'impédances de ligne, cas qui nous intéresse ici. Enfin, nous avons montré l'intérêt d'avoir une série de Debye pour rendre compte des effets dispersifs dans un code temporel et nous avons validé notre approche par une comparaison avec le code BLT fréquentiel.

Chapitre 3

Hybridation champ/câbles - Conducteurs de géométrie arbitraire

Sommaire

3.1 Introduction 100
3.2 Le formalisme de fil mince en FDTD 101
3.2.1 Le fil mince de R. Holland 101
3.2.2 Le modèle de fils minces de Berenger 103
3.3 Nouvelle approche : le formalisme pour des sections à géométrie arbitraire 105
3.3.1 Détermination des équations de ligne de transmission dans le cas d'une géométrie de conducteurs quelconques 105
3.3.2 L'équation de Laplace pour calculer les paramètres de ligne de transmission 107
3.4 Calcul des paramètres de la ligne de transmission 109
3.4.1 Paramètres dans la cellule FDTD (In-Cell) 109
3.4.2 Paramètres en espace ouvert 109
3.4.3 Paramètres avec une référence dans la cellule FDTD 110
3.5 Couplage avec le domaine 3D - Approximation du mode commun 111
3.6 Validation numérique 114
3.6.1 Le fil simple 114
3.6.2 La bifilaire 115
3.6.3 Multi-filaire 116
3.6.4 Ligne de transmission 117
3.6.5 Nœud de courant 119
3.6.6 Blindage et reprise de coupure 121
3.7 Conclusion 124

3.1 Introduction

Les réseaux de câbles jouent un rôle important dans l'étude de la compatibilité électromagnétique, d'une part, parce qu'ils sont les voies de couplage privilégiées et d'autre part, parce qu'ils assurent un rôle fonctionnel. En particulier, souvent, les câbles sont les principales sources de perturbations pour les équipements.

En définitive, la bonne modélisation des réseaux de câbles et de leur complexité est devenue un enjeu industriel majeur. L'introduction de modèles de fils minces, que ce soit en FDTD, (le fil mince de Holland [19] d'une part et le contour path de Taflove [20] d'autre part), ou dans d'autres méthodes [21, 22], a été nécessaire pour traiter de nombreux problèmes. Ces modèles ont ensuite évolué pour traiter la problématique multi-conducteurs dans la cellule [23], ou encore la position des fils dans le maillage (fils obliques) [24, 25, 26, 22].

Toutefois, il existe encore aujourd'hui, un certain nombre de points délicats non résolus, parmi lesquels nous pouvons citer :

- la gestion des nœuds de courant quand plusieurs ensembles de câbles non connectés suivent une même arête du maillage 3D, puis se séparent ;
- la gestion des routages de câbles complexes composés de conducteurs, de gaines isolantes, de blindages et de sur-blindages ;
- la modélisation des conditions aux limites, qui est un problème relativement simple dans le cas d'une bifilaire, mais qui devient nettement plus compliqué dans le cas d'un faisceau pouvant atteindre la centaine de câbles.

Dans ce chapitre, nous allons maintenant discuter d'une stratégie hybride FDTD-3D/MTL, dite « d'échelle », afin de donner une solution performante et précise pour traiter la complexité des réseaux de câbles dans les systèmes. Nous faisons dans un premier temps un rapide rappel sur le modèle de fil de R. Holland, puis nous présentons le formalisme de J.P. Berenger qui peut être vu comme l'extension du fil de R. Holland au cas multi-filaire. Ensuite, nous introduisons un nouveau formalisme dont le principe consiste à coupler un code de lignes de transmission qui modélise les conducteurs avec les champs du domaine 3D. Les paramètres de la ligne sont alors obtenus en résolvant un problème de Laplace. Enfin, pour terminer, nous présentons plusieurs applications numériques montrant la capacité de cette approche à traiter des problèmes complexes qu'il aurait été difficile de réaliser avec les approches actuelles.

3.2 Le formalisme de fil mince en FDTD

3.2.1 Le fil mince de R. Holland

Le formalisme de R. Holland (1981) [19] permet de traiter les fils minces parallèles aux arêtes du maillage. Les équations internes (Eq. 3.1a) qui définissent l'évolution du courant et de la charge, sont déterminées à partir de l'équation de Maxwell-Faraday. Pour arriver à cette formulation, l'hypothèse quasi-statique dans la cellule FDTD-3D est faite pour moyenner le champ dans la cellule. Les paramètres de la ligne de transmission apparaissent naturellement dans ce formalisme et se présentent comme l'intégration géométrique de la distribution du champ dans la cellule. Enfin, à cette première équation, est associée l'équation de conservation de la charge (Eq. 3.1b). Les équations sur les fils sont alors données par :

$$L_h \left(\partial_t i + \nu^2 \partial_l q\right) + Ri = \langle E_l \rangle \tag{3.1a}$$

$$\partial_t q + \frac{\sigma}{\varepsilon} q + \partial_l i = 0 \tag{3.1b}$$

Avec $L_h = (\mu/2\pi)\langle \ln(r/a) \rangle$, $\nu^2 = c^2/\varepsilon_r$ la vitesse de propagation dans le milieu en m/s, a le rayon du fil et $\langle x \rangle = \frac{1}{\mathcal{S}_l} \iint_{\mathcal{S}_l} x ds$, la valeur moyenne d'une quantité x dans la surface transversale. Les courants sur les fils induisent à leur tour un terme source $\boldsymbol{J}$ sur les champs, donné par $i = \mathcal{S}_l \langle J_l \rangle$

Dans son modèle, R. Holland propose une première approximation pour déterminer le paramètre L_h, en considèrent un cylindre de rayon R équivalent ($\mathcal{S}_l = \pi R^2$). Ainsi L_h devient :

$$L_h = \frac{\mu}{2\pi \mathcal{S}_l} \int_a^R r \ln(r/a) dr \tag{3.2}$$

qui donne lieu à l'approximation suivante :

$$L_h \approx \frac{\mu_0}{2\pi} \left(\ln\left(\frac{R}{a}\right) - \frac{1}{2} + \frac{a^2}{2R^2} \right) \tag{3.3}$$

La deuxième approximation qu'il a effectué, plus largement utilisée, tient compte de la géométrie de la grille FDTD-3D (Fig. 3.1(b)). Le terme de couplage L_h est alors obtenu par une moyenne dans la cellule, donnant :

$$L_h = \frac{\mu}{2\pi \mathcal{S}_l} \iint_{\mathcal{S}, r>a} \ln(r/a) ds \tag{3.4}$$

qui devient dans le cas où $a \ll \min(\Delta_o, \Delta_p)$ l'approximation suivante :

$$L_h = \frac{\mu_0}{2\pi} \left[\ln\left(\frac{\Delta_o^2 + \Delta_p^2}{a^2}\right) + \frac{\Delta_o}{\Delta_p} \arctan(\Delta_p/\Delta_o) + \frac{\Delta_p}{\Delta_o} \arctan(\Delta_o/\Delta_p) + \frac{\pi a^2}{4\Delta_o \Delta_p} - 3 \right] \tag{3.5}$$

Dans ce formalisme de fil minces, le champ incident est une donnée du domaine 3D ($\langle E_l \rangle$).

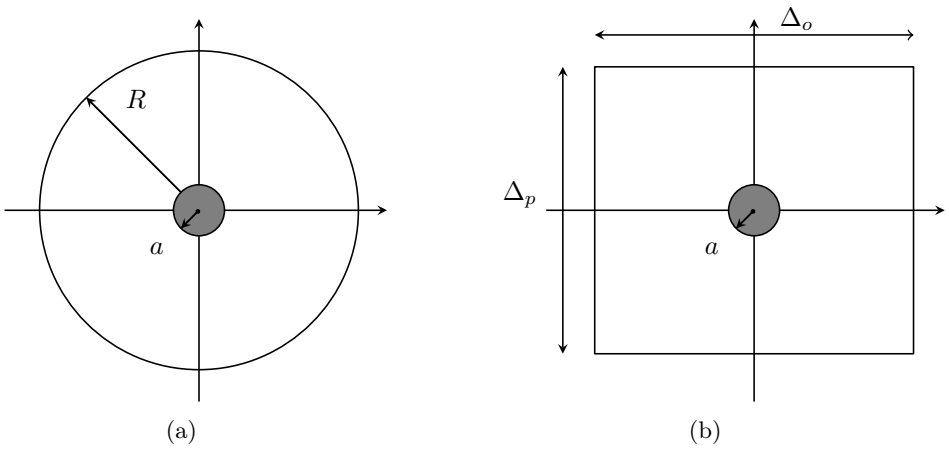


FIGURE 3.1 – Représentation de la géométrie équivalente de la cellule (a) et de la cellule de Yee (b) utilisée pour calculer l'inductance de couplage L.

Habituellement, les fils de R. Holland sont utilisés dans le cas où ils suivent les arêtes du maillage, c'est à dire que $\langle E \rangle|^n_{l+1/2} = E|^n_{l+1/2,o,p}$ avec l, o, p une permutation de $\{i, j, k\}$. Il est cependant possible de décaler le fil dans la maille. Dans ce cas de figure, le champ électrique peut être interpolé comme suit : (Eq. 3.6).

$$\begin{aligned} \langle E_m \rangle|^n_{l+1/2} \quad = \quad & E|^n_{l+1/2,o,p}\,(1-\alpha(o,m))(1-\alpha(p,m)) \\ & + E|^n_{l+1/2,o+1,p}\,\alpha(o,m)(1-\alpha(p,m)) \\ & + E|^n_{l+1/2,o+1,p+1}\,\alpha(o,m)\alpha(p,m) \\ & + E|^n_{l+1/2,o,p+1}\,(1-\alpha(o,m))\alpha(p,m) \end{aligned} \tag{3.6}$$

Avec $\alpha(x,m) = \delta(x,m)/\Delta_x$, $\delta(x,m)$ désignant la position du fil m dans la cellule suivant la direction x (Fig. 3.2). Dans le cas où l'on prend en compte la position du fil dans la maille, nous devons également considérer cela dans le rayonnement pour conserver une réciprocité dans le formalisme champ/câbles assurant la stabilité du schéma global [25]. Dans ce cas de figure, le terme source $\langle J_l \rangle$ est proportionnel à la surface de recouvrement.

$$\mathcal{S}_l\; J_m|^n_{l+1/2,o,p} = I|^n_{l+1/2}\,(1-\alpha(o,m))(1-\alpha(p,m)) \tag{3.7a}$$
$$\mathcal{S}_l\; J_m|^n_{l+1/2,o+1,p} = I|^n_{l+1/2}\,\alpha(o,m)(1-\alpha(p,m)) \tag{3.7b}$$
$$\mathcal{S}_l\; J_m|^n_{l+1/2,o+1,p+1} = I|^n_{l+1/2}\,\alpha(o,m)\alpha(p,m) \tag{3.7c}$$
$$\mathcal{S}_l\; J_m|^n_{l+1/2,o,p+1} = I|^n_{l+1/2}\,(1-\alpha(o,m))\alpha(p,m) \tag{3.7d}$$

L'utilisation de cette interpolation est très intéressante car elle permet l'utilisation du fil à l'intérieur de cellules du moment qu'il reste parallèle à un axe. Cependant, cette interpolation souffre d'une imprécision qui nuit à son utilisation [51].

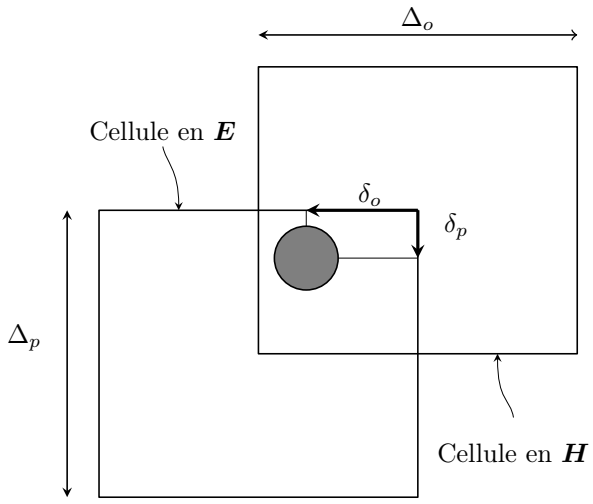


FIGURE 3.2 – Représentation de la géométrie de la cellule utilisée pour calculer l'inductance de couplage L quand le fil est décalé dans la cellule.

3.2.2 Le modèle de fils minces de Berenger

J.P. Berenger en 2000 [23] a étendu le principe du fil de R. Holland au cas où plusieurs conducteurs sont présents dans la maille. Dans ce cas de figure, l'équation sur la ligne et les termes de couplages champs/câble s'écrivent comme pour le modèle de R. Holland mais avec un formalisme matriciel (Eq. 3.8a). La principale difficulté de ce modèle est de donner une approximation correcte des paramètres de la ligne.

$$\boldsymbol{L_b}\left(\partial_t \boldsymbol{i} + \nu^2 \partial_l \boldsymbol{q}\right) + \boldsymbol{R}\boldsymbol{i} = \langle \boldsymbol{E_l} \rangle \tag{3.8a}$$

$$\partial_t \boldsymbol{q} + \frac{\sigma}{\varepsilon}\boldsymbol{q} + \partial_l \boldsymbol{i} = 0 \tag{3.8b}$$

$$\boldsymbol{i} = \mathcal{S}_l \langle \boldsymbol{J_l} \rangle \tag{3.8c}$$

Dans la définition de ce formalisme, notons que nous pouvons considérer l'interaction entre les conducteurs et les champs de deux manières. La première, proposée par J.P. Berenger, ne fait pas d'hypothèse sur le rayonnement et considère la position réelle des conducteurs pour déterminer le terme source sur la ligne ($\langle E_m \rangle$) et dans le domaine 3D ($\langle J_m \rangle$). Cette approche souffre tout de même de l'imprécision de l'interpolation (3.6) et (3.7).

Une autre approche consiste à utiliser le rayonnement du mode commun (Sec. 2.2.4 page 50). C'est cette approche que nous choisi d'adopter et que nous développerons dans le paragraphe (Sec. 3.3) pour notre méthode.

Dans son approche, J.P. Berenger propose de déterminer les paramètres de la ligne par l'intégration dans la cellule de l'équation différentielle autour du fil. Ainsi par identification

avec l'équation de ligne, nous avons :

$$L_{m,m} = \frac{\mu_0}{2\pi} \left\langle \ln\left(\frac{r}{a_m}\right) \right\rangle \tag{3.9a}$$

$$L_{m,k} = \frac{\mu_0}{4\pi} \left\langle \ln\left(\frac{r^2 + d_{m,k}^2 - 2rd_{m,k}\cos(\phi_{m,k} - \theta)}{a_m^2 + d_{m,k}^2 - 2a_i d_{m,k}\cos(\phi_{m,k} - \theta)}\right) \right\rangle \tag{3.9b}$$

Où les paramètres (a, d, ϕ, r, θ) sont donnés en figure 3.3 et $\langle x \rangle = \frac{1}{\mathcal{S}_l} \iint_{\mathcal{S}_l} x ds$ désigne l'intégration d'une quantité x sur la surface transversale.

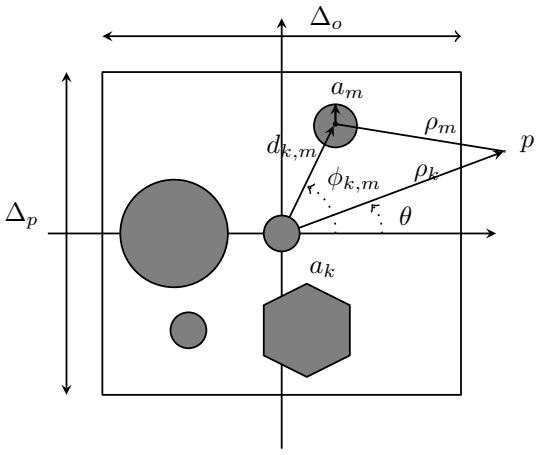


FIGURE 3.3 – Représentation de la géométrie de la cellule utilisée pour calculer l'inductance de couplage L.

$L_{m,m}$ est l'inductance du fil par rapport à la maille et s'obtient comme dans le cas de R. Holland. Dans le cas des inductances mutuelles $L_{m,k}$, J.P. Berenger donne l'approximation suivante :

$$\begin{aligned} L_{m,k} \approx \frac{\mu_0}{4\pi}[&\frac{1}{8\Delta_o\Delta_p}(G_{m,m} + G_{m,k} + G_{k,m} + G_{k,k} \\ &+ H_{m,m} + H_{m,k} + H_{k,m} + H_{k,k}) \\ &+ \frac{\pi a_k^2}{8\Delta_o\Delta_p}(1 - 2\ln(\frac{a_k}{d_{m,k}})) - \frac{3}{2}] \end{aligned} \tag{3.10}$$

avec :

$$
\begin{aligned}
G_{m,k} &= A_m B_k \ln\left(\frac{A_m^2 + A_k^2}{d_{m,k}}\right) \\
H_{m,k} &= A_m^2 \tan^{-1}\left(\frac{B_k}{A_m}\right) + A_m^2 \tan^{-1}\left(\frac{A_m}{B_k}\right) \\
A_m &= A - d_{k,m}\cos(\phi_{k,m}) \quad A_k = A - d_{k,m}\cos(\phi_{k,m}) \\
B_m &= B - d_{k,m}\sin(\phi_{k,m}) \quad B_k = A + d_{k,m}\sin(\phi_{k,m})
\end{aligned}
$$

3.3 Nouvelle approche : le formalisme pour des sections à géométrie arbitraire

Dans les deux formalismes précédents, le couplage de la ligne avec le 3D est obtenu par le moyennage dans la cellule, de l'équation de Maxwell Faraday. Les paramètres de la ligne sont donc représentés par une intégration géométrique ; puis une approximation est faite dans le cas où la ligne est petite par rapport à la cellule.

Dans la suite, nous allons présenter un formalisme qui permet de traiter des conducteurs de géométrie quelconque dans la maille. Ce formalisme est très similaire au formalisme de R. Holland et de J.P. Berenger mais il s'en démarque, d'une part dans la manière d'intégrer l'équation de Maxwell-Faraday et d'autre part dans la modélisation du couplage avec le domaine 3D. Nous verrons, de manière plus détaillée, le traitement effectué sur l'équation de Maxwell-Faraday, ainsi que les hypothèses faites pour poser ce formalisme.

De la même manière que pour le fil de R. Holland, nous pouvons faire dans un premier temps l'approximation d'un cylindre équivalent ou considérer la géométrie de la cellule (Fig. 3.3). L'une ou l'autre des deux considérations, ne présente pas de difficulté numérique particulière pour calculer les paramètres R, L, C et G .

3.3.1 Détermination des équations de ligne de transmission dans le cas d'une géométrie de conducteurs quelconques

Dans ce paragraphe nous allons donner les équations de ligne de transmission permettant de décrire des conducteurs de géométrie quelconque. Le développement fait ici a également pour but d'expliciter les conditions nécessaires pour permettre le couplage avec le domaine 3D et de donner le domaine de validité de l'approche.

Comme pour les formalismes de R. Holland et de J.P. Berenger, nous partons de l'équation de Maxwell-Faraday. Par un jeu d'écriture, nous la mettons en deux dimensions en utilisant le produit vectoriel avec la normale au plan ($\boldsymbol{u_l}$) qui définit notre espace 2D. Dans notre cas, ce plan est la section transversale de nos conducteurs. En espace libre, nous avons :

$$
\begin{aligned}
\boldsymbol{\nabla} \wedge \boldsymbol{E} &= -\mu_0 \partial_t \boldsymbol{H} \\
\boldsymbol{u}_l \wedge \boldsymbol{\nabla} \wedge \boldsymbol{E} &= -\mu_0 \partial_t \left(\boldsymbol{u}_l \wedge \boldsymbol{H}\right) \\
\boldsymbol{\nabla}_T E_l &= -\mu_0 \partial_t \left(\boldsymbol{u_l} \wedge \boldsymbol{H}_T\right) + \partial_l \boldsymbol{E}_T
\end{aligned} \tag{3.11}
$$

$\boldsymbol{u}_l$ est le vecteur longitudinal au fil et l'indice $_T$ dénote que la quantité vectorielle $\boldsymbol{f}$ est dans le plan orthogonal au vecteur $\boldsymbol{u}_l$.

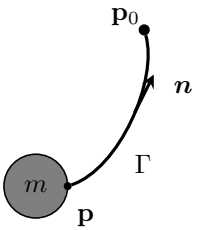


FIGURE 3.4 –

Nous pouvons intégrer cette équation sur un chemin Γ de p à p_0 dans le plan défini par le vecteur $\boldsymbol{u}_l$. Nous obtenons alors :

$$\int_\Gamma \boldsymbol{\nabla}_T E_l \boldsymbol{dl} \;=\; -\mu_0 \int_\Gamma \partial_t \left(\boldsymbol{u_l} \wedge \boldsymbol{H}_T\right) \boldsymbol{d\tau} + \int_\Gamma \partial_l \boldsymbol{E}_T \boldsymbol{d\tau}$$

qui s'écrit aussi :

$$E_l(p) - E_l(p_m) \;=\; -\mu_0 \partial_t \int_\Gamma \boldsymbol{n} \cdot \left(\boldsymbol{u}_l \wedge \boldsymbol{H}_T\right) d\tau + \partial_l \int_\Gamma \boldsymbol{n} \cdot \boldsymbol{E}_T d\tau \tag{3.12}$$

Nous avons $\boldsymbol{d\tau} = \boldsymbol{n} d\tau$ et $\boldsymbol{n}$ le vecteur longitudinal à Γ. Nous pouvons tout à fait choisir p sur la surface du conducteur. Nous pouvons également faire l'hypothèse que le potentiel est constant sur cette surface. Nous pouvons donc choisir n'importe quel point de la surface du conducteur comme point d'intégration. Nous noterons par la suite ce point p_m associé au conducteur m. Enfin, nous pouvons faire l'hypothèse que le champ suit une distribution quasi-statique, ce qui nous permet d'écrire $\boldsymbol{H}_T = \boldsymbol{\alpha}(p) i(l,t)$ et $\boldsymbol{E}_T = \boldsymbol{\beta}(p) q(l,t)$. Par exemple dans le cas d'une distribution avec une symétrie de révolution, nous avons $\boldsymbol{\alpha}(p) = \boldsymbol{u_\theta}/2\pi r$ et $\boldsymbol{\beta}(p) = \boldsymbol{u_r}/2\pi\varepsilon r$.

Sous ces hypothèses, l'équation précédente donne la première équation de lignes pour le conducteur m :

$$\begin{aligned} E_l(p_0) - E_l(p_m) &= -\mu_0 \partial_t \int_\Gamma \boldsymbol{n} \cdot \left(\boldsymbol{u}_l \wedge \boldsymbol{\alpha}(\tau) i(l,t)\right) d\tau + \partial_l \int_\Gamma \boldsymbol{n} \cdot \boldsymbol{\beta}(p) q(l,t) d\tau \\ &= L_m \partial_t i + \frac{1}{C_m} \partial_l q \end{aligned} \tag{3.13}$$

avec

$$L_m = -\mu_0 \int_\Gamma \boldsymbol{n} \cdot \left(\boldsymbol{u}_l \wedge \boldsymbol{\alpha}(\tau)\right) d\tau \tag{3.14a}$$

$$\frac{1}{C_m} = L\nu^2 = \int_\Gamma \boldsymbol{n} \cdot \boldsymbol{\beta}(p) q(l,t) d\tau \tag{3.14b}$$

Nous devons maintenant déterminer les valeurs de $E_l(p_m)$ et de $E_l(p_0)$. En ce qui concerne le terme $E_l(p_m)$, dans le cas où le conducteur est supposé parfait, celui est donné par $E_l(p_m) = 0$ et dans le cas où il est de conductivité finie, nous avons par la loi d'Ohm $E_l(p_m) = \rho J = \rho i/s = Ri$ avec s l'aire de la section transversale du conducteur. Nous pouvons donc écrire :

$$L\partial_t i + Ri + \partial_l v = E_l(p_0) \tag{3.15}$$

Nous devons maintenant déterminer le terme $E_l(p_0)$ qui est lié au choix d'une référence. Cette référence peut être n'importe quel point de l'espace, du moment qu'il n'est pas à l'intérieur d'un conducteur. Nous distinguons deux cas de figure que nous allons détailler.

Si la référence est à la surface d'un autre conducteur k, le terme $E_l(p_0)$ est tout simplement obtenu par la loi d'Ohm $E_l(p_0) = E_l(p_k) = -R_k i$. Notons le signe négatif qui représente le courant de retour dans le conducteur de référence. Dans cette configuration l'inductance et la capacité donnée par les équations (3.14) devient l'inductance mutuelle L_{mk} et la capacité C_{mk} entre les conducteurs m et k.

Si la référence est un point quelconque de l'espace, nous devons faire l'hypothèse que le mode guidé est au moins TEM, ainsi, le champ électrique suivant le vecteur de propagation $\boldsymbol{u_l}$ en tout point de l'espace est nul par définition.

Dans ces deux cas de figure, nous pouvons mettre l'équation (3.15) sous la forme suivante en la généralisant à plusieurs conducteurs.

$$\boldsymbol{L}\partial_t \boldsymbol{i} + \boldsymbol{R}\boldsymbol{i} + \partial_l \boldsymbol{v} = 0 \tag{3.16}$$

équation qui qualifie l'évolution de la tension sur la ligne, donnée par :

$$\boldsymbol{C}\partial_t \boldsymbol{v} + \partial_l \boldsymbol{i} = 0 \tag{3.17}$$

Dans la construction des deux équations précédentes, le choix de la référence pour évaluer les paramètres de la ligne (L et C) n'est pas fixé. Dans la suite, nous verrons plusieurs choix possibles et nous étudierons leur pertinence.

3.3.2 L'équation de Laplace pour calculer les paramètres de ligne de transmission

Cette section n'est pas consacrée à la résolution d'un problème Laplace mais consiste à donner le principe du calcul des paramètres de ligne à partir du problème de Laplace suivant :

$$\boldsymbol{\nabla}_T \cdot (\varepsilon \boldsymbol{\nabla}_T v) = 0 \tag{3.18}$$

où $\boldsymbol{\nabla}_T$ définit le Laplacien 2D dans le plan transversal à la ligne. Il existe pour résoudre ce problème, plusieurs méthodes basées sur des formulations intégrales [34, 32, 33] ou sur des formulations différentielles que nous n'expliciterons pas ici.

Pour évaluer les paramètres L et C, nous résolvons deux problèmes de ce type :

– un problème en considérant les diélectriques pour déterminer la capacité C ;

– Un problème sans diélectrique pour évaluer une capacité C_0 qui nous permet d'obtenir L par $LCc_0^2 = 1$ où c_0 est la vitesse de la lumière dans le vide.

Le calcul de la capacité C par le problème de Laplace consiste à fixer sur chaque conducteur i un potentiel V_i et un potentiel $V = 0$ sur la référence. Nous déterminons alors le potentiel présent dans tout le domaine et donc le champ $\boldsymbol{E} = -\boldsymbol{\nabla}_T v$. La charge totale Q_i sur chaque conducteur i est données par l'intégrale sur la surface de $\varepsilon \boldsymbol{n} \cdot \boldsymbol{E}$. On détermine ensuite la capacité C_{ij} entre deux conducteurs i et j par $Q_i = C_{ij}(V_j - V_i)$.

3.4 Calcul des paramètres de la ligne de transmission

Dans cette section, nous allons nous intéresser au choix de la référence pour calculer les paramètres L et C. Ces paramètres sont importants car ils conditionnent le couplage entre le mode de ligne et le 3D. Nous discuterons de trois approches différentes en expliquant leurs motivations.

Dans ces différentes approches, pour évaluer les paramètres nous utiliserons le problème de Laplace précédemment décrit.

3.4.1 Paramètres dans la cellule FDTD (In-Cell)

Dans le formalisme de R. Holland comme dans le formalisme de J.P. Berenger, le terme de couplage L est déterminé en prenant la valeur moyenne de la distribution du champ dans la cellule. Pour cette raison, nous considérons parfois ambiguïté que la cellule FDTD est la référence de la ligne.

Nous pouvons alors nous demander quelle est la pertinence de considérer le bord de la cellule comme référence pour calculer les paramètres de la ligne de transmission, comme nous le montre la figure (Fig. 3.5).

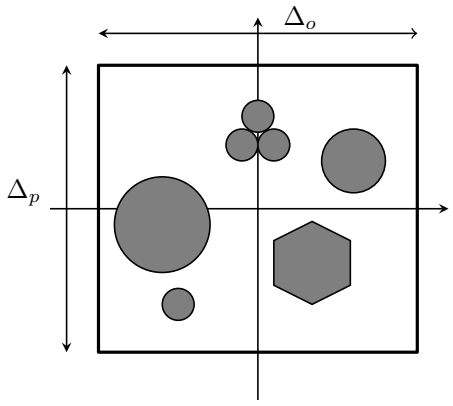


FIGURE 3.5 – Calcul des paramètres de la ligne de transmission en utilisant la cellule comme référence

L'hypothèse à vérifier est que le potentiel en tout point de la surface de la référence soit constant et que le terme $E_l(p_0)$ soit nul. Or, par hypothèse, la taille de la cellule est petite par rapport à la plus petite longueur d'onde présente. Ainsi la variation du champ est négligeable, ce qui nous permet de dire que la tension entre deux points sur le bord de la cellule est nulle (surface métallique). Nous pouvons donc tout à fait considérer le bord de la cellule comme une référence pour calculer les paramètres de la ligne de transmission.

3.4.2 Paramètres en espace ouvert

La deuxième possibilité redéfinit la notion de fil mince qui est généralement considéré comme un modèle local. En réalité, ce n'est pas la loi qui régit le comportement de la charge et du courant qui est local, mais c'est le couplage qui est localisé, ce qui est très important.

En considérant ce point, pourquoi choisir de borner le domaine pour calculer les paramètres de ligne. Il semblerait pertinent de prendre la valeur limite quand la référence tend vers l'infini.

Cette valeur peut être calculée par le problème de Laplace vu précédemment, pour un domaine non borné. Dans ce cas, la valeur recherchée est la matrice générale (sans référence). Notons cependant que dans le cas où il y a un seul conducteur il existe une formule analytique pour évaluer l'inductance.

L'avantage de ne pas considérer la cellule comme référence est que la section transversale de la ligne de transmission peut couvrir plusieurs cellules. En outre, cette approche est intéressante pour les fils obliques. En effet, l'approche qui considère la cellule comme référence est impropre dans ce cas de figure, car les bords de la cellule sont ambigus pour les fils obliques. Nous pourrions utiliser par exemple le formalisme de fil oblique de C. Guiffaut [26] pour prendre en compte le couplage oblique avec le domaine 3D. Toutefois, nous ne traiterons ici que le cas des lignes qui suivent les arêtes du maillage 3D.

3.4.3 Paramètres avec une référence dans la cellule FDTD

La dernière configuration étudiée (Fig. 3.6) traite le cas particulier où les fils sont très proches d'une référence globale (moins d'une maille). Dans ce cas de figure, nous pouvons considérer cette référence globale comme la référence pour le calcul des paramètres de la ligne de transmission. Cette approche est très intéressante car elle est courante dans les problèmes industriels.

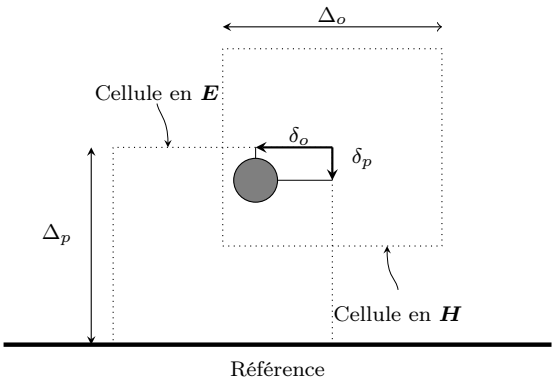


FIGURE 3.6 – Représentation d'un fil proche d'un paroi.

Cette approche ne peut pas considérer le cas où l'ensemble définit par la ligne de transmission et la référence globale est plus grand que la maille de Yee. En effet, dans ce cas de figure, le champ TEM se propage dans la ligne de transmission mais également dans le domaine 3D. Ce dernier va alors induire sur la ligne un terme source erroné.

3.5 Couplage avec le domaine 3D - Approximation du mode commun

Dans notre formalisme, nous avons choisi de faire le couplage de la ligne avec le domaine 3D par le mode commun. Dans cette approximation, le couplage du domaine 3D vers le modèle de ligne de transmission se fait par le champ incident longitudinal au chemin central $E_l^i(p_c)$, c'est-à-dire le chemin définit par les barycentres des sections transversales des conducteurs de la ligne. Ce champ incident est ensuite introduit dans les équations de ligne de transmission par le modèle de Agrawal. Pour conserver la symétrie du couplage, le terme source dans le domaine 3D, provenant de la ligne de transmission, est donné par le courant de mode commun, c'est-à-dire la somme des courants ramenés sur ce même chemin central. Ce couplage nous permet d'écrire les équations liées à la ligne :

$$\boldsymbol{L}\partial_t \boldsymbol{i} + \boldsymbol{R}\boldsymbol{i} + \partial_l \boldsymbol{v} = E_l^i(p_c) \tag{3.19a}$$

$$\boldsymbol{C}\partial_t \boldsymbol{v} + \partial_l \boldsymbol{i} = 0 \tag{3.19b}$$

et le terme J_l^s dans les équations de Maxwell donné par :

$$i_c = \mathcal{S}_l J_l^s \tag{3.20}$$

Avec p_c la position du chemin central et $i_c = \sum_{m=1}^{M} i_m$ le courant de mode commun défini comme la somme des courants traversés par chaque conducteur. Notons bien que les quantités $E_l^i(p_c)$ et J_l^s sont des quantités scalaires identiques pour tous les conducteurs.

Nous aurions pu considérer la vraie position des conducteurs dans le couplage champ câbles. Cependant nous avons fait le choix du mode commun car nous pensons que l'erreur commise par cette approximation est inférieure ou de l'ordre de l'erreur d'interpolation. De plus, nous pouvons démontrer que l'erreur commise par le rayonnement est négligeable si nous nous plaçons à une distance suffisante de la ligne.

Pour quantifier l'erreur commise sur le rayonnement des courants de la ligne, en utilisant l'approximation du mode commun, nous nous proposons de comparer le champ magnétique rayonné sous l'hypothèse quasi-statique, par un conducteur équivalent traversé par le courant de mode commun et le rayonnement exact des câbles. Nous définissons alors l'erreur de rayonnement e par la norme de la différence du champ magnétique normalisé par le courant de mode commun. Cette erreur peut être majorée par :

$$e = \left\| \frac{\partial_l \boldsymbol{H} - \boldsymbol{H_c}}{i_c} \right\| \leq \left\| \frac{\partial_l \boldsymbol{H} \cdot \boldsymbol{u_r}}{i_c} \right\| + \left\| \frac{\partial_l (\boldsymbol{H} - \boldsymbol{H_c}) \cdot \boldsymbol{u_\theta}}{i_c} \right\| \tag{3.21}$$

où $\boldsymbol{H_c}$ désigne le champ rayonné par le mode commun et $\boldsymbol{H}$ le champ théorique. Nous allons maintenant déterminer les deux termes de majoration. Pour cela, nous calculons le champ rayonné pour les câbles réels et pour un conducteur équivalent associé au mode commun. Puis nous donnons une approximation asymptotique pour les projections du champ sur les directions $\boldsymbol{u_r}$ et $\boldsymbol{u_\theta}$ comme définies sur la figure 3.7.

Le champ magnétique rayonné par un élément de courant sous l'hypothèse statique est donné par la relation de Biot et Savart $d\boldsymbol{H} = \frac{i(m)dl}{4\pi \|\boldsymbol{MP}\|^2} \boldsymbol{u_l} \wedge \boldsymbol{u_{MP}}$. Ainsi dans la configuration

associée au mode commun, nous aboutissons à la relation suivante où les notations sont données en figure 3.7.

$$\partial_l \boldsymbol{H_c} = \frac{1}{4\pi}\frac{\boldsymbol{u_\theta}}{r^2}\sum_{m=1}^{M} i_m = \frac{1}{4\pi}\frac{\boldsymbol{u_\theta}}{r^2} i_c \tag{3.22}$$

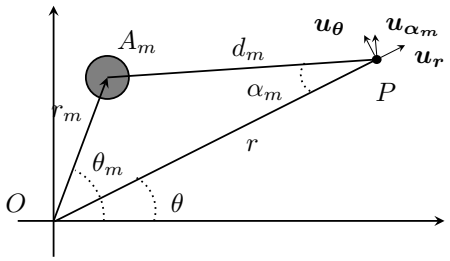


FIGURE 3.7 – Rayonnement par rapport au repère de la cellule d'un fil décalé.

De même le champ rayonné par la configuration réelle est donné par la relation suivante :

$$\begin{aligned} \partial_l \boldsymbol{H} &= \frac{1}{4\pi}\sum_{m=1}^{M} i_m \frac{\boldsymbol{u_{\alpha_m}}}{d_m^2} \\ &= \frac{1}{4\pi}\sum_{m=1}^{M} i_m \frac{\sin(\alpha_m)\boldsymbol{u_r} + \cos(\alpha_m)\boldsymbol{u_\theta}}{d_m^2} \end{aligned} \tag{3.23}$$

Enfin, les identités dans le triangle $A_m OP$ (3.26) nous permettent de réécrire le champ en :

$$\partial_l \boldsymbol{H}\cdot\boldsymbol{u_r} = \frac{1}{4\pi}\sum_{m=1}^{M} i_m \frac{r_m \sin(\theta_m-\theta)}{d_m^3} \tag{3.24}$$

$$\partial_l \boldsymbol{H}\cdot\boldsymbol{u_\theta} = \frac{1}{4\pi}\sum_{m=1}^{M} i_m \frac{\cos\alpha_m}{d_m^2} \tag{3.25}$$

$$d_m^2 = r_m^2 + r^2 - 2r_m r\cos(\theta_m-\theta) \tag{3.26a}$$
$$r_m^2 = d_m^2 + r^2 - 2d_m r\cos(\alpha_m) \tag{3.26b}$$
$$r_m \sin(\theta_m-\theta) = d_m \sin(\alpha_m) \tag{3.26c}$$

Pour déterminer le comportement quand la distance r devient grande par rapport à l'encombrement ($r \gg \max_m r_m$), nous commençons par considérer le champ magnétique suivant la direction $\boldsymbol{u_r}$. Ainsi, le développement asymptotique nous permet d'écrire :

$$\begin{aligned}
\|4\pi\partial_l \boldsymbol{H}\cdot\boldsymbol{u_r}\| &= \left\|\sum_{m=1}^{M} i_m \frac{r_m \sin(\theta_m-\theta)}{d_m^3}\right\| \\
&= \left\|\sum_{m=1}^{M} i_m \frac{\sin(\alpha_m)}{d_m^2}\right\| \\
&\leq \left\|\sum_{m=1}^{M} i_m \frac{1}{d_m^2}\right\| \\
&\leq \left\|\sum_{m=1}^{M} i_m \frac{1}{(r-r_m)^2}\right\| \qquad (3.27)
\end{aligned}$$

Car nous avons :

$$d_m^2 = r_m^2 + r^2 - 2r_m r\cos(\theta_m-\theta) \geq r_m^2 + r^2 - 2r_m r = (r-r_m)^2$$

Donc lorsque $r_m \ll r$ la composante radiale est majorée par :

$$\|4\pi\partial_l \boldsymbol{H}\cdot\boldsymbol{u_r}\| \leq \left\|\frac{i_c}{r^2}\right\| \qquad (3.28)$$

Ainsi l'amplitude de $\partial_l \boldsymbol{H}\cdot\boldsymbol{u_r}$ décroît très rapidement avec r.

De la même manière, nous pouvons donner le comportement asymptotique de la projection sur $\boldsymbol{u_\theta}$ du champ magnétique quand la distance r devient grande par rapport à l'encombrement ($r \gg \max_m r_m$).

$$\begin{aligned}
\|4\pi\partial_l\left(\boldsymbol{H}-\boldsymbol{H_c}\right)\cdot\boldsymbol{u_\theta}\| &= \left\|\sum_{m=1}^{M} i_m\left[\frac{\cos\alpha_m}{d_m^2}-\frac{1}{r^2}\right]\right\| \\
&\leq \left\|\sum_{m=1}^{M} i_m\left[\frac{1}{d_m^2}-\frac{1}{r^2}\right]\right\| \\
&\leq \left\|\sum_{m=1}^{M} i_m\left[\frac{1}{(r-r_m)^2}-\frac{1}{r^2}\right]\right\| \qquad (3.29)
\end{aligned}$$

Donc lorsque $r_m \ll r$ la composante radiale est majorée par 0. Ainsi l'amplitude de $\|4\pi\partial_l\left(\boldsymbol{H}-\boldsymbol{H_c}\right)\cdot\boldsymbol{u_\theta}\|$ tend vers 0 lorsque r devient grand.

L'approximation du mode commun permet donc de représenter correctement le rayonnement d'un ensemble de câbles si la distance entre le point observé et la ligne est suffisamment grande par rapport à la distance des conducteurs par rapports au barycentre de la ligne. On peut supposer que ceci est le cas lorsqu'on se place à l'extérieur de la cellule et si la câble est interne à celle-ci.

3.6 Validation numérique

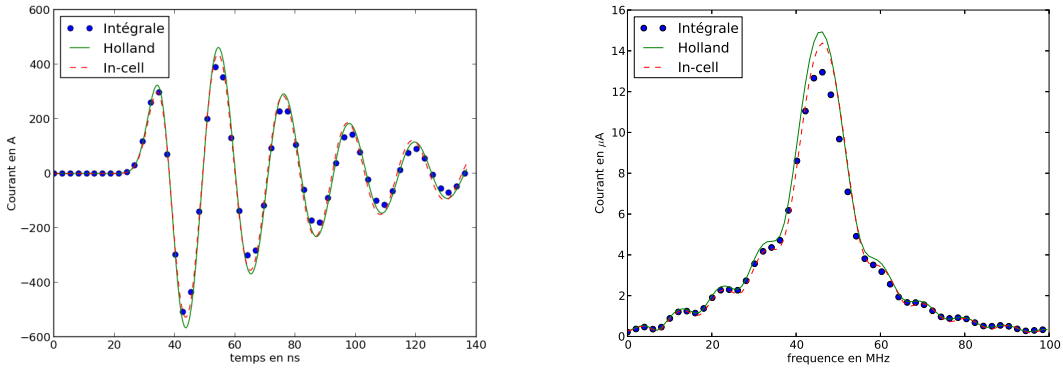
Dans cette partie, nous allons montrer numériquement la capacité de notre approche à traiter des problèmes concrets. Nous commençons par des cas relativement simples tels que l'excitation d'un fil, d'une bifilaire et d'une multifilaire composée de neuf conducteurs. Ces cas tests ont pour but de positionner notre modèle de fil par rapport au modèle de R. Holland et de J.P. Berenger. Puis nous présentons un certain nombre de cas tests qui illustrent les limites de notre approche. Dans les configurations choisies nous étudions premièrement le cas où les éléments filaires se retrouvent proches d'une référence globale. Cette configuration est importante car les fils sont fixés dans la pratique à un support qui représente une référence plus ou moins globale. Ensuite, nous donnons deux cas tests qui illustrent la souplesse apportée par cette approche. Pour cela, nous traitons, d'une part, le cas du nœud de courant, où deux fils non connectés suivent une même arête de maille pour se séparer et suivre deux directions très différentes et d'autre part, une ligne de transmission composée de plusieurs niveaux de blindages et des gaines diélectriques qui traversent une paroi.

3.6.1 Le fil simple

Le premier cas test que nous allons considérer est le cas d'un fil d'une longueur totale de 3m et de rayon 1cm. Le but de ce cas test est de comparer le modèle de R. Holland avec notre approche où les paramètres de la ligne sont calculés en considérant le bord de la cellule comme référence. Une méthode intégrale temporelle [52] présentée (Sec. E page 246) est choisie pour donner un résultat de référence. Le fil est placé parallèlement à l'axe x le long des arêtes d'un maillage FDTD-3D de pas spatial $dl = 20\text{cm}$ et illuminé par une onde plane dont le champ électrique $E_x(t) = 10^5 \exp\left(-\alpha^2(t-\tau)^2\right)$, avec $\tau = 3 \cdot 10^{-7}\text{s}$ et $\alpha = 0.2 \cdot 10^9\text{s}^{-1}$, est longitudinal au fil. On considère le fil en circuit ouvert ($i = 0$) à ses deux extrémités.

La figure (Fig. 3.8) représente le courant obtenu au milieu du fil pour les différentes approches.

Le modèle de couplage proposé donne un résultat comparable au formalisme de fil mince de R. Holland et à la solution de référence (méthode intégrale). Cet exemple valide donc notre approche sur un cas élémentaire.



(a) (b)

FIGURE 3.8 – Courant au milieu du fil pour différentes approches numériques.

3.6.2 La bifilaire

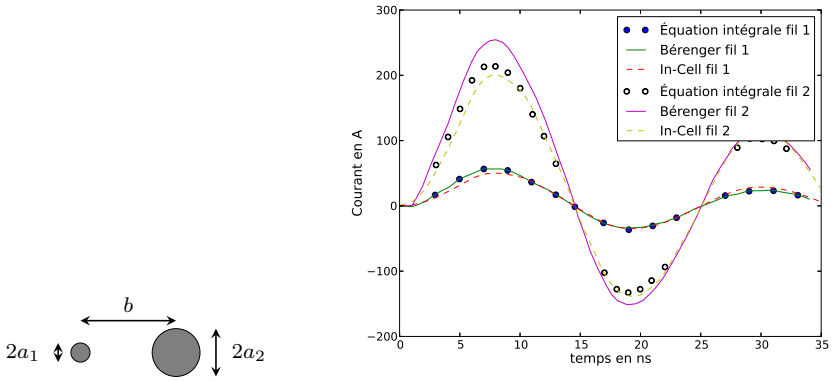
Les deux applications numériques qui suivent sont directement tirées des travaux de J.P. Berenger sur la modélisation des multifilaires [23].

La bifilaire est le premier cas test qui utilise la capacité multi-filaire de notre approche. Le but principal de ce cas test est de montrer les capacités de notre modèle pour traiter des problèmes à plusieurs conducteurs. Nous comparons les résultats obtenus avec le formalisme proposé par J.P. Berenger et avec la méthode intégrale temporelle. La bifilaire choisie est asymétrique d'une longueur totale de 3m ayant un premier conducteur de rayon $a_1 = 1$mm et un deuxième de rayon $a_2 = 1$cm. Les deux fils sont séparés de 2cm. Cette bifilaire est excitée par une onde plane dont la composante de champ électrique est donnée par :

$$E_x(t) = 10^6 \left(1 - \exp(-t \cdot 10^9/5)\right) \tag{3.30}$$

La ligne bifilaire est placée parallèlement à l'axe x dans une cellule FDTD-3D carrée de 20cm de coté. Celle-ci est en circuit ouvert ($i = 0$) à ses deux extrémités. La figure (Fig. 3.9) donne le courant au centre de la bifilaire par trois approches différentes :

- le formalisme multifilaire de J.P. Berenger ;
- un calcul intégral ;
- notre approche hybride.



(a) (b)

FIGURE 3.9 – Présentation de la géométrie (a) du cas test de la bifilaire et représentation du courant médian (b) pour différentes méthodes numériques.

De manière générale, nous avons une très bonne concordance entre notre approche et celle de J.P. Berenger, et cela, malgré le fait de n'avoir pas pris en compte la position des conducteurs dans la cellule pour déterminer le champ incident. Cette approximation est tout à fait satisfaisante car les deux conducteurs sont très proches de l'axe de la grille.

On remarque également qu'il n'y a pas de grande différence entre l'approximation d'un cylindre équivalent pour calculer les paramètres de la ligne dans notre approche hybride 1D/3D et l'utilisation de la géométrie de la cellule de Yee. Nous avons fait la même constatation pour les cas tests que nous allons aborder dans la suite, ainsi, dans un souci de clarté, nous ne parlerons plus de ce résultat commun à toutes nos simulations.

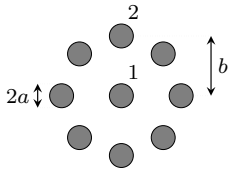
Note : Le fait que l'utilisation d'une géométrie circulaire ou cubique ne change pas la solution, nous interpelle. Ce comportement est lié à la nature même du modèle où l'hypothèse de stationnarité du champ dans la géométrie est plus importante que la géométrie elle-même.

Cette impression sera confortée par le résultat obtenu pour le modèle de Laplace sans référence que nous verrons plus loin (Sec. 3.6.4 page 117).

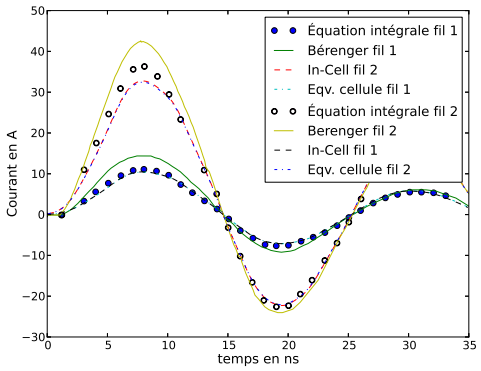
3.6.3 Multi-filaire

Ce cas test est composé de neuf fils qui suivent les arêtes du maillage. Il présente la problématique de la protection d'un fil (fil central) à un champ incident par huit fils l'entourant qui forment donc une espèce de blindage. Tous les fils ont le même rayon $a = 1\text{mm}$. Les huit fils de protection sont répartis uniformément à une distance b du fil central. La longueur totale de la ligne est de 3m et est en circuit ouvert à ses deux extrémités. Cette ligne est également placée dans une cellule de 20cm de côté et excitée par une onde plane (5.1). La figure 3.10 représente le courant relevé au milieu de deux conducteur pour différentes mé-

thodes indiqué sur la figure 3.10, pour différentes méthodes dans le cas où $b = 2$cm. Dans cet exemple, l'encombrement de la ligne (b) reste petit par rapport à la dimension de la cellule.



(a)



(b)

FIGURE 3.10 – Présentation de la géométrie (a) du cas test à 9 fils (effet de blindage) et représentation du courant médian (b) pour différentes méthodes numériques pour $d = 5$cm. Berenger : modèle de J.P. Berenger, Eqv. cellule : modèle de R. Holland en ne considérant qu'un fil et In-Cell : approche hybride proposée

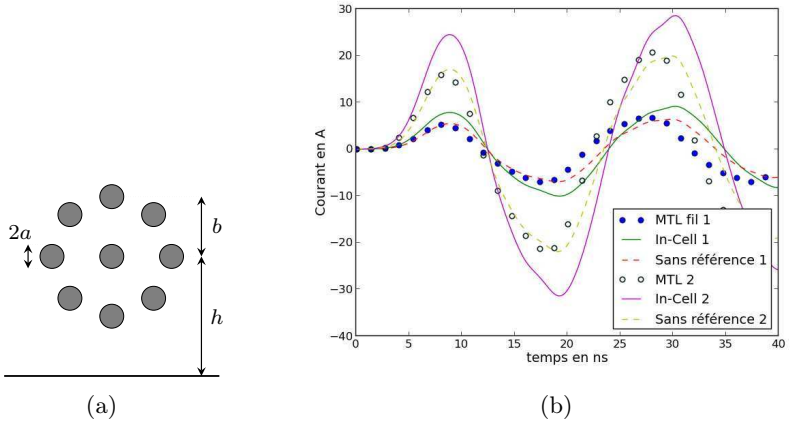
Nous avons une très bonne concordance entre les résultats obtenus par notre méthode, par l'approche de J.P. Berenger et le formalisme d'équation intégrale. Par contre le modèle de R. Holland, en tenant compte d'un conducteur équivalent pour représenter le faisceau de câbles ne suffit pas, comme nous pouvions nous y attendre. Ce cas test est intéressant, car il nous rapproche de certaines problématiques industrielles, même si nous restons encore très loin de la réalité à cause du nombre de conducteurs mais surtout de par la taille du faisceau. Dans ce cas test, nous sommes de l'ordre du centimètre d'encombrement, alors que dans la réalité avionique, par exemple, l'encombrement peut atteindre la dizaine de centimètres. En revanche, dans certaines configurations, les interstices, par lesquelles cheminent les conducteurs, n'excèdent guère quelques centimètres. Nous devons donc voir comment se comporte cette approximation dans le cas où les conducteurs sont proches de la paroi et/ou remplissent de manière plus importante la cellule.

3.6.4 Ligne de transmission

Dans cet exemple, nous reprenons la configuration géométrique de la ligne précédente à laquelle nous ajoutons un plan de masse métallique dans le domaine 3D. La ligne est placée à une hauteur h de celui-ci. La figure 3.11 représente les courants relevés au milieu de la ligne.

Nous avons choisi la hauteur, de manière à ce que la distribution de champ entre la ligne et le plan de masse soit correcte (fil au moins à trois mailles du plan). Dans ce cas de figure, nous pouvons choisir d'utiliser le modèle en espace ouvert, dont l'utilisation est rendue pertinente par la présence du plan. Nous pouvons aussi utiliser le modèle de Laplace

dans la cellule. Pour cet exemple, nous avons choisi de considérer la solution obtenue par la résolution d'une équations de ligne de transmission comme solution de référence.



(a) (b)

FIGURE 3.11 – Présentation de la géométrie (a) du cas test de la ligne à 9 fils en regard d'un plan de masse et représentation du courant médian (b) pour différentes méthodes numériques.

Dans cette configuration, le modèle de fil en espace ouvert donne un résultat satisfaisant malgré une erreur marquée dans la vitesse de phase. Cette différence dans la vitesse de phase n'est pas surprenante car la dispersion du schéma 3D est bien plus grande qu'en 1D, même pour ce cas de figure. Le modèle de Laplace dans la cellule donne un résultat un peu moins bon certainement lié a une mauvaise prise en compte du champ se propageant entre la ligne et le plan dans le modèle 3D.

Fil proche d'une paroi

Dans ce cas, nous étudions un fil qui est parallèle et proche d'une surface métallique parfaitement conductrice et supposée infinie. L'objectif de cet exemple est de voir les limites de l'approche hybride proposée pour traiter la problématique où les conducteurs sont proches d'une paroi.

Un fil de rayon $a = 1$cm de longueur totale de 3m est placé au dessus d'une plaque métallique. La longueur du coté de la plaque carrée est de 18.4m et le fil se situe à une maille (20cm) au dessus de la plaque. Le système est agressé par une onde plane définie comme pour le cas test (Sec. 3.6.1 page 114).

Le résultat (Fig. 3.12) de notre approche hybride en prenant la cellule comme référence pour l'évaluation des paramètres de la ligne (noté "Laplace dans la cellule"), est comparé avec une approche lignes de transmission où les paramètres sont obtenus en considérant la plaque infinie comme référence (noté "MTL"). De plus, nous avons choisi de déterminer le champ incident de deux manières différentes. La première, notée « analytique », où le champ incident est calculé de manière analytique en considérant un plan infini et la deuxième, notée

"numérique", où le champ incident est évalué le long du fil, sans considérer celui-ci dans le calcul, par le schéma FDTD dans le domaine 3D.

Nous proposons également deux autres résultats numériques. Le premier est obtenu en utilisant le modèle de R. Holland pour la prise en compte de la ligne et l'autre, noté « Laplace référence globale », est le résultat de notre approche hybride en considérant le plan métallique comme référence dans le calcul des paramètres de la ligne.

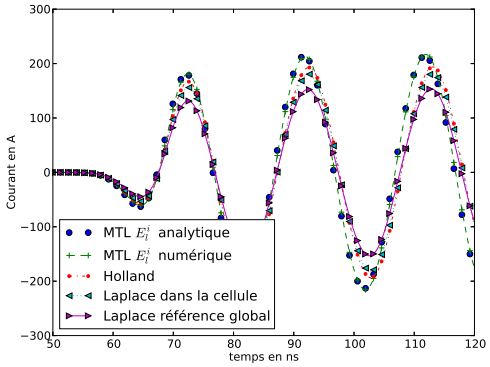


FIGURE 3.12 – Représentation du courant au milieu d'un fil proche d'un plan métallique pour différentes méthodes numériques.

Nous constatons dans ce cas de figure que le caractère fini de la plaque n'influe pas trop sur le calcul du champ incident. Ainsi, la différence que l'on peut observer entre les différents modèles et le résultat de référence (MTL) ne peut pas être attribuée à une erreur sur le champ incident. On remarque également que, dans cette configuration, le modèle de R. Holland donne un résultat sensiblement meilleur que notre approche hybride. Notons aussi que le choix du plan métallique comme référence dans notre modèle hybride conduit à une solution moins précise que pour le choix de la cellule. Cette différence est certainement due au fait que nous sommes dans le cas limite où cette approche peut être utilisée. En effet, comme le fil se trouve sur l'arête juste au-dessus du plan de masse, le champ, propagé par la ligne de transmission, va l'être dans la ligne de transmission mais également dans le 3D. Pour que cette approche trouve tout son sens, il faudrait que le fil se trouve plus proche de la paroi.

3.6.5 Nœud de courant

Dans cet exemple, nous nous intéressons au cas du nœud de courant. Ce problème représente les cas où un certain nombre de conducteurs suit un même chemin dans le domaine 3D, puis, à un point donné, se sépare pour suivre des directions très différentes. L'étude du nœud de courant est très importante dans de nombreux problèmes industriels. Par exemple, dans la majorité des problématiques en champ fort, l'étude du courant de mode commun est suffisante pour décrire le comportement global et détecter les risques de défaillance. Cette approximation du mode commun modélise les routages de fils par un fil équivalent

qui représente correctement ce courant de mode commun. Cependant, faire cette approximation suppose que le courant de mode commun respecte la loi de Kirchhoff à un nœud de courant, ce qui n'est pas évident. En réalité, la loi de Kirchhoff est garantie conducteur par conducteur. Ainsi, la modélisation de l'ensemble des conducteurs est rendue obligatoire pour modéliser correctement la répartition du courant à un nœud.

Pour vérifier la capacité de notre approche hybride en prenant comme référence la cellule pour traiter ce type de problème, nous étudions une configuration qui ressemble à une antenne dipolaire. Celle-ci se compose de deux branches, de longueurs égales $p = 2.5\text{m}$, raccordées à une bifilaire de longueur $l = p = 2.5\text{m}$ et dont les fils internes sont espacée de $e = 1.5\text{cm}$. Le rayon de tous les fils de la géométrie est de $a = 5\text{cm}$. Ce système est excité par un champ électrique incident $E(t)$ parallèle aux deux branches, tel que :

$$E(t) = -2.0\alpha^2(t-\tau)\exp\left(-\alpha^2(t-\tau)^2\right) \tag{3.31}$$

avec $\tau = 2\cdot 10^{-8}\text{s}$ et $\alpha = 0.2\cdot 10^{9}\text{s}^{-1}$.

La figure 3.13(a) donne le courant au milieu de la branche gauche pour notre approche hybride (noté "In-Cell" sur la figure), la méthode de R. Holland en considérant un conducteur équivalent pour la bifilaire et la méthode intégrale. La figure 3.13(b) donne le courant au milieu de l'un des deux conducteurs de la bifilaire et le courant de mode commun i_c.

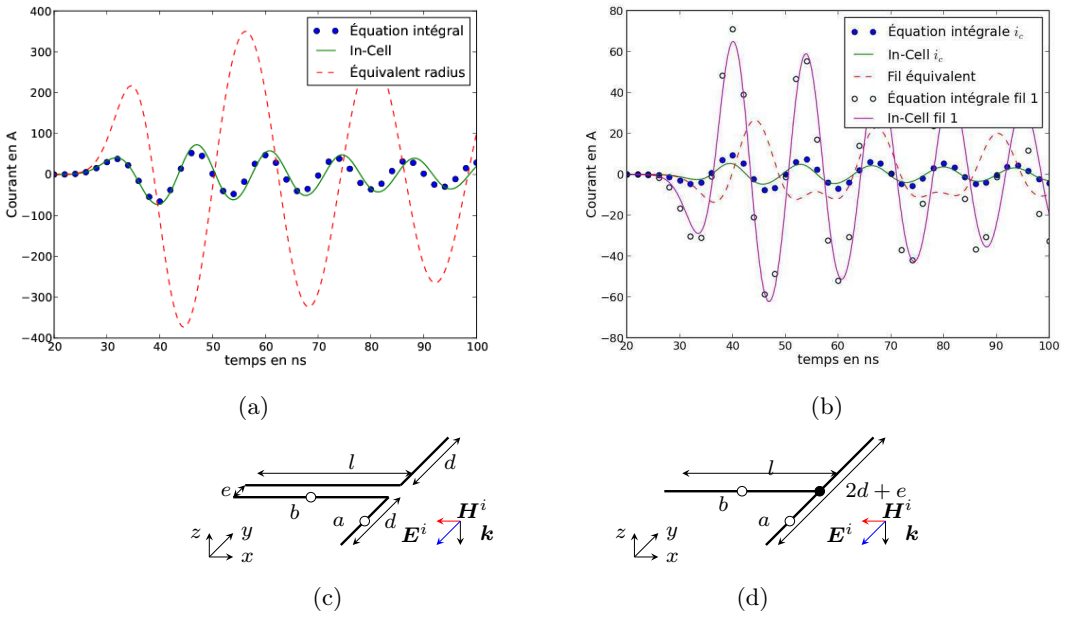


FIGURE 3.13 – Présentation de la géométrie de l'antenne dipolaire (c) et la géométrie avec l'utilisation du fil équivalent (d) et représentation du courant médian sur le segment de gauche (a) et le courant médian sur la bifilaire (b).

Sur ces résultats, nous remarquons une bonne concordance entre les solutions obtenues avec notre approche et la méthode intégrale. En revanche, dans ce cas de figure, l'approxi-

mation de la bifilaire par un conducteur équivalent n'est pas pertinente, ce qui est principalement dû à l'erreur faite au nœud de courant. Sur cet exemple, nous voyons très nettement l'intérêt de notre approche hybride qui est plus générale que l'approximation du modèle de R. Holland en permettant d'avoir accès au courant du mode différentiel. De ce fait, l'utilisation de notre formulation est très intéressante pour modéliser toute la complexité d'un équipement terminal car nous avons accès aux courants sur tous les conducteurs.

3.6.6 Blindage et reprise de coupure

Dans cette section, nous présentons deux autres cas tests pour montrer les capacités offertes par notre approche hybride. Nous n'avons donc pas fait d'étude comparative sur les résultats obtenus dans cette section. Nous nous sommes limités à une étude qualitative.

a. La reprise de coupure

La reprise de coupure est un problème industriel récurrent qui concerne le cas de figure où une ligne de transmission traverse une paroi. Habituellement la ligne de transmission traverse la paroi par le biais d'un connecteur et il est d'usage de reprendre à la masse tous les conducteurs de référence, ce n'est pas toujours le cas. En effet, il se peut que certains conducteurs soit connectés avec un certain nombre d'éléments parasites (résistance, inductance, etc...) introduit par le connecteur. Modéliser une reprise de coupure est donc un problème difficile, même dans le cas où il n'y a pas d'élément parasite.

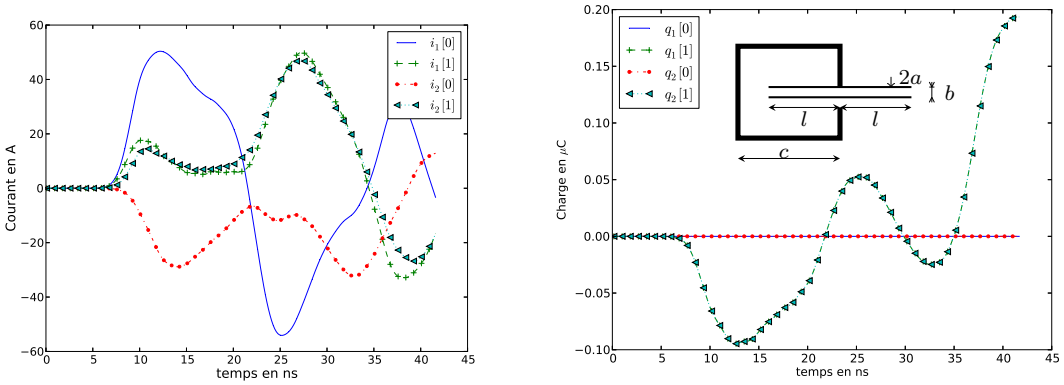
Dans l'exemple que nous proposons, nous considérons une bifilaire, composée de deux fils espacés de $d = 2$cm et de rayon $a = 1$mm, qui pénètre dans une enceinte fermée (cube métallique de longueur d'arête $c = 2.8$m). Le fil fait une longueur de $2l = 4$m également répartie à l'intérieur et à l'extérieur de l'enceinte fermée. A la reprise de coupure, l'un des conducteurs est repris à la masse (connecté directement à la surface). L'autre conducteur, quant à lui, ne subit pas de modification, comme cela est indiqué sur la figure 3.14(b).

La figure 3.14(a) nous donne le courant à une demie maille de part et d'autre de la surface. Nous remarquons que le courant, qui n'est pas repris à la masse, n'est que peu modifié à la traversée de la paroie, alors que le courant du conducteur repris à la masse, est très nettement différent de part et d'autre de la paroi. La figure 3.14(a) nous donne la charge portée par chacun des conducteurs. Pour le conducteur repris à la masse, nous trouvons une charge nulle qui traduit bien de la mise à la masse de celui-ci.

Cet exemple montre encore une fois l' intérêt de notre approche à pouvoir tenir compte de la totalité des conducteurs de la ligne dans la simulation pour obtenir des solutions pertinentes aux problèmes physiques que l'on se pose.

b. Les fils gainés

Cet exemple consiste en la reprise du cas test à neuf conducteurs (Sec. 3.6.3 page 116), auxquels nous avons ajouté une gaine de diélectrique de rayon $d = 2a = 2$mm et de permittivité relative $\varepsilon_r = 4.2$ correspondant approximativement à un caoutchouc silicone. Nous comparons qualitativement le résultat par rapport au cas où il n'y a pas de gaine. Nous



(a) (b)

FIGURE 3.14 – Présentation de la géométrie du cas test « reprise de coupure » (b) et représentation du courant (a) à une demie-maille de part et d'autre de la reprise de coupure, ainsi que la charge portée (b) par chacun des conducteurs.

voyons très nettement l'influence des gaines caoutchouc sur le comportement global des courants sur les conducteurs. Nous pouvons noter en particulier, un ralentissement de l'onde de propagation, due certainement au gaines diélectriques, que nous voyons clairement sur le courant du fil 2.

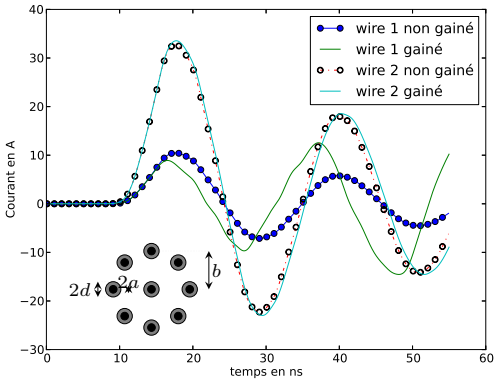


FIGURE 3.15 – Présentation de la géométrie du cas test à 9 fils avec diélectrique ($a = 1$mm, $d = 2$mm et $b = 2$cm) et représentation du courant médian pour différentes méthodes numériques.

Cet exemple montre encore la capacité de notre méthode hybride à traiter des cas complexes avec des fils gainés. L'intérêt de notre approche est que la méthode 3D de calcul des champs n'est pas impactée par la complexité de la géométrie de la ligne de transmission. La

complexité des câblages pris en compte dans nos simulations est reportée dans le calcul des paramètres de la ligne de transmission qui est effectué par la résolution d'un problème de Laplace.

3.7 Conclusion

Dans ce chapitre nous avons présenté une approche hybride dans le domaine temporel entre une méthode 3D de type FDTD pour calculer les champs électromagnétiques et une méthode FDTD 1D pour modéliser les lignes de transmission. L'intérêt de cette approche réside dans le fait de pouvoir prendre en compte des faisceaux de câbles de géométrie complexe avec ou sans diélectrique, ainsi que des réseaux multi-branches réalistes. Nous pouvons en particulier introduire des nœuds de courant permettant des connexions fil à fil à l'intérieur de chaque faisceau. Un certain nombre de configurations ont été présentées et nous ont permis de valider notre approche.

La stratégie mise en œuvre, nous permet d'introduire d'autres modèles sur les lignes de transmission sans modifier le code 3D. Nous pourrions par exemple étendre les modèles de conditions aux limites pour modéliser des équipements terminaux en utilisant un modèle circuit.

Pour le moment, nous nous sommes surtout focalisé sur des fils qui suivent le maillage. Nous avons vu que le couplage de la ligne avec le champ ne se fait qu'au moyen d'un mode commun. Cette approche pourrait être envisagée sans trop de difficultés de mise en œuvre sur des faisceaux de câbles dont la position serait indépendante du maillage.

Chapitre 4

Hybridation forte entre lignes de transmission et modèle 3D

Sommaire

4.1 Introduction 126
4.2 Stratégie d'échange de données dans une méthode FDTD - Parallélisation de code MTL 127
4.2.1 Principe général de la parallélisation 127
4.2.2 Hybridation coopérante 128
4.2.3 Hybridation collaborative 129
4.2.4 Validation numérique 130
4.3 Hybridation de domaines 134
4.3.1 Extrapolation de l'hybridation 1D/1D 134
4.3.2 Continuité du champ électromagnétique transversal 136
4.3.3 Calcul du champ dans le plan d'hybridation 137
4.4 Validation numérique et étude paramétrique 141
4.4.1 Influence de la longueur d'intégration 141
4.4.2 Limites fréquentielles de l'approche 142
4.4.3 Hybridation et sol de conductivité finie 145
4.4.4 Prise en compte d'une source rayonnante 147
4.5 Conclusion 152

4.1 Introduction

Dans le chapitre précédent, nous avons étudié une stratégie d'hybridation 1D/3D, dite d'échelle, entre une méthode 3D et une méthode de résolution des équations de ligne de transmission pour traiter des faisceaux de câbles réalistes. Dans ce chapitre, nous allons toujours nous intéresser à une hybridation entre une méthode 3D et une méthode de ligne de transmission, mais cette fois dans un cadre multi-domaines.

L'objectif est de mettre en place une stratégie d'hybridation permettant de découper une grande scène où certaines zones peuvent être modélisées par la théorie des lignes de transmission. Cette approche a pour vocation de modéliser les effets des réseaux entre les domaines en évitant pour cela, l'utilisation d'une méthode 3D.

Avant de présenter le principe de l'approche hybride 1D/3D envisagée pour la décomposition de domaines, nous allons étudier l'hybridation 1D/1D de deux lignes en restant dans la théorie des lignes de transmission. Cette étude nous permettra de définir les quantités sur les équations de ligne de transmission dont il faudra tenir compte dans l'hybridation. Cette hybridation peut être également vue comme une parallélisation. Cette première étude nous permettra aussi, d'envisager les solutions informatiques de manière indépendante de la problématique d'hybridation. Elle nous permettra également d'appréhender une architecture logicielle souple, modulaire et non intensive (vision boite noire). Les aspects informatiques seront plus largement développés dans la dernière partie de ce mémoire (Chapitre 7 page 192).

Ensuite, nous nous intéressons à la stratégie d'hybridation 1D/3D sur une ligne de transmission. Nous commençons par étendre simplement l'hybridation 1D/1D à l'hybridation 1D/3D, mais il s'avère que cette approche n'est pas satisfaisante pour traiter correctement le problème. Nous avons donc proposé une solution qui corrige les défauts de l'approche précédente et nous avons validé cette méthode sur un ensemble de cas tests.

4.2 Stratégie d'échange de données dans une méthode FDTD - Parallélisation de code MTL

Dans cette section, nous allons présenter une stratégie permettant de paralléliser ou d'hybrider un code de ligne de transmission. L'étude de cette stratégie revêt plusieurs points intéressants :

- La parallélisation d'une méthode FDTD-MTL nous permet de poser le problème des échanges de données dans l'hybridation. Nous verrons que plusieurs approches sont possibles et équivalentes. Cependant, certain choix permettent d'avoir des algorithmes plus simples et plus rapides ;
- L'apport de l'hybridation permet un gain certain sur le temps calcul et sur le coût mémoire ;
- La stratégie d'hybridation 1D/1D est le premier pas vers l'hybridation 1D/3D. En effet, dans le cas de notre décomposition de domaines, nous conservons une partie de la ligne dans le domaine 3D, nous pouvons espérer résoudre l'hybridation 3D/1D de la même manière que l'hybridation 1D/1D.

4.2.1 Principe général de la parallélisation

L'idée générale de la parallélisation (Fig. 4.1) d'un code temporel est de découper la géométrie du problème initial en plusieurs sous domaines. La solution globale est garantie en appliquant une condition d'interface entre les domaines. L'intérêt de cette approche est de pouvoir exploiter des architectures distribuées (Chapitre 7 page 192) qui permettent d'adresser plus de mémoire. Parfois cela est crucial. En effet, si le cas ne passe pas sur une architecture simple, nous avons un véritable bénéfice à utiliser une architecture distribuée car elle donne un résultat.

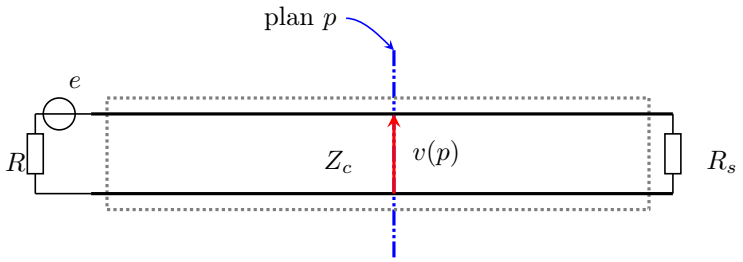
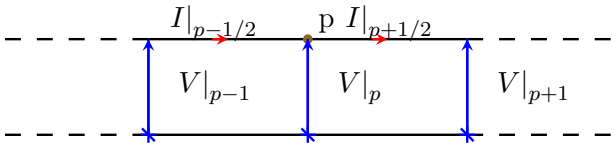


FIGURE 4.1 – Principe de découpage pour la parallélisation

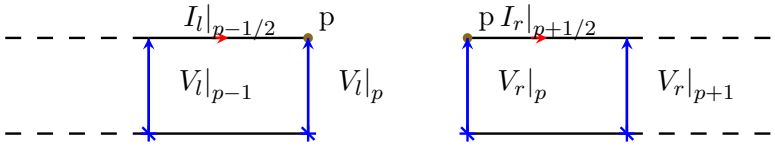
Le principe de l'hybridation dans la méthode de résolution des équations de ligne de transmission décrite dans le section 2.3 page 57 s'appuie fortement sur le décalage spatial des inconnues dans le schéma leap-frog rappelé ci-dessous :

$$I|_{p+1/2}^{n+1/2} = -Q_i \cdot\ I|_{p+1/2}^{n-1/2} + Q_v \cdot \left(V|_{p+1}^{n} - V|_{p}^{n}\right)$$
$$V|_{p}^{n+1} = -K_v \cdot\ V|_{p}^{n} + K_i \cdot \left(I|_{p+1/2}^{n+1/2} - I|_{p-1/2}^{n+1/2}\right)$$

Supposons que nous voulons résoudre ce système d'équations sur une ligne de transmission simple formée d'un seul fil (Fig. 4.2(a)) en découpant celle-ci en deux zones distinctes. Nous allons choisir une interface qui dans notre cas se limite à un point (p). La partie gauche par rapport à cette interface sera résolue par une instance du code, la partie droite par une autre instance.



(a) Ligne de transmission initiale découpée au point p



(b) Ligne de transmission découpée en deux au point p

FIGURE 4.2 – Illustration du découpage d'une ligne de transmission dans le but de faire un calcul parallèle.

La particularité de cette approche est que nous avons la même inconnue $V|_p$ calculé par deux instances différentes du code (Fig. 4.2(b)). Nous devons donc garantir l'intégrité [1] de cette inconnue. De plus, pour évaluer à l'itération suivante cette inconnue, nous devons disposer des inconnues $I|_{p-1/2}$ et $I|_{p+1/2}$ qui ne sont pas localisées dans les mêmes instances du code. Rappelons que chaque instance du code n'a pas accès à l'espace mémoire de l'autre. Ainsi, pour résoudre ce problème de localisation spatiale des inconnues et garantir leur intégrité, nous devons échanger des informations au niveau de cette interface. Cet échange d'information peut se faire de deux manières ; la première dite coopérante l'autre collaborative. Elles se distinguent dans la manière de calculer la tension au point d'interface.

Nous allons nous intéresser à ces deux approches afin de déterminer la plus performante. Ce choix sera ensuite repris dans l'hybridation 1D/3D.

4.2.2 Hybridation coopérante

Dans l'hybridation coopérante, chaque instance du code va calculer la tension en p. Elles sont en concurrence sur ce point. Pour pouvoir calculer correctement cette tension, nous devons au préalable échanger (d'un code à l'autre) les courants d'extrémités (Fig. 4.3).

1. l'intégrité désigne le fait de garantir la cohérence de la donnée à chaque instant où elle est utilisée

$$V_l|_p^{n+1} = -K_v \cdot V_l|_p^n + K_i \cdot \left(I_r|_{p+1/2}^{n+1/2} - I_l|_{p-1/2}^{n+1/2} \right)$$
$$V_r|_p^{n+1} = -K_v \cdot V_r|_p^n + K_i \cdot \left(I_r|_{p+1/2}^{n+1/2} - I_l|_{p-1/2}^{n+1/2} \right)$$

où r représente la première instance du code et l la deuxième. La figure suivante illustre l'hybridation coopérante :

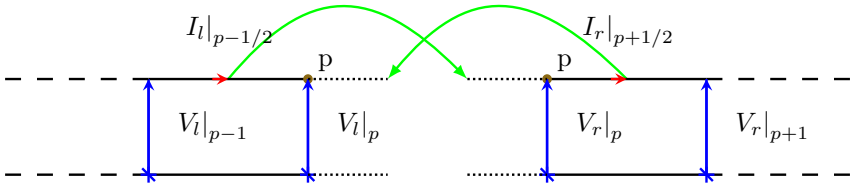


FIGURE 4.3 – Illustration d'une hybridation coopérante

Notons que dans cette approche nous avons fait l'hypothèse que les paramètres linéiques de la ligne étaient constants et égaux à gauche et à droite du point d'hybridation. Nous aurions pu aussi utiliser un nœud de courant pour l'interface et proposer une hybridation similaire en tenant compte des équations sur i et v décrites en ce nœud p, est données par :

$$I_m|_{p+1/2}^{n+1/2} = -Q_{i,m} \cdot I_m|_{p+1/2}^{n-1/2} + Q_{v,m} \cdot \left(V_m|_p^n - V_m|_{p-1}^n \right)$$
$$V_m|_p^{n+1} \sum_{m=1}^{M} A_m = - V_m|_p^n \sum_{m=1}^{M} B_m + \sum_{m=1}^{M} I_m|_{p-1/2}^{n+1/2}$$

avec m représentant le numéro de l'unité de calcul (ou branche) au point d'hybridation :

$$Q_{i,m} = \left(\frac{R_m \Delta t - 2L_m}{R_m \Delta t + 2L_m} \right), \; Q_{v,m} = \frac{2}{R_m \Delta t + 2L_m} \frac{\Delta t}{\Delta l_m}$$
$$A_m = \left(\frac{G_m \Delta t + 2C_m}{4} \right) \frac{\Delta l_m}{\Delta t}, \; B_m = \left(\frac{G_m \Delta t + 2C_m}{4} \right) \frac{\Delta l_m}{\Delta t}$$

Remarquons sur ces équations que chaque instance du code doit connaître les paramètres linéiques de chaque extrémité pour pouvoir calculer la tension en p. De plus, l'intégrité de la variable $V_m|_p^{\cdot}$ est garantie en exploitant la loi de Kirchhoff donnée par $\sum_{m=1}^{M} i_m(p) = 0$.

4.2.3 Hybridation collaborative

L'hybridation collaborative consiste à échanger à la fois les courants et les tensions. Dans cette forme d'hybridation, la première instance du code envoi à l'autre le courant afin qu'elle puisse calculer la tension au point p. Ensuite la deuxième instance envoi cette tension à l'autre (Fig. 4.4). L'hybridation collaborative peut se formaliser de la manière suivante :

$$V_r|_p^{n+1} = -K_v \cdot V_r|_p^n + K_i \cdot \left(I_r|_{p+1/2}^{n+1/2} - I_l|_{p-1/2}^{n+1/2} \right)$$
$$V_r|_p^n = V_l|_p^n$$

où r représente la première instance du code et l la deuxième.

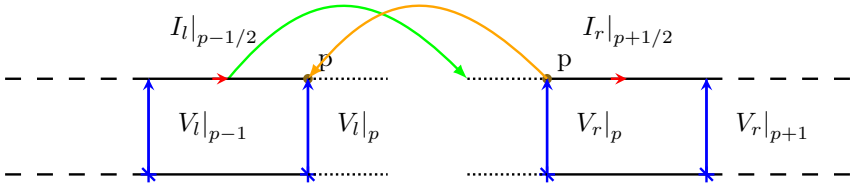


FIGURE 4.4 – Illustration d'une hybridation collaborative

En terme de parallélisation, l'approche coopérante est plus pertinente. En effet, dans cette approche il y a moins de temps d'« attente » qui est un facteur de ralentissement dans un calcul parallèle. La figure 4.5(a) illustre la succession des étapes réalisées en parallèle dans l'approche coopérante. Dans cette approche, chaque code réalise la même étape que les autres. Il n'y a donc pas de perte de temps si la charge est équilibrée.

En revanche dans l'approche collaboratrive 4.5(b), chaque instance de code doit attendre que les autres finissent leur travail pour pouvoir en bénéficier. Cette attente engendre une perte de temps (boite grise dans la figure). Dans les approches de calcul parallèle, il est préférable quand cela est possible de recalculer une donnée en parallèle que d'attendre et échanger celles-ci.

Hormis le risque de perte de temps dans l'approche collaboratrive, cette approche souffre d'un autre problème. En effet, des instances du codes ne sont pas symétriques. Nous devons donc implémenter deux algorithmes différents dans le même code, ce qui implique plus de complexité et donc plus de risque d'erreurs.

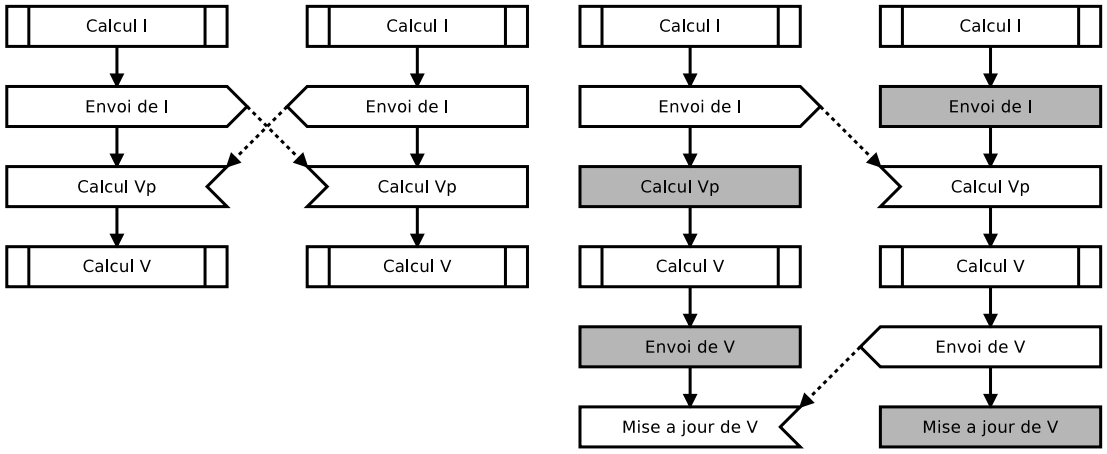
4.2.4 Validation numérique

Pour illustrer ces différentes stratégies d'hybridation, nous allons étudier leur comportement sur un cas test simple. Ce cas test est composé d'une ligne de transmission d'impédance caractéristique réelle. Nous allons simuler le comportement de cette ligne de transmission en utilisant trois méthodes (un seul code de ligne, deux codes en collaboration et deux codes en coopération).

La figure 4.6 trace le courant au point $l = 40$m et $l = 60$m pour les trois cas de calculs. Nous rappelons que le point d'hybridation est placé au milieu de la ligne c'est-à-dire à $p = 50$m. Nous constatons une parfaite concordance entre les trois courbes, ce qui valide les deux techniques d'hybridation.

En ce qui concerne les performances en temps de calcul, le tableau 4.1 reprend le temps de calcul pour 20 calculs successifs.

Le résultat obtenu peut sembler surprenant. En effet la parallélisation du code n'apporte que très peu de gain en temps de calcul, voire une perte avec l'approche collaborative.



(a) Représentation graphique des étapes de la boucle principale pour une hybridation coopérante

(b) Représentation graphique des étapes de la boucle principale pour une hybridation collaboratrive. Les boites grises représentent le temps perdu à attendre que le collaborateur réalise son action

FIGURE 4.5 – Comparaison des différentes étapes de la boucle principale entre les deux approches d'hybridation. Cette comparaison illustre la perte de temps (boîte grise) faite dans l'approche collaborative.

	directe	coopérante	collaborative
temps	$29''829$	$29''409$	$32''818$
gain	1	1.014	0.909

TABLE 4.1 – Évaluation et comparaison du temps de calcul limite pour les trois approches (directe, coopérante, collaborative)

La raison est très simple, la mise en place d'un code parallèle nous impose de faire des échanges de données en utilisant un protocole de communication. La transmission de ces messages n'est pas instantanée. En général, le temps de transfert d'un message est estimé par $T = t_i + size(msg) \cdot t_o$ où t_i est le temps d'initialisation et t_o le temps de transfert par octet. Nous pouvons donc évaluer le gain de la parallélisation du code à $g = \frac{2}{1+\frac{4(t_o+t_i)}{lt_c}}$. Un calcul parallèle ne peut donc être performant que si et seulement si $l \cdot t_c > 4(t_o+t_i)$, avec l la taille du problème et t_c la durée d'une itération. Dans le cas de notre exemple, nous sommes à la limite de cette contrainte et donc nous ne pouvons pas espérer de gain flagrant en temps.

Pour obtenir l'expression du gain g nous faisons les hypothèses suivantes :

Hypothèse 4.2.1. *le temps de calcul d'une itération est proportionnel à la taille du problème. Ainsi $T_{cal}(l) = n \cdot l \cdot t_c$ où n est le nombre d'itérations, l la taille du problème (nombre d'éléments unitaire de calcul) et t_c le temps d'une itération.*

Hypothèse 4.2.2. *Nous supposons que le temps perdu dans cette approche est unique-*

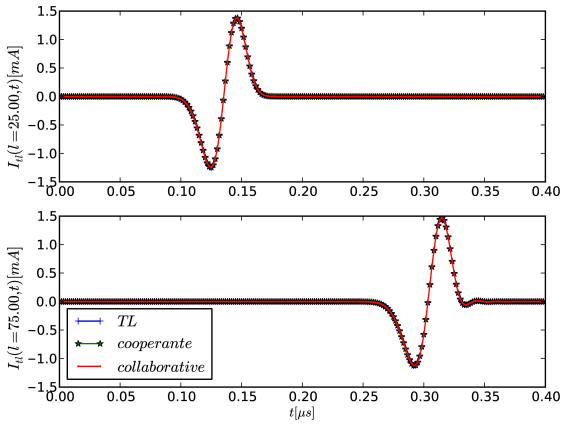


FIGURE 4.6 – Comparaisons des deux courants évaluées le long de la ligne de transmission pour les trois approches (directe, coopérante, collaborative)

ment lié au transfert d'information d'un code à l'autre (le temps de calcul de la tension d'hybridation est négligé). Le temps propre d'hybridation pour l'approche coopérante est : $T_{hyb} = 2n(t_o + t_i)$ *avec* t_i *le temps d'initialisation et* t_o *le temps de transfert par octet.*

Nous définissons alors le gain par :

Définition 4.2.1. *Le gain en temps de calcul de l'hybridation est défini comme suit :*

$$g = \frac{T_{cal}(l)}{(T_{cal}(l/m) + T_{hyb})} \tag{4.1}$$

avec T_{cal} *le temps utile de calcul,* T_{hyb} *le temps propre à l'hybridation et* m *le degré de parallélisation.*

Sous les hypothèses précédentes, nous obtenons alors :

Proposition 4.2.1. *Le gain en temps de calcul est donné par :*

$$g = \frac{2}{1 + \frac{4(t_o+t_i)}{lt_c}}$$

Si nous reprenons le cas de calcul et si nous utilisons une discrétisation plus fine (facteur 2), nous augmentons la charge de calcul (l) et donc ainsi le gain. Le tableau 4.2 montre les temps et les gains obtenus avec cette nouvelle configuration. Nous remarquons maintenant que nous avons un gain supérieur à 2, qui correspond à la limite théorique calculée précédemment. Cette limite théorique peut être même dépassée car elle ne prend pas en compte un certain nombre d'éléments comme le temps d'initialisation de l'unité de calcul, le temps d'écriture des données, etc...

	seule	coopérante	collaborative
temps	$61''$	$29''$	$32''$
gain	1	2.10	1.9

TABLE 4.2 – Évaluation et comparaison du temps de calcul pour les trois approches (seule, coopérante, collaborative) sur un problème plus volumineux

4.3 Hybridation de domaines

Dans la première section de ce chapitre, nous avons vu comment faire une hybridation 1D/1D d'une méthode temporelle de résolution des équations de lignes de transmission. Dans cette section, nous allons étendre cette approche à une hybridation 1D/3D utilisant deux méthodes basées sur des modèles différents (Maxwell 3D et lignes de transmission). Dans cette hybridation nous faisons l'hypothèse que les méthodes de chaque sous-modèle traitent une partie de la ligne.

Nous montrons dans un premier temps des approches « naïves » directement inspirées de l'hybridation 1D/1D, qui s'avèrent insuffisantes pour l'hybridation 1D/3D. Nous proposons alors de tenir compte d'une condition de continuité du champ pour corriger ce problème, et enfin, nous donnons un certain nombre de validations numériques.

4.3.1 Extrapolation de l'hybridation 1D/1D

Dans cette section, nous allons donner deux premières approches pour connecter plusieurs lignes entre elles, qui s'inspirent des stratégies collaborative et coopérante décrites précédemment.

a. Extrapolation du principe d'hybridation coopérante

Dans cette approche, nous cherchons à exploiter le principe de l'hybridation coopérante (Sec. 4.2.2 page 128). Pour pouvoir appliquer ce principe, nous devons avoir de part et d'autre du plan d'hybridation un modèle décrivant la même ligne de transmission.

Nous pouvons extrapoler le principe de coopération comme illustré sur la figure 4.7. L'intérêt de cette approche est que nous ne manipulons pas directement les inconnues v et q mais i. D'un point de vue physique, cette approche est basée sur le fait que le courant calculé de part et d'autre de l'interface par des méthodes différentes est le même.

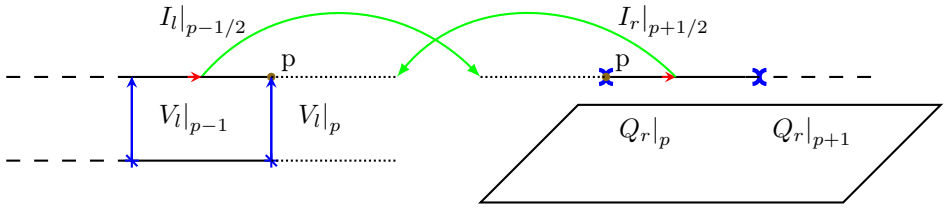


FIGURE 4.7 – Extrapolation du principe d'hybridation concurrente pour l'hybridation FDTD/MTL.

Cependant la méthode présentée montre une faiblesse. En effet, lors des transferts des données entre les domaines nous n'échangeons qu'un courant sur la ligne sans l'environnement électromagnétique autour de celui-ci. Ceci ne pose aucun problème pour le transfert du 3D vers le 1D, mais n'est pas suffisant pour le transfert du 1D vers le 3D, comme nous avons pu le constater sur des exemples.

b. Extrapolation du principe d'hybridation collaborative

Dans cette approche, nous nous inspirons de l'approche collaborative définie précédemment (Sec. 4.2.3) qui échange à l'interface l'inconnue sur la charge q ou la tension v. La difficulté est de savoir si nous devons assurer la continuité sur la charge q ou la tension v entre les domaines.

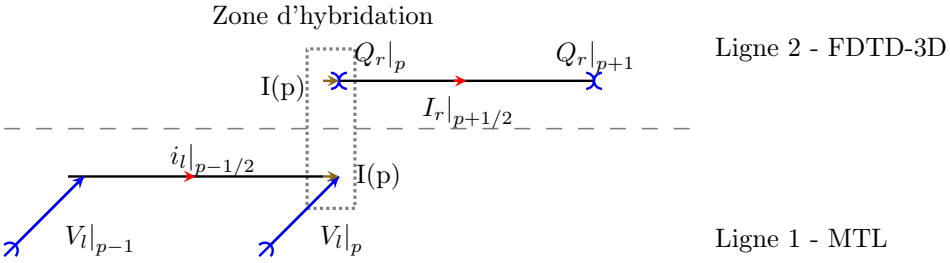


FIGURE 4.8 – Schéma de principe de l'hybridation FDTD/MTL.

Avant d'étudier les deux cas de figure, nous rappelons que la charge est liée à la différence de potentiel par la relation constitutive suivante $q = C \cdot v$. Dans le cas de la ligne de transmission, la capacité et la tension sont bien définies. Dans le cas du fil de R. Holland, la capacité C_h est liée à l'inductance de couplage L_h de R. Holland par la relation suivante :

$$\begin{aligned} \nu &= \frac{1}{\sqrt{L_h C_h}} = \frac{c_0}{\sqrt{\mu\varepsilon}} \\ \Rightarrow \quad C_h &= \frac{\mu\varepsilon}{c_0^2 L_h} \end{aligned} \tag{4.2}$$

où nous rappelons que l'inductance de R. Holland peut être approchée par :

$$L_h = \frac{\mu_0}{2\pi}\left(\ln\left(\frac{R}{a}\right) - \frac{1}{2} + \frac{a^2}{2R^2}\right) \tag{4.3}$$

avec πR^2 = aire de la cellule ($dx \cdot dy$ pour un fil suivant l'axe z). De cette manière, R. Holland voit les fils comme une ligne de transmission entre le fil et la cellule de Yee excitée par le champ électrique longitudinal moyen dans la cellule $\langle E_s \rangle$. La rétroaction de cette ligne de transmission dans le 3D est garantie par le principe de réciprocité [25]. Ainsi, la différence de potentiel de la ligne de transmission est définie entre le fil et la cellule.

Note : Il faut être vigilant de ne pas confondre la ligne de transmission définie par R. Holland et plus généralement dans la modélisation des fils minces et la ligne de transmission globale. Ces deux lignes de transmission sont totalement distinctes.

En définitive, nous avons deux stratégies possibles pour faire ce changement de variables : soit en imposant la continuité en tension $v_l = v_h$, soit en imposant la continuité de la charge $q_l = q_h$.

Quelque soit l'approche utilisée, nous créons une désadaptation liée à la différence du modèle de part et d'autre du plan d'hybridation. Dans la théorie des lignes de transmission, nous cherchons la solution au problème de Maxwell sous la forme d'onde électrique, qui est une forme partiellement intégrée des équations de Maxwell. En revanche, dans un code 3D, nous résolvons la forme différentielle des équations de Maxwell. Ainsi, dans le modèle de ligne de transmission, la variation du courant représente une variation du courant et du champ magnétique dans tout le plan transversal, alors que dans le domaine 3D, elle reste localisée à la cellule.

Ainsi pour répondre de manière satisfaisante au problème d'hybridation sur une ligne de transmission, nous devons assurer aussi la continuité du champ électrique et magnétique à l'interface d'hybridation et non uniquement la continuité des courants. Nous allons voir dans les prochaines sections quel est le sens de cette continuité dans le cas d'une ligne de transmission mais également comment nous pouvons pratiquement assurer cette continuité.

4.3.2 Continuité du champ électromagnétique transversal

Dans la théorie des lignes de transmission, la cartographie du champ électromagnétique est implicitement établie à travers le courant i, la tension v et la géométrie de la ligne qui définit les paramètres de celle-ci. Ceci est dû au fait que, dans le cas de l'approximation TEM, les champs électrique et magnétique dans le plan transversal de la ligne suivent une distribution quasi-statique. Nous avons alors des relations qui relient ces champs avec les courants et tensions établis sur la ligne.

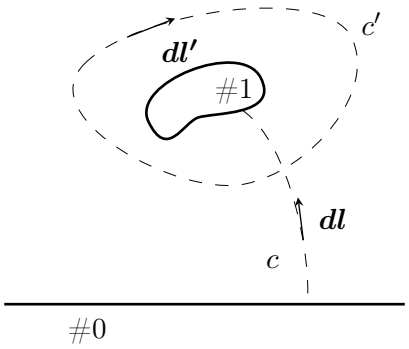


FIGURE 4.9 – Illustration d'un câble au-dessus d'un plan parfaitement conducteur et des surfaces définissant la différence de potentiel et de courant.

En ce qui concerne la tension, comme décrit par l'équation 4.4 et illustrée par la figure (Fig. 4.9), la différence de potentiel $v(l,t)$ entre 2 conducteurs est donnée par l'intégration du champ électrique sur n'importe quel chemin c, dans le plan transversal, entre ces deux conducteurs. En ce qui concerne le courant $i(l,t)$, celui-ci est donné par l'intégrale curviligne du champ magnétique sur un chemin fermé c' qui entoure le conducteur (Eq. 4.5).

$$v(l,t) = -\int_c \boldsymbol{E_t} \cdot \boldsymbol{dl} \tag{4.4}$$

$$i(l,t) = \oint_{c'} \boldsymbol{H_t} \cdot \boldsymbol{dl'} \tag{4.5}$$

Ainsi garantir la continuité de la tension et du courant est équivalent à garantir la continuité du champ électromagnétique dans le plan transverse à la ligne.

Dans le cas d'une hybridation du 3D vers le 1D, nous pouvons directement utiliser la charge q et le courant i calculés dans le 3D pour les injecter dans le 1D et ainsi assurer la continuité des courants, charges mais aussi des champs transverses dans le modèle de ligne de transmission. En effet, les champs électriques et magnétiques, comme nous venons de le voir, découleront directement des courants et des charges.

En revanche, dans une hybridation 1D vers 3D, la théorie des lignes de transmission ne donne pas directement accès aux champs dans le plan transversal à la ligne. Ainsi, afin d'assurer la continuité de ces champs dans le domaine 3D, où le modèle de fil est local à la maille, nous devons calculer ceux-ci à partir des courants et des charges de la ligne de transmission pour les imposer dans le domaine 3D.

4.3.3 Calcul du champ dans le plan d'hybridation

Nous avons vu dans la section précédente que nous avions une désadaptation de la ligne de transmission à l'interface entre les deux sous-domaines 1D et 3D. Nous avons expliqué que cette désadaptation était due au fait que nous ne garantissions pas continuité du champ électromagnétique tangentiel à l'interface entre les deux sous-domaines 1D et 3D. Afin de garantir cette continuité, nous devons calculer les champs tangents implicitement définis dans la théorie des équations de lignes de transmission et les appliquer au domaine 3D. Pour cela, nous utiliserons une formule de rayonnement des courants.

a. Formalisme intégral

Nous pouvons expliciter le champ dans le plan d'interface entre les domaines 3D et 1D, en exploitant un formalisme intégral qui permet de calculer le champ rayonné par une distribution de courant. Soit la figure 4.10 qui représente le plan dans lequel nous voulons calculer le champ électromagnétique associé à un fil de rayon a, supposé en espace libre et portant une distribution de courant $i(l,t)$. Le champ au point d'observation $\boldsymbol{M} = (\rho, \theta, 0)$ décrit dans le repère cylindrique $(\boldsymbol{u}_\rho, \boldsymbol{u}_\theta, \boldsymbol{u}_z)^T$ centré sur le plan transverse avec le vecteur $\boldsymbol{u}_z$ qui suit l'axe du fil, s'écrit sous la forme suivante [53] :

$$\boldsymbol{H}(r,t) \cdot \boldsymbol{u}_\theta = \frac{1}{4\pi} \int_{-\infty}^{0} \left\{ \frac{r}{R^3} i(l, t - R/c) + \frac{r}{cR^2} \partial_t i(l, t - R/c) \right\} dl \tag{4.6a}$$

$$\boldsymbol{E}(r,t) \cdot \boldsymbol{u}_r = \frac{1}{4\pi\varepsilon} \int_{-\infty}^{0} \left\{ \frac{3rl}{R^5} \int_0^t i(l, t - R/c) dt + \frac{3rl}{cR^4} i(l, t - R/c) + \frac{rl}{c^2 R^3} \partial_t i(l, t - R/c) \right\} dl \tag{4.6b}$$

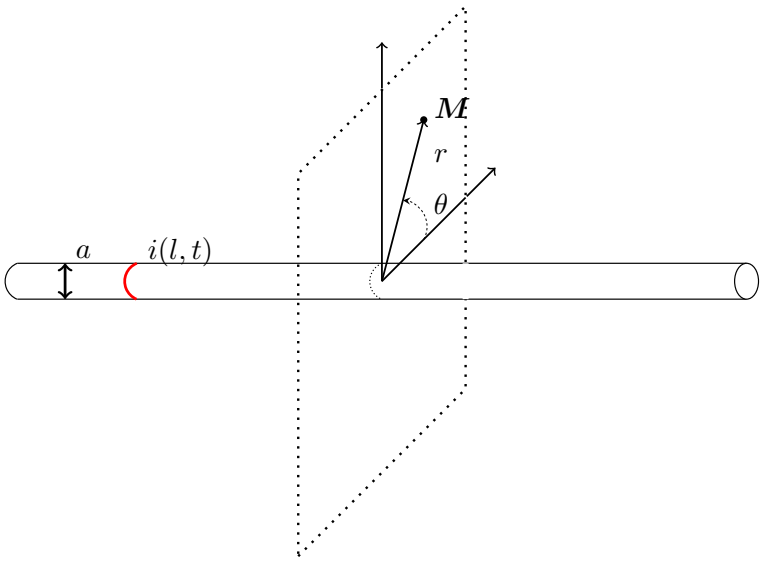


FIGURE 4.10 – Représentation du fil en espace libre et du plan d'intérêt

où l désigne la position d'un courant sur le fil et R la distance de ce même courant au point d'observation, définie par $R = \sqrt{r^2 + l^2}$. Dans cette formulation, le rayon du fil est négligé et nous exploitons la relation $\partial_R i(l, t - R/c) = -\frac{1}{c}\partial_t i(l, t - R/c)$.

Nous pouvons donc calculer le champ électromagnétique dans le plan de l'interface d'hybridation par la formule intégrale précédemment donnée.

Dans les formules précédentes, nous pouvons raisonnablement approcher ces intégrales en ne considérant que l'environnement proche du plan de l'interface d'hybridation ($l \leq p$) En effet, la contribution d'un élément de courant sur le champ électrique et magnétique décroît très rapidement ($1/R^2$) et devient donc très vite négligeable. Nous pouvons alors déterminer une longueur p telle que l'on puisse approcher correctement le champ électromagnétique dans le plan de l'interface d'hybridation par :

$$\hat{\boldsymbol{H}}(r,t) \cdot \boldsymbol{u}_\theta = \frac{1}{4\pi} \int_{-p}^{0} \left\{ \frac{r}{R^3} i(l, t - R/c) + \frac{r}{cR^2} \partial_t i(l, t - R/c) \right\} dl \tag{4.7a}$$

$$\hat{\boldsymbol{E}}(r,t) \cdot \boldsymbol{u}_r = \frac{1}{4\pi\varepsilon} \int_{-p}^{0} \left\{ \frac{3rl}{R^5} \int_0^t i(l, t - R/c) dt + \frac{3rl}{cR^4} i(l, t - R/c) + \frac{rl}{c^2 R^3} \partial_t i(l, t - R/c) \right\} dl \tag{4.7b}$$

Il y a en fait deux techniques différentes pour introduire ces champs dans l'interface. La première comme nous venons de le voir utilise une formulation intégrale pour calculer le champ électromagnétique dans le plan d'hybridation. Cette approche impose d'échanger le champ électromagnétique sur tout un plan, ce qui peut pénaliser les performances de la méthode. La deuxième approche (Fig. 4.12), que nous avons utilisée, consiste à définir une zone de recouvrement du domaine 1D sur le domaine 3D sur laquelle, nous imposons dans le domaine 3D les courants calculés dans le domaine 1D. De ce fait, les champs électromagnétiques dans le plan d'hybridation sont directement calculés dans le domaine 3D à l'aide

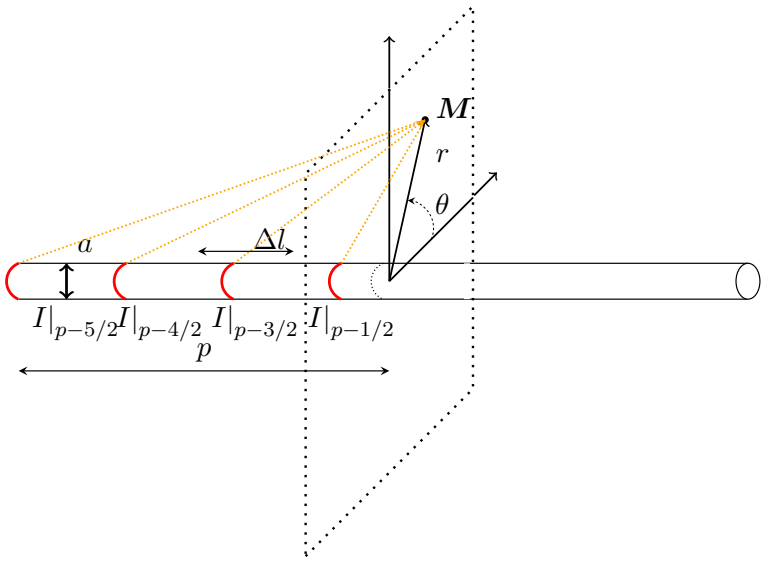


FIGURE 4.11 – Représentation du fil en espace libre et du plan d'intérêt

des courants du domaine 1D. De manière pratique, nous utilisons les P derniers courants obtenus par le code de ligne de transmission pour les appliquer comme termes source sur la ligne dans le domaine 3D. Dans cette configuration, le plan d'hybridation se trouve au bout de la ligne de transmission à $P + 1/2$ segments de la ligne dans le domaine 3D. La difficulté de la méthode reste l'évaluation de P que nous quantifierons dans notre premier exemple numérique.

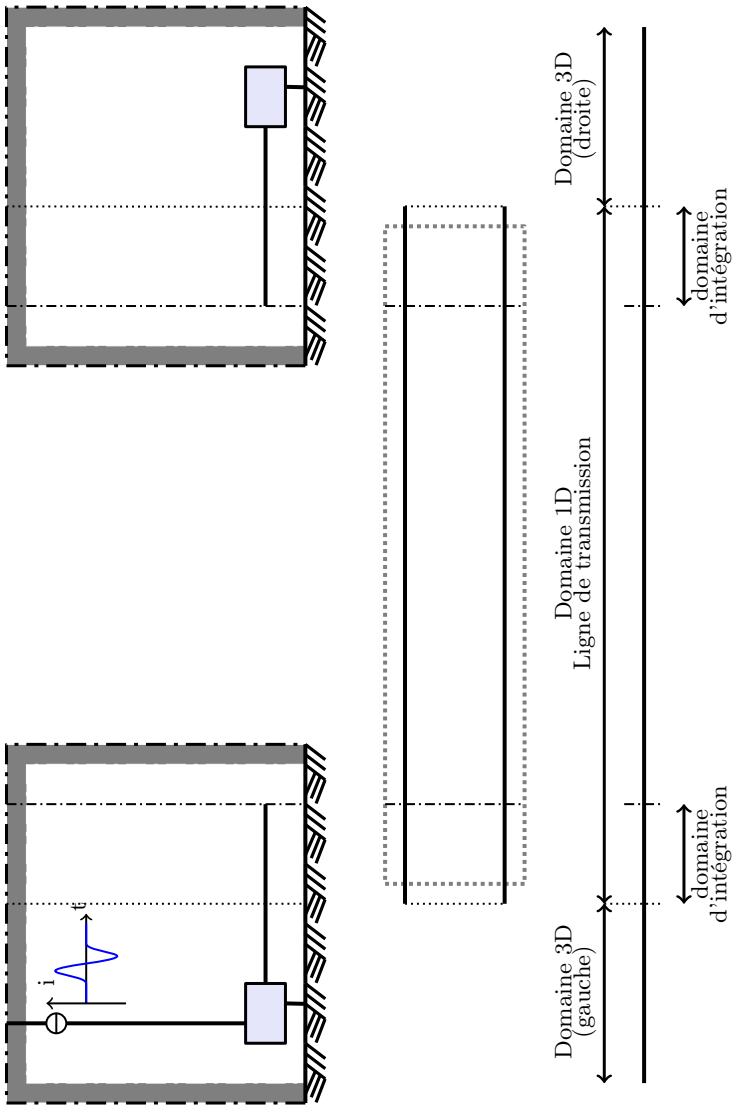


FIGURE 4.12 – Représentation des différentes zones de calcul pour une hybridation 3D-1D-3D.

4.4 Validation numérique et étude paramétrique

Dans cette section, nous allons présenter différents cas tests permettant de valider l'approche hybride 1D/3D que nous avons proposé dans le contexte multi-domaines. En particulier, nous présenterons les limites de son utilisation mais également ses avantages. Les exemples, étudiés dans cette section, sont relativement simples pour permettre de traiter chaque point délicat particulier à notre hybridation. Nous verrons entre autre les points suivants :

- l'influence sur la solution, du paramètre P qui caractérise la longueur de recouvrement du domaine 1D sur le domaine 3D ;
- la capacité de l'approche hybride à traiter des problèmes très basses fréquences, comme les problèmes foudre ;
- la capacité à prendre en compte des lignes de transmission au-dessus de sols de conductivité finie ;
- la pertinence à bien représenter les effets indirects de la foudre sur un réseau de lignes.

4.4.1 Influence de la longueur d'intégration

Le cas test relativement simple, est composé d'un fil de longueur totale $L = 200\text{m}$ de rayon $a = 0.1\text{m}$ et adapté sur ses deux extrémités. Ce fil est placé au-dessus d'un plan de masse ($h = 2\text{m}$) que nous supposons infini. Le système est agressé par une source de tension $v(t)$ dont l'évolution temporelle est une dérivée de gaussienne normalisée et placée à l'extrémité gauche du fil. La fonction $v(t)$ est donnée par :

$$v(t) = -A\alpha(2(\alpha(t-\tau))^2 - 1)e^{(\alpha(t-\tau))^2} \tag{4.8}$$

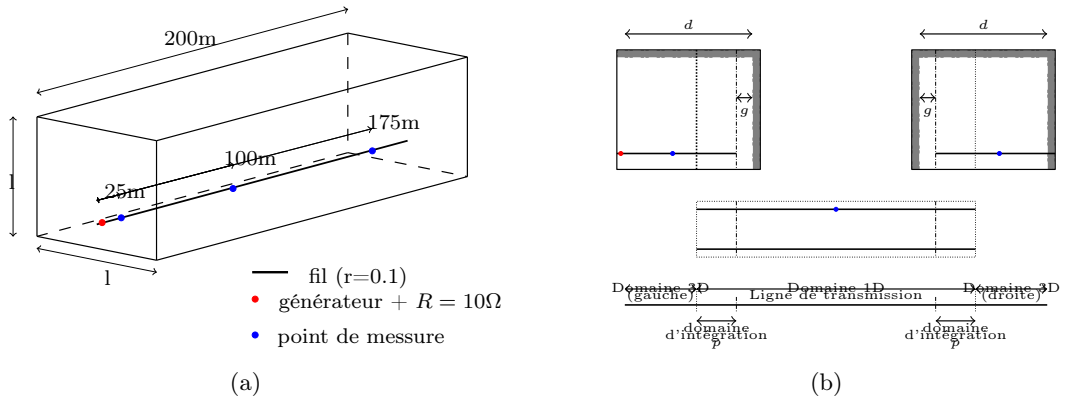
avec $A = 2e^{0.5}/(\alpha + \sqrt{2})$, $\alpha = 13.3\text{Ms}^{-1}$ et $\tau = 1.25\mu\text{s}$.

Ces paramètres permettent d'avoir une longueur d'onde minimale 100 fois plus grande que le pas de maillage $\Delta = 1\text{m}$. La figure 4.13(a) donne la géométrie utilisée dans le 3D. Sur cette géométrie, nous considérons le domaine de calcul autour du fil avec une longueur l que nous ajusterons au besoin. Nous utilisons des conditions absorbantes de type ML qui ne posent pas de problèmes dans les configurations où des fils traversent la couche absorbante, ce qui est le cas quand nous adaptons les fils à l'infini.

Nous utilisons la stratégie d'hybridation sur ce cas test en découpant le problème en trois domaines, un domaine 3D pour modéliser chaque extrémité et un modèle de ligne pour la partie centrale. La distance de recouvrement est définie par le paramètre p. La longueur du fil dans la zone 1D est ensuite ajustée de manière à avoir la bonne longueur de fil $L_{1d} = 100+2p$.

Le premier résultat (Fig. 4.14) nous donne une idée très claire de l'influence de la longueur de la zone de recouvrement sur la qualité de la solution hybride. Celle-ci s'exprime par la « transparence » du point d'hybridation, ou en d'autre terme, par le fait qu'au point d'hybridation, on doit avoir entre les solutions des deux domaines, un coefficient de transmission de un et un coefficient de réflexion nul.

Nous constatons sur les résultats que plus la longueur de la zone de recouvrement est importante, plus l'hybridation est transparente sur la solution. Pour $p = 5\text{m}$ le courant réfléchi (Fig. 4.14(a)) est relativement faible et imperceptible pour $p = 15\text{m}$, le calcul de



(a) (b)

FIGURE 4.13 – Description de la géométrie du cas test. (a) représente la géométrie 3D utilisée pour représenter un fil adapté au-dessus d'un plan de masse. (b) représente le découpage utilisé pour la stratégie d'hybridation 3D-1D-3D d'un fil adapté au-dessus d'un plan de masse.

l'erreur 4.14(c) ne fait que confirmer ce résultat. En ce qui concerne le courant transmis à travers le point d'hybridation 4.14(b), nous constatons un comportement similaire. Le calcul de l'erreur fait pourtant ressortir une erreur en transmission sur le premier front d'onde plus important même pour $p = 15$m. Cette erreur est liée à une petite différence de vitesse de phase entre le 3D et le 1D qui introduit un léger décalage temporel. De l'ensemble des expériences numériques menées sur ce point, on peut dire que la longueur de la zone de recouvrement nécessaire pour avoir une erreur satisfaisante, ne dépasse pas quelques longueur caractéristique de la ligne de transmission qui dans notre cas est la hauteur fil/sol h.

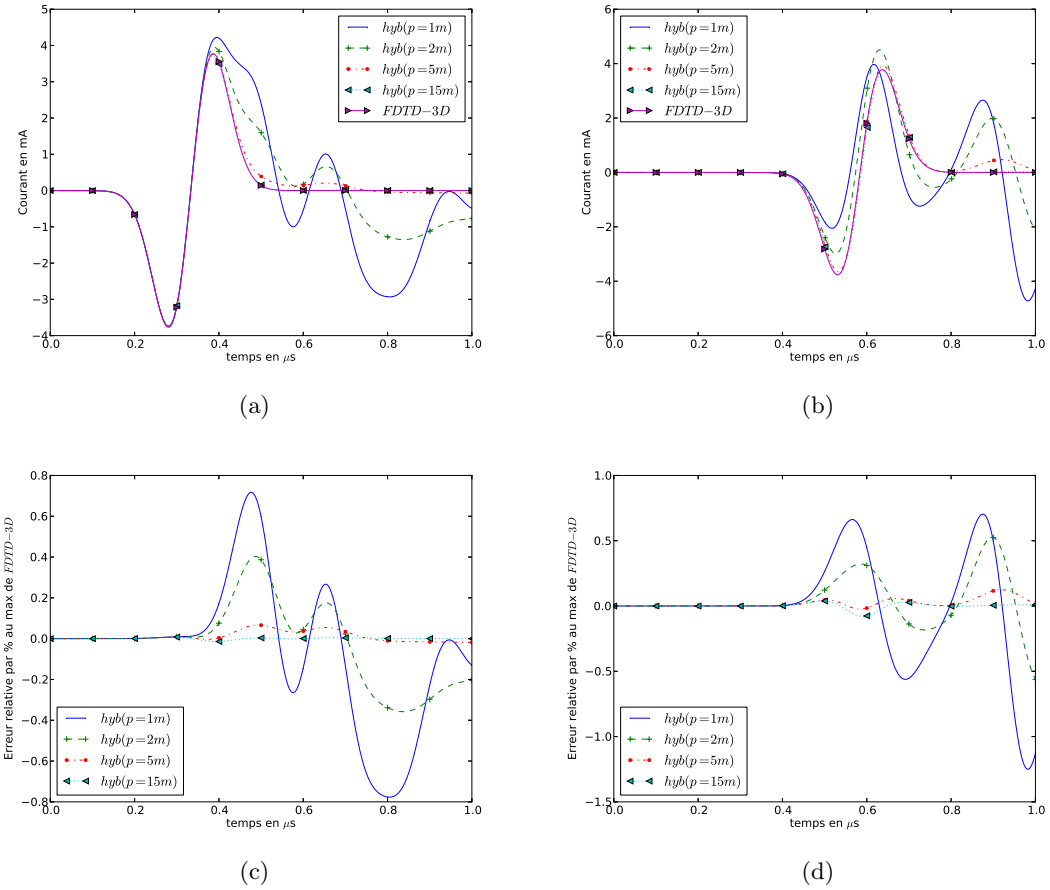
Le deuxième résultat important de ce cas test est l'évaluation des performances en temps de calcul mais aussi en mémoire. Nous pouvons noter sur ce point que l'utilisation de l'approche hybride permet un gain en mémoire significatif permettant d'adresser des problèmes plus grands avec un gain en temps qui reste intéressant 4.3.

	temps d'exécution	Mémoire [Mb]
FDTD-3D	40'913"	208.9
Hybrid FDTD/MTL	6'813"	9.855
MTL	0'741"	0.897

TABLE 4.3 – Temps de calcul et mémoire nécessaire pour traiter le cas test adapté 4.13 pour les FDTD-3D, hybride FDTD/MTL 3D-1D-3D et MTL

4.4.2 Limites fréquentielles de l'approche

Dans cet exemple, nous cherchons à évaluer la robustesse de l'approche pour un problème basse fréquence. Pour réaliser ces cas tests, nous allons utiliser la même géométrie, à ceci près, que nous mettons les deux extrémités du fil en court-circuit pour créer un système



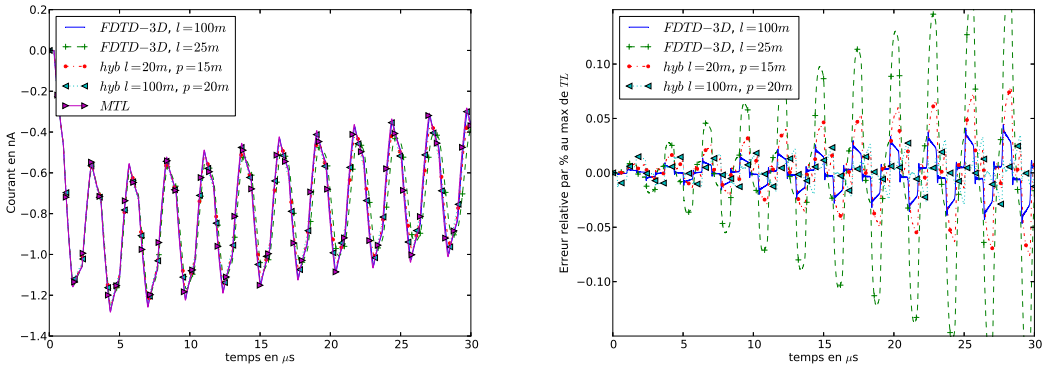
(a) (b)

(c) (d)

FIGURE 4.14 – Caractérisation de l'influence de la longueur de la zone de recouvrement sur la qualité de l'hybridation par comparaison des courants réfléchis et transmis par le point d'hybridation. Les figures (a) et (b) représentent le courant obtenu de part et d'autre du premier point d'hybridation ($l = 25$m et $l = 100$m) pour différentes longueurs de zone de recouvrement p. Les figures (c) et (d) représentent l'erreur commise par rapport au cas de référence 3D relativement à la valeur maximale $e = (I - I_r)/\max \|I_r\|$, où I est le courant calculé par notre méthode hybride et I_r, le courant de référence donné par un calcul FDTD 3D complet.

résonnant. De plus, nous utilisons un générateur en tension dont l'évolution temporelle est une bi-exponentielle $f(t) = \exp(-\alpha t) - \exp(-\beta t)$, avec $\alpha = 1.5 \cdot 10^4 \mathrm{s}^{-1}$ et $\beta = 0.95 \cdot 10^6 \mathrm{s}^{-1}$. L'utilisation d'un système résonnant est particulièrement intéressant à plusieurs égards. D'une part, cela permet de vérifier la nature bidirectionelle de l'hybridation et d'autre part, cela fait ressortir les défauts de l'hybridation par l'accumulation d'une erreur à chaque passage d'un front d'onde par le point d'hybridation (interface des 2 domaines).

La figure 4.15(a) donne le courant à 100m, nous retrouvons en valeur médiane la tendance de la biexponentielle avec la superposition des différents échos du front. Nous avons du mal à distinguer la différence entre les courants obtenus par le code de ligne (MTL), ceux obtenus en FDTD 3D avec une section transversale de $l = 100$m et par l'hybridation 3D-1D-3D également avec $l = 100$m et avec $p = 20$m. En revanche, si nous diminuons la taille de la section transversale l, les résultats se dégradent. En particulier pour le cas FDTD 3D avec $l = 25$m, nous constatons un défaut en amplitude et en vitesse de phase qui se traduit par un retard temporel. Le calcul de l'erreur $((f - f_r)/\max(\|f_r\|))$ nous donne une idée plus nette de l'erreur commise. En particulier pour une stratégie d'hybridation avec une section transversale insuffisante pour décrire la ligne de transmission, nous avons un résultat relativement correct ; ce qui est très intéressant car cela relâche la contrainte sur les parties 3D qui ne sont donc plus dimensionnées par rapport au problème de ligne de transmission mais par rapport à la modélisation des impédances terminales. Ce résultat est logique car, comme la ligne de transmission est correctement décrite sur la partie centrale, nous faisons une erreur que sur les domaines 3D, à l'inverse d'un calcul FDTD global.



(a) (b)

FIGURE 4.15 – Courants obtenus à 100m (a) par différentes approches (FDTD-3D, MTL, hybride FDTD/MTL 3D-1D-3D) et l'erreur (b) commise par rapport au cas TL pour traiter un problème de ligne de transmission en résonance pour une source de type biexponentiel.

En définitive, ce cas test nous a permis de montrer le caractère bidirectionnel de l'hybridation, mais également que nous pouvons utiliser celle-ci pour traiter les problèmes basses fréquences. Enfin, ce cas test montre que l'utilisation de la méthode la plus adaptée pour chaque sous-domaine assouplit les contraintes sur la taille du maillage nécessaire (paramètre

l), tout en conservant une bonne précision.

4.4.3 Hybridation et sol de conductivité finie

Le troisième cas test traite le même problème que celui vu précédemment (ligne de transmission en résonance excitée par une biexponentielle), dans le cas où le plan de masse est remplacé par un sol de conductivité finie $\sigma = 0.1\text{S/m}$. Nous utilisons la décomposition en une série de Debye présenté dans la (Sec. 2.4 page 71) pour approcher les paramètres de la ligne dans la méthode des équations de lignes (MTL). Nous faisons deux décompositions en série de Debye. Une avec deux termes et l'autre avec 5 termes dans la série.

Dans la géométrie 3D, nous avons considéré un sol de profondeur finie et nous prenons une section transversale de 100 mètres et un pas de maillage d'un mètre pour le domaine 3D. Les dimensions finies du sol et le pas de maillage ont une influence sur le domaine de validité en fréquence de notre modèle numérique. En particulier dans notre exemple, nous pouvons considérer le domaine de validité limité à une gamme de fréquence entre 201kHz et 503kHz.

Note : La dimension finie du sol et le pas de maillage ont une influence sur le domaine de validité, en particulier sur la gamme de fréquences qui sera correctement représentée dans le sol.

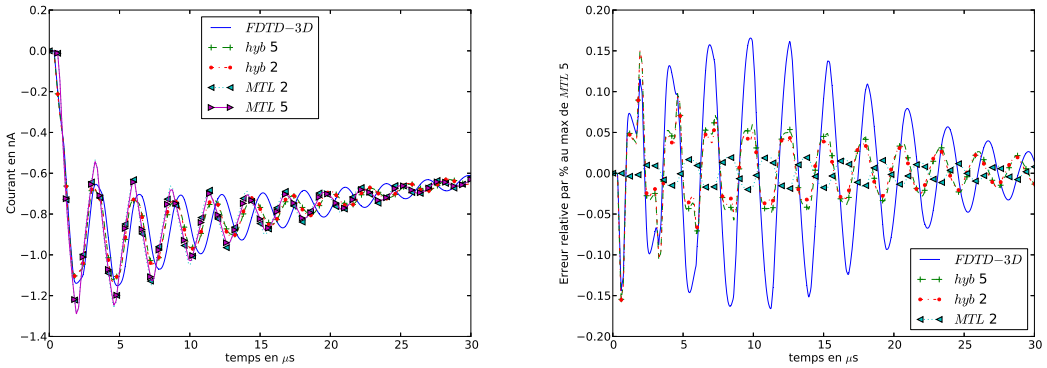
Cette limitation fréquentielle s'explique par l'épaisseur de peau. La nature finie de la conductivité implique que le courant s'établit à la « surface » du sol jusqu'à une certaine profondeur. De plus, cette profondeur dépend de la fréquence.

Ainsi, si nous voulons correctement représenter le comportement physique du sol nous devons correctement représenter cet effet de peau. En d'autres mots, il faut suffisamment discrétiser la plus petite épaisseur de peau (par exemple $dl \leq \delta_{min}/10$) et avoir suffisamment de place pour que la plus grande puisse se former (par exemple $P \geq 4\delta_{max}$ ce qui permet de représenter 89% du courant).

Si nous appliquons ces règles à un problème foudre que nous considérons dans la bande de fréquences $(1\text{kHz}, 1\text{MHz})$ et sachant que l'épaisseur de peau peut être approximée par $\delta = (\sigma\mu\pi f)^{-1/2}$, nous obtenons les valeurs suivantes : $P \geq 4\delta_{max} \approx 636.61\text{m}$ et $dl \leq \delta_{min}/10 \approx 0.503\text{m}$.

Notons également que la longueur transversale ne peut pas être choisie indépendamment de la fréquence. Si le courant peut s'infiltrer jusqu'à 600 mètres de profondeur, il y a de grande chance que le courant en surface du sol s'étende à une distance du même ordre de grandeur. Dans une première approximation, nous pouvons prendre comme dimension de la section transversale les paramètres suivants : une hauteur l au dessus du sol et une largeur P soit $\frac{(P+l)P}{dl^3}$ cellules par mètre de ligne de transmission. Ce qui ramené à notre cas vaut 736.9 millions de cellules. Pour donner un ordre de grandeur, cela correspond approximativement à un cube de 903 mailles de côté pour traiter 200 mètres de lignes de transmission...

Sur les résultats obtenus pour cet exemple, nous constatons que nous avons un défaut d'amplitude entre le modèle de ligne de transmission et le calcul FDTD complet ou notre approche hybride, qui n'est pas important. En revanche, nous avons un décalage temporel



(a) (b)

FIGURE 4.16 – Courants obtenus à 100m (a) par différentes approches (FDTD-3D, MTL, hybride) et l'erreur (b) commise par rapport au cas MTL pour traiter un problème d'un fil en résonance au-dessus d'un sol de conductivité $\sigma = 0.1$S/m pour une source de type biexponentiel.

entre le modèle de ligne de transmission et les autres méthodes qui s'accentue avec le temps et qui devient assez important. Cette erreur traduit une différence dans la vitesse de phase que nous retrouvons sur les courbes fréquentielles. Si nous considérons seulement la solution du calcul FDTD complet avec la solution de la méthode de ligne de transmission, nous remarquons une différence importante. Cette différence est beaucoup moins marquée entre la solution obtenue par notre approche hybride et celle de la méthode de ligne de transmission. Ceci s'explique par le fait que dans notre approche hybride nous limitons les erreurs de dispersion du schéma FDTD aux domaines 3D, alors que dans un calcul complet, ces erreurs sont sur la globalité de la scène. Nous pouvons donc conclure de cet exemple que notre approche hybride améliore la précision de la solution en limitant les erreurs de dispersion du schéma FDTD.

Le courant en fréquentiel (Fig. 4.17) donne une autre information intéressante qui est l'apparition d'un double lobe sur la solution pour l'approche hybride, alors que nous avons un seul lobe pour les autres approches. Cette différence est difficile à expliquer et peut avoir plusieurs causes. En effet, cela peut provenir d'une différence de vitesse de phase entre les courants calculés sur les deux domaines que nous cherchons à hybrider. Plus précisément, le courant calculé par le modèle de ligne de transmission et qui est utilisé comme terme source dans le 3D, se propage plus vite que le champ dans le domaine 3D. Nous pouvons aussi expliquer cette différence par le fait que le courant de retour de la ligne de transmission est obtenu par la réflexion du champ induit par le courant source. Or, dans le cas d'un sol de conductivité finie, le courant qui s'écoule dans le sol ne peut être entièrement décrit de cette manière.

En conclusion, malgré certaines remarques, nous pouvons dire que la stratégie d'hybridation se comporte relativement bien dans le cas où nous avons une ligne sur un sol de

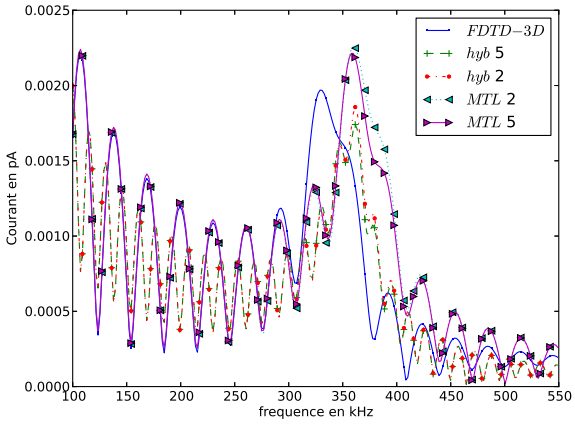


FIGURE 4.17 – Représentation fréquentielle du courant obtenu à 100m par différentes approches (FDTD-3D, MTL, hybride) pour traiter un problème d'un fil en résonance au-dessus d'un sol de conductivité $\sigma = 0.1$S/m pour une source de type biexponentiel.

conductivité finie.

4.4.4 Prise en compte d'une source rayonnante

Dans tous les cas test que nous venons de voir, nous avons utilisé une source placée sur la ligne de transmission (excitation d'un mode différentiel). Dans ce cas test, nous voulons étudier l'influence du rayonnement d'une source globale qui affecte tout le domaine (Ex. onde plane, antenne, canal de foudre). Cette étude doit nous permettre de définir quels termes de rayonnement nous pouvons négliger dans une hybridation de domaines.

Dans ce cas test, nous avons choisi d'utiliser une géométrie relativement simple 4.18. Cette géométrie décrit un fil en court circuit sur ces deux extrémités et est excitée par une antenne modélisée par un fil sur lequel nous avons placé une source. Nous avons également introduit deux plaques métalliques à chaque extrémité cherchant à représenter des structures rattachées aux extrémités du fil.

Les points bleus représentent des points de mesure et le point rouge, la source de courant dont la forme temporelle est une dérivée de gaussienne. Enfin, les deux traits discontinus représentent les interfaces entre les trois domaines utilisées dans le cas de l'hybridation. Sauf dans les cas où cela est explicité, la longueur d'intégration est de 15m pour un pas de maillage de 1m.

a. Influence du rayonnement

Dans le cas test étudié, nous comparons les solutions obtenues en utilisant une approche FDTD sur la globalité du domaine 3D ($FDTD-3D$) et une approche hybride 3D-1D-3D (hyb). Nous observons le courant en deux points de mesure, dont le premier est dans le premier domaine 3D du modèle hybride et le second dans le deuxième domaine 3D.

Dans le cas de l'hybridation 3D-1D-3D, nous négligeons deux contributions :

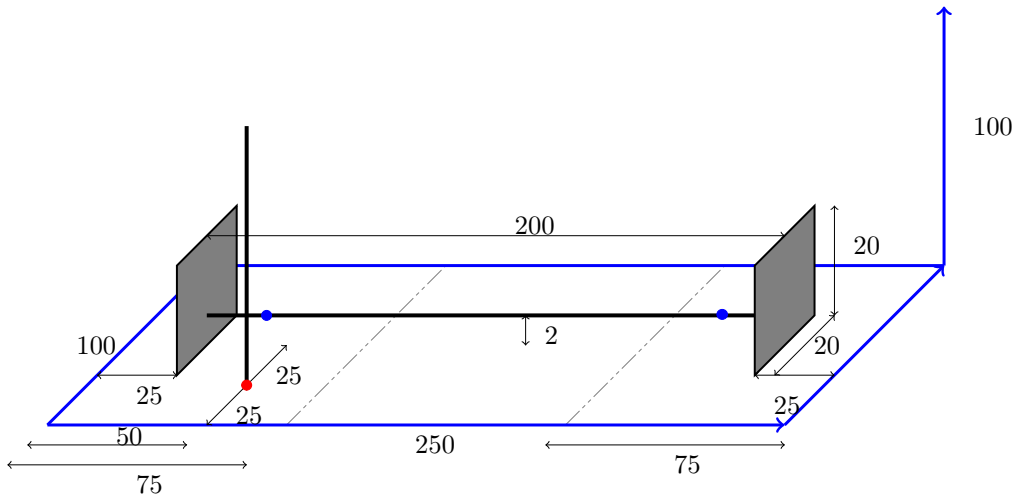


FIGURE 4.18 – Description de la géométrie du cas test avec une source rayonnante.

- la contribution liée au champ électromagnétique qui induit un terme source distribué le long de la ligne dans le modèle de ligne de transmission ;
- la contribution liée au champ électromagnétique qui induit un terme source rayonnée entre les différents domaines 3D.

En ce qui concerne le rayonnement du fil sur les domaines 3D, celui-ci est nul car nous somme dans l'hypothèse d'une ligne de transmission.

Avant d'analyser les résultats obtenus numériquement (Fig. 4.20), nous allons étudier le comportement global du système. La figure 4.19 nous donne une représentation symbolique de la position du front d'onde (flèche) à différents instants.

Le champ électrique rayonné par l'antenne va entrer en interaction avec la ligne de transmission. Comme le champ électromagnétique a une structure radiale, le couplage de ce champ avec la ligne va dépendre de la position sur la ligne. Plus nous sommes proche du point source 'c', plus le champ électrique va être orthogonal à la ligne et le couplage va être faible. En revanche, plus nous sommes loin du point source (Ex. point 'b'), plus le champ électrique va être parallèle à la ligne et le couplage va donc augmenter. Par ailleurs, l'amplitude décroît d'autant plus que nous s'éloignons du point source. En définitive, la contribution du terme source distribuée le long de la ligne risque d'être peu significatif. Enfin, le champ rayonnée par la source va entrer en interaction avec les structures terminales (plaques) et induire un autre terme source qui sera dominant par rapport aux autres.

Nous pouvons aussi visualiser le comportement global du courant sur la figure 4.19 qui nous donne la « position » du front d'onde à différents instants. Dans un premier temps, le courant est faible ($\approx [t_1 = 25/c, t_2 = 25\sqrt{2}/c]$). Puis, le champ entre en interaction avec la structure de gauche et génère une onde électrique qui va se propager de la gauche vers la droite. Cette onde arrive approximativement en 'a' au temps $t_3 \approx 25(1+\sqrt{2})/c = 0.2\mu s$ et en 'b' au temps $t_4 \approx (25\sqrt{2}+175)/c = 0.70\mu s$. Ensuite, le champ source atteint la deuxième structure et produit une onde électrique qui se propage de la droite vers la gauche. Cette onde arrive en 'b' au temps $t_5 = (\sqrt{25^2+175^2}+25)/c = 0.67\mu s$ et en 'a' au temps

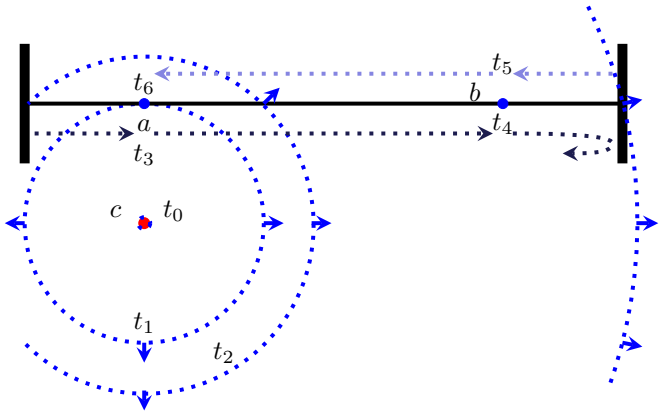


FIGURE 4.19 – Représentation symbolique du rayonnement de la source et des deux principales ondes se propageant sur la ligne de transmission

$t_6 = (\sqrt{25^2 + 175^2} + 175)/c = 1.17\mu s$. Enfin, ces deux ondes vont résonner entre les deux structures.

Si nous traitons le problème par simulation numérique, nous obtenons le courant en 'a' (Fig. 4.20(a)), et en 'b' (Fig. 4.20(b)). Nous retrouvons sur ces résultats les différentes phases décrites précédemment. Sur le courant en 'a', nous pouvons distinguer trois phases. La première phase, de 0 à 1µs, correspond au premier passage de l'onde (gauche vers droite). Il y a une petite erreur sur la solution hybride liée au fait de ne pas prendre en compte le terme source distribué dans la zone centrale (ligne de transmission). La deuxième phase, de 1µs à 1.5µs, correspond principalement au couplage de la source avec l'extrémité droite. Ce couplage n'est pas pris en compte dans l'approche hybride et nous remarquons effectivement une différence sur les résultats. La troisième phase, de 1.5µs à 1µs, correspond à la première réflexion de l'onde sur la plaque de droite. Cette réflexion est bien prise en compte dans l'approche hybride, comme le montre les résultats.

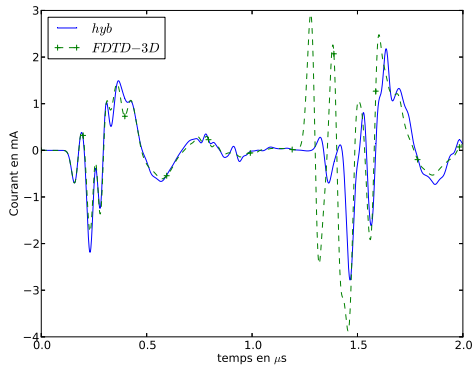
En conclusion nous notons que le fait de ne pas prendre en compte le rayonnement de la source dans le deuxième domaine de calcul introduit une erreur non négligeable liée à la tension de fermeture du modèle d'Agrawal.

Dans une deuxième configuration nous avons simplement surélevé les deux plaques, de manière à ce que la ligne soit en circuit ouvert, dans ce cas de figure la tension de fermeture est nulle. Nous ne faisons donc pas d'erreur sur cette tension en ne prenant pas en compte le champ dans le deuxième domaine 3D. Les résultats obtenus 4.21 par l'approche hybride et le calcul 3D complet montrent une relative bonne concordance.

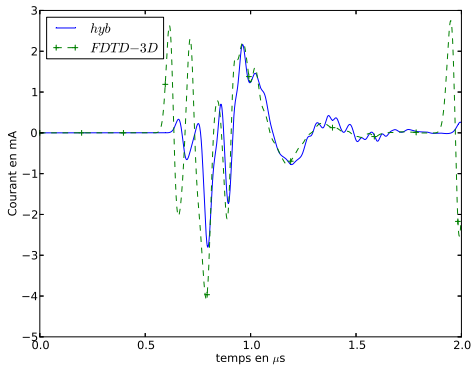
Ces résultats montre l'importance à tenir compte du rayonnement de la source dans tout le domaine, en particulier sur les structures connectées à la ligne de transmission pour correctement prendre en compte la tension de fermeture.

b. Conclusion

Dans cette section, nous avons cherché à valider numériquement la stratégie d'hybridation par découpage de domaines proposée. Nous avons tout d'abord étudié l'influence de la

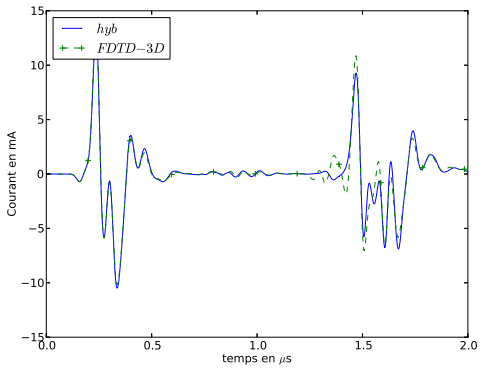


(a)

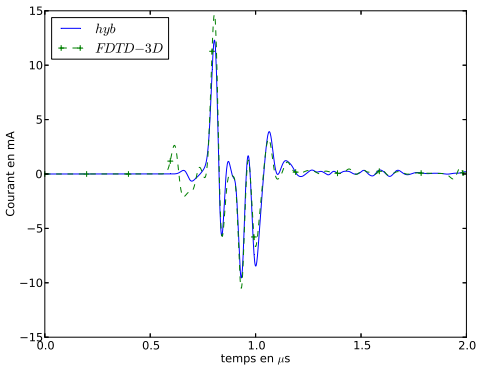


(b)

FIGURE 4.20 – Représentation du courant à gauche (a) de la ligne et à droite de la ligne (b) pour le cas 3D et hybride 3-1-3.



(a)



(b)

FIGURE 4.21 – Représentation du courant à gauche (a) de la ligne et à droite de la ligne (b) pour le cas 3D et hybride 3-1-3. Dans le cas où la ligne est en circuit ouvert à ces deux extrémités

longueur de recouvrement sur la qualité de la solution. L'expérience montre qu'une longueur de l'ordre de la hauteur de la ligne est suffisante pour avoir une bonne solution. Ce qui est tout à fait raisonnable par rapport à la longueur de la ligne.

Nous avons également étudié le comportement de l'hybridation en présence d'un sol de conductivité finie. Cette configuration est importante car récurrente dans les configurations réalistes. Cette étude nous a permis de voir que la stratégie d'hybridation se comporte très bien dans ce cas. De plus, les simulations effectuées ont montré que les modèles de sols dans le 3D et le 1D devaient être le plus proche possible pour avoir une solution hybride correcte. Ceci est difficile à réaliser car nous n'avons pas le même type de modèle numérique de sol en 1D et en 3D.

La dernière étude est consacrée à l'analyse et à l'influence du rayonnement d'une source globale. Le résultat montre très clairement que le champ incident rayonné par une source globale doit être considéré dans tous les domaines.

4.5 Conclusion

Dans ce chapitre nous avons présenté une stratégie d'hybridation de domaines 1D/3D mettant en œuvre la résolution d'une équation de ligne de transmission et la résolution des équations de Maxwell en 3D.

Plus précisément, l'hybridation de domaine consiste à découper géométriquement un problème pour utiliser la méthode la plus adaptée pour chaque sous domaine. Cette approche peut être motivée par plusieurs points ; d'une part un objectif de précision. D'autre part, un objectif de coût calculs.

Les résultats numériques ont montré le gain en temps et en mémoire de l'approche hybride proposée. Cependant elle reste délicate à mettre en œuvre dans le cas où nous avons une source globale qui illumine toute la scène. Dans ce cas de figure l'étude d'une stratégie d'hybridation mixte pourrait être envisagée pour passer outre cette difficulté. Cette approche consisterait à utiliser une stratégie d'hybridation en domaines disjoints pour calculer le rayonnement d'un sous domaine sur un autre et une hybridation de domaine pour prendre en compte les modes conduits.

Chapitre 5

Mise en œuvre de la stratégie d'hybridation sur une maquette réaliste d'un site de lancement

Sommaire

5.1 Introduction 154
5.2 Présentation du cas étudié – Positionnement par rapport au problème initial 155
5.2.1 Positionnement par rapport au problème initial 155
5.2.2 La maquette numérique 157
5.3 Études numériques 160
5.3.1 Source haute fréquence de type onde plane avec un sol parfaitement conducteur 160
5.3.2 Source basse fréquence de type onde plane avec un sol parfaitment conducteur 166
5.3.3 Source basse fréquence avec une injection à la base d'un mât avec un sol réaliste 167
5.3.4 Comportement d'un réseau multifilaire avec une injection de courant à la base d'un mât 170
5.4 Conclusion 173

5.1 Introduction

Dans le premier chapitre (Sec. 1.2.1 page 20), nous avons présenté l'intérêt des études numériques pour concevoir en amont les grands systèmes. Ces études numériques sont très utiles pour analyser des grands sites comme les bases de lancement spatial, où l'approche normative n'est pas prévue ni adaptée pour ce type de configuration. Cependant, comme nous l'avions annoncé dans notre premier chapitre, les outils de modélisation numérique actuels ne permettent pas encore de traiter tous les aspects de ce type de problème. Nous avons donc proposé deux stratégies d'hybridation (Sec. 3 page 99) et (Sec. 4 page 125) pour parfaire la modélisation et ainsi pouvoir apporter une solution à ce type de problème.

Dans ce chapitre, nous allons mettre en œuvre ces deux stratégies d'hybridation sur une maquette simplifiée d'une base de lancement spatial, inspirée du site Soyouz qui a déjà fait l'objet d'études à l'ONERA. Le premier objectif est de montrer ce que les stratégies d'hybridation proposées apportent de plus à la modélisation de grands sites terrestres, pour mieux rendre compte de la phénoménologie. Le deuxième objectif est de montrer les points qui pourraient être améliorés pour tenir compte plus précisément de la complexité d'infrastructures de grande taille. Ces points encore limitant aujourd'hui, peuvent être liés aux stratégies d'hybridation elles-mêmes, mais peuvent également être associés à d'autres limitations numériques plus générales.

Nous décrivons dans un premier temps la maquette numérique que nous avons utilisé pour faire nos simulations, puis, nous présentons différentes études numériques menées sur cette maquette. Enfin, nous terminons le chapitre en effectuant un bilan sur les capacités de notre approche hybride à étudier un grand site terrestre.

5.2 Présentation du cas étudié – Positionnement par rapport au problème initial

5.2.1 Positionnement par rapport au problème initial

Nous avons présenté dans le premier chapitre (Sec. 1.2.1 page 20) la problématique des grands sites terrestres. Dans ce chapitre, nous allons étudier une maquette « réaliste » du site de lancement spatial Soyouz (Fig. 5.1).



FIGURE 5.1 – Photo de l'ensemble Soyouz en Guyane, proposé par ENJOY SPACE. [8]

Ce choix a été motivé par les études menées récemment par l'ONERA/DEMR pour le compte du CNES et de Vitrociset (pour l'ESA) [1, 2], sur le site de Soyouz au Centre Spatial Guyanais, à Kourou. Ces différentes études nous ont permis d'avoir un recul suffisant pour définir une maquette convenable du site pour faire des simulations numériques.

Durant ces études, l'objectif était d'aider la définition des systèmes de protection du lanceur à un impact direct de la foudre sur le pas de tir (position la plus vulnérable). L'utilisation de mât permet de limiter le risque d'un impact direct sur le lanceur Soyouz. Cependant, pour que la protection soit complète, nous devons également vérifier que le champ électromagnétique dans le lanceur soit inférieur à un certain seuil et limiter les différences de potentiel. Ceci s'effectue à l'aide d'une simulation numérique dans laquelle nous ne considérons généralement pas les bâtiments aux alentours du lanceur, car, au vu de leur distance par rapport à celui-ci ($\approx$ 500m), nous supposons leurs influence négligeable. Cependant ces bâtiments sont interconnectés par un réseau relativement complexe, de différents types de conducteur (énergie, fluide, rail, etc).

Dans les études précédentes, le rôle de ce réseau a donc été négligé. Cependant, il peut transporter de l'énergie à une grande distance et il peut aussi entrer en résonance, maintenant ainsi un niveau énergétique élevé aux fréquences propres du réseau. Nous allons donc étudier la quantité de courant qui s'écoule dans le réseau de connexion. Ceci nous informera sur le dimensionnement utile pour le système de protection et sur le risque d'une défaillance dans la globalité du système.

Nous avons vu précédemment (Sec. 4.4 page 141) que le champ de rayonnement a une contribution importante dans notre approche hybride, et en particulier, qu'il était crucial de correctement prendre en compte le champ dans tous les domaines 3D, de manière à cor-

rectement modéliser la tension de fermeture (Sec. 2.2.3 page 46). Une solution envisageable pour garantir cela, consisterait à modéliser la source de manière globale, c'est-à-dire dans tous les domaines. Pour une application foudre, nous pourrions utiliser les courants calculés le long du canal de foudre par le code 3D et faire rayonner ceux-ci [54], afin d'évaluer les champs électrique et magnétique sur une surface source [55] dans chaque domaine de calcul. Dans cette approche, nous négligeons le rayonnement des bâtiments, qui peut se justifier en considérant leur dimension petite par rapport à la longueur d'onde.

Cette approche se complique dans le cas d'un impact direct, car dans ce cas de figure, la structure foudroyée fait partie intégrante du canal de foudre et, à ce titre, rayonne. Il est alors plus délicat de trouver une formule analytique pour ce rayonnement. Nous pourrions, pour pallier cette difficulté, utiliser une surface où nous relevons les courants électriques et magnétiques [56] dans la zone foudroyée pour calculer le rayonnement du système. Le calcul de la diffraction d'un objet sur un autre peut alors être pris en compte par des surfaces de Huygens ou de rayonnement [56]. Toutefois, cette approche est encore difficile à adapter en présence d'un sol.

La figure (Fig. 5.2) illustre les différentes interactions dans le cas où l'on découpe le problème initial en deux sous-domaines 3D connectés par une ligne de transmission.

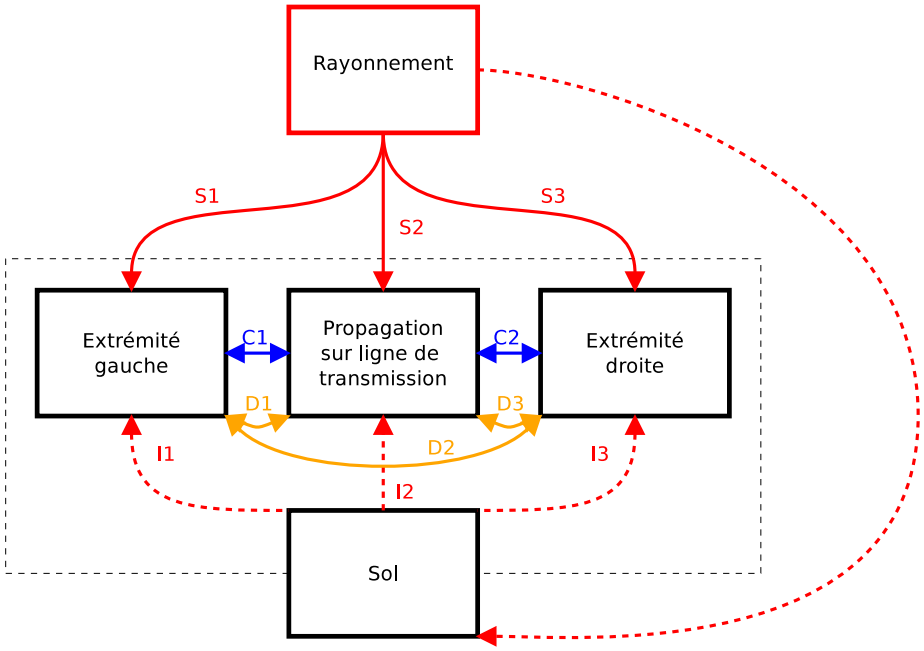


FIGURE 5.2 – Interaction entre les différents sous-domaines de calcul.

Nous distinguons plusieurs types d'interaction entre les différents sous-domaines. La première interaction est le terme source S global à tous les sous-domaines. À cette source est associée une source image I plus ou moins complexe suivant le type de sol. Dans le cas 3D, le terme image est obtenu par la réflexion de la source S sur le sol. La difficulté est alors d'injecter correctement le terme source dans chaque sous-domaine. Pour cela, nous pourrions utiliser par exemple un terme de Fresnel [57]. Pour la ligne de transmission, la

difficulté est de correctement représenter l'influence du sol sur les paramètres de la ligne et sur la source image. Nous avons déjà traité (Sec. 2.4 page 71) l'influence du sol sur les paramètres de la ligne. Par ailleurs, pour prendre en compte l'influence du sol sur la source image, il existe plusieurs modèles en basse fréquence qui permettent de calculer celui-ci. Nous pouvons citer par exemple, la forme de Cooray [58] et de Rubinstein [59] qui donne une bonne approximation de ce terme source, en un temps raisonnable. Enfin, nous distinguons deux types de couplage supplémentaires entre les différents sous-domaines qui sont la conduction C à travers la ligne de transmission (modélisé par le couplage du mode TEM) et la diffraction D entre les différents sous-domaines.

Dans les travaux menés jusqu'alors, la modélisation est faite par un fil vertical sur lequel on impose un générateur. Par rapport aux résultats que l'on a montré dans le chapitre précédent pour valider notre approche hybride, nous ne sommes pas en mesure de valider complètement notre approche sur ce type de configuration. Cependant, nous avons tout de même souhaiter étudier le comportement et évaluer l'influence des différentes interactions avec un autre type de source. Nous avons choisi tout d'abord, d'utiliser une onde plane avec différentes incidences et polarisations. Ce type de configuration peut simuler l'agression du site de lancement par un champ rayonné par une antenne ou par une IEMN. La figure 5.3 donne la convention utilisée pour définir l'onde plane. Le vecteur d'onde est défini par $\boldsymbol{k} = -\boldsymbol{u_\rho}$ et la polarisation est suivant $\boldsymbol{u_\theta}$ ou $\boldsymbol{u_\varphi}$.

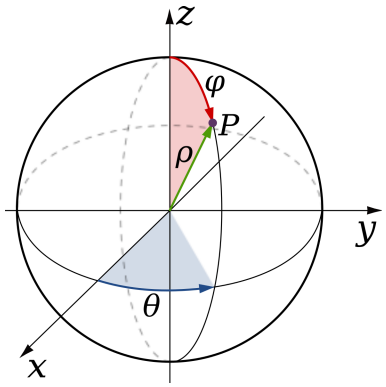


FIGURE 5.3 – Convention de représentation d'une onde plane en colatitude et longitude

5.2.2 La maquette numérique

La base Soyouz est composée principalement de 5 bâtiments (Fig. 5.1).
- la zone de lancement ;
- le bâtiment d'assemblage ;
- le centre de contrôle de lancement et de stockage ;
- la zone de stockage de fluide « loxlin » ;
- et une zone technique « kérosène ».

Nous nous sommes intéressés aux 3 bâtiments, les plus proches de la zone de lancement qui sont : le bâtiment d'assemblage, le bâtiment loxlin et la zone technique kérosène (Fig. 5.5).

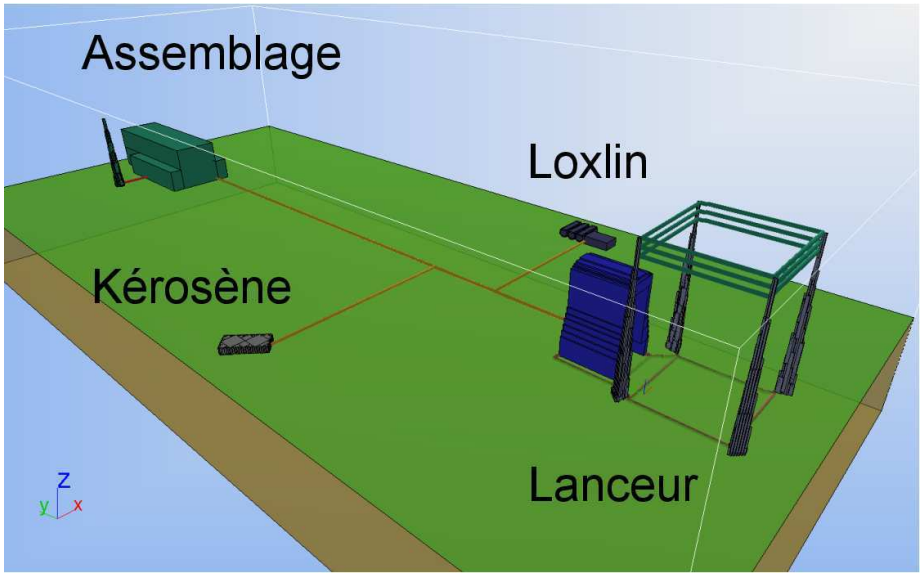


FIGURE 5.4 – Vue d'ensemble de la maquette numérique

A partir de ces bâtiments, nous avons défini une maquette qui nous semble représenter au mieux, la complexité de la base. Dans cette maquette, nous avons largement simplifié la zone avant du pas de tir, en particulier nous n'avons pas représenté le carnau, dont le but est l'évacuation des gaz d'échappement durant le décollage. Cette zone représente une partie du site importante dans la définition de la protection foudre. En effet, nous y trouvons des mâts qui sont actuellement les principaux éléments pour évacuer le courant lors d'un foudroiement dans la zone lanceur. Ces mâts ont des longueurs différentes et reposent sur des sols de nature électrique différents (terre argileuse et remblais) qui rend l'équilibrage du courant de fuite dans les mâts délicat. Cependant, malgré l'importance de cette zone, celle-ci n'est pas dimensionnée dans le comportement électrique du réseau. C'est pourquoi nous n'avons pas modélisé cet élément dans nos simulations.

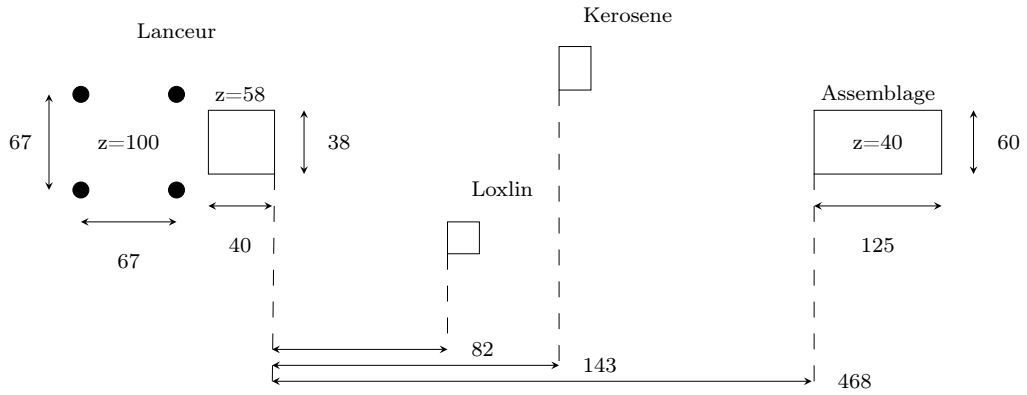
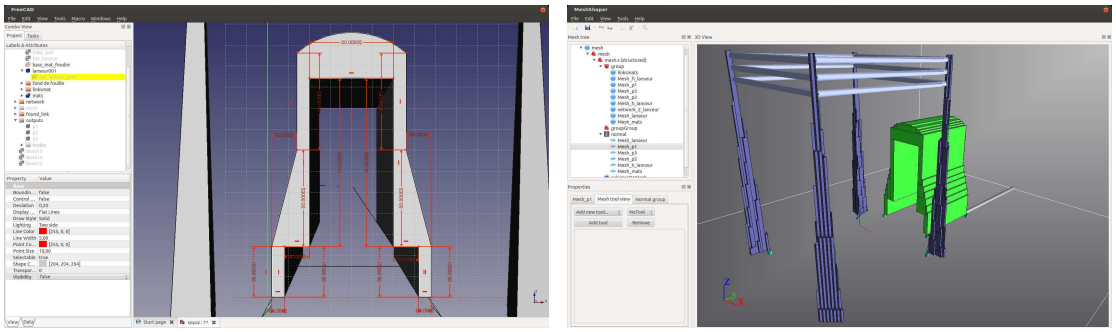


FIGURE 5.5 – Cotation de la maquette numérique

Pour nos simulations, dans un premier temps nous avons considéré un sol parfaitement conducteur. Dans cette configuration, nous ne pouvons pas introduire le réseau de terre entre les mâts. Dans une deuxième étape où nous prendrons en compte un sol de conductivité finie, nous introduirons un réseau de terre qui relie chaque mât et la coiffe.

La première étape de conception de cette maquette numérique a été de construire le modèle 3D ou modelage à partir de logiciel FreeCAD [60], qui est un logiciel de modélisation 3D gratuit. Ce logiciel initialement orienté pour le génie mécanique vise aujourd'hui d'autres disciplines comme l'architecture tout en restant simple d'utilisation.



(a) FreeCAD - le modèle CAO (b) MeshShaper - Le maillage FDTD

FIGURE 5.6 – Représentation de la maquette à différentes étapes du processus de modélisation. La première étape de modelage (a) est celle dans laquelle nous définissons les différents éléments (ici la coiffe du lanceur). La deuxième étape est la construction du maillage pour les FDTD-3D (b).

Ensuite, toujours à l'aide de FreeCAD, à partir d'un modelage, nous avons généré un maillage surfacique triangulaire de la structure à partir d'un critère géométrique (type STL). Ce type de maillage n'a pas de très bonnes propriétés pour être utilisé dans une méthode numérique. Cependant, il est suffisant pour être prétraité par le mailleur FDTD Nash [1] qui va le convertir en un maillage structuré pour la méthode FDTD-3D. En particulier, cet outil est capable de construire des surfaces et des volumes cartésiens à partir d'un ensemble de surfaces quelconques. Ces surfaces peuvent définir la géométrie de la scène considérée mais également des éléments plus « numériques » liés aux outils de modélisation comme des surfaces sur lesquelles nous nous appuyons pour faire rayonner le champ (surfaces de Huygens). Pour compléter cet outil, nous y avons ajouté la capacité de définir un volume à partir d'un demi espace (surface coupant le domaine de calcul). Cette capacité nous a permis de construire très simplement des zones volumiques de diélectriques homogènes à partir de l'interface air/sol par exemple.

1. Nash est un outil développé par AxesSim permettant la génération de maillage structuré pour des outils de simulation basés sur la méthode des différences finies, à partir de maillage non structuré de type triangulaire.

5.3 Études numériques

5.3.1 Source haute fréquence de type onde plane avec un sol parfaitement conducteur

Dans cette première étude numérique, nous avons considéré uniquement deux bâtiments définis par la zone de lancement et le bâtiment d'assemblage. Ces deux bâtiments sont connectés par une ligne de transmission dont les conducteurs sont de rayon $r = 10\text{cm}$ et placée au-dessus d'un plan de masse à une hauteur $h = 2\text{m}$. Ce système est agressé par une onde plane de type dérivée de gaussienne normalisée :

$$E(t) = -A\alpha(2(\alpha(t-\tau))^2 - 1)e^{(\alpha(t-\tau))^2} \tag{5.1}$$

avec $A = 2e^{0.5}/(\alpha + \sqrt{2})$, $\alpha = 13.3\text{Ms}^{-1}$ et $\tau = 1.25\mu\text{s}$. Ces paramètres permettent d'avoir une longueur d'onde principale 100 fois plus grande que le pas de maillage $\Delta = 1\text{m}$. La figure 5.7 représente la forme d'onde ainsi que sa transformée de Fourier.

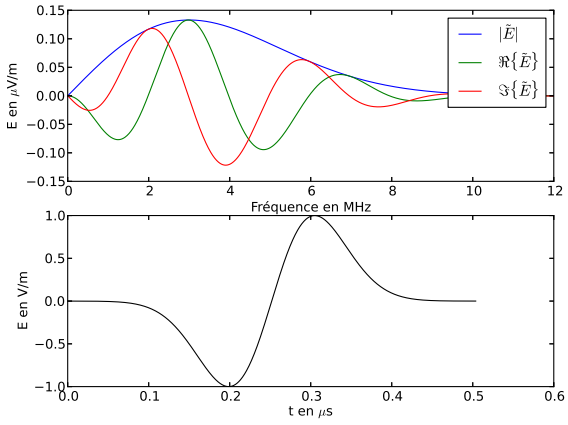


FIGURE 5.7 – Représentation de la forme d'onde de type dérivée de gaussienne et de sa transformée de Fourier

Nous avons calculé le courant de part et d'autre de la ligne de transmission qui relie les deux bâtiments. Dans l'approche hybride, nous définissons trois zones (Fig. 5.8), dont deux zones 3D de taille identique autour des deux bâtiments et une troisième entre les bâtiments modélisée par une ligne de transmission.

La figure 5.9 nous donne le courant sur la ligne reliant les deux bâtiments, dans le cas où nous avons une onde plane se propageant verticalement et polarisée de manière à ce que le champ électrique soit orthogonal à la ligne ($\theta = 0$, $\varphi = 0$, $\boldsymbol{E} = E\boldsymbol{u_x}$). Dans cette configuration, le champ électrique ne se couple pas avec la ligne de transmission. Ainsi la comparaison entre la méthode 3D et hybride représente uniquement l'erreur liée au fait de ne pas avoir pris en compte le rayonnement des bâtiments sur la ligne et entre les bâtiments. N'oublions pas que l'erreur peut également être introduite par l'hybridation elle-même. La figure 5.9(a) donne le courant à 10m du lanceur et la figure 5.9(c) donne l'erreur par rapport

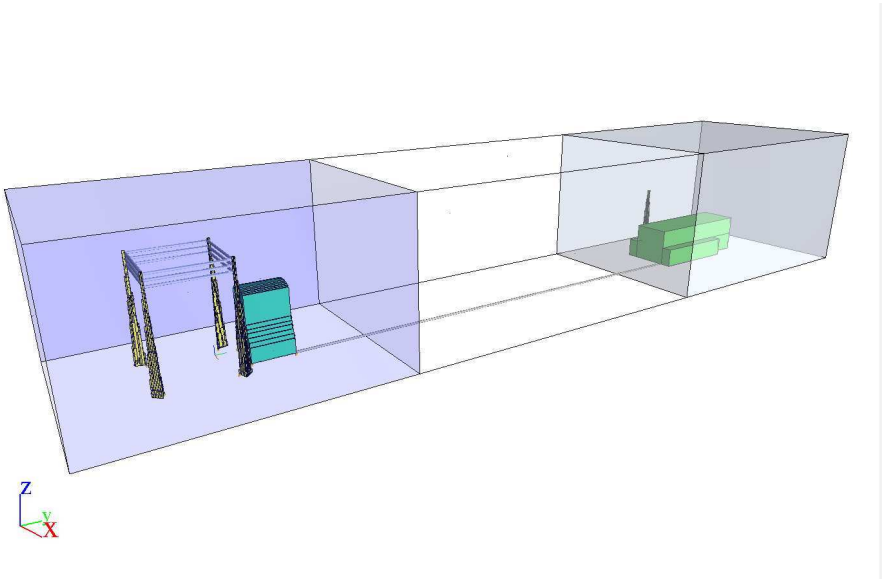


FIGURE 5.8 – Représentation des différentes zones de l'hybridation 3D-1D-3D de l'ensemble de lancement Soyouz.

au courant obtenu par un calcul entièrement 3D ($(I - I_{3D})/\max I_{3D}$). Nous avons une très bonne concordance (moins de 4% d'erreur) entre les courants obtenus par le calcul 3D complet et l'approche hybride, et cela, malgré le fait de ne pas avoir tenu compte de la diffraction des bâtiments.

La figure (Fig. 5.9(b)) représente le courant à 10m du bâtiment d'assemblage. Nous avons également une très bonne concordance entre les deux courants obtenus par le calcul FDTD complet et l'approche hybride. Naturellement, dans le cas où nous considérons uniquement la zone proche du lanceur dans la simulation, nous n'avons pas accès à ce courant.

La dernière figure représente le courant à 10m du lanceur dans le domaine fréquentiel. Nous retrouvons un pic à 2.25MHz, qui correspond à une résonance en $\lambda/2$ d'une longueur de 66.7m. Cette fréquence est probablement liée aux fils tendus aux sommets des mâts, la longueur de ces lignes étant de 67m.

La figure 5.10 nous donne le courant dans la ligne reliant les deux bâtiments, dans le cas où nous avons une onde plane se propageant verticalement et polarisée de manière à ce que le champ électrique soit parallèle à la ligne ($\theta = 0$, $\varphi = 0$, $\boldsymbol{E} = E\boldsymbol{u_y}$). Dans cette configuration, le champ électrique se couple « entièrement » avec la ligne de transmission.

Dans ce cas, pour évaluer la contribution du champ électrique le long de la ligne, nous avons considéré deux configurations d'hybridation. La première configuration (courbe hybridation), tient compte de la contribution du champ incident le long de la ligne. Dans la deuxième configuration (courbe hybridation approchée), nous avons négligé ce terme source. La figure 5.10(a) donne le courant à 10m du lanceur pour ces deux configurations. Nous constatons globalement (Fig. 5.10(c)) que l'erreur sur les solutions obtenues par un calcul sur le domaine global et l'approche hybride est plus importante dans le cas où nous négligeons le champ le long de la ligne. Nous pouvons obtenir alors une différence de l'ordre de

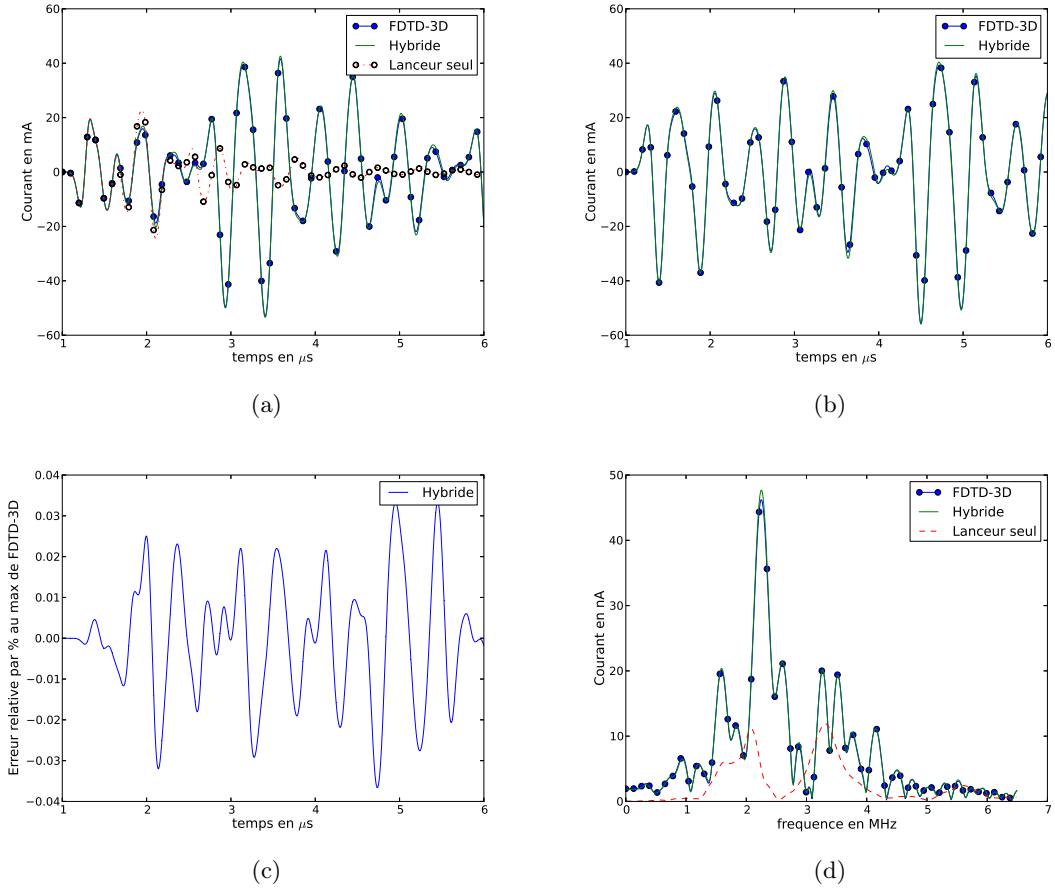


FIGURE 5.9 – Courant sur la ligne reliant les deux bâtiments à 10m du lanceur (a) et à 10m du bâtiment d'assemblage de la fusée (b) pour une agression par une onde plane de type dérivée de gaussienne d'incidence ($\theta = 0°$, $\varphi = 0°$, $\boldsymbol{E} = E\boldsymbol{u_x}$). Spectre du courant à 10m du lanceur (d) et erreur relative (c) par rapport au résultat 3D.

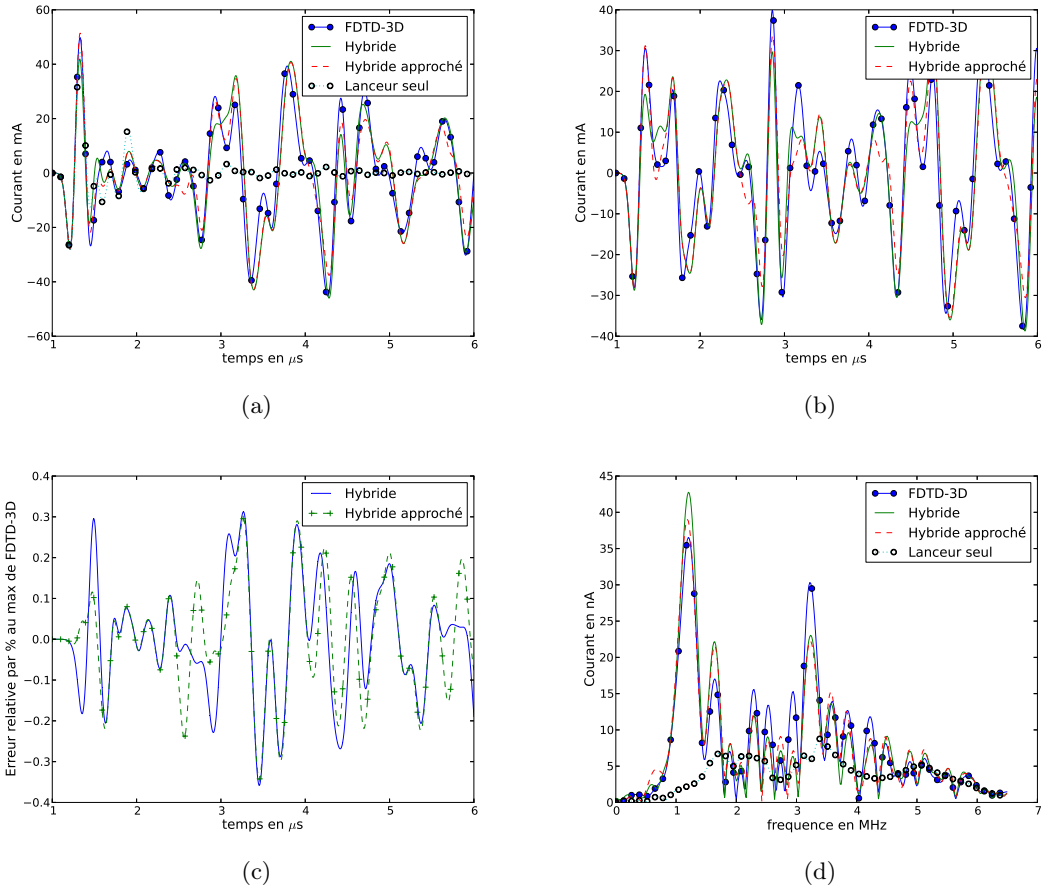


FIGURE 5.10 – Courant sur la ligne reliant les deux bâtiments à 10m du lanceur (a) et à 10m du bâtiment d'assemblage de la fusée (b) pour une agression par une onde plane de type dérivée de gaussienne d'incidence ($\theta = 0°$, $\varphi = 0°$, $\boldsymbol{E} = E\boldsymbol{u_y}$). Ainsi que le spectre du courant à 10m du lanceur (d) et l'erreur relative (c) par rapport au résultat 3D.

20% entre les deux solutions. Dans l'autre configuration, les deux solutions ont une meilleure concordance et l'erreur commise est de l'ordre de 5 à 10%. Contrairement à l'exemple précédent où l'onde plane ne se couplait pas directement sur la ligne de transmission, nous voyons très nettement dans ce cas, l'importance de la prise en compte du terme source lié au champ, dans le modèle de ligne de transmission.

La représentation fréquentielle (Fig. 5.10(d)) du courant à 10m du lanceur fait ressortir, sur cet exemple, une fréquence particulière aux alentours des 1.3MHz. Cette fréquence correspond à la fréquence de résonance en $\lambda/2$ d'une longueur de 115m qui est entre la hauteur des mâts (100m) et la longueur du bâtiment d'assemblage (125m). La fréquence à 3.1MHz donne une longueur de 48m, qui est proche de la dimension des bâtiments.

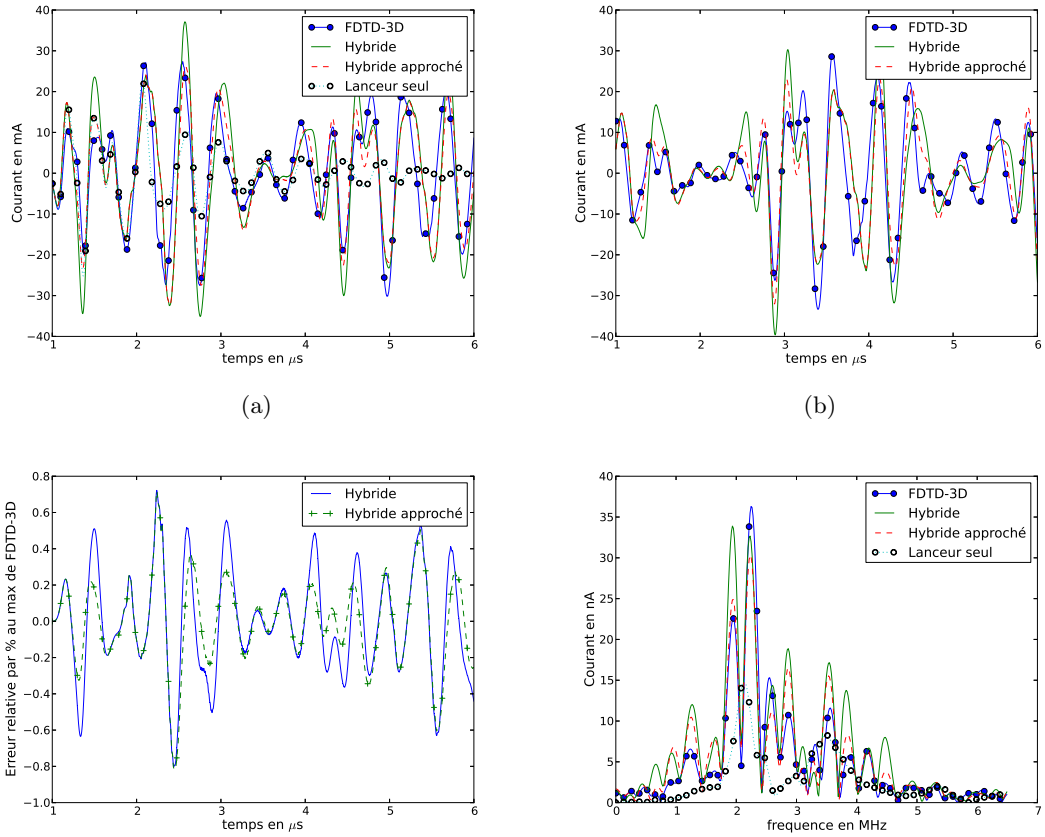


FIGURE 5.11 – Courant sur la ligne reliant les deux bâtiments à 10m du lanceur (a) et à 10m du bâtiment d'assemblage de la fusée (b) pour une agression par une onde plane de type dérivée de gaussienne d'incidence ($\theta = 30°$, $\varphi = 60°$, $\boldsymbol{E} = E\boldsymbol{u_\varphi}$). Spectre du courant à 10m du lanceur (d) et erreur relative (c) par rapport au résultat 3D.

Le dernier cas de figure 5.11 nous donne le courant dans le cas où nous avons pris une

onde plane d'incidence oblique ($\theta = 30°$, $\varphi = 60°$, $\boldsymbol{E} = E\boldsymbol{u}_{\boldsymbol{\varphi}}$). Nous pouvons faire sensiblement les mêmes conclusions et remarques que dans le cas précédent. L'erreur sur le courant (Fig. 5.11(d)) est sensiblement identique. De plus, nous retrouvons dans la représentation fréquentielle du courant à 10m du lanceur les tendances des deux premiers résultats : deux fréquences dominantes aux alentours à 1.9MHz (79m) et 2.2MHz (68m). Dans la gamme de fréquences de 3 à 6MHz, nous retrouvons un comportement similaire aux deux premiers cas. En revanche dans ce cas, la fréquence de résonance à 1.3MHz est moins marquée.

Au delà des résultats numériques, l'étude des performances informatiques en terme de calcul (temps et mémoire) est important. Le tableau 5.1 donne le temps et la mémoire consommés par les deux approches (3D complet et hybride 3D-1D-3D). Nous constatons un gain notable en temps et en mémoire. Nous pouvons calculer le gain théorique en mémoire et en temps en utilisant l'élément unitaire (cellule de Yee) comme unité de référence de mémoire et de temps. Le gain en mémoire s'obtient en faisant le rapport du nombre d'élément total dans les deux approches. Le gain en temps s'obtient en faisant le rapport des deux plus grands domaines (les plus lents) de chaque approche. Nous remarquons que le gain obtenu expérimentalement est inférieur au gain théorique. Il y a deux principaux facteurs à cela. Premièrement, dans le calcul théorique, nous n'avons pas pris en compte le temps nécessaire à l'échange des données. Deuxièmement, nous n'avons pas pris en compte les couches absorbantes dans le calcul théorique du gain. Or le nombre d'éléments dans les couches absorbantes est grand et ne peut être négligé. En effet, nous constatons que, dans le cas traité entièrement en 3D, nous avons 8.4 éléments de calcul utile pour 1 élément PML. Dans l'approche hybride, nous avons 7.0 éléments de calcul utile pour 1 élément PML. Nous avons donc proportionnellement plus de PML dans l'approche hybride que dans l'approche directe. De plus, le calcul dans les zones PML est plus complexe. Tout ceci explique principalement la différence obtenue sur les temps pour les deux approches globale et hybride.

	temps d'exécution	Mémoire [Gb]
FDTD-3D seul	1h10′	1.6
Méthode hybride	28′	1.1
Gain (théo./calculé)	3.1/2.46	1.55/1.45

TABLE 5.1 – Temps de calcul et mémoire nécessaire pour traiter le cas test Soyouz 5.8 pour les méthodes FDTD 3D et hybride 3D-1D-3D

En conclusion de cet exemple, la stratégie d'hybridation nous permet bien de correctement prendre en compte l'influence du réseau d'interconnexion. De plus, elle nous permet un gain important en temps et en mémoire. Ce gain est lié à la fois à une parallélisation (calcul dans les différents domaines) mais également à l'utilisation d'une méthode plus adaptée, avec un formalisme plus « compact » pour modéliser la partie centrale. Cependant, cette approche présente une difficulté dans le cas des problèmes basses fréquences comme la foudre. En effet, à cause de la longueur d'onde liée à la source, le domaine global n'est pas très grand en nombre de longueur d'onde et donc il faut tenir compte de toutes les interactions entre les objets présents dans la scène de calcul, ce qui n'est pas actuellement fait dans notre

approche hybride.

5.3.2 Source basse fréquence de type onde plane avec un sol parfaitment conducteur

Nous ne sommes pas capable à l'heure actuelle de simuler de manière satisfaisante un problème foudre avec notre approche hybride. Cependant, nous avons estimé important d'évaluer quand même notre méthode hybride sur une source basse fréquence. Pour faire cette évaluation, nous avons utilisé une onde plane se propageant verticalement et polarisée de manière à ce que le champ électrique soit parallèle à la ligne ($\theta = 0$, $\varphi = 0$, $\boldsymbol{E} = E\boldsymbol{u_y}$). La forme d'onde dans ce cas de figure est donnée par une biexponentielle qui possède un spectre foudre :

$$E(t) = -e^{-\alpha t'} - e^{-\beta t'} \tag{5.2}$$

avec $\alpha = 4.5 \times 10^4$s, $\beta = 0.95 \times 10^6$s et $t' = t - 300/c_0$.

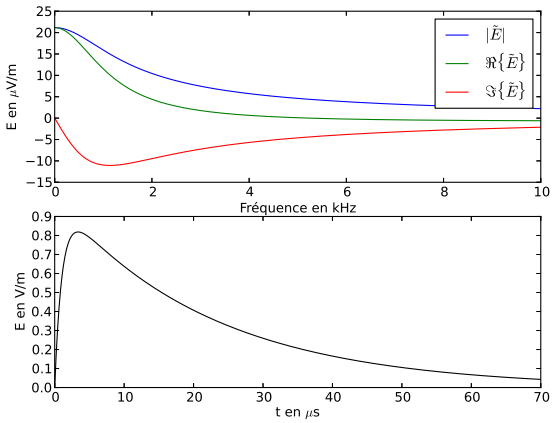
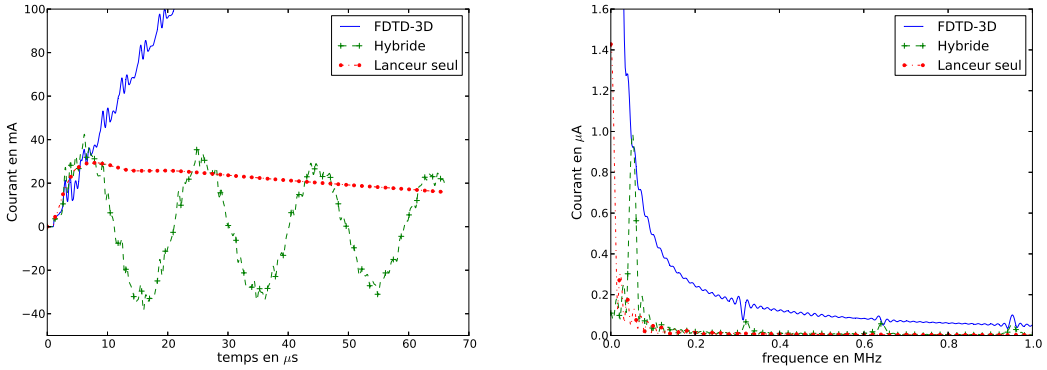


FIGURE 5.12 – Représentation de la forme d'onde de type biexponentielle et de sa transformée de Fourier

La figure 5.13 donne le courant à 10m du lanceur pour différentes configurations et la figure 5.13(b) donne la réponse fréquentielle de ce même courant.

Il est difficile de conclure sur ce résultat. En effet, si nous regardons les résultats pour une configuration où nous avons uniquement le lanceur avec ceux où nous tenons compte de l'ensemble des bâtiments par notre approche hybride, nous obtenons un comportement global identique (décroissance du courant suivant une loi biexponentielle). Avec l'approche hybride, nous retrouvons en plus la résonance entre les deux bâtiments. Ces deux résultats semblent donc cohérents entre eux même s'ils sont très différents d'un calcul FDTD global. Cependant, l'interprétation des fréquences de résonance est complexe. Nous pouvons mesurer quatre fréquences ($f_1 = 953.4$kHz, $f_2 = 643.6$kHz, $f_3 = 321.8$kHz, $f_4 = 52.2$kHz). La deuxième fréquence correspond à la résonance du réseau ($f_3 = 321.8\text{kHz} \rightarrow 466\text{m}$), f_1 et f_2 sont les harmoniques de cette fréquence de résonance. En revanche, f_4 ne représente aucune longueur dans la simulation. Cette fréquence correspond à une longueur de résonance en $\lambda/2$

de 2.87km. Il est probable que cette fréquence traduise le comportement basse fréquence du domaine de calcul et des couches absorbantes.



(a) (b)

FIGURE 5.13 – Courant sur la ligne reliant les deux bâtiments à 10m du lanceur (a) et le spectre (b) pour une agression globale par une onde plane de type biexponentielle d'incidence ($\theta = 0$, $\varphi = 0$, $\boldsymbol{E} = E\boldsymbol{u_y}$).

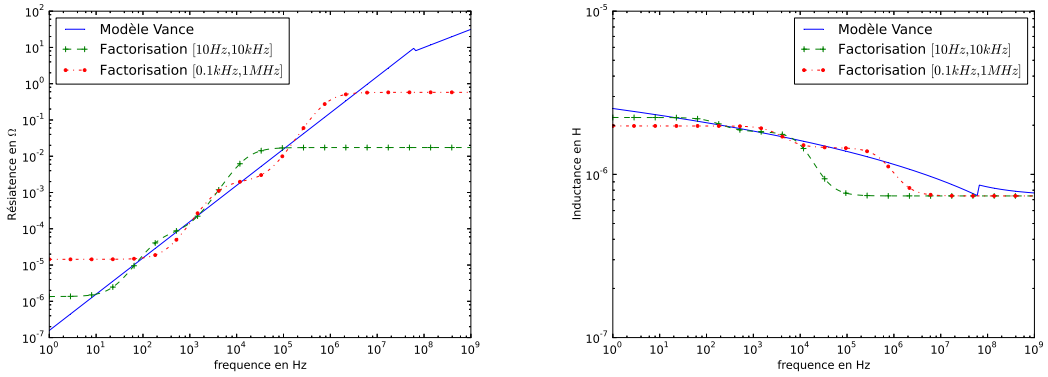
5.3.3 Source basse fréquence avec une injection à la base d'un mât avec un sol réaliste

Dans ce cas test, nous n'utilisons pas de source de type onde plane, mais nous faisons une injection de courant à la base d'un mât. Cette configuration est très proche de ce qui est fait dans le cas d'un essai sur le terrain. De plus pour ce cas test, nous utilisons un sol de conductivité finie $\sigma = 1\text{mS/m}$ qui correspond à une conductivité moyenne réaliste. Enfin, le sol est considéré sur une profondeur de 50m. Cette configuration permet de s'affranchir des effets du rayonnement de la source et nous espérons ainsi valider une partie de notre approche hybride, lorsqu'elle est dédiée aux problèmes basses fréquences de type foudre.

La figure 5.14 nous donne l'impédance de la ligne obtenue par l'approximation de E. F. Vance (Chapitre 2.2 page 33) avec deux décompositions en série de Debye à deux termes. La première décomposition est réalisée par une optimisation sur l'intervalle fréquentielle $[0.1\text{kHz}, 1\text{MHz}]$; la deuxième est faite sur l'intervalle $[10\text{Hz}, 10\text{kHz}]$. La première décomposition sera utilisée pour la source haute fréquence (dérivée de gaussienne), la deuxième pour les basses fréquences (biexponentielle). Nous avons choisi de mieux modéliser la partie basse fréquence du spectre de chacune des sources car l'impédance de la ligne croit très rapidement et les hautes fréquences vont donc être très vite absorbées. En définitive, nous pouvons dire que le comportement global du système est dirigé par les basses fréquences.

Nous avons également ajouté un fond de fouille sous la zone du lanceur qui relie chaque angle du bâtiment et des mâts entre eux. La figure 5.15 représente le maillage utilisé.

Dans le calcul, nous n'utilisons pas une source globale que nous pouvons imposer de manière cohérente dans tous les domaines. Nous négligeons donc l'influence du rayonnement



(a) (b)

FIGURE 5.14 – Représentation de la dépendance en fréquence des paramètres de ligne de transmission ainsi que des décompositions en série de Debye associées.

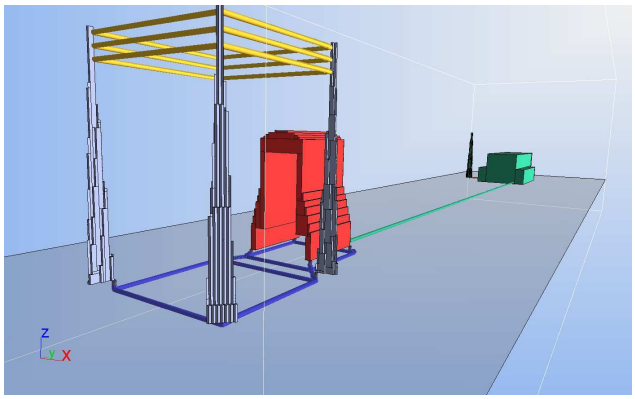


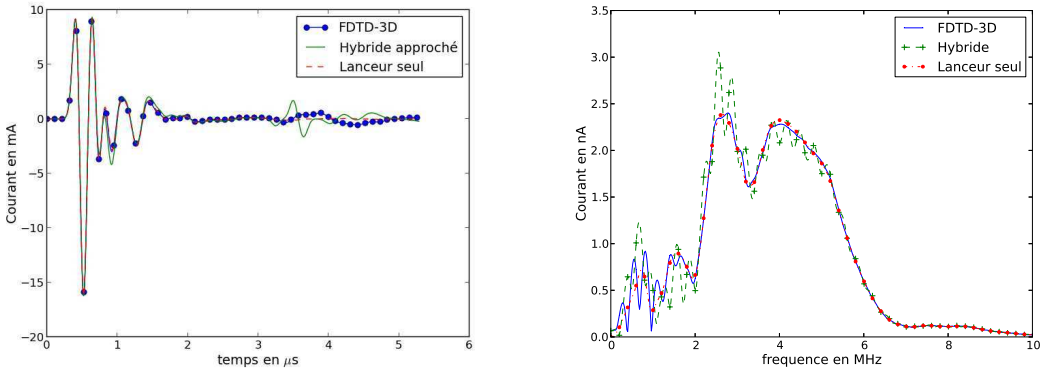
FIGURE 5.15 – Représentation du maillage avec un sol de conductivité finie.

direct d'un bâtiment sur l'autre. Nous avons vu dans la présentation du modèle de Agrawal (Sec. 2.2.3 page 46) que négliger les tensions de fermeture aux deux extrémités de la ligne, induites par un champ incident, revient à négliger le champ magnétique sur toute la longueur de la ligne. Cette approximation peut donc être importante et nous avons constaté numériquement l'influence de cette contribution dans les validations numériques de notre stratégie d'hybridation (Sec. 4.4.4 page 147).

Une erreur liée à la non prise en compte du champ rayonné dans la deuxième méthode 3D devrait se manifester par l'apparition d'une d'onde parasite rétrograde. Nous pouvons dans une première approximation estimer que cette onde rétrograde arrivera en un point d'observation sur la ligne à l'instant $t_2 = 2(L-l)/\nu + t_1$ où ν est la vitesse de propagation le long de la ligne et t_1 est l'instant où la première onde progressive arrive au point d'observation, placé à une distance l du lanceur.

La figure 5.16(a) représente le courant à 10m du lanceur pour différentes configurations pour une source de type dérivée de gaussienne (Eq. 5.1). Sur ce résultat, nous pouvons mesurer l'instant t_1 de passage de l'onde progressive $t_1 \approx 0.40\mu s$, ce qui nous permet d'estimer l'instant où l'onde rétrograde arrive en ce même point, soit $t_2 \approx 3.28\mu s$. Cette valeur est très proche de ce que nous pouvons observer sur nos résultats de simulation ($\approx 3.2\mu s$). Nous observons également un défaut aux alentours de $1\mu s$ qui résulte de l'apparition d'une onde parasite au niveau du plan d'hybridation.

Malgré le bon comportement de l'approche hybride par rapport au calcul global 3D, il existe une différence entre les solutions qui est certainement attribuée au fait de ne pas avoir tenu compte du rayonnement entre les bâtiments. Or, nous savons que nous ne pouvons pas raisonnablement négliger cette contribution à basse fréquence.



(a) (b)

FIGURE 5.16 – Courant sur la ligne reliant les deux bâtiments à 10m du lanceur (a) et le spectre (b) pour une injection de type dérivée de gaussienne à la base d'un mât.

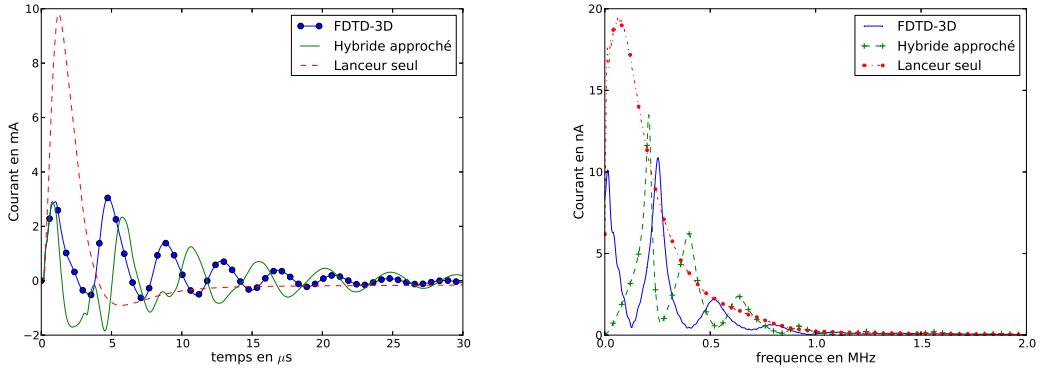
La deuxième figure 5.17 représente le même problème avec une source caractéristique de la foudre ; c'est-à-dire une forme biexponentielle (Eq. 5.2).

Nous considérons premièrement le cas où nous faisons un calcul en ne considérant que

l'environnement proche du lanceur (lanceur seul). Dans ce cas de figure, nous constatons que les très basses fréquences sont très vite absorbées par le sol. Maintenant, si nous considérons la stratégie d'hybridation nous constatons également ce phénomène, mais nous voyons également l'influence du réseau de ligne qui résonne. Nous pouvons estimer la fréquence de résonance en $\lambda/2$ par $f_c = c_0/(2L) \approx 320$kHz. Dans le cas d'un calcul 3D, nous mesurons une fréquence de résonance aux alentours des 250kHz et dans le cas avec l'approche hybride, nous avons 210kHz. La stratégie d'hybridation nous permet donc de prendre en compte ce phénomène.

Nous voyons également dans ce cas de figure l'apparition de l'onde rétrograde parasite liée à la non prise en compte du rayonnement d'un domaine sur l'autre dans l'approche hybride. Nous constatons également que les fréquences de résonance sont décalées dans les basses fréquences et sont moins atténuées, ce qui traduit certainement une erreur dans la décomposition en série de Debye.

En définitive pour cette problématique, la stratégie d'hybridation permet de prendre en compte l'influence du réseau de connexion entre les bâtiments, sans pour autant donner une solution satisfaisante sur les courants. Néanmoins, ces résultats préliminaires montrent que le réseau joue un rôle important dans l'écoulement du courant et que l'utilisation de la seule zone proche du lanceur est insuffisante pour étudier son comportement même face à un impact foudre. Toutefois, malgré le gain apporté par l'approche dans la modélisation de ce type de configuration, la prise en compte du champ rayonné d'un domaine 3D sur un autre semble important.



(a) (b)

FIGURE 5.17 – Courant sur la ligne reliant les deux bâtiments à 10m du lanceur (a) et le spectre (b) pour une injection de courant de type biexponentiel à la base d'un mât.

5.3.4 Comportement d'un réseau multifilaire avec une injection de courant à la base d'un mât

Dans ce dernier exemple, nous étudions le comportement d'un réseau multi filaire reliant les différents bâtiments. Dans cette configuration, nous allons considérer les quatre bâti-

ments (lanceur, assemblage, loxlin et kérosène cf. figute 5.5 page 158). Ces bâtiments sont posés sur un sol de conductivité moyenne $\sigma = 1\text{mS/m}$ pour une profondeur de 80m. Nous ajoutons entre les bâtiments un réseau de connexion dont l'encombrement (distance entre les conducteurs) est plus petit que la taille de la cellule du maillage 3D considéré pour cet exemple (centimètre par rapport au mètre). Ce réseau est en court-circuit à chacune de ces terminaisons ; ce qui correspond à une connexion idéale sur les bâtiments. La figure 5.18 donne les différents liens à travers le réseau.

Chacun de ces liens est composé d'au moins deux conducteurs reliant un bâtiment à un autre en passant par des embranchements. Dans ces embranchements, nous relions les conducteurs de part et d'autre par un lien idéal.

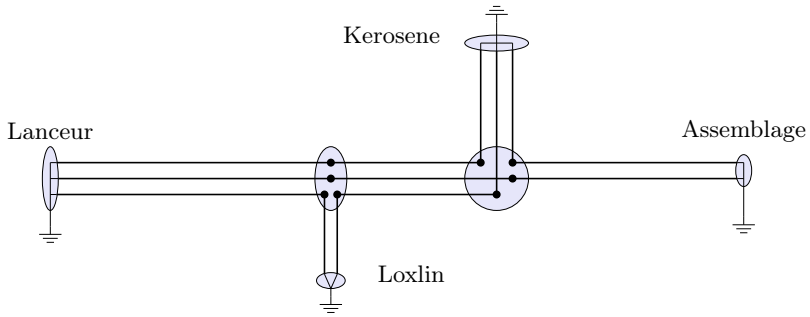
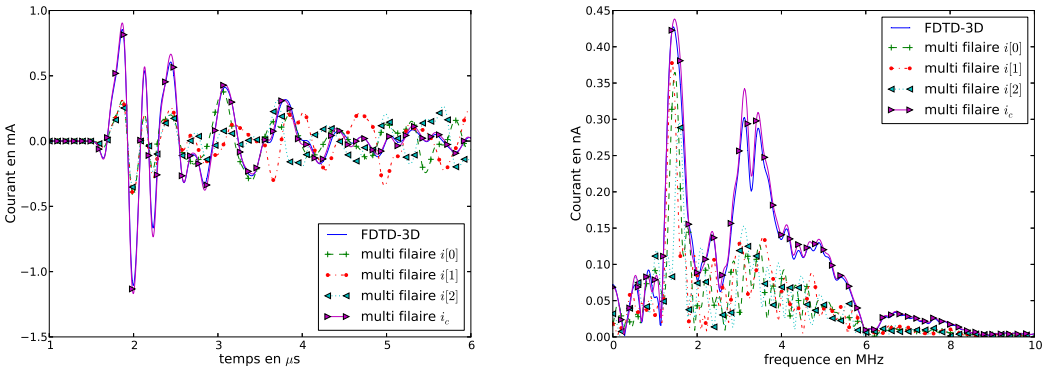


FIGURE 5.18 – Réseau multi-filaire entre les bâtiments lanceur, assemblage, loxlin et kérosène

Dans ce cas de figure, nous avons utilisé une source de type dérivée de gaussienne. La principale difficulté est qu'il existe peu d'alternative pour traiter cette problématique ; nous n'avons pas pu confronter les résultats obtenus par notre stratégie d'hybridation champ/-câble avec d'autres approches.

La figure 5.19 représente comme dans les cas tests précédents le courant à 10m du lanceur pour la stratégie d'hybridation. Elle représente également le courant de mode commun ($i_c = \sum i_j$) et le courant « équivalent » obtenu en modélisant le réseau par un câble équivalent (un fil simple).

Le premier constat que nous pouvons faire, c'est que le courant de mode commun dans ce cas de figure, donne un résultat proche du courant de mode commun calculé par l'approche multi filaire. Le deuxième constat que nous pouvons faire est que malgré le fait que le courant de mode commun décroît de manière significative le courant qui circule dans chacun des conducteurs reste important et d'amplitude relativement constante. Le caractère transitoire de la source est perdu et une partie de l'énergie résonne dans le réseau, ce qui n'est pas sans conséquence pour des équipements potentiels.



(a) (b)

FIGURE 5.19 – Courant sur la ligne multi-conducteurs reliant les deux bâtiments à 10m du lanceur (a) et le spectre (b) pour une injection de courant de type dérivée de gaussienne à la base d'un mât.

5.4 Conclusion

L'utilisation de notre stratégie d'hybridation 1D/3D sur une maquette simplifiée d'une infrastructure terrestre de grande taille comme la base de lancement Soyouz a montré plusieurs choses.

1. La stratégie d'hybridation permet de prendre en compte l'influence du réseau en un temps comparable à celui nécessaire pour étudier uniquement l'environnement proche du lanceur. Il semble que considérer uniquement la zone proche du lanceur conduise à une majoration des valeurs qui nous intéressent. Cependant, les fréquences de résonance ou le courant transporté sur le réseau global peuvent entraîner des dommages importants sur les systèmes connectés au lanceur. Il est donc important de les prendre en compte dans la simulation.
2. La stratégie d'hybridation de domaine doit prendre en compte les sources globales (canal de foudre) dans tous les sous-domaines. Pour cela, dans le cas d'un impact indirect, le rayonnement de la source peut être calculé analytiquement [54]. Cependant, ce calcul peut devenir complexe à généraliser dans le cas où nous avons un sol de conductivité finie, car dans ce cas, le calcul du rayonnement fait intervenir une intégrale de Sommerfeld qui converge lentement. L'utilisation de formulations approchées est aussi possible, comme par exemple, la formule de Rubinstein qui permet d'avoir un bon compromis entre temps de calcul et précision.
3. Les travaux menés dans cette thèse n'ont pas complètement couvert la problématique foudre. Les principales validations se sont portées sur des sources hautes fréquences. Il pourrait être intéressant d'évaluer ces stratégies avec des techniques permettant de s'affranchir de la problématique basse fréquence dans le domaine 3D.
4. En dernier lieu, la stratégie d'hybridation champ / câble, qui permet de prendre en compte un routage multi-filaire dans la cellule de Yee, est très intéressante. Cependant, il est très difficile de vérifier numériquement la pertinence de toutes les hypothèses. Une validation expérimentale semble la plus adaptée pour ce type de configuration.

Deuxième partie

Re-visite des solutions informatiques pour l'hybridation et le calcul parallèle

Chapitre 6

L'informatique, outil pour la science

Sommaire

6.1 Introduction 176
6.2 Bonne pratique de la programmation 177
6.2.1 Lisibilité du code - le codage 177
6.2.2 Pérenniser son travail - Les tests unitaires 177
6.2.3 Architecture robuste - Programmation orientée objet (POO) . . . 178
6.2.4 Conclusion 179
6.3 Les nouvelles orientations de l'informatique 181
6.3.1 Vitesse de calcul ou vitesse de mise en œuvre 181
6.3.2 Mélanger les styles pour allier performance et confort 183
6.3.3 Mise en œuvre d'une programmation parallèle 184
6.4 Conclusion 191

6.1 Introduction

Ce chapitre, qui traite de l'informatique pour les sciences, est un peu en marge de mon travail de thèse. Cependant, j'ai voulu présenter cette problématique pour plusieurs raisons. D'une part, le problème informatique occupe une grande partie du temps de travail des numériciens. Il n'est donc pas superflu de l'aborder. De plus le scientifique n'est pas un informaticien ; il est donc parfois un peu perdu devant une notion aussi large. C'est pourquoi, il me semble important d'en présenter l'essentiel. Dans un premier temps, nous parlerons de ce qui peut être appelé les « bonnes pratiques de l'informatique » qui décrivent ce qui est considéré par une grande partie de la communauté informatique, comme une base solide pour le développement. Dans cette partie, nous n'en présenterons qu'un minimum et invitons chacun à être curieux. Dans un deuxième temps, nous aborderons les dernières avancées et innovations en informatique qui changent notre vision de la programmation.

6.2 Bonne pratique de la programmation [61, 62, 63]

Dans cette section, il n'est pas question de « comment gérer un projet » [64, 65] mais bien de se munir des bons outils pour qu'un projet informatique soit viable à long terme.

Je vais présenter un ensemble de règles et utilitaires pour qu'un projet informatique ne devienne pas inutilisable. Il est vrai que s'astreindre à une telle rigueur est contraignant. Mais si nous ne faisons pas l'effort ***une*** fois nous devrons faire un effort à ***chaque*** fois. Alors faisons le la ***première*** fois pour ne plus ***jamais*** le refaire. Un bon informaticien est un bon fainéant qui ne fait jamais ce qu'un ordinateur peut faire à sa place.

Il m'arrive encore de temps à autres de m'autoriser quelques passe-droits pour aller plus vite ou parce que je suis en retard. Mais à chaque fois une semaine, un mois, un an après je le paye.

Nous aborderons la bonne pratique de l'informatique suivant trois axes : la lisibilité du code, la pérennisation du travail et une architecture logicielle robuste.

6.2.1 Lisibilité du code - le codage

Quelque soit le langage il est important de suivre des règles de codage. Ces règles ne sont pas là pour nous astreindre mais elles sont le résultat d'une expérience collective qui ont plusieurs objectifs : permettre une lecture plus simple du code et cela quelque soit la personne qui l'a écrit. En général, choisir un bon éditeur (Vi, Emacs, Eclipse) et prendre un peu de temps pour le configurer correctement, est suffisant pour régler ce problème.

Un autre point important est le bon usage des commentaires. Un commentaire, pour être efficace, ne doit pas expliquer comment on fait mais ce qu'on fait. En particulier pour une fonction par exemple : expliquer ce qu'elle réalise comment elle réagit aux options, son domaine de validité etc...

Il y a de nombreux ouvrages qui présentent ces problématiques [66, 67]. Certains présentent même l'écriture de code illisible comme une solution pour protéger son emploi. En effet, qui osera congédier la seule personne capable de décoder une portion de code capitale [68] ! Mais attention à ce que nos propres codes ne deviennent pas un mystère pour nous-même.

6.2.2 Pérenniser son travail - Les tests unitaires

Quand nous développons, nous testons ce que nous faisons tout ou en partie. Mais bien trop souvent, ces tests sont faits manuellement. Nous prenons un cas test, nous exécutons puis nous vérifions que le résultat est bon. Le problème de cette approche est qu'elle n'est pas automatisée ni pérenne dans le temps. Il y a plusieurs conséquences directes :

– La portabilité : comme les tests sont faits sur une seule machine (celle du développeur) avec un compilateur, comment je peux garantir que mon outil va marcher sur toutes les machines et avec tous les compilateurs ? Notons que, comme tout programme informatique, les compilateurs sont buggés et implémentent plus ou moins partiellement le standard des langages informatiques [69], ainsi l'utilisation de plusieurs compilateurs permet d'avoir un code plus stable.

– La régression : La vie d'un outil ne se termine jamais. Nous ajoutons régulièrement de nouvelles fonctionnalités. Mais l'ajout d'une fonctionnalité entraîne forcément des modifications dans ce qui a déjà été réalisé. Ainsi, bien plus souvent qu'on ne le croit, cela provoque des pertes totales ou partielles d'une fonctionnalité d'un outil.

La solution qui nous permet de garantir que nous n'avons pas dégradé les capacités de notre outil, que ce soit pour un portage ou une évolution, est de repasser tous les tests. Il est évident que cela n'est envisageable que si ces tests sont automatisés. Une bonne manière d'automatiser les tests est de prendre la voie du « Développement Dirigé par les Tests » [70, 71, 72] qui est une technique de développement de logiciel qui préconise d'écrire les tests unitaires avant d'écrire le code source d'un logiciel.

6.2.3 Architecture robuste - Programmation orientée objet (POO)

Dans cette section, je veux présenter rapidement la problématique de production de code robuste. Il existe de nombreuses méthodes et approches pour aboutir à cet objectif. J'ai choisi de présenter une approche qui me semble performante : la programmation pseudo objet.

Cette programmation est concentrée autour des structures qui portent les données plus que sur la fonction recherchée. Cette approche voit la programmation comme la manipulation d'objets représentant un concept, une idée ou toute entité du monde physique. Ces objets peuvent être agrégés dans d'autres objets pour fournir des services toujours plus évolués. L'exemple de code FORTRAN 6.1 extrait de « Object-oriented scientific programming with fortran 90 » [73] nous montre comment utiliser les « types » pour définir une notion physique comme un ensemble de particules et les « subroutines » pour les manipuler.

```
1  module species_module
2   use distribution_module ; use partition_module
3   implicit none
4   type particle
5     private
6     real :: x, y, z, vx, vy, vz
7   end type particle
8   type species
9     real :: qm, qbm, ek
10    integer :: nop, npp
11    type (particle), dimension(:), pointer :: p
12  end type species
13 contains
14  subroutine species_distribution(this, edges, distf)
15    type (species). intent (out) :: this
16    type (sla,bpartition), intent (in) :: edges
17    type (dist,fcn), intent (in) :: distf
18    ! subroutine body
19  end subroutine species_distribution
20  ! additional member routines ...
21 end module species_module
```

Code Listing 6.1 – Résumé du module FORTRAN 90 définissant la notion d'ensemble de

particules.

Au final, nous ne manipulons plus directement les données, mais nous utilisons des services, des capacités associées à une structure de données. L'exemple de code FORTRAN (Src. H.2 page 260) est une bonne illustration de cette approche. De plus, si nous utilisons un langage supportant la programmation objet et le polymorphisme, deux structures déclinant les mêmes services peuvent être utilisées dans un même algorithme ; ce qui est toute la puissance du polymorphisme.

Il y a plusieurs avantages à ce type de programmation. D'une part, elle oriente le découpage du code, ce qui permet en général une plus grande réutilisabilité du travail. D'autre part, elle offre une grande souplesse car nous ne manipulons pas directement les données mais une interface, ce qui permet de faire de la programmation générique. Cette notion de programmation orienté object est un principe que nous allons plus amplement développer dans le dernier chapitre de cette partie (Sec. 7.2.3 page 196). Car cette approche est indispensable pour mettre en place une stratégie d'hybridation non intrusive.

6.2.4 Conclusion

Cette première section introductive de la partie sur l'informatique avait pour but de présenter trois notions qui me semblent importantes dans le développement informatique.

- l'utilisation de règles de codage qui permettent toujours un gain de temps sur le long terme [74],
- la mise en place de test unitaire pour toujours être persuadé que tout marche,
- et enfin, l'utilisation d'une architecture qui se révèle à l'usage à la fois naturelle et pratique.

J'ai voulu présenter les deux premiers points car ils me semblent importants dans la gestion de projet informatique (Fig. 6.1) et ont été utilisés tout au long de cette thèse. Le dernier point est plus important car la notion de programmation orientée object est centrale pour permettre la mise en place d'une approche hybride non-intrusive.

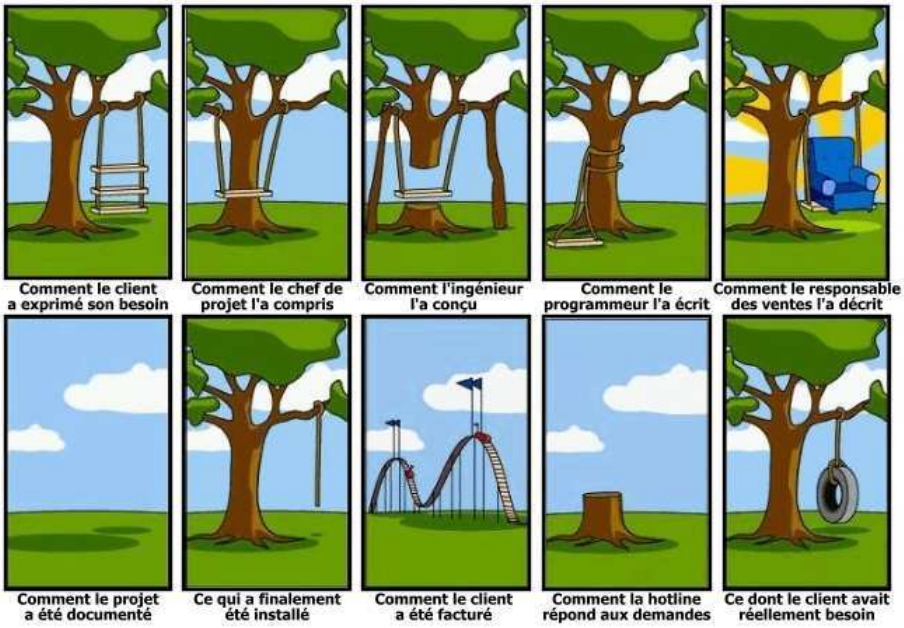


FIGURE 6.1 – Les problématiques de la gestion informatique (vision humoristique) [75]

6.3 Les nouvelles orientations de l'informatique

6.3.1 Vitesse de calcul ou vitesse de mise en œuvre

L'informatique est en perpétuelle évolution pour proposer des solutions toujours plus performantes et en particulier plus rapides. Il y a près de 50 ans les premiers programmes étaient en instruction machine [1] qui demandait au développeur une compréhension poussée de l'architecture machine et un temps infini pour réaliser les programmes les plus élémentaires. La tâche était ardue. Puis les langages de programmation « évolués » sont apparus FORTRAN (1954), puis le B (1969), le très célèbre C (1972), Java (1995) et le Python (1990). Il y a toujours eu une compétition entre les langages compilés en code machine natif (FORTRAN, C, etc...) et les langages interprétés (Java, Python). Les langages compilés avaient jusqu'à présent un avantage écrasant en terme de vitesse d'exécution. Mais le 21ème siècle semble être de meilleur présage pour les langages interprétés ; par exemple Java 7 a des performances de calcul proche du FORTRAN et de C/C++ (Tab. 6.1). La nouvelle famille d'interpréteur intitulée « just-in-time (JIT) compiler » ou compilateur à la volée ont des performances qui ne cessent de croître et donnent une alternative intéressante à l'optimisation dynamique d'instructions [76]. Il y a deux projets intéressants dans ce domaine, le compilateur à la volée de Python PyPy [77] et le langage Julia [78].

Cependant les langages interprétés restent plus exigeants en ressource et la plupart du temps moins rapides à l'exécution que les langages compilés en code natif. Ils gardent toutefois un intérêt notamment par leur facilité de mise en œuvre et la portabilité des programmes, qui peuvent la plupart du temps être lancés sans modification sur toute plate-forme où fonctionne l'interpréteur.

	|-	|—	25%	médian	75%	—|	-|
FORTRAN Intel	1.00	1.00	1.00	1.00	1.46	2.15	8.31
C GNU gcc	1.00	1.00	1.00	1.17	1.41	2.03	4.30
C++ GNU g++	1.00	1.00	1.00	1.22	1.66	2.00	2.00
Java 7	1.09	1.09	1.49	1.78	2.05	2.89	3.46
Go	1.85	1.85	4.06	5.31	8.50	15.18	15.46
Python	1.12	1.12	8.21	51.97	99.83	129.04	129.04

TABLE 6.1 – Quel langage de programmation est le plus rapide [79] ?

Une question importante à se poser est : pourquoi deux langages compilés (FORTRAN, C) peuvent avoir une différence de performance de près de 20% ? Ou encore pourquoi les langages les plus populaires sont les langages interprétés (Fig. 6.2) en dépit de leurs performances ? La réponse est dans le compilateur (ou interpréteur). Un compilateur est un programme qui va convertir une instruction de langage en un ensemble d'instructions machine. Ainsi plus un langage va offrir de souplesse de programmation, plus le compilateur va devoir être performant pour comprendre ce que fait le programmeur pour en déduire la meilleure séquence d'instructions machine. Il y a un équilibre entre performance et souplesse

1. Instruction machine souvent abrégée par instruction est l'opération élémentaire demandée par un programme à un processeur.

de programmation.

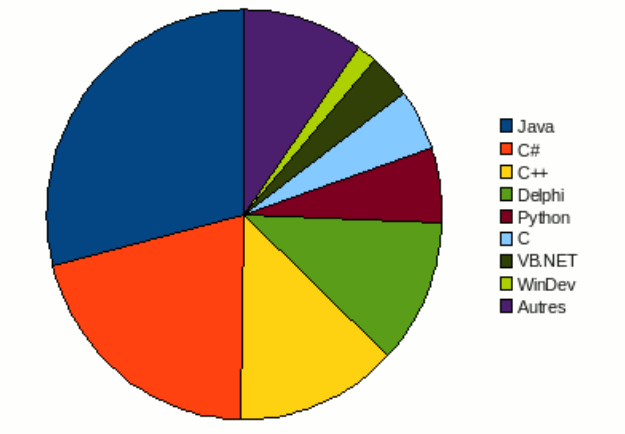


FIGURE 6.2 – Langages de programmation préférés

La comparaison (Sec. G) entre différents langages sur des exemples simples comme l'utilisation de tableaux ou la liste chaînée montre plusieurs choses :

- Plus le langage a une forme explicite des structures de données, plus le compilateur pourra trouver la séquence d'instruction optimale (exemple du tableau en paramètre d'une fonction (Sec. G.2)).
- En revanche plus le langage est souple (exemple de la liste chaînée (Sec. G.1) avec le pointeur de void en C ou typage dynamique fort en Python), plus la généralisation est simple et permet d'écrire moins de lignes de codes. Cependant cette généricité a un coût : possédant moins d'informations sur les données, il est plus délicat d'optimiser à l'extrême les performances.

La conséquence d'utiliser un langage « générique » ou avec des paradigmes plus évolués, c'est qu'il faut moins de lignes de codes pour réaliser une même fonctionnalité. Le tableau 6.2 nous donne une comparaison en différents langages. Nous remarquons que les 20% en temps de calculs gagnés en utilisant FORTRAN nous ont coûté 20% de plus de lignes de codes à écrire.

Langage	Lines ratio
c	1
C++	1
FORTRAN	0.8
Java	1.5
Python	6.5

TABLE 6.2 – Comparaisons du nombre de lignes utiles entre langages de programmation[79]

Il faut donc se poser la bonne question : vaut-il mieux avoir un langage rapide ou avoir un langage facile à écrire ? Nous pouvons considérer ce problème sous plusieurs points de vue. Nous avons choisi de le considérer suivant l'axe des coûts, induits et engendrés.

Supposons que nous voulons développer un outil dans un langage A, nous estimons qu'il nous faudra une année-homme de travail pour réaliser cet outil. Sachant que la moyenne de production d'un développeur est de 10 lignes de codes utiles par jour [80, 64], ce projet comprendra $x = 230*10 = 2300$ lignes de codes utiles à la fin du projet. Si je choisis un autre langage B, je peux obtenir (Tab. 6.2) le nombre de lignes nécessaires pour réaliser le même outil soit $y = r * x$. Par exemple, si je passe du FORTRAN au Python $y = 0.1231x = 283$ lignes de codes suffisent. Ainsi un projet d'une année en FORTRAN peut être ramené à un peu moins de deux mois.

Nous pouvons comptabiliser ce gain de temps en profit net mais pour un ingénieur / chercheur le calcul est un peu différent ; ce n'est pas un objectif qui est fixé (il n'y a pas de fin) mais c'est le temps que nous devons considérer. Donc moins de programmation entraîne plus de temps pour la recherche, etc ...

Les détracteurs de cette approche font deux reproches :

- Ils développent plus de 10 lignes de code par jour. Faites le décompte de votre production annuelle et vous verrez. De toute manière, si nous produisons plus vite il n'en reste pas moins que si nous avons moins de lignes de code à écrire nous gagnons du temps.
- C'est bien beau de faire des programmes en peu de lignes de code mais si leur exécution dure 100 fois plus de temps, nous n'avons rien gagné. Nous avons vu que cette idée reçue est mise à mal.

Pour conclure cette partie, il est évident qu'il n'y a pas de langage universel alliant à la fois performance d'exécution et souplesse de mise en œuvre. Cependant, nous devons nous poser la question : quel langage est le plus pertinent pour mon objectif, et pourquoi pas une solution hybride ?

6.3.2 Mélanger les styles pour allier performance et confort

Il est vrai qu'il faut un compromis entre vitesse de réalisation et vitesse d'exécution, car il est difficile de gagner sur les deux plans. En revanche, il existe des stratégies permettant de tirer partie des deux. En effet, dans la plupart des produits informatiques, le principe de Pareto [2] [81, 82] se vérifie. Ce principe empirique également appelé règle 80/20 dit que seul 20% du code est critique et consomme 80% du temps d'exécution total. Les 80% du code restant réalise par exemple des fonctionnalités ergonomiques (interface homme-machine (IHM), entrées/sorties, gestion des erreurs, etc...) ou d'usage ponctuel. Il est souvent considéré comme une perte de temps d'optimiser ces portions de code.

Cette stratégie consiste à utiliser un langage de haut niveau pour les 80% de code non critique. Nous ne passons donc qu'un temps limité à réaliser des fonctions ergonomiques, qui sont d'ailleurs souvent plus simple à réaliser dans ces types de langage. Puis, nous utilisons un langage le plus performant pour les 20% de code critique. Par exemple, en utilisant Python comme langage de haut niveau pour réaliser les 80% de code « ergonomique » puis pour optimiser les 20% restant, nous pouvons faire simplement des appels à des langages compilés. Nous pouvons par exemple utiliser la puissance du C en utilisant Cython [83]

2. Le principe de Pareto, aussi appelé loi des 80/20, est le nom donné à un phénomène empirique constaté dans certains domaines : environ 80 % des effets sont le produit de 20 % des causes.

qui est un « méta » langage qui simplifie l'écriture d'extension en C pour Python. Nous pouvons également exploiter les performances du FORTRAN en paquetageant de manière semi-automatique une bibliothèque FORTRAN avec f2py [84, 85] ou FWRAP [86]. Plus généralement et la technologie nous l'impose, nous pouvons utiliser un langage hôte comme le C/C++/JAVA/Python et implémenter les fonctions critiques sur GPGPU [87] via CUDA [88] ou plus généralement avec OpenCL [89]. Notons que cette approche permet de réaliser rapidement des prototypes de toutes les idées que je peux avoir, puis de passer un peu plus de temps sur les fonctionnalités réellement pertinentes (les plus utiles, les plus intéressantes, etc...) pour les optimiser et ainsi avoir une application souple et performante.

Pour illustrer mes propos, nous allons comparer trois implémentations différentes d'un code de ligne en temporel. Une version FORTRAN car ce langage est réputé performant. Une version Python/NumPy a également été implélemtée pour sa simplicité de mise en œuvre tout en offrant des performances raisonnables ; enfin une version Python/OpenCL, pour tirer partie des capacités offertes par le calcul GPU.

6.3.3 Mise en œuvre d'une programmation parallèle

Dans cette section, nous allons présenter différentes implémentations d'un code FDTD 1D pour les lignes de transmission. Ces différentes implémentations ont toutes cherché à utiliser au mieux les capacités offertes par un langage pour faire du calcul numérique performant. Nous terminerons cette présentation par une comparaison en terme de performance brute (temps de calcul).

Le code FORTRAN que nous allons présenter est la première version du code de ligne que nous avons développé. Cette version permet de traiter en temporel un réseau composé de plusieurs branches simples ; c'est-à-dire mono-filaire et de paramètres R, L, C et G constants.

Les fonctions centrales dans un algorithme FDTD sont les fonctions de mise à jour du champ électrique (i et v). Dans notre cas de figure, il y a deux fonctions, la fonction de mise à jour du courant et celle de la tension. Ces deux fonctions diffèrent peu l'une de l'autre (cas des valeurs en bout de ligne). La portion de code (Src. 6.2) nous donne les fonctions utilisées pour mettre à jour le courant le long de la ligne. La première fonction est `segment_eval_middel_i` ligne 255. Cette fonction calcule la nouvelle valeur du courant i en fonction de la tension à gauche et à droite ainsi que l'ancienne valeur du courant. De plus, le premier argument de cette fonction `this` du type `segment_t` représente les paramètres électriques de la ligne de transmission. Ces paramètres C_m et D_m sont calculés une seule fois et sont stockés dans la structure pour éviter un surcoût de calcul.

Cette fonction est définie pour des valeurs scalaires (un seul courant). Ainsi pour mettre à jour tous les courants le long de la ligne, nous devons faire une boucle. Nous pouvons faire une boucle explicite mais nous pouvons également utiliser la commande `elemental` dans la déclaration de la fonction. Cette commande FORTRAN indique au compilateur FORTRAN que cette fonction est « pure » [3] et « vectorisable » c'est à dire qu'elle peut être appelée en remplaçant chaque attribut par un tableau de même dimension. Dans ce cas de figure, c'est le compilateur qui implémente la boucle sur tous les éléments du tableau. Cette écriture

3. Une fonction dite pure en FORTRAN signifie qu'elle n'a pas d'effet de bord, c'est-à-dire qu'elle ne modifie l'état d'aucune autre variable (variable statique ou globale, lire ou écrire des données de ou vers un écran ou un fichier) autre que celle retournée par la fonction.

a bien des avantages ; d'une part, elle économise l'écriture d'une boucle, mais surtout elle informe FORTRAN qu'il peut faire un calcul parallèle optimisé sur les tableaux passés en argument.

```
248   !----- public interface -----!
249
250   !
251   ! I_{p+1/2}^{n+1/2} = f(I_{p-1/2}^{n+1/2}, V_{p-1}^{n}, V_{p}^{n
          })
252   !
253   ! update I on a segment
254   !
255   elemental function segment_eval_middel_i(this, v_left, i_last,
          v_right) result(i)
256     type(segment_t), intent(in) :: this
257     real(data_t), intent(in) :: v_left, i_last, v_right
258
259     real(data_t) :: i
260
261     i = - i_last * segment_get_Cm(this) + (v_left - v_right) *
            segment_get_Dm(this)
262   end function segment_eval_middel_i
263
264   !----- FDTD interface -----!
265
266   subroutine segment_advance_left_half_time(this, field, n)
267     type(segment_t), intent(in) :: this
268     type(field_t), intent(inout) :: field
269     integer, intent(in) :: n
270
271     call field_set_i(field,                          &
272          segment_eval_middel_i(this,                 &
273          field_get_v_left_with_source(field, n),     &
274          field_get_i_with_source(field, n),          &
275          field_get_v_right_with_source(field, n)) )
276   end subroutine segment_advance_left_half_time
```

Code Listing 6.2 – Implémentations de la procédure FDTD en FORTRAN.

C'est l'usage que nous en avons fait dans la subroutine `segment_advance_left_half_time` ligne 266. `this` représente toujours notre ligne et `n` est le numéro de l'itération temporelle. `field` est un conteneur générique qui contient deux tableaux, un pour le courant, l'autre pour la tension. Nous avons accès à ces deux tableaux par les fonctions `field_get_v_*` et `field_get_i_*` que nous utilisons pour calculer la nouvelle valeur du courant.

Cette écriture nous garantit la génération d'un code optimisé par le compilateur qui dans certain cas peut tirer partie d'une architecture parallèle.

Malgré la performance du langage FORTRAN, nous avons choisi de réimplémenter ce périmètre en Python/NumPy pour introduire les modèles dispersifs. Nous avons fait ce choix car l'introduction de nouveau modèle est plus simple dans les langages dits objets et permet d'économiser de nombreux branchements (`IF ELSE`) dans la boucle principale, ce qui est

toujours une source de gain. De plus, l'utilisation de la librairie NumPy permet de faire des calculs sur des tableaux avec des performances comparables aux langages compilés, ce qui nous a permis au passage d'introduire les lignes de transmission multifilaires car NumPy permet d'utiliser des tableaux de matrices comme des tableaux de scalaires.

Comme dans le code FORTRAN, les fonctions critiques sont les fonctions de mise à jour de la valeur du champ. L'approche objet de Python permet d'écrire les deux fonctions de mise à jour du courant et de la tension dans la même méthode `compute`. Dans cet exemple, les tableaux `x` et `y` sont à deux dimensions. La première dimension correspond aux conducteurs, la deuxième dimension correspond aux segments. Ainsi la ligne 190 peut être interprétée comme : $\boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{B}(\boldsymbol{y}_{i+1} - \boldsymbol{y}_i)$

```
1  class SemiYeeScheme(NumericalScheme):
2      """Solve semi-Yee Scheme
3
4          $ \alpha x + \beta \partial_{t}{x} + \partial_{l}{y} = s$
5
6      where:
7
8          * $x$ is solve unknow
9          * $y$ is cross solving unknow
10         * $\alpha$ and $\beta$ two real constants
11         * $s$ is source term
12
13     Note: Is vectorial problem so $\alpha$ and $\beta$ are square
14     matrix and $x$, $y$ two vector. They have all the same rank.
15     """
16     def comput(self, x, y, s=None):
17         """Compute unknow of semi-Yee Scheme without boundary
18
19         INPUTS:
20             s: same shape of x
21         """
22         x = self.xfx * x + self.xfy * (y[:, 1:] - y[:, :-1])
23         if s is not None:
24             x += self.xfs * s
25         return x
```

Code Listing 6.3 – Implémentations de la procédure FDTD en Python.

Nous avons également fait deux optimisations de ce code en utilisant deux technologies de calcul parallèle. La première optimisation utilise numexpr qui est une bibliothèque Python qui permet de faire du calcul vectoriel en utilisant un compilateur à la volée. Cette compilation à la volée permet d'exploiter les capacités multi-cœurs des ordinateurs actuels. Cette librairie est également capable d'exploiter la puissance de la librairie VML (Vector Math Library) de Intel si disponible. La deuxième optimisation OpenCL permet de faire du calcul sur GPU.

Ces deux extensions sont l'extension de la version Python, où seules les fonctions de calculs sont surchargées pour utiliser les capacités de calcul performantes. Ces deux exemples viennent donc illustrer l'argumentation faite auparavant sur le compromis vitesse de calcul

et vitesse de réalisation du code.

Nous pouvons voir la version Python NumPy comme un prototype facile et rapide à écrire qui offre tout de même une performance raisonnable. Une fois que nous avons fait ce prototype, nous pouvons passer un peu de temps à l'optimiser et nous verrons qu'il n'est pas utile de faire une optimisation poussée pour avoir un gain notable.

L'optimisation basée sur numexpr a nécessité l'écriture de 26 lignes de code (Src. 6.4) soit environ 0.2% du code. Pour faire cette optimisation, nous réutilisons le travail déjà réalisé dans l'implémentation Python (ligne 13) puis nous redéfinissons la méthode `comput` qui réalise le calcul. Dans cette fonction, nous utilisons les capacités de calcul offertes par numexpr qui ne présentent pas de grande difficulté.

```
1  #! /usr/bin/env python
2  # -*- coding:utf-8 -*-
3  '''
4  Created on 11 mai 2012
5
6  @author: nathanael
7  '''
8
9  import numerical_scheme_py
10 import numpy as np
11 import numexpr as ne
12
13 class SemiYeeScheme(numerical_scheme_py.SemiYeeScheme):
14     #================================================================
15     # Overwrite numerical_scheme_py.SemiYeeScheme
16     #================================================================
17     def comput(self, x, y, s=None):
18         xfx = self.xfx
19         xfy = self.xfy
20         y_1 = y[:, 1:]
21         y_2 = y[:, :-1]
22         if s is None:
23             return ne.evaluate("xfx * x + xfy * (y_1 - y_2)")
24         else:
25             xfs = self.xfs
26             return ne.evaluate("xfx * x + xfy * (y_1 - y_2) + xfs * s")
```

Code Listing 6.4 – Implémentations de la procédure FDTD en Python/numexpr.

La dernière optimisation que nous avons faite utilise les capacités de calcul offertes par OpenCL. Cette technologie nous permet de tirer partie des cartes graphiques pour faire des calculs vectoriels, mais également d'exploiter les capacités multi-cœurs des CPU modernes. Nous avons fait deux optimisations différentes qui utilisent OpenCL. La première est très similaire à la précédente et a été réalisée en 97 lignes de code soit 0.7% du code total. La portion de code 6.5 donne le code OpenCL utilisé pour faire cette optimisation.

Cette première optimisation n'est pas optimale d'un point de vue GPU car à chaque itération temporelle, nous allons faire des copies mémoire entre la mémoire RAM et la mémoire GPU ce qui coûte cher en temps. Cependant, cette approche ne modifie pas l'architecture

logicielle, ce qui a ces avantages. De plus, nous verrons par la suite que les temps de calcul obtenus par cette optimisation « non optimisée » rattrape les performances de FORTRAN.

Cependant, nous avons tout de même cherché à trouver une optimisation qui permet de tirer pleinement avantage de OpenCL et de l'architecture GPU. Cette optimisation est légèrement plus complexe car nécéssite la synchronisation des données locales et des données utilisées pour le calcul parallèle. Mais le principe est toujours le même, nous cherchons à faire un calcul massivement parallèle sur un vecteur. Cette optimisation a été réalisée en 225 lignes de code.

```
# [...]

class _NumericalSchemeCL(object):
    # [...]
    def get_pgr_str(self):
        "Return the OpenCL program."
        return """
    __kernel void semi_yee_scheme(__global float *x,
                                  __global const float *y,
                                  float xfx, float xfy,
                                  int nb_seg, int nb_con)
    {
      /* compute */
      int gid = get_global_id(0);
      x[gid] = xfx * x[gid] + xfy * (y[gid+1] - y[gid]);
    }
    """

numcl = _NumericalSchemeCL()

class SemiYeeScheme(numerical_scheme_py.SemiYeeScheme):
    # [...]
    def comput(self, x, y, s=None):
        # build buffer of OpenCL computation
        shape = x.shape
        x = x.reshape((x.size,))
        y = y.reshape((y.size,))
        x_buf = cl.Buffer(numcl._ctx, mf.COPY_HOST_PTR, hostbuf=x)
        y_buf = cl.Buffer(numcl._ctx,
                          mf.READ_ONLY | mf.COPY_HOST_PTR, hostbuf=y)
        # get update coefficient
        xfx = self.xfx[0, 0]
        xfy = self.xfy[0, 0]
        # compute new value
        numcl.pgr.semi_yee_scheme(numcl.queue,
                                  (x.size, 1), None,
                                  x_buf, y_buf,
                                  xfx, xfy,
                                  *self.get_nb_seg_con())
        # get output buffer
        dest = np.empty_like(x)
```

```
42          cl.enqueue_copy(numcl.queue, dest, x_buf)
43          dest = dest.reshape(shape)
44          return dest
```

Code Listing 6.5 – Implémentations de la procédure FDTD en Python/OpenCL.

Avant de comparer les différentes implémentations et optimisations, le tableau 6.3 reprend de manière synthétique la couverture fonctionnelle de chacune des implémentation et optimisation.

Implémentassion	fonctionnalités
FORTRAN	ligne de transmission mono-filaire nœuds multi-ligne paramètre électrique constant
Python/NumPy	ligne de transmission multi-filaire nœuds multi-lignes paramètre électrique constant paramètre électrique dispersif source 3D
Python/numexpr	ligne de transmission mono-filaire nœuds multi-lignes paramètre électrique constant
Python/OpenCL	ligne de transmission mono-filaire nœuds multi-lignes paramètre électrique constant

TABLE 6.3 – Périmètre fonctionnel des différents outils.

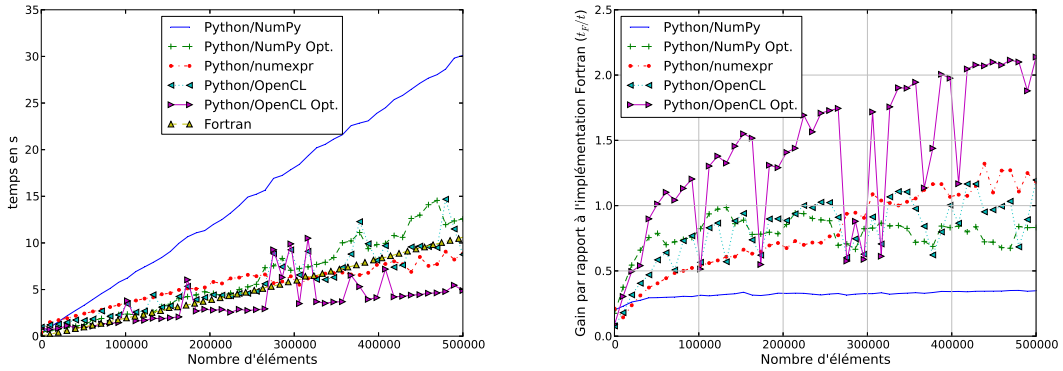
Pour comparer la performance en temps de calcul des différentes implémentations nous utilisons une simple ligne de transmission que nous allongeons progressivement. La figure 6.3 nous donne le temps total d'exécution en fonction du nombre d'éléments sur la ligne. Le résultat est sans équivoque, l'optimisation GPU apporte un gain significatif pour un coût en terme de travail relativement petit par rapport à l'ensemble du travail.

Ce résultat met également en évidence une autre chose importante, c'est que Python comme OpenCL a un temps de latence [4] qui est progressivement compensé par le gain de calcul. Cette latence est extrêmement faible pour FORTRAN, ce qui explique pourquoi cette implémentation est plus performante sur le petit cas.

Pour terminer, ce résultat met également en avant un autre point, le caractère lent souvent prêté au langage Python peut être compensé voire dépassé par l'utilisation de technologie plus performante avec souvent un gain de temps de développement et d'exécution.

Pour terminer cette partie, il faut se méfier de généraliser, tous les langages de programmation ne sont pas tous équivalents. Ils ont tous des spécificités propres et sont souvent pensés pour un besoin particulier. Il faut donc toujours se demander si le langage de programmation choisi est bien le plus adapté pour la problématique visée.

4. temps incompressible lié à l'initialisation d'un système



(a) Performance de trois implémentations d'un code FDTD1D

(b) Gain en performance des deux implémentations Python par rapport à l'implémentation FORTRAN

FIGURE 6.3 – Comparaison des performances en temps de calcul de trois implémentations différentes de code FDTD1D sur un même périmètre fonctionnel donné.

6.4 Conclusion

Cette petite partie est en aparté de mes travaux pour présenter un sujet qui me semble important. L'informatique est un domaine très actif et c'est un euphémisme. Chaque jour, de nouvelles méthodes, langages, technologies, naissent. Ces avancées sont là pour aider le développeur à travailler plus vite, plus efficacement pour produire des outils plus rapides et conviviaux. Mais il y a une autre réalité, un physicien ou un mathématicien n'a pas le temps, ni l'expertise de se tenir au courant de tout ce qui se fait dans cet univers de l'informatique.

Il est donc important de faire remonter les avancées majeures et pertinentes de l'informatique à ces personnes qui restent de grandes consommatrices de technologies informatiques.

Chapitre 7

Du calcul parallèle vers le calcul hybride

Sommaire

7.1 Introduction 193
7.2 Du besoin à la spécification 194
7.2.1 L'identification d'un besoin 194
7.2.2 Les solutions technologiques pour le calcul parallèle 194
7.2.3 Présentation de la méthode 196
7.3 Mise en œuvre d'un prototype 199
7.3.1 L'interface de communication 199
7.3.2 Mise en œuvre d'un worker 201
7.3.3 Définition de l'interface du worker 206
7.3.4 Mise en œuvre d'une stratégie d'hybridation 208
7.4 Exemples d'utilisation du prototype 211
7.4.1 Mise en œuvre du manager 211
7.4.2 Exemples de drivers 222
7.5 Conclusion 230

7.1 Introduction

Nous avons vu tout au long de ce mémoire que les approches hybrides permettent d'aborder de nouvelles classes de problème. Cependant, ces approches restent difficile à mettre en place de par la physique elle-même mais également de par les contraintes informatiques sous-jacentes.

Dans cette section, nous allons aborder la problématique de l'hybridation forte ; en particulier présenter une solution permettant de faire simplement du calcul hybride non ou peu intrusif, sans pour autant sacrifier les performances de calcul. De plus, la problématique de parallélisation (calcul parallèle) est implicite, bien qu'il ne s'agisse pas de faire un énième standard de calcul parallèle.

Nous commencerons par poser le problème et définir les spécificités de l'hybridation forte. Nous présenterons ensuite la solution informatique proposée pour répondre au mieux aux problématiques de souplesse de mise en œuvre, de non intrusivité et de performance. Nous terminerons ce chapitre par donner des exemples de mise en œuvre du manager.

7.2 Du besoin à la spécification

7.2.1 L'identification d'un besoin

La problématique du calcul hybride est un point de convergence entre l'ONERA et AxesSim. Chacun, riche de plusieurs expériences (Quercy, MaxwellTD, hybridation Alice-CRIPTE), entrevoit le potentiel d'une telle stratégie permettant de résoudre des problèmes concrets comme le sujet propre à cette thèse (hybridation 3D-1D). Cependant, un autre constat est fait : si cette approche est prometteuse, sa mise en œuvre est complexe de par des problématiques informatiques. Le cadre de cette thèse nous a permis de réfléchir sur cette problématique.

Mais avant de rentrer dans le vif du sujet, clarifions la notion d'hybridation. Les hybridations se classent suivant deux critères :

1. le type d'échanges entre les unités de calcul : unidirectionnel ou bidirectionnel.
2. la relation temporelle ou récurente des échanges qui peut être forte, (échange à chaque itération) ou faible (pour une exécution séquentielle avec un seul échange). La topologie électromagnétique nous montre que l'hybridation faible n'a pas de sens sans des hypothèses permettant de négliger une rétroaction.

Le tableau 7.1 illustre avec des exemples cette classification.

	unidirectionelle	bidirectionelle
forte	surface Huygens champ-total/champ-diffracté (TF/SF) - rayonnement en domaines disjoints par hybridation TF/SF	interaction entre domaines disjoints par double surface Huygens TF/SF- interaction champ/fil (modèle de R. Holland)
faible	hybridation champ/câble temps/fréquence (FDTD-3D/ BLT)	

TABLE 7.1 – Classification des différents types d'hybridation

L'hybridation faible unidirectionelle semble la plus simple à mettre en œuvre. Cette stratégie de calcul met en œuvre plusieurs outils distincts dans un scénario collaboratif séquentiel ; où un premier outil fait un premier calcul puis stocke un résultat intermédiaire qui sera utilisé comme donnée d'entrée pour un second outil. La principale difficulté est de définir un format d'échange commun ne laissant pas de place à l'interprétation.

En revanche l'hybridation forte, qu'elle soit uni ou bidirectionelle, demande d'une part de convenir d'un format d'échange, mais impose également de revoir en profondeur la structure interne des outils. Entre autre cela nécessite d'introduire un mécanisme informatique permettant la synchronisation et l'échange des données durant le déroulement du calcul.

7.2.2 Les solutions technologiques pour le calcul parallèle

Il existe plusieurs solutions technologiques (thread, processus, socket, OpenMP, MPI, ØMQ, OpenCL...) permettant le calcul parallèle. Chacune de ces solutions est propre ou optimale pour un type de calcul parallèle donné ou plus précisément pour un type d'architecture matérielle décrit ci-après :

- **Calcul multi-cœur :**
  Consiste à exécuter des instructions parallèlement ou de manière imbriquée[1] sur une même puce électronique : nous parlons parfois d'hyper threading ou de thread matérielle.
  Dans cette catégorie les processus partagent la mémoire au niveau du cache[2], cela permet de réaliser des échanges de données très rapides entre les processus.
- **Calcul multi-processeur :**
  Consiste à implanter sur une même architecture matérielle plusieurs unités de calcul qui peuvent échanger de manière rapide des données, par le partage de zone mémoire (vive) nécessitant la mise en œuvre d'un bus de données très complexe qui limite leur déploiement à très grande échelle.
  Dans cette catégorie, les processus partagent la mémoire et sont donc en concurrence directe sur ces zones. L'échange de données entre processus est donc rapide mais délicat.
- **Calcul distribué :**
  Dans cette approche, plusieurs ordinateurs (au sens large) sont organisés en cluster, grille etc, à travers un réseau de communication de type Ethernet (TCP/IP) par exemple.
  Dans cette configuration, les ordinateurs ne partagent pas d'espace mémoire mais échangent des messages à travers un réseau. Dans cette configuration, les partages de données sont coûteuses en temps mais les ressources mémoire sont plus simplement extensibles (scalabilité).
- **Les autres :**
  Processeur vectoriel, calcul sur processeur graphique (GPGPU), etc.

Il est tout à fait possible d'utiliser des technologies prévues pour une architecture sur une autre. En revanche, elle ne pourra pas tirer partie des avantages conférés par l'architecture matérielle. Dans certaines configurations, elle peut même dégrader les performances. Par exemple, Intel a implémenté le standard OpenMP pour des ordinateurs en réseau (Cluster OpenMP*). Dans cette implémentation, la mémoire partagée ne l'est que virtuellement car elle doit être synchronisée entre les différents ordinateurs. Dans ce cas de figure, une utilisation abusive de la mémoire partagée va ralentir les accès car ils devront être synchronisés à travers un réseau. Alors que sur une architecture multi-cœur serait indolore.

Il est important de distinguer les différents types de calculs parallèles car chacun a ces propres spécificités, contraintes et limitations. Dans cette section, nous allons plus particulièrement étudier le calcul distribué nécessitant l'utilisation de technologies permettant l'échange de messages à travers une interface réseau (socket, MPI, ØMQ). Nous nous arrêterons plus particulièrement sur MPI (Message Passing Interfce) et ØMQ (The Intelligent Transport Layer), plus adaptés au calcul scientifique à haute performance.

1. Parallélisme d'instruction ou chevauchement partiel d'instructions permettant de commencer l'exécution d'une instruction sur un microprocesseur avant que la précédente instruction ne soit finie (pipelines). [90]
2. Mémoire directement présente sur le microprocesseur

a. MPI - Message Passing Interface

MPI [91] conçu en 1993, est une norme définissant une bibliothèque de fonctions, disponible dans un grand nombre de langages. Cette norme s'est imposée comme standard pour le calcul parallèle sur système à mémoire distribuée (calcul distribué). Elle peut également être utilisée pour le calcul multi-processeur et multi-cœur sans pour autant en tirer avantage. Sur ce type d'architecture, OpenMP est plus adapté.

Un atout majeur de MPI est que de nombreux fabricants de super-calculateurs fournissent leur propre implémentation de MPI, particulièrement optimisée pour leur architecture matérielle. C'est une des raisons qui permettent à MPI d'avoir aujourd'hui cette place privilégiée pour le calcul scientifique parallèle.

b. ØMQ - The Intelligent Transport Layer

ØMQ (ZeroMQ ou zmq) [92] est une librairie de la famille des AMQP (Advanced Message Queuing Protocol) qui a plus pour vocation de fournir un standard d'échange de messages pour les applications. Il fournit une interface unifiée pour plusieurs protocoles de transports inter-thread et inter-processus pour du calcul parallèle en mémoire partagée ou via un réseau avec TCP (Transmission Control Protocol) ou encore PGM (Pragmatic General Multicast) pour une optimisation dans le cas de transferts multi-récepteurs.

ØMQ se distingue de MPI par le fait qu'il n'est pas basé sur la notion de mono-programme. Avec MPI, de manière schématique, nous exécuterons en même temps N fois le même programme. Dans une architecture basée sur ØMQ, toutes les entités peuvent être distinctes. MPI2 assouplit cette notion de mono-programme mais l'idée reste très présente.

c. OpenPALM

Dans cette section, nous avons présenté les solutions techniques de calcul parallèle permettant de faire du calcul hybride. Cependant, il existe déjà un certain nombre de solutions permettant de faire du calcul hybride. L'une d'entre elles, qui a retenu notre attention, est OpenPALM [93]. Cette technologie, issue du calcul météorologique, a été conçue pour faire du calcul hybride (modèle thermique de l'atmosphère, des continents et des océans).

OpenPALM est basé sur MPI, avec une interface graphique comme celle de LabView [3] [95] (branchement graphique de modules par des liens représentant les échanges entre modules).

Nous n'avons pas retenu cette solution car la définition d'un module pour OpenPALM lui est propre. En d'autres termes, il ne semble pas possible d'utiliser le même module pour une utilisation dans OpenPALM et pour l'utiliser seul ; ce qui est une contrainte qui s'est imposée à nous.

7.2.3 Présentation de la méthode

Quand nous pensons à une stratégie d'hybridation, nous voulons un outil capable de mixer plusieurs codes de calculs, permettant de tirer partie des avantages de chaque méthode et cela de manière presque automatique. Il semble que c'est plus un problème informatique.

3. LabVIEW est un logiciel de développement d'applications de la société américaine National Instruments, basé sur un langage de programmation graphique appelé langage G. [94]

La réalité est bien différente : d'une part, chaque méthode numérique a ses propres spécificités qui peuvent la rendre incompatible avec une autre et cela même en restant dans une même physique. De plus, chaque stratégie d'hybridation s'appuie sur un principe physique. Ainsi les types d'échanges entre les modules peuvent être très variés.

Les outils du domaine temporel n'échappent pas à cette règle. Nous devons parfois faire cohabiter des stratégies temporelles comme le leap-frog, le RK2, etc. avec des pas de temps différents. Il semble délicat de laisser vivre chaque outil de manière indépendante, sans la moindre connaissance des contraintes des autres outils mis en jeu. Cependant, comme nous voulons une architecture peu ou pas intrusive, nous sommes loin du compte si chaque outil doit avoir une pleine connaissance de la stratégie des autres. La définition d'un chef d'orchestre, qui seul connaît la mélodie et donne le rythme à chaque instrument, semble pertinent. Ainsi les instruments ne connaissent et ne jouent que leur partition sans se préoccuper des autres.

L'utilisation d'un « chef d'orchestre » permet de nous affranchir de l'intrusivité associée au principe physique et mathématique de chaque outil mis en jeu. Mais il y a une autre intrusivité, l'intrusivité informatique que l'on peut voir sous deux angles. D'une part, quel est l'effort, le travail à fournir pour hybrider un outil ? L'autre point de vue est la capacité qu'a un outil à fonctionner seul et dans une stratégie d'hybridation.

La figure 7.1 résume la solution proposée. Dans cette solution, les modules de calcul sont des « boites noires ». Par la suite nous parlerons de workers. Nous ajoutons à ces modules une interface. Cette interface doit être superficielle pour qu'elle soit la moins intrusive possible dans le module. Cette interface permet de diriger le worker par le biais d'un contrôleur (driver). Enfin, le rôle du chef d'orchestre est réalisé par le manager qui a une fonction d'administration.

Terminologie :

- worker : définit un module de calcul ou toute autre application que nous voulons hybrider. Un worker est executé au sein d'un processus qui lui est propre. C'est cette propriété qui donne le caractère parallèle de l'exécution. De plus chaque worker (processus) peuvent être executé sur des machines différentes pour distribuer la charge de calcul. Enfin chaque worker peut être des processus parallèles basés sur MPI ou OpenMP par exemple.
- interface : représente les capacités offertes par le worker au sein du manager.
- driver : définit comment est utilisé un worker par le biais de son interface pour réaliser la fonction attendue pour la stratégie d'hybridation. Le driver peut être vu comme la fonction principale (main) exécutée par chaque worker.
- l'application : (ou le manager) est le composant central qui organise les échanges à l'aveugle. Les drivers fournis par l'utilisateur donnent le comportement.

La stratégie proposée s'appuie sur la notion d'un manager qui va administrer et gérer l'exécution de chaque worker suivant la séquence de commandes définie par le driver. Cette vision hérite de l'approche généralement employée pour faire une application multi-méthode avec MPI. Le principe consiste à définir une fonction principale pour chaque méthode mise en jeu. Puis dans la fonction principale de l'application, nous faisons un aiguillage sur chaque méthode en fonction du numéro de processus. Cette approche présente l'inconvénient de devoir lier dans une seule application toutes les méthodes, ce qui peut s'avérer limitant dans

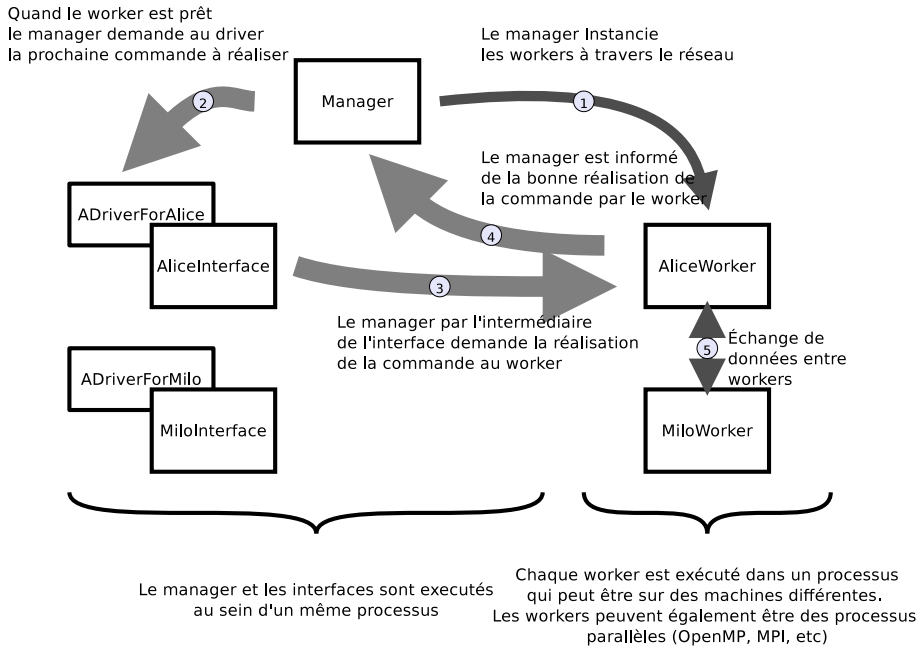


FIGURE 7.1 – Synoptique générale du manager illustré par une hybridation Alice/Milo.

certains cas de figure. L'autre limitation est, qu'en général, l'implémentation de chaque méthode est faite par l'approche dite « dirigée par le contrôle », ce qui en général produit un mauvais découpage fonctionnel de l'implémentation. Autrement dit, nous retrouvons des instructions liées à la gestion du calcul hybride, aux échanges de données dans des fonctions liées au calcul et cela dans des fonctions très profondes.

L'approche proposée impose un découpage fonctionnel plus atomique et fait remonter toutes les fonctions de communication et d'administration au plus haut niveau, ce qui permet de les mutualiser et de les rendre transparentes pour l'utilisateur. Notons, que si l'utilisateur ne voit plus les fonctions liées à l'hybridation, c'est qu'elles ne sont plus intrusives.

7.3 Mise en œuvre d'un prototype

Maintenant que nous avons décrit l'idée principale de notre approche, nous allons approfondir chaque point en détaillant certains aspects du prototype. Nous verrons les différents points de manière ascendante, c'est-à-dire de la plus atomique à la plus générale. Nous commencerons donc par la mise en place d'un protocole de communication. Ensuite, nous verrons la mise en œuvre d'un worker, puis la définition de son interface ; et enfin, la définition des drivers et leur utilisation.

7.3.1 L'interface de communication

Le but de cette interface de communication est de fournir une définition unifiée de la manière dont vont être échangés les messages entre les workers et le manager mais également entre les workers eux-mêmes. Nous verrons dans cette section l'interface pour échanger des messages entre le manager et les workers. L'autre point sera détaillé dans la mise en œuvre du worker (Sec. 7.3.2).

Pour réaliser le périmètre fonctionnel fixé, nous avons défini deux structures de données qui réalisent à un degré moindre le rôle de l'interface de communication. Elle a pour but d'établir un lien de communication avec l'application principale et de recevoir les commandes envoyées.

Nous illustrons notre discours avec des exemples extraits de notre implémentation Fortran. Notons cependant que cette implémentation a également été réalisée en Python et pourrait l'être dans d'autres langages (C, C++, etc). L'implémentation dans différents langages permet d'hybrider ensemble des outils réalisés dans des langages différents.

Fonctionnalité 7.3.1. *Hybridation entre codes de calcul développés dans des langages différents.*

La première structure de données définit une commande et fournit un certain nombre de fonctions permettant de manipuler cette commande. Dans notre exemple Fortran, cette structure est représentée par le type `command_t`, dont les principales fonctions de son interface sont données ci-dessous. Une commande envoyée par l'application principale est formatée sous la forme d'une chaîne de caractères composée du nom de la commande suivie des arguments protégés par des parenthèses. Ces arguments sont représentés sous leur forme textuelle et séparée par des virgules. Notons la ressemblance avec l'appel d'une fonction.

```
NOM_DE_LA_COMMAND('string',␣2,␣0.2)
```

Cette manière de faire nous impose d'interpréter la valeur des arguments (fonctions lignes 10 à 13 (Src. 7.1)). Mais elle permet également de nous affranchir des problèmes de codage entre architecture (32 bit et 64 bit par exemple).

Fonctionnalité 7.3.2. *Hybridation entre architectures différentes.*

La portion de code suivant (Src. 7.1) représente l'interface Fortran publique de la structure `command_t`.

```
1   !------------------------------ header -------------------------------------!
2   IMPLICIT NONE
3   PRIVATE ! all private
4
5   ! command public interface
6   PUBLIC command_t
7   ! property and get-set
8   PUBLIC command_name          ! return the command's name
9   PUBLIC command_len           ! return the number of arguments
10  PUBLIC command_get_arg,     &! return the i-th argument in string
11        command_get_arg_int,  &! ... as an integer
12        command_get_arg_real, &! ... as a real
13        command_get_arg_bool   ! ... as a boolean
```

Code Listing 7.1 – Interface du module Fortran dédiée à la gestion des commandes

Maintenant que nous avons défini la structure générale des commandes, nous allons définir plus précisément leurs rôles et comment nous les mettons en œuvre.

Nous définissons trois types de commandes :

- Les commandes fournies par le worker. Ces commandes sont spécifiées par l'utilisateur et implémentées dans le worker.
- Les commandes dites internes qui sont standardisées et implémentées dans l'interface. Ce sont principalement des commandes permettant l'échange d'informations entre les différents workers ainsi que des commandes permettant de modifier les paramètres du worker.
- Les commandes standardisées qui sont définies par le standard ; mais il revient à l'utilisateur de les implémenter. Ce sont des commandes qui ont pour vocation la gestion du worker et qui ne peuvent être systématiquement généralisées.

La deuxième structure matérialise le lien (canal de communication avec l'application principale). Cette structure réalise l'échange des commandes ainsi que l'implémentation des commandes standardisées. Toujours dans notre exemple Fortran (Src. 7.2), cette structure est représentée par le type `mti_shell_t`. Les fonctions associées à ce type permettent de recevoir les commandes provenant de l'application suivant le principe de question réponse. Le worker informe l'application principale par un message composé de son identifiant qui doit être unique, qu'il est disponible et prêt à recevoir l'instruction suivante. L'application principale peut traiter cette information sans équivoque pour envoyer la commande suivante.

Cette interface permet également les échanges entre les différents workers par l'échange de messages. Pour le moment, seulement des tableaux de réels peuvent être échangés.

```
1   !------------------------------ header -------------------------------------!
2   IMPLICIT NONE
3   PRIVATE ! all private
4
5   ! MTI Shell public interface
6   PUBLIC mti_shell_t
7   ! property and get-set
8   PUBLIC mti_shell_nameId
9   PUBLIC mti_shell_get_next_command
```

Code Listing 7.2 – Interface du module Fortran dédiée au worker

Nous pouvons tout à fait implémenter la convention d'échanges choisie dans d'autres langages. Ce qui permet de faire des hybridations avec des outils développés dans différents langages de programmation. Nous verrons plus loin un exemple de worker réalisé en Python.

Une fois cette interface définie, son implémentation en Fortran ou en Python peut être utilisée dans tous les workers. Nous ne sommes pas obligés de ré-implémenter à chaque fois cette interface d'échanges. Nous gagnons ainsi en temps mais surtout en fiabilité, car nous développons une seule fois, mais surtout nous mutualisons la correction de bugs.

7.3.2 Mise en œuvre d'un worker

Nous avons vu dans la section précédente deux des principaux composants utilisés pour réaliser un worker. Nous allons voir dans cette section comment nous les utilisons pour réaliser un worker. Le premier exemple en Fortran pour le code FDTD Alice, le deuxième en Python pour le code de ligne de transmission Milo.

Rappelons que le but du worker est d'exécuter à la réception d'une commande la fonction qui lui est associée. Il y a deux manières différentes d'aborder ce problème. Nous allons les aborder en nous appuyant sur deux exemples (Fortran et Python). De plus, dans les deux cas, nous devons faire une boucle sur les commandes reçues.

La première manière de faire est de réaliser un aiguillage. La première étape (lignes 12 à 21) consiste à créer un nouveau `shell` qui représente le worker définit par son identifiant et l'identifiant du manager. Dans cet exemple, par défaut, les commandes reçues ne sont pas affichées par le worker pour ne pas le ralentir (argument `print_command_=.FALSE.`). Puis nous faisons une boucle sur les commandes reçues (lignes 23 et 24 (Src. 7.3)). Ensuite nous réalisons l'aiguillage (structure de contrôle `SELECT CASE`) pour associer au nom de chaque commande la bonne action à réaliser. Enfin pour chaque commande (bloc `CASE`), nous devons interpréter les arguments de la commande puis nous réalisons l'action associée. Voyons par exemple, la commande `STOP` (lignes 51 à 54), qui est une commande standardisée et obligatoire car systématiquement appelée pour terminer le worker. Nous voyons en particulier qu'elle met la variable `finish` à faux pour terminer le worker et sortir de la boucle de réception des commandes.

Il est également possible comme pour la commande `INIT_STEP` de réaliser la commande directement dans le bloc (mise à jour du temps, des champs sur les faces et affichage de la progression). Cependant, il est préférable d'utiliser des fonctions. En effet, la plupart des actions réalisées sont des actions élémentaires de l'algorithme général (lignes 39 à 47). L'utilisation de fonction évite la recopie du code qui est toujours source de bugs. Bien entendu, nous pouvons avoir des actions qui ont des arguments en entrée comme la commande `GET_I_SLICE` ligne 56-64, qui a pour but de copier dans un tableau la valeur des courants sur un fil. Ces courants pourront par la suite être envoyés à un autre worker. Pour fonctionner, cette fonction a besoin de connaitre le numéro du fil, le segment du premier courant ainsi que le sens dans lequel nous allons relever les courants (vers la gauche ou la droite) et enfin, le nombre de courant relevé. Cette fonction est découpée en deux parties. La première partie est propre au worker. Elle a pour but d'allouer la mémoire d'une variable propre au worker

et de lire les arguments. Puis, la fonction de lecture des courants est appelée. Cette variable est associée au nom `I_SLICE` et pourra par exemple être envoyée par une commande `SEND` (lignes 68 à 83).

```
1   MODULE alice_main
2    USE alice_api
3    ! [...]
4
5  CONTAINS
6    ! [...]
7
8    SUBROUTINE run_shell(workerId, managerId)
9      USE mti_h
10     CHARACTER(len=*), INTENT(in) :: workerId, managerId
11
12     LOGICAL :: finish
13     ! shell command
14     TYPE(mti_shell_t), POINTER :: shell
15     TYPE(command_t) :: cmd
16
17     ! [...]
18
19     finish = .FALSE.
20     shell => new_mti_shell()
21     CALL init_mti_shell(shell, workerId, managerId, print_command_
           =.FALSE.)
22
23     DO WHILE(.NOT.finish)
24        cmd = mti_shell_get_next_command(shell)
25
26        SELECT CASE(command_name(cmd))
27        CASE('INIT_STEP')
28           it1 = it1 + 1
29           t=dt*(it1-1); tns=t*1.e9
30
31           ! remise a zero des champs E tangents aux faces
                 metalliques.
32           IF (nface /= 0) CALL rajE(ex,ey,ez,nface,face,source,t,
                 xe,ye, &
33                 ze,xh,yh,zh)
34
35           IF (MOD(it1,itfin/100).EQ.1) THEN
36              WRITE(*,"('[',I3,'%]')") it1*100/itfin
37           END IF
38
39        CASE('COMPUTE_H') ! Evaluation a t+dt/2.
40           CALL compute_H()
41        CASE('COMPUTE_I')
42           CALL compute_I_wires()
43
```

```
44          CASE('COMPUTE_Q') ! Evaluation a t+dt
45             CALL compute_Q_wires()
46          CASE('COMPUTE_E')
47             CALL compute_E()
48
49          ! [...]
50
51          CASE('STOP')
52             PRINT*, "stop alice done."
53             finish = .TRUE.
54             CALL clean_all()
55
56          CASE('GET_I_SLICE') ! (num_fil, first_seg, dir, nb_seg)
57             IF (ALLOCATED(var_i_fil_slice)) &
58                  DEALLOCATE(var_i_fil_slice)
59             ALLOCATE(var_i_fil_slice(command_get_arg_int(cmd, 4)))
60             CALL fil_get_slice_courant(fil,&
61                  num_fil=fil_get_num(fil, command_get_arg_int(cmd,
                        NUL_FIL_)),&
62                  num_seg=command_get_arg_int(cmd, NUL_SEG_),&
63                  i=var_i_fil_slice,&
64                  dir=command_get_arg_int(cmd, 3))
65
66          ! [...]
67
68          CASE(MTI_SEND_MSG_) ! (to, name, dim)
69             IF (command_len(cmd).GT.0) THEN
70                var_name = command_get_arg(cmd, MSG_NAME_)
71                IF (ALLOCATED(array)) &
72                     DEALLOCATE(array)
73
74                SELECT CASE(var_name)
75                CASE('I_SLICE')
76                   IF (ALLOCATED(array)) &
77                        DEALLOCATE(array)
78                   ALLOCATE(array(SIZE(var_i_fil_slice)))
79                   array = var_i_fil_slice
80
81                ! [...]
82                END SELECT
83             END IF
84
85             CALL mti_shell_send(shell, cmd, var_name, array)
86
87          CASE default
88             PRINT*, "*** Invalide commande '" // TRIM(command_name(
                   cmd))
89          END SELECT
90       END DO
91
```

```
92      CALL del_mti_shell(shell)
93    END SUBROUTINE run_shell
94  END MODULE alice_main
```

Code Listing 7.3 – Définition en Fortran d'un worker (Alice code FDTD 3D)

Bien que cette solution donne le comportement souhaité, cette approche n'est pas vraiment satisfaisante. Une des raisons réside dans l'utilisation de l'aiguillage lui-même dont la taille augmente à chaque extension du worker. Cela nuit à la lisibilité du programme mais surtout ne permet pas un découpage modulaire du worker. De plus, elle impose de reimplémenter pour chaque worker des parties comme la gestion des erreurs (lignes 87 à 89), l'envoi et la réception des messages entre worker ou simplement la gestion de la boucle de réception des commandes. Dans tous les cas, la réimplémentation d'une fonction nuit à la maintenance et aux évolutions futures du module de gestion de calcul hybride.

Une amélioration consisterait à associer à chaque traitement une fonction indépendante, un nom et les règles pour interpréter les arguments de la fonction. Nous avons ainsi le triplet suivant :

1. Nom de la fonction
2. Liste ordonnée des types pour convertir les arguments de la commande qui sont sous la forme d'une chaîne de caractères en la valeur binaire correspondante pour les passer en argument à la fonction.
3. Une référence vers la fonction à exécuter.

Puis, nous définissions une fonction généraliste qui prend en argument une liste du précédent triplet correspondant à l'ensemble des commandes fournies par le worker. Pour pouvoir faire des échanges de données entre les workers, nous introduisons deux fonctions appelées à la réception et à l'envoi d'une donnée (`set_attr`, `get_attr`). De plus, la fonction `clean_before_exit` optionnelle peut être introduite pour libérer la mémoire par exemple. Ces fonctions ont les rôles suivants :

- `clean_before_exit` est une fonction optionnelle qui est exécutée à la réception de la commande « STOP ». Le reste de l'implémentation est entièrement pris en charge par le module.
- `get_attr` est la fonction qui est appelée durant l'exécution de la commande d'envoi d'un message et a pour rôle de charger les données dans le tableau envoyé.
- `set_attr` est la fonction symétrique de `get_attr` pour la réception d'un message.

De cette manière, nous avons une totale indépendance entre l'implémentation du worker et les outils d'aide à la mise en place.

Dans l'exemple suivant, nous ne présentons pas comment nous avons implémenté les différents mécanismes présentés mais comment les utiliser pour définir un worker. Cet exemple est en Python pour le code de ligne de transmission.

Nous définissons la classe d'objet, ligne 1 (Src. 7.4) qui représente les workers de notre code de ligne. Cette classe hérite de deux mécanismes. Le premier (`PyWorker`) implémente les mécanismes internes du worker similaire à ceux présentés en Fortran (`comand_t` et `mit_shell_t`). Elle a également en charge la gestion des triplets définissant la commande. Nous verrons plus loin la définition de ce triplet. Le deuxième mécanisme hérité intitulé

`WithMsgExchange` a pour rôle d'implémenter une des solutions possible pour introduire le mécanisme d'échange de données (`set_attr`, `get_attr`).

Comme nous l'avons déjà évoqué, la classe `PyWorker` joue le même rôle que le type `mti_shell_t` utilisé dans l'exemple FORTRAN précédent, à ceci près, qu'elle implémente la définition du triplet (nom, type des arguments, fonction). Ainsi, une nouvelle commande du worker se définit à l'aide du décorateur de méthodes `@iscmd()`. Ce décorateur définit la méthode décorée comme une commande du même nom mais en lettre capitale. Ainsi par exemple, à la lignes 13-16, nous définissons la commande `INIT_STEP`. De plus, nous peuvons définir le type des arguments de la méthode dans ce décorateur. Par exemple à la ligne 28, nous définissons la commande `SET_OUTPUT` qui prend un seul argument de type string. Nous avons donc bien notre triplet :

- D'une fonction, celle qui est décorée par `@iscmd()`,
- D'un nom : défini par le nom de la fonction elle-même,
- Et d'une liste, ordonnée des types utilisés pour interpréter les arguments.

Ainsi à la réception de la commande `SET_OUTPUT('path/to/a/file')`, le worker (`PyWorker`) prend en charge l'interprétation de `path/to/a/file` en une chaîne de caractères et l'appel de la méthode `set_output` avec le bon argument. La définition du worker se limite donc à ce que réalise le worker et plus à comment il interagit avec le manager et les autres workers.

Très rapidement le décorateur `@iscmd_argv()` a le même rôle que `@iscmd()`, à ceci près, que le nombre d'argument est variable. Dans ce cas de figure, il revient au worker d'interpréter les arguments passés en chaîne de caractères en fonction du contexte.

```
1  class MiloWorker(PyWorker, WithMsgExchannge):
2      def __init__(self, nameid, manager_addr=None,
3                   print_comand=False):
4          super(MiloWorker, self).__init__(nameid, manager_addr,
5                                           print_comand)
6          self.output_file = None
7          self.output_format = 'ah5'
8          self.debug = False
9
10     #=================================================================
11     # My MTI worker interface
12     #=================================================================
13     @iscmd()
14     def init_step(self):
15         ""
16         self.network.simu._get_algo().timer.next()
17
18     @iscmd_argv()
19     def compute_i(self, *args):
20         "Update I value."
21         self.network.simu._get_algo().comput_half_time_i(*args)
22
23     @iscmd_argv()
24     def compute_v(self, *args):
25         "Update V Value."
26         self.network.simu._get_algo().comput_half_time_v(*args)
```

```
27
28      @iscmd(str)
29      def set_output(self, file_name):
30          "set the output file."
31          self.output_file = file_name
```

Code Listing 7.4 – Définition pythonique d'un worker (Milo code de ligne de transmission)

Ce worker se lance comme suit, où `nameid` désigne l'identifiant du worker et `manager_addr` celui du manager. Puis, nous lançons la méthode généraliste `run()` qui réalise la boucle sur les commandes reçues.

```
1 with MiloWorker(nameid, manager_addr) as milo:
2     milo.run()
```

Dans cet exemple, nous n'avons pas parlé des échanges de messages et nous ne voyons rien qui s'y rapporte. Dans l'amélioration proposée, nous avons défini deux fonctions `get_attr` et `set_attr` qui ont pour rôle de faire l'interface avec les fonctions d'envoi et de réception. Cependant, ces deux fonctions peuvent dépendre de la stratégie utilisée pour stocker les messages par exemple dans un attribut, dans un dictionnaire, etc . . .. La classe `WithMsgExchange` définit la manière dont sont stockés les messages et implémente les deux fonctions d'interface. Le worker Milo n'a plus qu'à utiliser cette interface.

Par exemple la commande `SEND(worker_id, 'output_file')` permet d'envoyer le contenu de l'attribut `output_file` défini ligne 6 au worker répondant à l'identifiant `worker_id`. L'utilisateur n'a rien d'autre à faire que de déclarer sous la forme d'attribut, les quantités qu'il veut pouvoir échanger.

La deuxième approche apporte une plus grande souplesse de par le découpage en fonctions, ainsi qu'une plus grande portabilité par le découpage modulaire non intrusif. En effet, du moment que l'interface des mécanismes internes du worker (échange de messages, procédure de fin, etc) sont conservés, leur implémentation peut changer ; nous avons une bonne compatibilité ascendante.

7.3.3 Définition de l'interface du worker

Nous avons vu dans la section précédente, comment nous définissons un worker qui réalise les commandes. Ce worker sera utilisé par un manager pour réaliser une application hybride. Cependant ce manager, pour pouvoir correctement interagir avec le worker, doit connaître son interface, mais également tous les mécanismes liés au contrôle de son environnement, par exemple fichier de configurations ou de données. Ces informations ne peuvent pas être mutualisées car propre à chaque worker.

L'exemple suivant (Src. 7.5) donne la forme de l'interface du worker Milo. Nous distinguons deux parties dans cette interface écrite en Python. La première partie, composée uniquement de la fonction `__init__`, elle a pour rôle d'initialiser le contexte du worker. Nous retrouvons notamment deux chemins de fichier (`network_file` et `output_file`) ainsi que deux arguments supplémentaires (`*args` et `**kw`). Sans rentrer dans le détail maintenant, les deux premiers arguments sont des attributs locaux à l'interface. Les deux autres arguments sont utilisés pour paramétrer l'exécution du worker et configurer son environnement.

Nous verrons plus en détail ces arguments par la suite. La deuxième partie (lignes 11 à 25 (Src. 7.5)) est la définition à proprement parlé de l'interface du worker. Remarquons la similitude (utilisation du décorateur `@iscmd`) entre la définition du worker (Src. 7.4) et la définition de son interface (Src. 7.5).

La question qui se pose est pourquoi ne pas directement utiliser la définition du worker pour définir son interface. Pour cela, il suffit de changer le rôle du décorateur `@iscmd` en fonction du contexte d'utilisation. Il y a deux inconvénients à cette solution. D'une part elle nuit à la modularité de l'interface et du worker et rend la lecture du code bien plus compliquée. Nous ne pourrons guère que faire la différence entre ce qui est du worker ou de son interface. De plus, si nous utilisons le worker comme interface, cela impose que l'interface et donc le manager, chargent le worker ainsi que toutes les dépendances ; ce qui peut être contraignant alors qu'à priori l'interface ne l'utilise pas. En revanche, il pourrait être intéressant d'étendre la notion d'interface de manière à éliminer ce doublon de codage. Par exemple, définir des mécanismes capables de déduire du worker son interface. Ou encore que le worker durant son initialisation (exécution à distance) fournisse au manager son interface. Cela pourrait être fait en ajoutant une commande interne spécifique.

```
1  class MiloInterface(IsSimulation):
2      def __init__(self, network_file, output_file, *args, **kw):
3          self.network_file = network_file
4          self.output_file = output_file
5          super(MiloInterface, self).__init__(*args, **kw)
6          self.printCommand = False
7
8      #================================================================
9      # Stable features
10     #================================================================
11     @iscmd()
12     def init_step(self):
13         pass
14
15     @iscmd()
16     def compute_i(self, includes=None, excludes=None):
17         pass
18
19     @iscmd()
20     def compute_v(self, includes=None, excludes=None):
21         pass
22
23     @iscmd()
24     def set_output(self, filename):
25         pass
26     # [...]
```

Code Listing 7.5 – Définition de l'interface d'un worker (Milo code de ligne de transmission)

Revenons sur la méthode `__init__` qui a pour rôle d'initialiser l'interface et de créer le contexte d'utilisation du worker. Pour bien comprendre le mécanisme d'initialisation d'un worker par son interface, nous allons étudier une fonction (Src. 7.6) qui construit une in-

terface pour le code Milo. Cette fonction est intéressante car elle reprend les principaux mécanismes.

```
def milo(module=module_milo,
         network_file=AttachedFile('milo.py'),
         output_file='results.h5',
         attached_files=None,
         runargs=['--nameid=:py:self.nameid::',
                  '--addr=:py:self.manager.nameid::'],
         **kw):
    """Build a new MiloInterface with default arguments.
    """
    # [...]
```

Code Listing 7.6 – Interface de la fonction d'initialisation de l'interface de Milo

Nous remarquons que tous les arguments sont optionnels car définis avec une valeur par défaut. Le rôle des différents arguments est donné ci-dessous.

1. `module` définit comment exécuter le worker. Ce composant réalise à peu près le rôle d'un fork.
2. `network_file` est exactement le même que celui vu en argument de la méthode `__init__`. Comme son nom l'indique, cette variable définit le fichier de description du réseau étudié. Nous constatons que le chemin vers ce fichier est encapsulé dans la notion de fichier attaché. Cette encapsulation a pour rôle d'informer le manager que ce fichier sera copié du contexte du manager au contexte du worker.
3. `output_file` définit le nom du fichier de résultat produit par le code. Le nom du fichier n'est pas encapsulé car il est créé par le worker directement dans son contexte.
4. `attached_files` permet de définir une liste de fichiers qui sera copiée dans le contexte du worker, des paramètres RLCG stockés dans un fichier indépendant par exemple.
5. `runargs` est la liste des arguments utilisés à l'exécution du module. Rappelons, qu'un worker a besoin de connaître son identifiant qui est fourni par le manager et l'adresse de ce manager pour pouvoir communiquer avec celui-ci. Ces deux informations sont utilisées pour réaliser la communication. Ces deux informations devront être connues du worker avant que toute connection avec le manager ne soit possible. Dans notre cas de figure, ces informations sont passées par les arguments de ligne de commande `--nameid` et `--addr`.
   Ces deux informations sont déterminées à l'exécution du manager. Nous devons donc évaluer leur valeur. Nous avons fait le choix de faire une substitution directement dans une chaîne de caractères. Nous remplaçons le bloc limité par `:py:` et `::` par la valeur retournée par l'évaluation du contenu du bloc. Les limiteurs ont été choisis pour ne pas entrer en collision avec la syntaxe Python.
6. `**kw` permet d'utiliser les options avancées de l'interface qui ne sont pas décrites ici.

7.3.4 Mise en œuvre d'une stratégie d'hybridation

Nous avons vu dans la section précédente, comment nous définissons un worker pour nos stratégies d'hybridation. Maintenant nous allons voir comment utiliser ces workers, plus

précisément leur interface, pour mettre en place nos stratégies d'hybridation. La stratégie proposée s'appuie sur la notion d'un manager qui va administrer et gérer l'exécution de chaque worker à travers un « driver ». Cette vision hérite de l'approche généralement employée pour faire une application multi-méthode avec MPI, dont le principe consiste à définir une fonction principale pour chaque méthode mise en jeu. Puis dans la fonction principale de l'application, un aiguillage est fait sur chaque méthode en fonction du numéro de processus. Cette approche, comme nous l'avons déjà évoquée, présente un certain nombre d'inconvénients. Dans cette section, nous allons présenter comment nous définissons un driver. La définition d'un manager est l'association d'un ou plusieurs drivers avec l'interface d'un worker. Nous verrons ici de manière très succincte la définition d'un manager. Des exemples plus complets sont présentés dans la suite (Sec. 7.4).

Un « driver » est un composant informatique qui doit associer à l'interface d'un worker la séquence d'instructions que va réaliser ce worker. Un driver est donc en quelque sorte une simple fonction qui réalise une succession d'appels de fonctions d'un worker particulier.

Nous avons choisi d'utiliser la notion de generator pour décrire cette liste de commandes. La syntaxe recommandée (Src. 7.7) est la suivante. Nous définissons une fonction qui prend un seul argument. Cet argument est l'instance d'une interface d'un worker sur laquelle vont se réaliser l'exécution des commandes. Cette fonction sera directement utilisée par le manager. Ainsi le manager, après avoir initialisé le worker, appellera cette fonction avec le bon worker en argument. Cette fonction retourne le generator à proprement parlé qui est aussi une fonction. Cette fonction ne prend aucun argument en paramètre. A l'intérieur du generator, nous distinguons deux types d'instructions, les instructions exécutées dans le generator `var = 3 + 2` par exemple et les commandes qui seront envoyées au worker `yield worker.cmd(var)`. Notons la présence du mot clé `yield` qui signifie que la valeur sera retournée dans notre cas au manager. Il est impératif que ce qui suit le mot clé `yield` soit des commandes valides, c'est-à-dire des méthodes décorées par `@iscmd()`.

```
1 def driver(worker):
2   def driver_():
3     var = 3 + 2
4     yield worker.cmd(var)
5   return driver_
```

Code Listing 7.7 – Exemple d'un driver

Nous pouvons paramétrer nos drivers comme le montre l'exemple (Src. 7.8). Pour réaliser ce paramétrage, nous ajoutons un nouveau niveau à notre encapsulation fonctionnelle. Cette fois nous définissons une fonction qui retourne un « driver » paramétré par l'argument « var ».

```
1 def build_driver(var):
2   def driver(worker):
3     def driver_():
4       var += (var % 2 - 1 ) * -1 # must be odd
5       yield worker.cmd(var)
6     return driver_
7   return driver
```

Code Listing 7.8 – Exemple d'un driver paramétré

Cette dernière forme est celle que j'ai la plus utilisée car elle est relativement simple à mettre en place comparée à un itérateur [4] par exemple. De plus, je trouve que les fonctions sont relativement lisibles. Nous faisons bien la distinction entre les commandes qui vont être envoyées au worker et les portions de code qui vont être exécutées dans le driver.

Nous allons maintenant présenter un outil permettant d'optimiser l'exécution d'une boucle fixe de commandes. Dans un grand nombre d'applications, nous réalisons une boucle de commandes fixes. Dans les applications temporelles, c'est la boucle temporelle. L'exemple (Src. 7.9) représente un driver simplifié d'un code FDTD. Dans cet exemple, nous allons envoyer au worker `nb` fois le couple (`eval_E()`, `eval_H()`) soit de manière générale un total de ('nombre de commandes dans la boucle' * 'nombre d'itérations') messages à envoyer. Ce qui peut considérablement ralentir l'exécution.

```
1 def build_driver_time_loops(nb):
2   def driver_time_loops(worker):
3     def driver_time_loops_():
4       for t in range(nb):
5         yield worker.eval_E()
6         yield worker.eval_H()
7     return driver_time_loops_
8   return driver_time_loops
```

Code Listing 7.9 – Exemple d'une boucle réalisée par le manager

Comme la boucle principale ne comporte pas de branchement ou d'évolution en fonction de l'itération, nous pouvons envoyer la séquence des commandes et réaliser la boucle sur le worker : nous parlerons de boucle délocalisée, implicitement sur le worker. Nous n'envoyons plus que ('nombre de commandes dans la boucle' + 3), le gain est immédiat. L'exemple (Src. 7.10) illustre son utilisation.

```
1 def build_driver_externe_loops(nb):
2   def driver_externe_loops(worker):
3     def driver_externe_loops_():
4       list_of_cmds = [worker.eval_E(), worker.eval_H()]
5       for cmd in delocated_range(list_of_cmds, nb):
6         yield cmd
7     return driver_externe_loops_
8   return driver_externe_loops
```

Code Listing 7.10 – Exemple d'une boucle réalisée par le worker

Cette approche permet un gain important en terme de temps de calcul. Bien entendu, cette approche se limite au cas de boucles où il n'y a pas de branchement ou d'évolution en fonction de l'itération.

4. En génie logiciel, l'itérateur est un patron de conception (design pattern) comportementale.
Un itérateur est un objet qui permet de parcourir tous les éléments contenus dans un autre objet, le plus souvent un conteneur (liste, arbre, etc). Un synonyme d'itérateur est curseur, notamment dans le contexte des bases de données.

7.4 Exemples d'utilisation du prototype

Dans cette section, nous allons voir un ensemble d'exemples d'utilisation du « manager ». L'objectif est de présenter les capacités offertes par le manager, mais également de présenter comment nous pouvons utiliser le même worker dans différents contextes. La réutilisation du worker permet de mutualiser le code, ce qui est toujours un gain en productivité.

Nous ferons cette présentation en deux parties. D'une part, nous présentons un ensemble de cas d'utilisation du « manager » montrant les différentes capacités offertes. Dans un deuxième temps, nous présentons deux ensembles de drivers qui correspondent aux deux stratégies d'hybridation proposées dans cette thèse.

7.4.1 Mise en œuvre du manager

Dans cette section, nous allons voir un ensemble de programmes utilisant le manager. Ces différents programmes sont écrits en Python et sont décomposés en deux grandes parties.

La première partie importe l'ensemble des modules utilisés dans le programme et la deuxième partie définit la stratégie d'hybridation. Nous distinguons trois types de modules importés (voir par exemple (Src. 7.11)) :

- les modules fournis par Python et sa communauté comme `os`, `math`, `numpy` ou `random`. Les noms de ces modules sont suffisamment explicites pour comprendre leurs rôles.
- le manager lui même (`mti`) et des utilitaires qui lui sont associés.
- les modules qui définissent les « interfaces » (Sec. 7.3.3 page 206) et les « drivers » (Sec. 7.3.4 page 208) prédéfinis pour les différents « workers » (Sec. 7.3.2 page 201) (outils de calcul) qui seront utilisés. Dans nos exemples, nous allons mettre en oeuvre `milo`, qui est le code de ligne en temporel et `pyalice` qui est l'interface de Alice code FDTD-3D de ONERA.

La deuxième partie du code réalise le programme à proprement parlé, qui peut être vu comme un scénario d'utilisation des différents workers. Ces scénarii peuvent être de diverses natures :

- ils peuvent simplement utiliser un seul worker,
- définir des paramètres globaux déterminés à l'exécution du code,
- construire des simulations paramétriques voire stochastique,
- faire collaborer entre eux plusieurs workers,
- etc.

Dans la suite de cette partie, nous allons aborder de manière progressive ces différents cas de mis en œuvre du prototype de manager. Nous présentons dans un premier temps les différents scénarii d'utilisation du manager avec une complexité croissante. Nous terminons par présenter un scénario d'hybridation pour réaliser une hybridation en découpage de domaine utilisée dans l'étude numérique de la base de lancement Soyouz. Puis dans un deuxième temps, nous présentons les deux jeux de drivers que nous avons utilisés pour réaliser les validations numériques des deux approches d'hybridation qui sont l'hybridation en découpage de domaines et l'hybridation champ /câbles.

a. Appel du code de ligne à travers le manager

Le premier exemple que nous allons présenter va simplement calculer le courant et la tension le long d'une ligne de transmission excitée par une source de Thévenin et chargée par une résistance. Cet exemple ne présente pas un grand intérêt scientifique ou technique. En revanche, il permet de présenter les premiers mécanismes du manager et des outils qui lui sont associés.

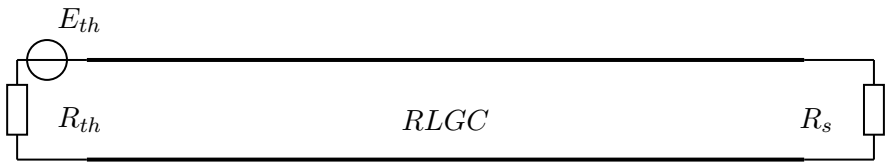


FIGURE 7.2 – Représentation du réseau utilisé.

La première partie de cette exemple, représente la description du réseau (Fig. 7.2). Dans ce fichier (Src. 7.11), nous commençons par définir des paramètres généraux de la simulation. Nous allons voir ces différents paramètres :

```
1  # Milo network file
2
3  simu.set_algo_opt('force', True)
4  simu.duration(end_time=3.8e-06, dt=5.6e-10)
5
6  E_th = phy.sources.DGauss(alpha=4.442883e+06, tau=7.554779e-07)
7  R_th = 339.375221742
8
9  tube_model = phy.CstPhyModel(n=1,
10                               R=0.98,     L=1.2e-06,
11                               C=1.05e-11, G=4.8e-08)
12
13 tube_geo = topo.new_tube_length(10, 6.0)
14 tube = topo.new_tube(tube_geo, tube_model)
15
16 left = topo.new_junction((tube.plug_at('left'),))
17 left.all_together(model=phy.Thevenin(1, [R_th], E_th))
18
19 right = topo.new_junction((tube.plug_at('right'),))
20 right.all_together(model=phy.Matched())
21
22 outputs.new_output_request('output_i', tube, 'i')
23 outputs.new_output_request('output_v', tube, 'v')
```

Code Listing 7.11 – Fichier de description du réseau pour Milo scripté en Python.

– ligne 3 nous demandons à Milo de passer outre tous les problèmes qu'il pourrait rencontrer, par exemple un pas de temps trop grand.
– ligne 4 nous définissons les paramètres temporels de la simulation.

– lignes 6 et 7 nous définissons les paramètres de notre source de Thévenin que nous utiliserons plus loin.
– lignes 9 à 11 nous définissons les paramètres de la ligne de transmission.
– ligne 13 et 14 nous définissons notre tube composé d'une géométrie (10 segments d'une longueur de 6 mètres) et d'un modèle RLCG.
– lignes 16 à 20 nous définissons la condition aux limites de notre problème ; une source de Thévenin pour celle de gauche et une adaptation pour la droite.
– lignes 22 et 23 nous faisons une demande de sortie en courant et tension.

Nous devons maintenant lancer le calcul sur ce réseau. Nous pourrions le faire directement en ligne de commande, où `network_1.py` est le réseau que nous venons de décrire et `output.h5` est le fichier où vont être stockés les résultats.

```
1 # milo -i network_1.py -o output.h5
```

Cependant pour l'exercice, nous allons utiliser le manager (Src. 7.12). Cet exemple va nous permettre de voir comment nous manipulons des fichiers à travers le manager, instancier un worker et lui associer un driver.

Comme nous l'avons déjà évoqué, nous importons les différents modules qui nous seront utiles. Puis, nous définissons une fonction `main` qui contiendra le corps du code.

À la ligne 16, nous créons un fichier attaché. Ce fichier est attaché au worker « Milo » ligne 17. Cette déclaration permet de copier le fichier `network_1.py` dans le contexte d'utilisation du code. Enfin nous définissons comment le worker sera utilisé. Dans notre cas, nous voulons réaliser une exécution « simple », dans ce cas nous pouvons utiliser le driver `build_driver_alone` fourni par la worker. Enfin (ligne 19), nous pouvons créer notre manager et l'exécuter.

```
1  """Build a simple simulation.
2  """
3
4  import os
5  # Import manager tools
6  from mti import AttachedFile, link_sim, Manager
7  # Import Milo simulation tools (mtl code)
8  from milo.scripts.mti import interface as milo_mti_interface
9
10 def main(nb_time):
11     """The simulation is executed exactly as if I would do in
12     command line.
13
14     $ milo -i input_file -o output.h5
15
16     """
17     input_files = (AttachedFile('network_1.py'),)
18     milo_sim = milo_mti_interface.milo(attached_files=input_files)
19     milo_sim.set_driver(milo_sim.build_driver_alone(nb_time))
20     Manager([milo_sim]).run()
21
22 if __name__ == '__main__':
23     main(nb_time = 300)
```

Code Listing 7.12 – Définition du manager qui appelle simplement le code de ligne

Les personnes vigilantes remarqueront que nous avons défini deux fois la durée de la simulation ; une fois dans le fichier réseau (Src. 7.11) et une fois dans le manager (Src. 7.12). Cette double définition vient du fait que les codes ont un existant en dehors du manager. Dans cette utilisation, la définition de la durée de la simulation peut être définie dans le fichier. Dans ce cas de figure, c'est le manager qui a la main sur l'exécution. L'introduction de la durée de la simulation introduit le problème de la gestion des paramètres globaux, c'est-à-dire des paramètres définis par le manager et qui sont souvent partagés par tous les workers d'une même simulation. Nous allons voir par la suite les simulations paramétriques qui permettent de passer outre ce problème.

b. Utilisation du manager pour faire une simulation paramétrique

Nous allons étendre l'exemple précédent en y ajoutant la notion de paramètre. Cette notion permet de définir la valeur numérique d'un paramètre par le manager à l'exécution. De plus, un paramètre peut être partagé entre plusieurs workers. Cette capacité est très intéressante par exemple pour avoir une valeur cohérente de dt dans toutes les simulations.

Nous allons commencer par modifier le fichier réseau vu précédemment pour y introduire des paramètres. Un paramètre est simplement introduit par des accolades {}. Par exemple à la ligne 5, `{dt}` sera remplacé par la valeur numérique du paramètre nommé 'dt'.

Il n'y a pas de limitation sur le nom de paramètres. C'est l'utilisateur qui peut définir les paramètres dont il a besoin (nb_steps, Zl, R, L, C, G). Ils doivent juste être définis au moment de l'exécution. Les paramètres peuvent se résumer à un ensemble de clés valeur défini par l'utilisateur.

```
1  # Milo network file with parameters
2  # A parameter are define by following syntax '{parameter_name}'
3
4  simu.set_algo_opt('force', True)
5  simu.time_scaling(dt={dt}, nb_time_step={nb_steps})
6
7  E_th = phy.sources.DGauss(alpha=4.442883e+06, tau=7.554779e-07)
8  R_th = {Zl}
9
10 tube_model = phy.CstPhyModel(n=1, R={R}, L={L}, C={C}, G={G})
11
12 tube_geo = topo.new_tube_length(10, 6.0)
13 tube = topo.new_tube(tube_geo, tube_model)
14
15 left = topo.new_junction((tube.plug_at('left'),))
16 left.all_together(model=phy.Thevenin(1, [R_th], E_th))
17
18 right = topo.new_junction((tube.plug_at('right'),))
19 right.all_together(model=phy.Matched())
20
21 outputs.new_output_request('output_i', tube, 'i')
```

```
22 outputs.new_output_request('output_v', tube, 'v')
```

Code Listing 7.13 – Fichier de description du réseau pour Milo scripté en Python avec des paramètres.

Nous allons maintenant modifier le fichier du manager pour définir la valeur des paramètres. Nous allons faire deux modifications principales. Premièrement ligne 12, nous définissons une fonction qui retourne la valeur de chaque paramètre utilisé. La fonction appelée ligne 39 permet de mettre toutes les variables locales dans un dictionnaire. Ce dictionnaire sera utilisé par le manager pour remplacer chaque paramètre par sa valeur. Deuxièmement, le fichier `network_params.py` est cette fois encapsulé dans une notion de fichier template. Ce qui informera le manager qu'en plus d'être attaché, il doit remplacer les paramètres par leur valeur numérique donnée en argument.

```
1  """Build a simple simulation.
2  """
3
4  import os
5  # for math operators
6  import math
7  # Import manager tools
8  from mti import AttachedFile, link_sim, Manager
9  # Import Milo simulation tools (mtl code)
10 from milo.scripts.mti import interface as milo_mti_interface
11
12 def build_parameters():
13     "Build parameters to fill milo file."
14     dt = 5.6e-10
15     nb_steps = 6
16
17     dl = 6.0
18
19     tl_center_len = 10
20     R= 0.98
21     L= 1.2e-06
22     C= 1.05e-11
23     G= 4.8e-08
24
25     Zl = math.sqrt(L/C)
26     return locals()
27
28 def main(nb_time):
29     """The simulation is executed with some parameters.
30     """
31     parameters = build_parameters()
32
33     input_files = (TemplateFile('network_params.py', parameters),)
34     milo_sim = milo_mti_interface.milo(attached_files=input_files)
35     milo_sim.set_driver(milo_sim.build_driver_alone(parameters['nb_time']))
36     Manager([milo_sim]).run()
```

```
37
38 if __name__ == '__main__':
39     main(nb_time = 300)
```

Code Listing 7.14 – Fichier de définition de la simulation avec génération de paramètres.

La paramétrisation est un outil très intéressant pour les études numériques. Dans la suivant, nous voulons faire un exemple de simulation paramétrique où nous étudions une ligne de tranmission au dessus d'un sol de conductivité finie. Nous cherchons entre autre à évaluer le rôle du rayon et de la hauteur de la ligne sur le comportement global. Pour faire cette étude, nous modifions la fonction (lignes 16 à 39) qui calcule les paramètres de manière à prendre en compte une ligne au-dessus d'un sol (ligne 23). De plus, nous voulons faire cette étude pour toutes les combinaisons possibles de rayons et de hauteurs $r \in [0.01, 0.02, 0.02]$ et $h \in [1.0, 1.5, 2.0]$. Nous faisons donc une boucle sur les éléments du produit au sens ensembliste (ligne 56).

Nous avons apporté une autre modification. Pour faire la différence entre les différents résultats, nous allons placer les résultats dans des répertoires différents. Pour modifier le repertoire de travail, nous utilisons l'argument `workingdir` du worker (ligne 47). Dans notre exemple, la valeur de cette variable est définie à la ligne 60. Ainsi, chaque résultat sera rangé dans le répertoire `workingdir/test_3/r{valeur_de_r}_h{valeur_de_h}`. Nous saurons sans ambiguïté à quel résultat correspond chaque répertoire.

```
1  """Build a simple simulation.
2  """
3
4  import os
5  # for math operators
6  import math
7  # Import manager tools
8  from mti import AttachedFile, link_sim, Manager
9  # Import Milo simulation tools (mtl code)
10 from milo.scripts.mti import interface as milo_mti_interface
11 # Tools to compute TL paramters
12 import milo.phy.model.by_geometry
13 # for complex loops
14 from itertools import product
15
16 def build_parameters(freq=1e6,
17                      h=5.0, r=0.03,
18                      rau=100., mu=1., eps=1.):
19     """Build parameters to fill milo file.
20
21     Build parameters for a tube above ground at given frequency.
22     """
23     from milo.phy.model.by_geometry import CableAboveGround_VanceF
24     TL = CableAboveGround_VanceF(r, h, rau, mu, eps)
25
26     omega=2.0*numpy.pi*freq
27
```

```
28      R = TL.get_R(omega)
29      L = TL.get_L(omega)
30      C = TL.get_C(omega)
31      G = TL.get_G(omega)
32      Zl = numpy.abs(TL.get_Z0(omega))
33
34      dt = 5.6e-10
35      nb_steps = 6
36
37      dl = 6.0
38      tl_center_len = 10
39
40      return locals()
41
42  def main(nb_time, r, h, working_dir):
43      """The simulation is executed with some parameters.
44      """
45      parameters = build_parameters(r=r, h=h)
46
47      input_files = (TemplateFile('network_params.py', parameters),)
48      milo_sim = milo_mti_interface.milo(workingdir=working_dir,
49                                         attached_files=input_files)
50      milo_sim.set_driver(milo_sim.build_driver_alone(parameters['nb_time']))
51      Manager([milo_sim]).run()
52
53  if __name__ == '__main__':
54      # run sim for RxH
55      R = (0.01, 0.02, 0.02)
56      H = (1.0, 1.5, 2.0)
57      for r, h in product(R, H):
58          # parameters are using to build simulation name
59          test_case_name = "workingdir/test_3/r%f_h%f" % (r, h)
60          main(nb_time=300, r=r, h=h,
61               working_dir=test_case_name)
```

Code Listing 7.15 – Fichier de définition de la simulation paramétrique.

De la même manière, il est tout à fait possible de faire une analyse de type Monte Carlo (Src. 7.16) en utilisant des outils de génération de nombre pseudo aléatoire (ligne 15 et 62).

```
1  """Build a simple simulation.
2  """
3
4  import os
5  # for math operators
6  import math
7  # Import manager tools
8  from mti import AttachedFile, link_sim, Manager
9  # Import Milo simulation tools (mtl code)
10 from milo.scripts.mti import interface as milo_mti_interface
11 # Tools to compute TL paramters
```

```
12 import milo.phy.model.by_geometry
13 # for complex loops
14 from itertools import product
15 # Generate pseudo-random numbers
16 import random
17
18 def build_parameters(freq=1e6,
19                      h=5.0, r=0.03,
20                      rau=100., mu=1., eps=1.):
21     """Build parameters to fill milo file.
22
23     Build parameters for a tube above ground at given frequency.
24     """
25     from milo.phy.model.by_geometry import CableAboveGround_VanceF
26     TL = CableAboveGround_VanceF(r, h, rau, mu, eps)
27
28     omega=2.0*numpy.pi*freq
29
30     R = TL.get_R(omega)
31     L = TL.get_L(omega)
32     C = TL.get_C(omega)
33     G = TL.get_G(omega)
34     Zl = numpy.abs(TL.get_Z0(omega))
35
36     dt = 5.6e-10
37     nb_steps = 6
38
39     dl = 6.0
40     tl_center_len = 10
41
42     return locals()
43
44 def main(nb_time, r, h, working_dir):
45     """The simulation is executed with some parameters.
46     """
47     parameters = build_parameters(r=r, h=h)
48
49     input_files = (TemplateFile('network_params.py', parameters),)
50     milo_sim = milo_mti_interface.milo(workingdir=working_dir,
51                                        attached_files=input_files)
52     milo_sim.set_driver(milo_sim.build_driver_alone(parameters['nb_time']))
53     Manager([milo_sim]).run()
54
55 if __name__ == '__main__':
56     # Monte Carlo method on (r,h)
57     nb_points = 500
58     r_mu, r_sigma = (0.01, 0.2)
59     h_mu, h_sigma = (2.0, 0.3)
60
61     i = 0
```

```
62      while (i < nb_points):
63          r = random.gauss(r_mu, r_sigma)
64          h = random.gauss(h_mu, h_sigma)
65          if (h <= 0 or h <= r):
66              # break no physique value
67              break
68          i += 1
69          test_case_name = "workingdir/test_3/%d_r_%f_h%f" % (i, r, h)
70          main(nb_time=300, r=r, h=h, working_dir=test_case_name)
71
72      # post process result here.
```

Code Listing 7.16 – Fichier de définition de la simulation utilisant une méthode de Monte Carlo.

c. Utilisation du manager pour faire une simulation hybridée

Dans cette dernière partie consacrée à la mise en œuvre du manager, nous allons présenter la mise en place d'une hybridation de domaines. La configuration présentée est tout à fait analogue à ce que nous avons fait dans les différentes validations numériques.

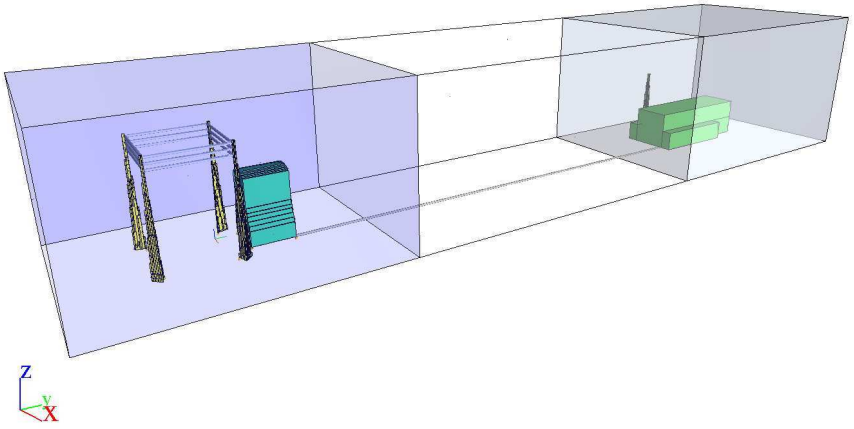


FIGURE 7.3 – Représentation des différentes zones de l'hybridation 3D-1D-3D de l'ensemble de lancement Soyouz.

Le scénario considéré vise à étudier une partie de la base Soyouz par une approche hybride en découpage de domaines (Fig. 7.3). Dans cette approche, il y a deux domaines 3D, l'un correspond au lanceur l'autre à la zone d'assemblage. Entre ces deux bâtiments, il y a une zone plus « simple » que nous modélisons par une ligne de transmission. Le scénario de simulation est donné par le code source (Src. 7.16).

Nous retrouvons des éléments similaires dans les différents cas déjà présentés. Notons que dans ce cas de figure, la fonction `build_parameters` (ligne 13), qui a pour but de définir tous les paramètres de la simulation, est définie dans le fichier `params.py` (ligne 11). La fonction `build_parameters` est appelée avec trois arguments. Le premier `genetype='biexpo'` indique que nous voulons une source de type biexponentiel. Le deuxième argument est `dl=1.0`, cette information permet de déterminer le pas de temps qui doit être dans notre cas global à tous les workers. Le dernier paramètre `metalsol=True` indique que nous sommes dans une configuration où le sol est considéré comme parfaitement conducteur. Cette information nous permet de choisir les conditions aux limites dans les domaines 3D et de calculer les paramètres RLCG de la ligne de transmission. Nous n'entrerons pas plus dans le détail de cette fonction.

Ensuite nous définissons le répertoire de travail. Dans notre cas de figure, les résultats de la simulation seront rangés dans `"simulation/%Y-%m-%d/"` où `%Y` sera par l'année courante, `\%m` par le mois courant et `%d` par le jour (year month day). Nous avons déjà utilisé ce type de capacité pour différencier les résultats d'une simulation paramétrique. Dans ce cas de figure, les résultats sont rangés dans des répertoires triés par date, ce qui permet de ne pas écraser un résultat par inadvertance.

Dans la suite nous définissons trois workers, un pour chaque domaine. La définition des workers ne change pas. Nous définissons un répertoire de travail. Notons que dans notre cas, nous avons choisi d'utiliser un répertoire différent pour chaque domaine (workingdir plus left, right ou network, ligne 24, 30 et 34). Puis nous définissons les fichiers associés à chaque domaine (lignes 25-28, 31-32, 35-37). Ces fichiers correspondent aux maillages, aux réseaux et autres.

De la ligne 39 à 41, nous faisons une étape spécifique aux simulations hybrides. Cette étape consiste à informer le manager que deux workers vont échanger des données. Après cette déclaration, c'est le manager qui va prendre en charge toutes les étapes de création, d'administration et de destruction des canaux de communication.

```
1  #! /usr/bin/env simanager
2  #
3  # Builds hybrid simulation topologie
4  #
5  # +----------+    +----------+    +--------------+
6  # | lanceur *----* network *----* assemblage |
7  # +----------+    +----------+    +--------------+
8  #   (left)                          (right)
9  #
10
11 from params import build_parameters
12
13 parameters = build_parameters(genetype='biexpo', dl=1.0,
14                                metalsol=True)
15
16 workingdir = "simulation/%Y-%m-%d/"
17
18 # import some simulations drivers
19 from sym import mti_drivers
```

```
20
21
22 # Build all client
23 left = ext.alice.alice(workingdir = workingdir + 'left',
24                        attached_files = [
25         TemplateFile('left/fort.1', parameters),
26         TemplateFile('left/fort.2', parameters)])
27
28
29 network = ext.milo.milo(workingdir=workingdir + 'network',
30       network_file = TemplateFile('network.py',
31                                                   parameters))
32
33 right = ext.alice.alice(workingdir = workingdir + 'right',
34                        attached_files = [
35         TemplateFile('right/fort.1', parameters),
36         TemplateFile('right/fort.2', parameters)])
37
38 # link client togather
39 link_sim(left, network)
40 link_sim(right, network)
41
42 # build and set drivers for each clients
43 overlaps = parameters['overlaps']
44 nb_time_step = parameters['nb_time_step']
45 left.set_driver(mti_drivers.build_alice_driver\
46                     (nb_time_step,
47                      [(network, 2, 1)],
48                      overlaps=overlaps))
49
50 network.set_driver(mti_drivers.build_milo_driver\
51                        (nb_time_step,
52                         ebounds=[(left,  'ligne_1', 'left'),
53                                  (right,  'ligne_1', 'right')],
54                         overlaps=overlaps))
55
56 right.set_driver(mti_drivers.build_alice_driver\
57                     (nb_time_step,
58                      [(network, 3, -1)],
59                      overlaps=overlaps))
60
61 # Run simulation
62 Manager([left, network, right]).run()
```

Code Listing 7.17 – Source du manager pour une hybridation hybride en découpage de domaine

L'avant dernière étape consiste à spécifier les drivers pour chaque worker. Cette étape est similaire aux cas précédents. Nous allons cependant nous attarder sur les arguments passés au constructeur de drivers. Ces arguments sont importants car propres aux drivers et à la

stratégie d'hybridation.

De la ligne 46 à 48, nous définissons le driver pour la zone de gauche (lanceur). Le premier paramètre est le nombre d'itérations. Notons que nous retrouvons cette information pour tous les drivers, ce qui est normal. Le deuxième argument définit la liste des lignes hybridées (ligne 46). Dans notre cas, nous avons `(network, 2, 1)`, ce qui signifie que l'extrémité droite (1) du fil numéro 2 est connectée avec le réseau worker 'network'. Pour l'autre domaine 3D (ligne 67), c'est le fil numéro 3 qui est hybridé avec le réseau mais par son extrémité gauche (-1). Le dernier paramètre permet de définir le nombre de segments sur lequel nous allons faire le recouvrement. Comme pour le nombre d'itérations, ce paramètre est global à tous les workers.

Si nous considérons maintenant le driver pour le code de ligne (lignes 49-53), nous retrouvons les paramètres globaux (nombre d'itérations et recouvrements). Mais contrairement aux drivers pour les domaines 3D, ici nous avons deux hybridations sur la même ligne de transmission ('ligne_1') : une hybridation a son extrémité gauche avec le domaine de droite et une hybridation à son extrémité droite avec le domaine de gauche. Nous avons ainsi une hybridation bidirectionnelle 3D-1D-3D.

La dernière étape consiste toujours à exécuter tous les workers à travers le manager.

7.4.2 Exemples de drivers

Dans la partie précédente, nous avons présenté différents scénarii d'utilisation du manager. Dans la suite, nous allons nous pencher sur la définition des drivers. Nous présenterons les drivers permettant d'hybrider Alice le code FDTD-3D de ONERA et Milo le code de ligne FDTD-MTL développés durant ces travaux. Les deux configurations présentées sont d'une part les drivers pour mettre en place la stratégie d'hybridation en découpage de domaines (Sec. 4.3 page 134) et d'autre part les drivers pour mettre en place la stratégie d'hybridation champ câbles (Sec. 3 page 99). Dans les deux cas de figure, ces deux stratégies d'hybridation sont construites suivant le principe coopérant (Sec. 4.2.2 page 128) qui permet d'avoir un algorithme symétrique et légèrement plus performant.

Nous commençons par présenter les drivers qui permettent de mettre en place la stratégie d'hybridation en découpage de domaines. Ces drivers sont ceux que nous venons d'utiliser dans le dernier exemple d'utilisation du manager. Dans un deuxième temps, nous présenterons les drivers utilisés pour mettre en place la stratégie d'hybridation champ / câbles.

a. Drivers pour l'hybridation en découpage de domaines

Nous étudierons en premier lieu les deux drivers permettant de faire une hybridation en découpage de domaines. Rappelons que le principe de cette hybridation consiste à échanger les courants en bout de ligne avec un autre code. Dans le cas où nous faisons une hybridation du 3D vers le 1D nous ne pouvons échanger que le courant terminal. En revanche dans le sens 1D vers 3D nous devons échanger plusieurs courants de manière à expliciter le champ électromagnétique se propageant le long de la ligne.

Le premier driver (Src. 7.18) définit le comportement du code de ligne. Le deuxième driver (Src. 7.19) définit le comportement du code 3D. Ces deux drivers sont construits suivant le même prototype de fonction `driver_name(nb_steps, ebound, overlaps)`. Où

`nb_steps` définit le nombre d'itérations temporelles, `ebound` permet d'identifier l'extrémité de la ligne sur laquelle nous faisons une hybridation. La définition de l'extrémité diffère en fonction de la méthode. Le dernier paramètre est l'`overlaps`, définit la longueur (nombre de segments) sur laquelle nous allons imposer les courants dans le 3D.

Le premier driver est consacré au code de lignes de transmission (lignes 29-51). Nous retrouvons les étapes principales d'une procédure FDTD :

```
1  #! /usr/bin/env python
2  # -*- coding: utf-8 -*-
3  """Define two sets of driver to couple together Alice 3D FDTD and
4  Milo MTL method.
5  """
6
7  import numpy as np
8
9
10 def build_milo_driver(nb_steps, ebounds, overlaps):
11     """Return driver for milo simulation connected to 3D method at
12     some boundaries.
13
14     INPUTS:
15      nb_steps: The number of time steps
16      ebounds: The list of 3D connected boundary
17         [(from_sim, tube_id, boundary_id),]
18
19        - from_sim: co simulation instance
20        - tube_id: tube name id
21        - boundary_id: 'left' or 'right'
22
23      overlaps: The number of segment overlaps
24
25     """
26     def build_milo_driver_(client):
27         def build_milo_driver__():
28
29             main_steps = []
30             main_steps.append(client.init_step())
31             main_steps.append(client.compute_i())
32
33             for to_sim, tube_id, boundary_id in ebounds:
34                 main_steps.append(client.assign_val('I_SLICE',
35                                                     tube_id,
36                                                     boundary_id,
37                                                     overlaps))
38                 main_steps.append(client.send_msg(to_sim,
39                                                   'I_SLICE',
40                                                   overlaps))
41
42             main_steps.append(client.compute_v())
43
```

```
44              for from_sim, tube_id, boundary_id in ebounds:
45                  main_steps.append(client.recv_msg(from_sim, 'I', 1))
46                  main_steps.append(client.compute_v_boun(tube_id,
47                                                          boundary_id))
48
49              for cmd in client._iter_default_loops_(nb_steps,
50                                                     main_steps):
51                  yield cmd
52
53          return build_hilo_driver__
54      return build_hilo_driver_
```

Code Listing 7.18 – Drivers Milo pour la stratégie d'hybridation en découpage de domaines

- `init_step` ligne 30
- `compute_i` ligne 31
- `compute_v` ligne 42

Entre ces lignes sont intercalées deux boucles sur les conditions aux limites, qui sont liées à un autre code. La première boucle (ligne 33 à 40) permet de lire et d'envoyer les courants calculés par le code de lignes. Ces courants sont relevés sur la ligne `tube_id` à l'extrémité `boundary_id` sur une longueur `overlaps`. Les courants relevés sont sauvegardés dans une variable intermédiaire `I_SLICE`. Par la suite (ligne 39) nous envoyons le contenu de cette variable au worker `to_sim`. La troisième valeur de la fonction `send_msg` indique la taille du message, dans ce cas c'est bien l'overlaps.

De la ligne 44 à 47 nous faisons la deuxième boucle qui permet de recevoir le courant qui provient de l'autre worker `from_sim`. Ce courant est stocké dans une variable intermédiaire `I`. Dans un deuxième temps, ce courant est utilisé pour calculer la tension à l'extrémité `boundary_id` de la ligne `tube_id`.

Notons qu'il n'y a pas de contrainte sur le worker qui va recevoir et envoyer les messages. Il peut être un code FDTD-3D ou tout autre chose.

Pour terminer, les lignes 49 et 51 permettent d'exécuter les instructions directement sur le worker en utilisant la notion de boucle distante (Sec. 7.3.4 page 210). Ce mécanisme est géré par la fonction `_iter_default_loops_`. De plus, cette fonction prend également en charge les étapes d'initialisation et de fermeture de Milo sans que l'utilisateur ne s'en préocupe.

Le deuxième driver est consacré à la méthode FDTD-3D du worker Alice. Nous retrouvons des commandes similaires à celles utilisées pour le code de lignes mais adaptées à la méthode FDTD-3D (`compute_h` ligne 82 et `compute_e` ligne 108). Nous retrouvons également les commandes permettant de gérer les fils de R. Holland (`compute_i` ligne 83 et `compute_q` ligne 107).

En ce qui concerne les mécanismes d'échange des courants aux extrémités (lignes 85-94 et 96-105), nous avons un algorithme symétrique à celui vu précédemment. Nous relevons le courant sur le fil `wire_id` à l'extrémité `end_id` et nous l'envoyons. Rappelons que pour l'hybridation du 3D vers le 1D, nous avons besoin que du courant terminal. Ce qui explique que la taille du message envoyé est de 1. Dans la deuxième partie, nous recevons les courants puis nous les imposons sur le fil.

```
56 def dipole_by_end(wire_id, end_id, overlaps):
57     "Build right value for alice"
58     return wire_id, -end_id * overlaps, end_id
59
60 def build_alice_driver(nb_steps, ebound, overlaps):
61     """Return driver for Alice simulation connected to MTL method at
62     some wires.
63
64     INPUTS:
65      nb_steps: The number of time steps
66      ebounds: The list of MTL connected boundary
67       [(wire_id, seg_id, dir, milo),]
68
69        - wire_id: The wire id connected
70        - end_id:  The segment number on the wire connected.
71        - milo: co simulation instance
72
73      overlaps: The number of segment overlaps
74
75     """
76     def build_alice_driver_(alice):
77         def build_alice_driver__():
78
79             main_steps = []
80             main_steps.append(alice.init_step())
81             main_steps.append(alice.write_temp())
82             main_steps.append(alice.compute_h())
83             main_steps.append(alice.compute_i())
84
85             for milo, wire_id, end_id in ebound:
86                 wire_id, seg_id, dir = dipole_by_end(wire_id,
87                                                      end_id,
88                                                      overlaps)
89                 main_steps.append(alice.get_i_slice(wire_id,
90                                                     seg_id + np.sign(seg_id),
91                                                     1, dir))
92                 main_steps.append(alice.send_msg(milo,
93                                                  'I_SLICE',
94                                                  1))
95
96             for milo, wire_id, end_id in ebound:
97                 wire_id, seg_id, dir = dipole_by_end(wire_id,
98                                                      end_id,
99                                                      overlaps)
100                main_steps.append(alice.recv_msg(milo,
101                                                 'I_SLICE',
102                                                 overlaps))
103                main_steps.append(llice.set_i_slice(wire_id,
104                                                    seg_id,
105                                                    dir))
```

```
106
107             main_steps.append(alice.compute_q())
108             main_steps.append(alice.compute_e())
109             main_steps.append(alice.update_sources())
110
111             # run the main loops
112             for cmd in alice.main_loops(nb_steps, main_steps)
113                 yield cmd
114
115         return build_alice_driver__
116     return build_alice_driver_
```

Code Listing 7.19 – Drivers Alice pour la stratégie d'hybridation en découpage de domaines

Ces deux drivers sont construits suivant le principe coopérant car les deux boucles sont parfaitement symétriques. Ces deux boucles sont la traduction aux deux méthodes du même algorithme.

b. Drivers pour l'hybridation champ / câbles

Nous venons de voir les deux drivers permettant de mettre en place la stratégie d'hybridation en découpage de domaines. Avant de conclure, nous allons présenter les deux drivers qui permettent de mettre en place la stratégie d'hybridation champ /câbles. Cette présentation, hormis le fait qu'elle montre concrètement comment nous mettons en place cette stratégie, permet également d'illustrer la polyvalence du manager.

Rappelons que le principe de cette approche est de calculer le champ électrique incident le long d'un chemin dans le 3D. Puis nous utilisons ce champ incident dans la théorie de lignes de transmission pour mettre à jour le courant. Enfin ce courant est utilisé dans le code 3D pour mettre à jour le champ. Le principe est algorithmiquement parlant très proche du précédent. Il diffère en seulement deux points. D'une part dans le 3D nous ne calculons plus les courants et les charges mais seulement le champ incident. Deuxièmement, les échanges entre workers ne se font pas seulement sur une partie du fil mais sur toute la longueur.

Ainsi ligne 157 (Src. 7.20) au lieu de recevoir le courant de bout de ligne, nous recevons le champ incident sur toute la longueur de la ligne. Et ligne 168 au lieu de relever le courant au bout du fil, nous relevons le champ électrique le long du chemin de ce fil. En dehors de ces deux différences, les deux drivers sont très proches des deux précédents.

```
124 def load_bundle(t):
125     "Load bundle t and evaluate default arguments."
126     return_current = None
127     if len(t) == 3:
128         tube_name, tube_len, sim_3d = t
129         way = 'direct'
130     elif len(t) == 4:
131         tube_name, tube_len, sim_3d, way = t
132     elif len(t) == 5:
133         return t
134     else:
135         raise Exception()
```

```
136     return tube_name, tube_len, sim_3d, way, return_current
137
138 def build_milo_driver_with_bundle(nb_steps, bundles):
139     """Return driver for Milo simulation for compute thin wire
140     with MTL method.
141
142     INPUTS:
143      nb_step: The number of time steps
144      bundles: [(tube_name, tube_len, 3d_sim, [way 1.0 or -1.0])]
145
146     """
147     def build_milo_driver_(client):
148         # List of command using in one time step
149         def build_milo_driver__():
150
151             main_steps = []
152             main_steps.append(client.init_step())
153
154             for bundle in bundles:
155                 tube_name, tube_len, sim_3d, way, return_current \
156                     = load_bundle(bundle)
157                 main_steps.append(client.recv_msg(sim_3d,
158                                                   'E',
159                                                   tube_len))
160                 main_steps.append(client.set_tube_e_inc(tube_name,
161                                                         way))
162
163             main_steps.append(client.compute_i())
164
165             for bundle in bundles:
166                 tube_name, tube_len, sim_3d, way, return_current \
167                     = load_bundle(bundle)
168                 main_steps.append(client.assign_val('I_EQUIVALENT',
169                                                     tube_name,
170                                                     way))
171                 main_steps.append(client.send_msg(sim_3d,
172                                                   'I_EQUIVALENT',
173                                                   tube_len))
174
175             main_steps.append(client.compute_v())
176
177             for cmd in client._iter_default_loops_(nb_steps,
178                                                    main_steps):
179                 yield cmd
180
181         return build_milo_driver__
182     return build_milo_driver_
183
184
185
```

```
186
187 def build_alice_driver_with_bundle_light(nb_steps, bundles):
188     """Return driver for Alice simulation where the wire are
189     comuted by MTL method.
190
191     INPUTS:
192      nb_steps: the numbers of time steps
193      bundles: The list of bundles define by the wire id them length
194
195     """
196     def build_alice_driver_(alice):
197         def build_alice_driver__():
198             main_steps = []
199             main_steps.append(alice.init_step())
200             main_steps.append(alice.write_temp())
201             main_steps.append(alice.compute_h())
202
203             # Evaluate the incident E field along wire.
204             # This value are evaluated and stored at I location.
205             main_steps.append(alice.compute_i_inc('e'))
206             # For each bundles get the incident E field
207             # and send it to Milo client
208             for wire_id, wire_len, milo_client in bundles:
209                 main_steps.append(alice.get_i_slice(wire_id, 1,
210                                                     wire_len, 1))
211                 main_steps.append(alice.send_msg(milo_client,
212                                                  'I_SLICE',
213                                                  wire_len))
214
215             # For each bundles receive the common currant
216             # and set it into the wire path.
217             for wire_id, wire_len, milo_client in bundles:
218                 main_steps.append(alice.recv_msg(milo_client,
219                                                  'I_SLICE',
220                                                  wire_len))
221                 main_steps.append(alice.set_i_slice(wire_id,
222                                                     1, dir=1))
223
224             main_steps.append(alice.compute_e())
225             main_steps.append(alice.update_sources())
226
227             # run the main loops
228             for cmd in alice.main_loops(nb_steps, main_steps,
229                                         **main_loops_kw)
230                 yield cmd
231
232         return build_alice_driver__
233     return build_alice_driver_
```

Code Listing 7.20 – Drivers pour la stratégie d'hybridation champ câbles

Cet exemple montre tout l'avantage d'une programmation orientée objet (type plus fonctions), interface dans un worker. Cette utilisation permet de mettre en place très simplement de nombreux scénarii d'utilisation et stratégies d'hybridation.

Le manager et l'ensemble des outils qui ont été mis en place dans ce prototype permettent un véritable gain en productivité. Le gain de temps est fait sur deux plans. D'une part, nous passons moins de temps dans des problématiques informatiques. D'autre part, le fait de pouvoir scripter le manager en Python offre de très grande capacité de paramétrisation de la simulation. Pour terminer, le gain en productivité apporté par le manager et Python ne se fait pas au détriment du temps de calcul, car le temps consommé dans le manager et Python reste très faible par rapport au temps passé dans les méthodes numériques.

7.5 Conclusion

Dans cette section, nous avons présenté un prototype de manager permettant de faire interagir plusieurs composants logiciels différents avec un minimum d'intrusivité. Ce prototype nous a permis de clarifier le besoin pour une hybridation forte entre méthodes temporelles. Cette étude nous a conduit à repenser l'architecture logicielle qui nous a éloignée des architectures classiques basées sur MPI pour nous rapprocher de solutions plus flexibles mais abordables.

La mise en place de manager a permis de définir une approche peu intrusive qui se limite à la fonction principale du programme. De plus, l'utilisation de langage optimisé pour les workers dans un processus indépendant permet d'avoir une exécution performante et naturellement parallèle. En dernier lieu, l'utilisation d'un langage haut niveau comme Python pour définir le manager permet une grande souplesse de mise en œuvre pour faire de simples études paramétriques et des hybridations multi-méthodes.

Conclusion générale et perspectives

Dans la première partie de ce mémoire, nous avons présenté certains aspects généraux sur l'étude numérique des grands sites terrestres. La taille et la complexité des systèmes industriels montrent les limites de la modélisation numérique. Pour pallier cette difficulté, l'utilisation d'une approche hybride dans le domaine temporel FDTD-3D/MTL-1D a été envisagé. Pour mettre en place cette stratégie d'hybridation, plusieurs points ont du être exploré.

Nous avons, dans un premier temps, présenté la modélisation des lignes de transmission, puis étudié et donné un schéma différences finies dans le domaine temporel pour résoudre celle-ci. Dans l'objectif de faire une hybridation en découpage de domaines pour l'étude d'un grand site terrestre, nous avons aussi cherché à modéliser les lignes de transmission au-dessus d'un sol de conductivité finie. Cette modélisation nous a conduit à utiliser une technique d'optimisation globale pour faire une décomposition en une série de Debye. Ce choix a été fait dans le but de trouver une procédure capable d'obtenir, en un temps raisonnable, une série de Debye qui soit la plus proche de la fonction recherchée et qui permette également de conserver la stabilité du schéma FDTD. L'approche proposée donne satisfaction, mais pourrait être améliorée sur deux points. Le premier point consiste sur la mise en place d'une heuristique d'optimisation globale incorporant le nombre de pôles dans la recherche. Le deuxième point concerne l'utilisation d'un critère de stabilité, moins restrictif que la simple extension du cas à un pôle qui reste la seule configuration où nous avons une démonstration de la stabilité.

Nous avons ensuite présenté deux approches d'hybridation de ligne de transmission avec une méthode 3D. La première approche proposée se consacre à la modélisation de câbles complexes qui sont immergés dans le domaine 3D. Cette approche prend tout son sens dans un contexte industriel, où le nombre de câbles ne cesse de croître et où la maîtrise de la problématique CEM est parfois cruciale. La deuxième approche permet une hybridation en découpage de domaines. Cette approche a pour but d'exploiter la méthode numérique la plus « adaptée » pour chaque sous-domaine. Le choix peut se faire sur un critère de précision du résultat numérique et/ou de performances en terme de vitesse de calcul.

L'hybridation d'échelle a pour but la modélisation de structure linéique complexe, petite par rapport à la longueur d'onde et par rapport à la discrétisation numérique. Cette approche poursuit les travaux initiés par R. Holland et J.P. Berenger sur la modélisation des fils

minces. L'approche proposée modélise les structures complexes par une ligne de transmission dont les paramètres sont calculés par la résolution d'une équation de Laplace. Ensuite, les deux systèmes sont couplés par le principe de Agrawal sur le mode commun. Cette approximation est très largement suffisante. En définitive, l'hybridation champ/câbles a démontré numériquement sa validité et sa souplesse de mise en œuvre. Cette approche pourra être étendue aux fils obliques de C. Guiffaut pour ne plus être limitée au seul cas où les câbles sont parallèles aux axes de la grille.

L'hybridation multi-domaine a été envisagée suivant différents points de vue (continuité simple des inconnues aux extrémités). Toutes ces approches tentent de donner une approximation au principe fondamental de la continuité du champ électromagnétique dans le plan d'interface entre les deux domaines 1D et 3D. Dans le cas d'une ligne de transmission, ce principe se traduit en la continuité du mode de ligne TEM. Pour garantir la continuité de ce mode, du domaine 3D vers le domaine 1D, il suffit d'échanger les quantités i et q. En revanche, pour garantir cette continuité du domaine 1D au domaine 3D, nous devons expliciter le champ électromagnétique implicite dans la théorie des lignes de transmission. Pour calculer ce champ électromagnétique, plusieurs approches ont été envisagées. La première consiste à utiliser un formalisme de Green. Cette approche permet d'avoir un formalisme exact du champ dans le plan d'hybridation. Cependant, cette approche devient très complexe, dès lors que le problème sort des cas canoniques. C'est pourquoi, nous utilisons une méthode 3D pour calculer le champ dans le plan d'hybridation. Le principe consiste à relever les courants obtenus par la méthode de lignes de transmission et à les faire rayonner dans le domaine 3D en les appliquant sur la structure de la ligne de transmission. Cette approche a démontré, sur plusieurs applications numériques, sa validité. Cependant, elle reste complexe à mettre en œuvre, surtout si nous devons/voulons modéliser toutes les interactions/couplages entre les différents éléments du système.

La dernière partie de cette thèse est consacrée aux aspects informatiques. Bien que l'informatique ne soit pas une finalité en soi, elle revêt un caractère particulier pour notre discipline. Il est donc utile de regarder ce qui se fait dans les autres branches de l'informatique et en particulier sur les méthodologies de développement qui nous aident à mieux gérer notre travail informatique. Toutefois, cette partie est surtout consacrée à l'étude, sur un plan informatique, de l'hybridation forte. Nous avons pu ainsi développer un prototype de « manager » limitant l'intrusivité dans les codes de calcul et permettant une vision boite noire, dans laquelle chaque code mis en jeu dans une stratégie d'hybridation, est indépendant des autres. Ce prototype a donné satisfaction pour les besoins de ces travaux de thèse ; cependant, des évolutions et améliorations sont possibles pour offrir plus de souplesse et pourquoi pas une optimisation dynamique de la parallélisation.

Annexe A

Équations de Maxwell

A.1 Formulation différentielle des équations de Maxwell

Les phénomènes électromagnétiques en tout point $\boldsymbol{M}$ d'un espace donné sont déterminés par les densités de charges de courant ainsi que les champs électromagnétiques. Le champ électromagnétique en un point $\boldsymbol{M}$ de l'espace et à un instant t est constitué des quatre champs vectoriels suivants :

- un champ électrique $\boldsymbol{E}(\boldsymbol{M},t)$ en V/m et une induction électrique $\boldsymbol{D}(\boldsymbol{M},t)$ en C/m^2.
- un champ magnétique $\boldsymbol{H}(\boldsymbol{M},t)$ en A/m et une induction magnétique $\boldsymbol{B}(\boldsymbol{M},t)$ en T.

Ces différentes grandeurs vérifient les équations dites de Maxwell suivantes :

$$\boldsymbol{\nabla} \wedge \boldsymbol{E} + \partial_t \boldsymbol{B} = 0 \tag{A.1}$$
$$\boldsymbol{\nabla} \wedge \boldsymbol{H} - \partial_t \boldsymbol{D} = \boldsymbol{J} \tag{A.2}$$
$$\boldsymbol{\nabla} \cdot \boldsymbol{D} = \rho \tag{A.3}$$
$$\boldsymbol{\nabla} \cdot \boldsymbol{B} = 0 \tag{A.4}$$

Avec $\rho(\boldsymbol{M},t)$ la densité de charge en C/m^3 et $\boldsymbol{J}$ la densité de courant en A/m^2. Ces deux grandeurs sont liées par l'équation de la conservation de la charge (Eq. A.5) qui découle des équations de Maxwell.

$$\boldsymbol{\nabla} \cdot \boldsymbol{J} + \partial_t \rho = 0 \tag{A.5}$$

Enfin, les lois de comportement des matériaux ou les relations constitutives permettent de relier les champs électromagnétiques à leur induction.

A.2 Milieux diélectriques et magnétiques

Ces milieux sont des milieux linéraires, homogènes et isotopes caractérisés par la permittivité électrique ϵ en F/m et la perméabilité magnétique μ en H/m, telles que :

$$\boldsymbol{D} = \epsilon \boldsymbol{E} \tag{A.6}$$
$$\boldsymbol{B} = \mu \boldsymbol{H} \tag{A.7}$$

Dans le cas du vide, la permittivité électrique ϵ_0 et la perméabilité magnétique μ_0 respectent par convention $\epsilon_0 \mu_0 c^2 = 1$ où c est la vitesse de la lumière dans le vide. Ce qui donne les valeurs numériques suivantes :

$$\epsilon_0 = \frac{1}{36\pi} 10^{-9} \mathrm{F/m} \tag{A.8}$$
$$\mu_0 = 4\pi 10^{-7} \mathrm{H/m} \tag{A.9}$$

De plus dans la majorité des cas, nous décrivons la permittivité électrique et la perméabilité magnétique relativement à celles du vide c'est-à-dire que :

$$\epsilon = \epsilon_0 \epsilon_r \tag{A.10}$$
$$\mu = \mu_0 \mu_r \tag{A.11}$$

A.3 Milieux conducteurs

Les milieux conducteurs sont régis par la loi d'Ohm locale suivante :

$$\boldsymbol{J} = \sigma \boldsymbol{E} \tag{A.12}$$

Où σ est la conductivité du milieu exprimée en Siemens par mètre. Pour la plupart des métaux, cette grandeur est de l'ordre de $10^7 \mathrm{S/m}$.

Notons que dans la plupart des cas, nous utilisons des métaux idéalisés dits de conductivité infinie. Ce qui impose que le champ électrique $\boldsymbol{E}$ est nul.

Annexe B

Méthode FDTD

B.1 Principe de la méthode FDTD

Le schéma numérique habituellement utilisé pour la résolution des équations de Maxwell a été proposé par K. Yee en 1966 [35]. Il s'agit d'un schéma aux Différences Finies de type Leap-Frog (saute-mouton) en espace et en temps. Ses principales propriétés sont :

- **Discrétisation en temps**
  Le schéma temporel de type de Leap-Frog consiste à calculer le champ électrique $\boldsymbol{E}$ à l'instant $n\Delta t$ et le champ magnétique $\mathbf{H}$ à l'instant $\left(n+\frac{1}{2}\right)\Delta t$, où $n \in \mathbb{N}$ et Δt est le pas de temps choisi pour la discrétisation.
- **Discrétisation en espace**
  De la même manière et ce, pour des raisons de stabilité numérique, les $\boldsymbol{E}$ et les $\mathbf{H}$ sont calculés à un demi pas spatial de décalage. Les équations de Maxwell sont donc discrétisées suivant un schéma Différences Finies, sur un maillage structuré de forme cubique (différences finies dans les trois directions de l'espace). Les inconnues E_x, E_y, E_z, H_x, H_y et H_z sont placées comme cela est indiqué sur la figure B.1.

Pour comprendre le principe, intéressons-nous aux composantes en x des champs électrique et magnétique. Les composantes des vecteurs des champs électrique et magnétique s'écrivent respectivement aux temps $t = (n+\frac{1}{2})\Delta t$ et $t = n\Delta t$, de la manière suivante :

$$\varepsilon \left(\partial_t E_x\right)^{n+\frac{1}{2}} + \sigma\, E_x^{n+\frac{1}{2}} = \left(\partial_y E_z\right)^{n+\frac{1}{2}} - \left(\partial_z H_y\right)^{n+\frac{1}{2}} \tag{B.1a}$$

$$\mu \left(\partial_t H_x\right)^{n} = -\left(\partial_y E_z\right)^{n} + \left(\partial_z E_y\right)^{n} \tag{B.1b}$$

ce qui donne par discrétisation en temps :

$$\varepsilon \frac{E_x^{n+1} - E_x^{n}}{\Delta t} + \sigma \frac{E_x^{n+1} + E_x^{n}}{2} = \left(\partial_y H_z\right)^{n+\frac{1}{2}} - \left(\partial_z H_y\right)^{n+\frac{1}{2}} \tag{B.2a}$$

$$\mu \frac{H_x^{n+\frac{1}{2}} - H_x^{n-\frac{1}{2}}}{\Delta t} = -\left(\partial_y E_z\right)^{n} + \left(\partial_z E_y\right)^{n} \tag{B.2b}$$

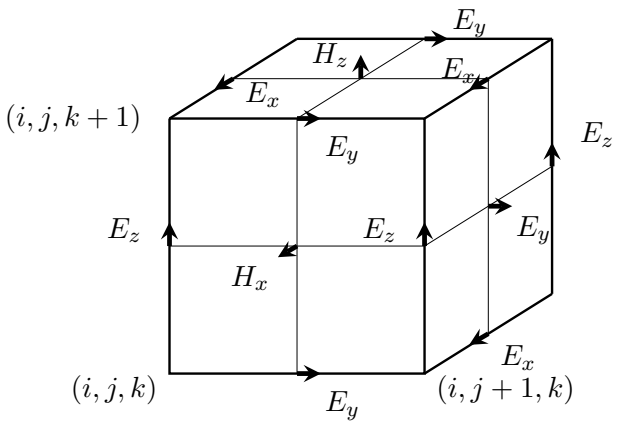


FIGURE B.1 – Décalage spatial du schéma de Yee.

Nous retrouvons des résultats similaires pour les autres composantes.

Pour la discrétisation en espace, nous supposons que les nœuds du maillage sont définis par les indices i, j, k. Les champs sont alors évalués en $E_x(i+\frac{1}{2}, j, k)$, $E_y(i, j+\frac{1}{2}, k)$, $E_z(i, j, k+\frac{1}{2})$, $H_x(i, j+\frac{1}{2}, k+\frac{1}{2})$, $H_y(i+\frac{1}{2}, j, k+\frac{1}{2})$, $H_z(i+\frac{1}{2}, j+\frac{1}{2}, k)$.
Par des développements de Taylor, les dérivées spatiales utilisées dans les relations précédentes deviennent :

$$\partial_y H_z = \frac{H_z(i+\frac{1}{2}, j+\frac{1}{2}, k) - H_z(i+\frac{1}{2}, j-\frac{1}{2}, k)}{\Delta y} \tag{B.3a}$$

$$\partial_z H_y = \frac{H_y(i+\frac{1}{2}, j, k+\frac{1}{2}) - H_y(i+\frac{1}{2}, j, k-\frac{1}{2})}{\Delta z} \tag{B.3b}$$

$$\partial_y E_z = \frac{E_z(i, j+1, k+\frac{1}{2}) - E_z(i, j, k+\frac{1}{2})}{\Delta y} \tag{B.4a}$$

$$\partial_z E_y = \frac{E_y(i, j+\frac{1}{2}, k+1) - E_y(i, j+\frac{1}{2}, k)}{\Delta z} \tag{B.4b}$$

L'algorithme du schéma numérique de type Leap-Frog est le suivant :

1. calcul du champ $\mathbf{H}^{n+\frac{1}{2}}$.
   Par exemple pour H_x, nous avons :

$$\mu \frac{H_x^{n+\frac{1}{2}} - H_x^{n-\frac{1}{2}}}{\Delta t} = -\frac{E_z^n(i, j+1, k+\frac{1}{2}) - E_z^n(i, j, k+\frac{1}{2})}{\Delta y} + \frac{E_y^n(i, j+\frac{1}{2}, k+1) - E_y^n(i, j+\frac{1}{2}, k)}{\Delta z} \tag{B.5}$$

2. calcul du champ $\boldsymbol{E}^{n+1}$ à partir de $\mathbf{H}^{n+\frac{1}{2}}$.
   Par exemple pour E_x, nous obtenons :

$$\varepsilon\frac{E_x^{n+1}-E_x^n}{\Delta t}+\sigma\frac{E_x^{n+1}+E_x^n}{2}=\frac{H_z^{n+\frac{1}{2}}(i+\frac{1}{2},j+\frac{1}{2},k)-H_z^{n+\frac{1}{2}}(i+\frac{1}{2},j-\frac{1}{2},k)}{\Delta y}$$
$$-\frac{H_y^{n+\frac{1}{2}}(i+\frac{1}{2},j,k+\frac{1}{2})-H_y^{n+\frac{1}{2}}(i+\frac{1}{2},j,k-\frac{1}{2})}{\Delta z} \tag{B.6}$$

Remarque B.1.1. *Nous avons $E_x^n(i+1/2,j,k)$, $E_y^n(i,j+1/2,k)$, $E_z^n(i,j,k+1/2)$ sont les approximations de la solution exacte $\boldsymbol{E}_x(\mathbf{x},t)$, $\boldsymbol{E}_y(\mathbf{x},t)$ et $\boldsymbol{E}_z(\mathbf{x},t)$ au point $(x_{i+1/2},y_j,z_k,t^n)$ $(x_i,y_{j+1/2},z_k,t^n)$ et $(x_i,y_j,z_{k+1/2},t^n)$.De même, $H_x^{n+1/2}(i,j+1/2,k+1/2)$, $H_y^{n+1/2}(i+1/2,j,k+1/2)$, $H_z^{n+1/2}(i+1/2,j+1/2,k)$ sont les approximations de la solution exacte $\mathbf{H}_x(\mathbf{x},t)$,$\mathbf{H}_y(\mathbf{x},t)$ et $\mathbf{H}_z(\mathbf{x},t)$ au point $(x_i,y_{j+1/2},z_{k+1/2},t^{n+1/2})$, $(x_{i+1/2},y_j,z_{k+1/2},t^{n+1/2})$ et $(x_{i+1/2},y_{j+1/2},z_k,t^{n+1/2})$.*

B.2 Les fils minces

Le formalisme de fils minces présenté ici a été introduit par R. Holland en 1981[19]. Ce modèle permet de décrire des éléments linéiques conducteurs qui suivent les arêtes du maillage différence finie.

Le fil mince de R. Holland explicite le comportement du courant et de la charge dans le fil qui dérive directement des équations de Maxwell. Le formalisme s'appuie sur l'hypothèse quasi-statique qui se justifie par l'hypothèse de départ que la taille de toutes les cellules est petite par rapport à la plus petite longueur d'onde du spectre d'excitation. L'hypothèse quasi-statique permet de simplifier grandement le formalisme de couplage entre le champ électromagnétique et le fil mince.

La première équation différentielle liant i et q sur un fil dérive de l'équation de Faraday (Eq. A.1) que nous pouvons écrire dans un repère cylindrique (l,r,θ) qui suit le parcours du fil de rayon a. La composante en θ du champ magnétique s'écrit alors :

$$\partial_l E_r-\partial_r E_l=\mu\partial_t H_\theta \tag{B.7}$$

Que l'on peut intégrer sur un rayon r et simplifier en exploitant le fait que le fil est supposé parfaitement conducteur, c'est-à-dire que le champ électrique $E_l(a)=0$. Nous pouvons également considérer les fils de conductivité finie (R_s), dont la composante du champ électrique longitudinal au fil s'obtient par la loi d'Ohm $E_l(a)=iR_s/dl$. Ce qui nous donne :

$$E_l(a)-E_l(r)+\int_a^r\partial_l E_r dr=-\mu\int_a^r\partial_t H_\theta dr \tag{B.8}$$

L'hypothèse quasi-statique nous permet d'écrire les deux relations suivantes :

$$E_r = \frac{q}{2\pi\epsilon r} \tag{B.9}$$

$$H_\theta = \frac{i}{2\pi r} \tag{B.10}$$

Où q est la charge par unité de longueur sur le fil et i le courant traversant le fil. Ce qui nous permet d'écrire :

$$\partial_t i + \left(\frac{1}{\epsilon\mu}\right)^2 \partial_l q = \frac{E_l(r) - E_l(a)}{L} \tag{B.11}$$

où L désigne l'inductance de R. Holland en H/m définie comme suit :

$$L = \frac{\mu}{2\pi} \ln\left(\frac{r}{a}\right) \tag{B.12}$$

Dans le cas d'un schéma aux différences finies, les cellules sont parallélépipédiques. Il convient donc de prendre le champ moyen dans le rectangle (défini par Δx et Δy dans le cas où la direction du fil est suivant l'axe z). Ce qui conduit à :

$$\partial_t i + \left(\frac{1}{\epsilon\mu}\right)^2 \partial_l q = \frac{\langle E_l(r)\rangle - E_l(a)}{L_c} \tag{B.13}$$

Où L_c désigne l'inductance de R. Holland ramenée à la cellule définie par :

$$L = \frac{\mu}{4\pi}\left[\ln\left(\frac{\Delta x^2 + \Delta y^2}{4a^2}\right) + \frac{\Delta y}{\Delta x}\arctan\left(\frac{\Delta x}{\Delta y}\right) + \frac{\Delta x}{\Delta y}\arctan\left(\frac{\Delta y}{\Delta x}\right) - \frac{\pi a^2}{\Delta x \Delta y} - 3\right] \tag{B.14}$$

La seconde équation différentielle liant i et q sur un fil s'obtient en intégrant volumiquement l'équation de la conservation de la charge (Eq. A.5).

$$\partial_t q + \partial_l i = 0 \tag{B.15}$$

Les deux équations (B.11) et (B.15) décrivent l'évolution de la charge et du courant sur un fil que l'on peut résoudre avec un schéma de type différence finie.

Annexe C

Méthode FVTD

Dans la section précédente, nous avons rapidement présenté la méthode schéma aux Différences Finies qui a très largement démontré sa robustesse. Cependant, elle présente l'inconvénient majeur d'être non conforme à la géométrie. La méthode des volumes finis répond à ce problème.

C.1 Principe de la méthode FVTD

Contrairement à la méthode des différences finies qui met en jeu des approximations des dérivées, la méthode des volumes finis exploitent des approximations d'intégrales directement basées sur la forme dite forte de l'équation à résoudre. En particulier, elle donne lieu à la formulation conservative c'est-à-dire qu'elle exprime la conservation d'une quantité $f(\mathbf{U})$. Ainsi, l'évolution dans le temps de la quantité sur un domaine $\mathcal{D}$ correspond à l'ensemble des flux $\phi(\mathbf{U})$ sur sa frontière.

La loi de conservation peut donc s'écrire :

$$\int_{\mathcal{D}} \partial_t f(\mathbf{U}) d\tau + \int_{\partial\mathcal{D}} \phi(\mathbf{U}) d\sigma = 0 \tag{C.1}$$

Si l'on considère un domaine infinitésimal, nous obtenons :

$$\partial_t f(\mathbf{U}) + \boldsymbol{\nabla} \cdot F(\mathbf{U}) = 0 \tag{C.2}$$

avec F une fonction telle que $F(\mathbf{U}) \cdot n = \phi(\mathbf{U})$ et n la normale sortante de la frontière $\partial\mathcal{D}$.

Nous pouvons appliquer ce principe aux équations de Maxwell homogènes (Eq. A.1) pour notre présentation de la méthode. Ce système peut s'écrire sous la forme conservative suivante :

$$\alpha \partial_t \mathbf{U} + \boldsymbol{\nabla} \cdot F(\mathbf{U}) = 0 \tag{C.3}$$

Avec $\mathbf{U} = (E_x, E_y, E_z, H_x, H_y, H_z)^t$, $\alpha = \begin{bmatrix} \epsilon I & 0 \\ 0 & \mu I \end{bmatrix}$, Id est la matrice identité de $\mathbb{R}^3$ et

$$F(\mathbf{U}) = (F_1(\mathbf{U}); F_2(\mathbf{U}); F_3(\mathbf{U})) \tag{C.4}$$

avec $F_1(\mathbf{U}) = \begin{pmatrix} 0 \\ H_z \\ -H_y \\ 0 \\ -E_z \\ E_y \end{pmatrix}$, $F_2(\mathbf{U}) = \begin{pmatrix} -H_z \\ 0 \\ H_x \\ E_z \\ 0 \\ -E_x \end{pmatrix}$, $F_3(\mathbf{U}) = \begin{pmatrix} H_y \\ -H_x \\ 0 \\ -E_y \\ E_x \\ 0 \end{pmatrix}$ ainsi F est une combinaison de trois fonctions linéaires de $\mathbb{R}^6$.

Le système d'équation s'écrit alors :

$$\alpha\partial_t\mathbf{U} + \partial_x F_1(\mathbf{U}) + \partial_y F_2(\mathbf{U}) + \partial_z F_3(\mathbf{U}) = 0 \tag{C.5}$$

Pour résoudre le problème sur un domaine spatial Ω donné, nous réalisons une discrétisation en un ensemble de volumes V de telle manière que $\bigcup V = \Omega$. Nous n'avons plus qu'à calculer l'intégrale de l'équation C.5 sur chaque volume V.

Remarquons que le terme temporel de C.5 peut être mis sous la forme d'une divergence ce qui nous permet de faire une résolution volume fini à la fois en espace et en temps, ce que l'on fait généralement. Cependant nous pouvons utiliser un schéma différent pour résoudre en temps, de Euler, de Runge-Kutta par exemple.

Nous n'entrerons pas plus dans la description de la méthode des volumes finis ; des développements complets sont donnés dans [96].

C.2 Les fils minces

Dans la méthode des volumes finis, le formalisme de fil mince peut être similaire à celui proposé par R. Holland pour les différences finies. De la même manière, nous avons d'une part le système d'équation aux dérivées partielles (Eq. B.11) et (Eq. B.15) du fil décrivant la variation de i et q et d'autre part une interaction entre le fil et le domaine 3D qui se matérialise par des termes sources sur le fil et dans le domaine 3D. C'est sur ce point que diffère légèrement le modèle de R. Holland appliqué aux volumes finis.

Pour avoir un couplage optimal entre le fil mince et le domaine 3D, il est recommandé d'utiliser des volumes en prisme qui longent le fil comme le montre la figure C.1, ou définir un cylindre d'interaction du fil avec le domaine 3D [96] qui permet également de définir des fils obliques.

Dans le cas d'un fil qui suit les arêtes d'un prisme, nous obtenons pour le k-ième segment l'équation (B.11) qui est moyenne sur l'ensemble des volumes adjacents, soit :

$$\partial_t i_k + c^2\partial_l q_k = \frac{\langle \boldsymbol{E}\cdot\boldsymbol{dl}\rangle_k}{\langle L\rangle_k} \tag{C.6}$$

avec $\langle \boldsymbol{E}\cdot\boldsymbol{dl}\rangle_k = \frac{\sum_{p=1}^{\text{nbv}} \boldsymbol{E}_p\cdot\boldsymbol{dl}}{\text{nbv}}$, nbv étant le nombre de volumes autour du fil, $\boldsymbol{E}_p\cdot\boldsymbol{dl}$ la projection du champ électrique au centre du volume p suivant la direction $\boldsymbol{dl}$ du fil. Enfin, $\langle L\rangle_k = \frac{\sum_{p=1}^{\text{nbv}} L_p}{\text{nbv}}$ est l'inductance moyenne.

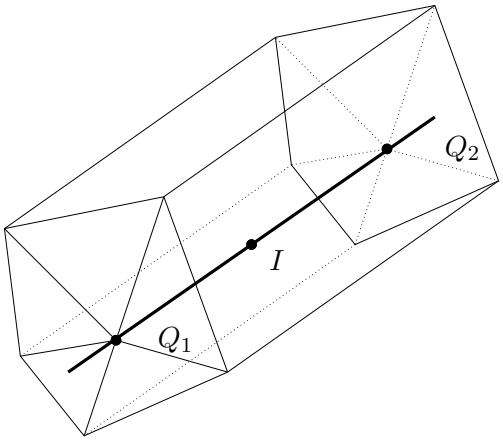


FIGURE C.1 – Définition des mailles autour d'un segment filaire.

En ce qui concerne le terme source dans le 3D, nous n'entrerons pas dans les détails mais le principe est d'introduire une fraction de la densité de courant $\boldsymbol{J} \cdot \boldsymbol{dl} = \frac{I}{S}$ dans l'équation de Maxwell Ampère pour chaque volume adjacent au fil, avec S la section du fil.

Annexe D

Principe du champ total / champ diffracté appliqué aux lignes de transmission

Le principe du champ total et du champ diffracté est couramment utilisé. Nous pourrions utiliser ce même principe en utilisant comme milieu de propagation les lignes de transmission. Pour pouvoir évaluer cette approche, nous devons définir la notion de 'champ total champ diffracté', pour les lignes de transmission.

Le fondement du champ total / champ diffracté (TF/SF acronyme de Total-Field / Scattering Field) découle de la linéarité des équations de Maxwell. Cette linéarité nous permet d'écrire les champs magnétique et électrique de la manière suivante :

$$\boldsymbol{E}_t = \boldsymbol{E}_i + \boldsymbol{E}_s \ , \quad \boldsymbol{H}_t = \boldsymbol{H}_i + \boldsymbol{H}_s \tag{D.1}$$

Avec $\boldsymbol{E}_i$ et $\boldsymbol{H}_i$ champ induit connu dans tout l'espace et à tous les pas de temps. Ce champ doit exister dans l'espace libre, sans matériau. $\boldsymbol{E}_s$ et $\boldsymbol{H}_s$ sont les champs diffractés. Ils représentent en général l'information primaire recherchée. $\boldsymbol{E}_t$ et $\boldsymbol{H}_t$ quant à eux représentent les champs totaux.

Note : Le fait qu'il ne doit pas y avoir de matériau à l'interface est ce qui empêche d'utiliser ce principe pour faire de l'hybridation avec une ligne de transmission qui traverse cette interface.

Le principe du TF/SF est également valable pour une forme partiellement intégrée des équations de Maxwell. Notamment pour les lignes de transmission (Eq. 2.43). Ainsi, dans le cas d'une ligne de transmission le principe TF/SF se formalise de la manière suivante :

$$v_t = v_i + v_s \ , \quad i_t = i_i + i_s \tag{D.2}$$

Nous continuerons de présenter le principe du TF/SF en l'appliquant aux lignes de transmission.

L'idée directrice est de découper l'espace en trois zones : deux zones de champ diffracté (partie gauche et droite) et une zone de champ total (partie centrale).

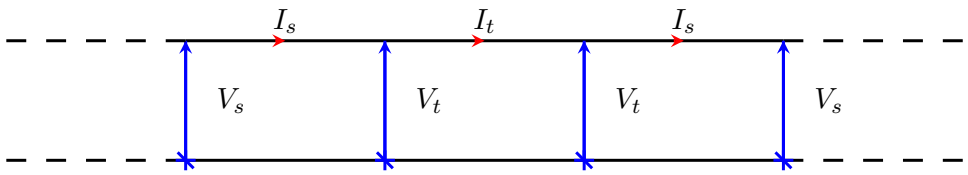


FIGURE D.1 – Découpage en champ total et diffracté sur une ligne de transmission.

Les différentes zones se résolvent avec le même système (Eq. 2.45). En l'appliquant au champ total ou au champ diffracté, il peut se mettre sous cette forme dans la cas d'une ligne homogène :

$$I|_{p-1/2}^{n+1/2} = -Q_i \cdot\ I|_{p-1/2}^{n-1/2} + Q_v \cdot \left(V|_p^n - V|_{p-1}^n \right)$$
$$V|_p^{n+1} = -K_v \cdot\ V|_p^n + K_i \cdot \left(I|_{p+1/2}^{n+1/2} - I|_{p-1/2}^{n+1/2} \right)$$

avec :

$$Q_i = \left(\frac{R\Delta t - 2L}{R\Delta t + 2L} \right),\ Q_v = \frac{2}{R\Delta t + 2L} \frac{\Delta t}{\Delta l}$$
$$K_v = \left(\frac{G\Delta t - 2C}{G\Delta t + 2C} \right),\ K_i = \frac{2}{G\Delta t + 2C} \frac{\Delta t}{\Delta l}$$

Nous avons donc dans le cas du champ total :

$$I_t|_{p+1/2}^{n+1/2} = -Q_i \cdot\ I_t|_{p+1/2}^{n-1/2} + Q_v \cdot \left(V_t|_{p+1}^n - V_t|_p^n \right)$$
$$V_t|_p^{n+1} = -K_v \cdot\ V_t|_p^n + K_i \cdot \left(I_t|_{p+1/2}^{n+1/2} - I_t|_{p-1/2}^{n+1/2} \right)$$

et du champ diffracté :

$$I_s|_{p+1/2}^{n+1/2} = -Q_i \cdot\ I_s|_{p+1/2}^{n-1/2} + Q_v \cdot \left(V_s|_{p+1}^n - V_s|_p^n \right)$$
$$V_s|_p^{n+1} = -K_v \cdot\ V_s|_p^n + K_i \cdot \left(I_s|_{p+1/2}^{n+1/2} - I_s|_{p-1/2}^{n+1/2} \right)$$

En revanche, à l'interface entre la zone de gauche et la zone centrale l'algorithme va résoudre le système suivant, qui ne représente pas la réalité physique :

$$I_s|_{p+1/2}^{n+1/2} = -Q_i \cdot\ I_s|_{p+1/2}^{n-1/2} + Q_v \cdot \left(V_t|_{p+1}^n - V_s|_p^n \right)$$
$$V_t|_p^{n+1} = -K_v \cdot\ V_t|_p^n + K_i \cdot \left(I_t|_{p+1/2}^{n+1/2} - I_s|_{p-1/2}^{n+1/2} \right)$$

En revanche, en exploitant la linéarité du problème $v_t = v_i + v_s$, $i_t = i_i + i_s$ et par identification avec la résolution du problème en champ total, nous mettons en avant les

termes correctifs à appliquer. Bien entendu ces termes correctifs ne sont possibles que dans la mesure où nous pouvons calculer le champ incident dans tout l'espace et à chaque point de temps indépendamment de l'état du système.

$$\begin{aligned} I_s|_{q-1/2}^{n+1/2} &= -Q_i \cdot\ I_s|_{q-1/2}^{n-1/2} + Q_v \cdot \left(V_t|_q^n - V_s|_{q-1}^n \right) - Q_v \cdot\ V_i|_q^n \\ V_t|_q^{n+1} &= -K_v \cdot\ V_t|_q^n + K_i \cdot \left(I_t|_{q+1/2}^{n+1/2} - I_s|_{q-1/2}^{n+1/2} \right) - K_i \cdot\ I_i|_{q-1/2}^{n+1/2} \end{aligned}$$

Nous pouvons faire de même pour l'autre interface :

$$\begin{aligned} I_s|_{q+1/2}^{n+1/2} &= -Q_i \cdot\ I_s|_{q+1/2}^{n-1/2} + Q_v \cdot \left(V_s|_{q+1}^n - V_t|_q^n \right) + Q_v \cdot\ V_i|_q^n \\ V_t|_q^{n+1} &= -K_v \cdot\ V_t|_q^n + K_i \cdot \left(I_s|_{q+1/2}^{n+1/2} - I_t|_{q-1/2}^{n+1/2} \right) + K_i \cdot\ I_i|_{q+1/2}^{n+1/2} \end{aligned}$$

Le principe TF/SF peut être mis sous une forme synthétique représentant ce que l'on pourrait appeler un générateur d'ondes orientées. Les ondes ainsi générées sont les suivantes :

$$\begin{aligned} w &= \frac{v_i - \xi Z i_i}{2\sqrt{Z}} \\ 0 &= \frac{v_i + \xi Z i_i}{2\sqrt{Z}} \end{aligned}$$

Avec Z l'impédance de normalisation réelle. ξ en revanche représente la direction prise par l'onde générée $\xi = +1$ pour une onde progressive $\xi = -1$ pour une onde rétrograde. Les termes correctifs peuvent donc être redéfinis comme suit :

$$\begin{aligned} I|_{p-\xi/2}^{n+1/2} &= -Q_i \cdot\ I|_{p-\xi/2}^{n-1/2} + Q_v \cdot \left(V|_{p+(1-\xi)/2}^n - V|_{p-(1+\xi)/2}^n \right) - \xi Q_v \cdot\ V_i|_p^n \\ V|_p^{n+1} &= -K_v \cdot\ V|_p^n + K_i \cdot \left(I|_{p+1/2}^{n+1/2} - I|_{p-1/2}^{n+1/2} \right) - \xi K_i \cdot\ I_i|_{p-\xi/2}^{n+1/2} \end{aligned}$$

Notons enfin qu'il est plus simple de définir ce générateur en utilisant simplement la tension ou le courant, en faisant l'approximation quasi-statique spatiale et temporelle suivante : $V_i|_{p-(1+\xi)/2}^n \approx -\xi Z\ I_i|_{p-\xi/2}^{n+1/2}$.

Annexe E

Modélisation de structures filaires par la méthode des équations intégrales en temporel

ee

$$\boldsymbol{E^t} = \boldsymbol{E^s} + \boldsymbol{E^i} \tag{E.1}$$

Équation de Maxwell-Faraday sous sa forme potentielle vecteur.

$$\boldsymbol{E^s} = -\boldsymbol{\nabla} V - \partial_t \boldsymbol{A} \tag{E.2}$$

Jauge de Lorentz

$$\partial_t v = -c^2 \boldsymbol{\nabla} \cdot \boldsymbol{A} \tag{E.3}$$

$$\partial_t \boldsymbol{E^s} = c^2 \boldsymbol{\nabla} \left(\boldsymbol{\nabla} \cdot \boldsymbol{A} \right) - \partial_t^2 \boldsymbol{A} \tag{E.4}$$

Champ total

$$\partial_t \boldsymbol{E^t} = \partial_t \boldsymbol{E^i} + c^2 \boldsymbol{\nabla} \left(\boldsymbol{\nabla} \cdot \boldsymbol{A} \right) - \partial_t^2 \boldsymbol{A} \tag{E.5}$$

Sur le conducteur le champ total est nul

$$\partial_t \boldsymbol{E^t} \cdot \boldsymbol{s}(p_s) = 0 \tag{E.6}$$

Ce qui donne lieu à :

$$\partial_t \boldsymbol{E^i} \cdot \boldsymbol{s}(p_s) = -c^2 \boldsymbol{\nabla} \left(\boldsymbol{\nabla} \cdot \boldsymbol{A} \right) \cdot \boldsymbol{s}(p_s) + \partial_t^2 \boldsymbol{A} \cdot \boldsymbol{s}(p_s) \tag{E.7}$$

Le potentiel vecteur est donné par la relation de Green

$$\boldsymbol{A} = \mu_0 \int_{\mathcal{C}_0} \boldsymbol{J} * G d\tau \tag{E.8}$$

Sur la surface du conducteur, on a :

$$I(p_s) = 2\pi R \boldsymbol{J}(p_s) \cdot \boldsymbol{s}(p_s) \tag{E.9}$$

$$\boldsymbol{A} = \mu_0 \int_{\partial \mathcal{C}_0} I\boldsymbol{s} * G ds \tag{E.10}$$

$$G = \delta \left(\frac{p, t - r/c}{4\pi r} \right) \tag{E.11}$$

$r = \|mp\|$, m point calculé et p point d'intégration

$$\begin{aligned} \partial_t \boldsymbol{E^i} \cdot \boldsymbol{s}(p_s) &= -\frac{\mu_0 c^2}{4\pi} \boldsymbol{\nabla} \left(\boldsymbol{\nabla} \cdot \int_{\partial \mathcal{C}_0} \frac{I(p, t - r/c)}{r} \boldsymbol{s}(p) ds \right) \cdot \boldsymbol{s}(p_s) \\ &\quad + \frac{\mu_0}{4\pi} \partial_t^2 \left(\int_{\partial \mathcal{C}_0} \frac{I(p, t - r/c)}{r} \boldsymbol{s}(p) ds \right) \cdot \boldsymbol{s}(p_s) \end{aligned} \tag{E.12}$$

Discrétisations du fil

$$\begin{aligned} \partial_t \boldsymbol{E^i} \cdot \boldsymbol{s}(p_s) &= -\frac{\mu_0 c^2}{4\pi} \boldsymbol{\nabla} \left(\boldsymbol{\nabla} \cdot \sum_{k=1}^{K} \int_{\partial \mathcal{C}_k} \frac{I(p_k, t - r/c)}{r} \boldsymbol{s}(p_k) ds \right) \cdot \boldsymbol{s}(p_s) \\ &\quad + \frac{\mu_0}{4\pi} \partial_t^2 \left(\sum_{k=1}^{K} \int_{\partial \mathcal{C}_k} \frac{I(p_k, t - r/c)}{r} \boldsymbol{s}(p_k) ds \right) \cdot \boldsymbol{s}(p_s) \end{aligned} \tag{E.13}$$

La formulation temporelle est obtenue en introduisant le self-term α_k.

$$\int_{\partial \mathcal{C}_k} \frac{I(p_k, t - r/c)}{r} \boldsymbol{s}(p) ds = \alpha_k \boldsymbol{s}(p_k) I(p_k) \tag{E.14}$$

Ce qui nous permet d'écrire la relation que nous pouvons résoudre avec un schéma temporel.

$$\begin{aligned} \partial_t \boldsymbol{E^i} \cdot \boldsymbol{s}(p_j) &= -\frac{\mu_0 c^2}{4\pi} \boldsymbol{\nabla} \left(\boldsymbol{\nabla} \cdot \sum_{k=1}^{K} \int_{\partial \mathcal{C}_k} \frac{I(p_k, t - r/c)}{r} \boldsymbol{s}(p_k) ds \right) \cdot \boldsymbol{s}(p_j) \\ &\quad + \frac{\alpha_j \mu_0}{4\pi} \partial_t^2 I(p_j, t) \\ &\quad + \frac{\mu_0}{4\pi} \partial_t^2 \left(\sum_{k=1, k \neq j}^{K} \int_{\partial \mathcal{C}_k} \frac{I(p_k, t - r/c)}{r} \boldsymbol{s}(p) ds \right) \cdot \boldsymbol{s}(p_j) \end{aligned} \tag{E.15}$$

Annexe F

Représentation BLT des équations de lignes de transmission

Cette annexe présente le formalisme de lignes de transmission proposé par Baum, Liu, Tesche. Ce formalisme permet de décrire le courant et la tension en bout de ligne en utilisant la notion d'onde de puissance ou onde électrique complexe [97]. Nous ne présenterons ici que le formalisme général, sans rentrer dans toutes les subtilités des termes sources distribuées. Ce formalisme est celui utilisé par le code CRIPTE de l'ONERA pour traiter les problèmes de lignes de transmission en fréquentiel.

Nous admettons que la tension et le courant peuvent être décomposés en une onde de tension complexe qui se propage dans la direction positive + et en une autre onde se propageant dans la direction négative − avec $e^{\pm\gamma l}$ le terme de phase selon la direction. $\gamma = jk$ est le nombre d'ondes complexes. On admet qu'en tout point l de la ligne que la tension (res. courant) est la somme de ces deux ondes :

$$\widetilde{\boldsymbol{V}}(l) = \widetilde{\boldsymbol{Z_c}}\left(e^{\widetilde{\gamma} l}\widetilde{\boldsymbol{W}}^{+} + e^{-\widetilde{\gamma} l}\widetilde{\boldsymbol{W}}^{-}\right) \tag{F.1}$$

$$\widetilde{\boldsymbol{I}}(l) = \left(e^{\widetilde{\gamma} l}\widetilde{\boldsymbol{W}}^{+} - e^{-\widetilde{\gamma} l}\widetilde{\boldsymbol{W}}^{-}\right) \tag{F.2}$$

Où $\widetilde{\boldsymbol{Z_c}}$ est l'impédance de la ligne.
On obtient ainsi la définition des ondes électriques complexes normalisées.

$$\widetilde{\boldsymbol{W}}^{+} = \frac{1}{2}e^{\widetilde{\gamma} l}\widetilde{\boldsymbol{Z_c}}^{-1}\left(\widetilde{\boldsymbol{V}}(l) + \widetilde{\boldsymbol{Z_c}}\widetilde{\boldsymbol{I}}(l)\right) \tag{F.3}$$

$$\widetilde{\boldsymbol{W}}^{-} = \frac{1}{2}e^{-\widetilde{\gamma} l}\widetilde{\boldsymbol{Z_c}}^{-1}\left(\widetilde{\boldsymbol{V}}(l) - \widetilde{\boldsymbol{Z_c}}\widetilde{\boldsymbol{I}}(l)\right) \tag{F.4}$$

Cependant, il est plus simple de considérer uniquement les ondes incidentes et réfléchies à chaque extrémité. Ainsi, pour les deux extrémités gauche (L) et droite (R) et où i désigne l'onde incidente et r l'onde réfléchie :

$$\widetilde{\boldsymbol{W}}_L^i = \widetilde{\boldsymbol{W}}^-(0) \qquad \widetilde{\boldsymbol{W}}_L^r = \widetilde{\boldsymbol{W}}^+(0)$$
$$\widetilde{\boldsymbol{W}}_R^i = \widetilde{\boldsymbol{W}}^+(L) \qquad \widetilde{\boldsymbol{W}}_R^r = \widetilde{\boldsymbol{W}}^-(L)$$

Le terme de propagation dans la ligne est décomposé en deux termes. Le premier terme corespond à la propagation de l'onde incidente, le deuxième terme lié à l'onde excitatrice.

$$\widetilde{\boldsymbol{W}}_L^r = \widetilde{\boldsymbol{W}}_R^i e^{\widetilde{\gamma}L} - \frac{1}{2}\widetilde{\boldsymbol{Z_c}}^{-1}\left(\widetilde{\boldsymbol{V}}(l_s) + \widetilde{\boldsymbol{Z_c}}\widetilde{\boldsymbol{I}}(l_s)\right)e^{-\widetilde{\gamma}l_s} \tag{F.5}$$
$$\widetilde{\boldsymbol{W}}_R^r = \widetilde{\boldsymbol{W}}_L^i e^{\widetilde{\gamma}L} + \frac{1}{2}\widetilde{\boldsymbol{Z_c}}^{-1}\left(\widetilde{\boldsymbol{V}}(l_s) + \widetilde{\boldsymbol{Z_c}}\widetilde{\boldsymbol{I}}(l_s)\right)e^{\widetilde{\gamma}(L-l_s)} \tag{F.6}$$

Que l'on peut mettre sous une forme matricielle

$$\widetilde{\boldsymbol{W}}^r = \boldsymbol{\Gamma}\widetilde{\boldsymbol{W}}^i - \boldsymbol{S} \tag{F.7}$$

Où $\boldsymbol{\Gamma}$ est la matrice de propagation et $\boldsymbol{S}$ le terme source.

De plus, nous pouvons lier la tension incidente à la tension réfléchie par le coefficient de réflexion.

$$\widetilde{\boldsymbol{W}}^r = \boldsymbol{\rho}\widetilde{\boldsymbol{W}}^i \tag{F.8}$$

Ce qui nous permet d'écrire :

$$\widetilde{\boldsymbol{W}}^i = [\boldsymbol{\Gamma} - \boldsymbol{\rho}]^{-1}\boldsymbol{S} \tag{F.9}$$

De plus, nous pouvons remarquer que la tension ou le courant (au signe près) total en bout de ligne sont donnés par :

$$\widetilde{\boldsymbol{V}} = \widetilde{\boldsymbol{Z_c}}(\widetilde{\boldsymbol{W}}^i + \widetilde{\boldsymbol{W}}^r) = \widetilde{\boldsymbol{Z_c}}\left[\mathbf{1} + \boldsymbol{\rho}\right]\left[\boldsymbol{\Gamma} - \boldsymbol{\rho}\right]^{-1}\boldsymbol{S} \tag{F.10}$$
$$\widetilde{\boldsymbol{I}} = (\widetilde{\boldsymbol{W}}^i - \widetilde{\boldsymbol{W}}^r) = \left[\mathbf{1} - \boldsymbol{\rho}\right]\left[\boldsymbol{\Gamma} - \boldsymbol{\rho}\right]^{-1}\boldsymbol{S} \tag{F.11}$$

Annexe G

Comparaisons entre différents langages de programmation

Dans cette annexe nous illustrons nos propos concernant les langages informatiques et en particulier deux aspects :

- La verbosité d'un langage. C'est le nombre de lignes nécessaire pour réaliser une fonction donnée.
- Le niveau d'abstraction du langage qui permet une plus grande réutilisabilité du code souvent au détriment des performances pures en exécution.

Nous illustrons ces deux points sur deux exemples : la liste chaînée qui est un élément récurrent en informatique. Le deuxième exemple concerne l'utilisation des tableaux.

G.1 La liste chaînée

La liste chaînée est un élément récurrent en informatique. Nous nous proposons d'implémenter une simple liste dans trois langages (FORTRAN, C et Python).

Langage	Nombre de lignes	Capacité de la liste
Fortran	73	seule des valeurs de type real sont manipulables par la liste. Le FORTRAN n'a pas de conteneur générique, ce qui permet au Fortran d'avoir de très bons compilateurs. Mais cette propriété oblige d'écrire plus de codes, une liste pour les réels une liste pour les complexes ...
C	66	Toute valeur qui peut être pointée par un `void*` peut être manipulée par cette liste. C'est un des avantages du C mais également sa plus grande faiblesse. En effet, la manipulation de pointeur est toujours délicate et source de bugs.
Python	29	Tous les objets peuvent être manipulés par cette liste. Toute la puissance de Python réside dans le non typage explicite d'interface, mais cette souplesse a un coût.
Python + librairie standard	0	Une autre puissance du Python c'est la définition de nombreuses librairies qui peuvent être réutilisées dans de nombreux contextes.

```
1   !
2  ! A Fortran list implementation
3  ! The lists' values can only be real.
4  !
5  module list_m
6    implicit none
7
8    type node_t
9       real :: val
10      type(node_t), pointer :: nxt => NULL()
11   end type node_t
12
13   type list_t
14      type(node_t), pointer :: root => NULL()
15   end type list_t
16
17 contains
18   ! add new value at end
19   subroutine append(list, val)
20     type(list_t), intent(inout) :: list
21     real, intent(in) :: val
22
23     type(node_t), pointer :: node, temp
24
25     allocate(node)
26     node%val = val
27
28     if (.not.associated(list%root)) then
```

```
29          list%root => node
30        else
31          temp => list%root
32          do while (associated(temp%nxt))
33             temp => temp%nxt
34          end do
35          temp%nxt => node
36        end if
37      end subroutine append
38
39      ! remove last value and return the value
40      function pop(list) result(res)
41        type(list_t), intent(inout) :: list
42        real ::  res
43
44        type(node_t), pointer :: last, prev
45
46        if (.not.associated(list%root)) then
47          res = 0.
48        else
49          last => list%root
50          nullify(prev)
51          do while(associated(last%nxt))
52             prev => last
53             last => last%nxt
54          end do
55
56          res = last%val
57          deallocate(last)
58          if (associated(prev)) then
59             nullify(prev%nxt)
60          end if
61        end if
62      end function pop
63    end module list_m
64
65    program test
66      use list_m
67
68      type(list_t) :: mylist
69      real :: val1 = 1
70      real :: val2 = 2.
71
72      call append(mylist, val1)
73      call append(mylist, val2)
74
75      print"('val 2 : ', f5.2)", pop(mylist)
76      print"('val 1 : ', f5.2)", pop(mylist)
77
78    end program test
```

```
1  /*
2   * A c list implementation using void pointer.
3   */
4
5  #include <stdlib.h>
6  #include <stdio.h>
7
8  typedef struct node
9  {
10   void *val;
11   struct node *nxt;
12 } node_t, *list_t;
13
14 /* Add new value at end */
15 list_t append(list_t list, void *val)
16 {
17   node_t* node = malloc(sizeof(node_t));
18
19   node->val = val;
20
21   node->nxt = NULL;
22
23   if(list == NULL)
24     {
25       return node;
26     }
27   else
28     {
29       node_t* temp=list;
30       while(temp->nxt != NULL)
31         {
32     temp = temp->nxt;
33         }
34       temp->nxt = node;
35       return list;
36     }
37 }
38
39 /* Return and remove last value */
40 void *pop(list_t list)
41 {
42   if(list == NULL)
43     {
44       return NULL;
45     }
46   else
47     {
48       node_t* last=list;
49       node_t* prev=NULL;
```

```
50
51        while(last->nxt != NULL)
52          {
53      prev = last;
54      last = last->nxt;
55          }
56
57        void *res = last->val;
58        free(last);
59        if (prev != NULL)
60    {
61      prev->nxt = NULL;
62    }
63
64        return res;
65      }
66  }
67
68  int main(int argc, char **argv)
69  {
70    list_t mylist = NULL;
71    int val1 = 1;
72    float val2 = 2;
73
74    mylist = append(mylist, &val1);
75    mylist = append(mylist, &val2);
76
77    printf("val 2: %f\n", *(float*)pop(mylist));
78    printf("val 1: %d\n", *(int*)pop(mylist));
79
80    return 0;
81  }
```

```
1  #! /usr/bin/env python
2  #
3  # A pure Python list implementation
4
5
6  class List:
7      "A pure Python list."
8      def __init__(self, value=None):
9          "Init List's root or List's node if 'value' is give."
10         if value is None:
11             self.last = None
12         else:
13             self.val = value
14             self.prev = None
15
16     def append(self, val):
17         "Append new value."
```

```
18          node = self.__class__(val)
19          if self.last is None:
20              self.last = node
21          else:
22              node.prev = self.last
23              self.last = node
24
25      def pop(self):
26          "Return and remove last value."
27          val = self.last.val
28          self.last = self.last.prev
29          return val
30
31
32
33
34  if __name__ == '__main__':
35      mylist = List()
36
37      mylist.append(1)
38      mylist.append(2.)
39      mylist.append("a string")
40      mylist.append(complex(0, 1))
41
42      print "val 4: %s" % (mylist.pop())
43      print "val 3: %s" % (mylist.pop())
44      print "val 2: %s" % (mylist.pop())
45      print "val 1: %s" % (mylist.pop())
```

```
1   #! /usr/bin/env python
2   #
3   # Python list exemple
4
5
6   if __name__ == '__main__':
7       mylist = list()
8
9       mylist.append(1)
10      mylist.append(2.)
11      mylist.append("a string")
12      mylist.append(complex(0, 1))
13
14      print "val 4: %s" % (mylist.pop())
15      print "val 3: %s" % (mylist.pop())
16      print "val 2: %s" % (mylist.pop())
17      print "val 1: %s" % (mylist.pop())
```

G.2 Les tableaux

```
! Exemple de t a b l e Python /Numpy
module tables_ext
  implicit none

  public add_table

  interface add_table
     module procedure add_table_real, add_table_int
  end interface

contains
  function add_table_real(a, b) result(c)
    real, dimension(:), intent(in) :: a,b
    real, dimension(size(a)) :: c
    c = a + b
  end function add_table_real

  function add_table_int(a, b) result(c)
    integer, dimension(:), intent(in) :: a,b
    integer, dimension(size(a)) :: c
    c = a + b
  end function add_table_int
end module tables_ext


program exemple
  use tables_ext

  real, dimension(3) :: a,b
  integer, dimension(3) :: i,j

  a = (/0.1, 0.2, 0.3/)
  b = (/0.4, 0.5, 0.6/)
  print*, "a+b = ", add_table(a, b)

  i = (/1,2,3/)
  j = (/4,5,6/)
  print*, "i+j = ", add_table(i, j)
end program exemple
```

```
$gfortran -i annexe_comp_langages_table.f90 -o
   annexe_comp_langages_table_f
$./annexe_comp_langages_table_f
 a+b =   0.50000000      0.69999999      0.90000004
 i+j =            5           7           9
```

```
/* Exemple de table c */

#include <stdlib.h>
```

```
4  #include <stdio.h>
5
6  void add_table(float *a, float *b, float **c, const int &n)
7  {
8    int i;
9    for (i=0; i<n; i++)
10     (*c)[i] = a[i] + b[i];
11 }
12
13 void add_table(int *a, int *b, int **c, const int &n)
14 {
15   int i;
16   for (i=0; i<n; i++)
17     (*c)[i] = a[i] + b[i];
18 }
19
20
21 int main()
22 {
23   int const n = 3;
24   float a[n] = {0.1, 0.2, 0.3};
25   float b[n] = {0.4, 0.5, 0.5};
26   int i;
27
28   float *c = (float*)malloc(sizeof(float)*n);
29
30   add_table(&(a[0]), &(b[0]), &c, n);
31   printf("a + b = ");
32   for (i=0; i<n; i++)
33     printf("%f ", c[i]);
34   free(c);
35
36   int j[n] = {1, 2, 3};
37   int k[n] = {4, 5, 5};
38
39   int *p = (int*)malloc(sizeof(int)*n);
40
41   add_table(&(j[0]), &(k[0]), &p, n);
42   printf("\nj + k = ");
43   for (i=0; i<n; i++)
44     printf("%i ", p[i]);
45   free(p);
46   printf("\n");
47
48   return 0;
49 }
```

```
1 $c++ annexe_comp_langages_table.c -o annexe_comp_langages_table_c
2 $./annexe_comp_langages_table_c
3 a + b = 0.500000 0.700000 0.800000
```

```
4 j + k = 5 7 8
```

```
1  #! /usr/bin/env python
2  #
3  # Exemple de table Python/Numpy
4
5  import numpy
6
7  def add(a, b):
8      return a+b
9
10 if __name__ == '__main__':
11
12     a = numpy.linspace(0, 1, 9).reshape((3,3))
13     b = numpy.linspace(-1, 0, 9).reshape((3,3))
14     print "a+b = ", add(a, b)
15
16     i = numpy.arange(3)
17     j = numpy.arange(3,6)
18     print "i+j = ", add(i, j)
19
20     print "3+4 = %d" % add(3, 4)
21
22     print "'1'+'1' = '%s'" % add('1', '1')
```

```
1 $python annexe_comp_langages_table.py
2 a+b = [[-1.   -0.75 -0.5 ]
3  [-0.25  0.    0.25]
4  [ 0.5   0.75  1.  ]]
5 i+j = [3 5 7]
6 3+4 = 7
7 '1'+'1' = '11'
```

Annexe H

La programmation orientée objet, application au calcul hybride

L'exemple suivant illustre comment nous pouvons découper l'implémentation de notre code de ligne de transmission en une bibliothèque composée de plusieurs modules plus ou moins élémentaires suivant le principe de la programation orienté object. Tous ces modules sont regroupés dans le module `tltd_h.f90`.

```
1  module tltd_h
2   !** main faitures
3   use define                ! some parameters
4   use segment_o             ! elementary part of a line (physical
        parameter)
5   use field_o               ! electric (i and v) field
6   use port_o                ! ends of a line
7   use boundary_o            ! boundary condition
8   use juction_o             ! specifique boundary (junction beatwen
        lines)
9   use tube_o                ! define a tube (segment + field + 2
        ports)
10  use network_o             ! set of tubes, junctions and boundarys
11  use output_request_o      ! define all output requests
12  use tltd_o                ! simulation manager
13
14  !** some specific sources
15  use generator_o
16  use dipole_o
17  use wave_source_o
18
19  !** some utilities
20  use generator_u
21  use boundary_u
22  use tltd_u
23  use reader_u
```

```
24
25    !** external dep
26    use h5lt
27  end module tltd_h
```

Code Listing H.1 – Point d'entrée de la bibliothèque pour le calcul sur ligne de transmission

Nous n'allons pas passer en revue chacun de ces modules pour en expliquer le contenu. Nous nous arrêterons simplement sur le module `network_o.f90` qui est le plus haut niveau de description d'un réseau. Néanmoins, nous arrivons à percevoir le rôle de chacun de ces modules qui suivent la même structure que le module `network_o` basé sur la programmation orientée objet, déjà présentée.

Le module `network_o` est important car il décrit le réseau de notre problème de ligne de transmission. Le réseau est définit par une structure `network_t` qui regroupe l'ensemble des tubes, jonctions et conditions terminales (lignes 49-59) qui composent le réseau. A cette structure sont associées des fonctions spécifiques permettant d'utiliser ce réseau. Des fonctions génériques (lignes 20-26) permettent de créer et libérer la mémoire. Le deuxième ensemble de fonctions (ligne 28-33) a pour but de créer le réseau à proprement parlé. Les fonctions lignes 35 à 39 permettent d'accéder aux différents éléments du réseau. Les deux dernières fonctions (lignes 41 et 42) ont pour but de mettre à jour les inconnues (un demi pas de temps).

```
13    !------------------------------ header
        ------------------------------------!
14    implicit none
15    private ! all private
16
17    !------------------------------ public interface
        --------------------------!
18    public network_t
19
20    public new_network
21    public free_network
22    public pprint_network
23    public check_network
24
25    public network_print_data
26    public network_save_str
27
28    public network_init_juctions
29    public network_init_tube
30    public network_init_boundary
31    public network_tube_plugin_on_boundary
32    public network_tube_plugin_on_junction
33    public network_get_adapted_boundary_of
34
35    public network_get_nb_tubes
36    public network_get_tube
37    public network_get_ptube
38    public network_get_pboundary
```

```
39    public network_get_pboundarys
40
41    public network_advance_left_half_time
42    public network_advance_right_half_time
43
44    !------------------------------ Types
         ------------------------------------!
45    !
46    ! A 'network' is a set of 'tubes', 'junctions' and 'generators'
         linked
47    ! together
48    !
49    type network_t
50       private
51       integer :: nb_tubes = 0
52       type(tube_t), dimension(:), allocatable :: tubes
53
54       integer :: nb_juctions = 0
55       type(juction_t), dimension(:), allocatable :: juctions
56
57       integer :: nb_boundaries = 0
58       type(boundary_t), dimension(:), allocatable :: boundaries
59    end type network_t
```

Code Listing H.2 – Interface du module Fortran dédiée à la gestion d'un réseau de ligne

Ce découpage ne présente pas de grande difficulté et suit une logique de découpage en entités cohérentes, en limitant l'accès direct au structure interne, ce qui permet de garantir un minimum d'intégrité des données.

Mais le principal intérêt de ce découpage est de pouvoir utiliser ces fonctions dans différents contextes d'utilisation. En effet quelque soit le contexte d'utilisation (seul ou hybridé), notre outil de calcul de ligne de transmission devra au moins être capable :

- de créer un réseau (lignes 28-33)
- de réaliser le calcul à proprement parler (lignes 41 et 42)
- mais également d'accéder aux structures internes du réseau (lignes 35 à 39) pour avoir un contrôle fin sur les tubes et les conditions aux limites, par exemple pour pouvoir faire de l'hybridation ou tout simplement écrire les résultats dans un fichier.

Nous pouvons encapsuler cet ensemble de fonctions dans une bibliothèque informatique qui pourra être utilisée en fonction des besoins. La ligne de commandes suivantes construit la bibliothèque dynamique (options `-shared`) nommée `libtltd.so`, composée de l'ensemble des modules compilés.

```
1  IFORT -Fpic -shared -Wl,-soname,libtltd.so -o libtltd.so
       amelet_hdf_u.o boundary_o.o boundary_u.o define.o dipoles_o.o
       field_o.o generator_o.o generator_u.o juction_o.o main_u.o
       network_o.o output_request.o port_o.o power_wave_o.o
       reader_sdf_o.o reader_u.o segment_o.o tltd_h.o tltd_o.o tltd_u.
       o tube_o.o wave_source.o
```

Nous pouvons utiliser cette bibliothèque `libtltd.so` dans différents contextes, par exemple pour résoudre un problème de ligne de transmission en temporelle. Nous pouvons également l'utiliser comme moteur 1D pour les fils de R. Holland, ou enfin pour mettre en place une stratégie d'hybridation de découpage de domaines.

Bibliographie

[1] D. Prost (Onera) V. Enjalbert (DGA) L. Mohedano (APAVE) F. Issac, E. Bachelier. Space launching site protection against lightning hazards. *Aerospace Lab Journal*, December 2012.

[2] E. Bachelier, F. Issac, S. Bertuol, JP. Parmantier, and JC. Alliot. Numerical EM simulations for the definition of the lightning protection systems of the future VEGA and SOYOUZ launching pads. *ESA Workshop on Aerospace EMC*, March 2009.

[3] C. GARY. Approche complète de la propagation multifilaire en haute fréquence par utilisation des matrices complexes. bulletin de la direction des études et recherches. Technical report, EDF, 1976.

[4] F. Rachidi. *Effets électromagnétiques de la foudre sur les lignes de transmission aériennes - Modélisation et simulation*. PhD thesis, École Polytechnique Fédérale de Lausanne, 1991.

[5] David ORZAN. *Couplage externe et interne entre un champ électromagnétique et un réseau de lignes multifilaires*. PhD thesis, École Polytechnique Fédérale de Lausanne, 1998.

[6] A DERI, G TEVAN, and G VARJU. Calcul du couplage inductif entre lignes de transmission. Technical report, Rail International, November 1985.

[7] MORIN Eric. *Modélisation d'un réseau électrique de tramway : du composant au système*. PhD thesis, Electronique, Electrotechnique, Automatique, Télécommunication, Signal, January 2005.

[8] enjoyspace. Soyouz en guyane. http://www.enjoyspace.com/fr/dossiers/soyouz-en-guyane, 2012. [Online ; accessed 30-Mai-2012].

[9] Hirf se - high intensity radiated field synthetic environment.

[10] A. Claudi Y. Méndez Hernández, G. Drobnjak and M. Kizilcay. An engineering approach in modeling lightning effects on megawatt-class onshore wind turbines using emtp and models. *PIERS*.

[11] L.C. Montanes Bellosta, M.A. Garcia Garcia, A. Llombart Estopinan, M. Sanz Badia, and M. Garcia-Gracia. Simulation of surges on power lines using spice and emtp : a comparative study. In *Electrotechnical Conference, 1998. MELECON 98., 9th Mediterranean*, volume 1, may 1998.

[12] Kane Yee. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *Antennas and Propagation, IEEE Transactions on*, 14 (3) :302 –307, may 1966. ISSN 0018-926X.

[13] X. Ferrieres, A. Cosnuau, J.C. Alliot, and P. Leca. Code solving the Maxwell equations in the time domain, implanted on distributed-memory type machine. *Aerospace Science and Technology*, 1(6) :425–439, 1997.

[14] X. Ferrieres and B.L. Michielsen. Efficient solution of the transmission line euations for large networks. 7ième colloque International CEM, Toulouse, 2-4 mars, 1994.

[15] D. Lacour. *Etude du couplage d'une onde électromagnétique sur une structure complexe par décomposition en sous-domaines.* Thèse de doctorat de l'Université Paris XI, 1999.

[16] D. Lacour, X. Ferrieres, P. Bonnet, V. Gobin, and JC. Alliot. Application of multi-domain decomposition method to solve EMC problem on an aeroplane. *Electronics Letters*, 33(23) :1932–1933, 1997.

[17] V. Mouysset. *Une méthode de sous-domaines pour la résolution des équations de Maxwell instationnaire en présence d'un ensemble non-connexe d'objets diffractant.* Thèse de Doctorat de l'Université Paul Sabatier de Toulouse, 2006.

[18] V. Mouysset, P. Borderies, X. Ferrieres, and P.A. Mazet. Simulation of multiple scattering scenes for time domain maxwell's equations by an hybrid and parallel method. PIERS proceeding, Hangzhou, china, march 24-28, 2008.

[19] R. Holland and L. Simpson. Finite-difference analysis of emp coupling to thin struts and wires. *Electromagnetic Compatibility, IEEE Transactions on*, EMC-23(2) :88 –97, may 1981. ISSN 0018-9375.

[20] A. Taflove and S. C. Hagness. *Computational Electrodynamics : The Finite-Difference Time-Domain Method.* Norwood, MA : Artech House.

[21] X. ; Pernet S. ; Michielsen B.L. ; Rogier F. ; Degond P. Pebernet, L. ; Ferrieres. Discontinuous galerkin method applied to electromagnetic compatibility prob- lems : introduction of thin wire and thin resistive material models. *Science, Measurement and Technology, IET*, pages 395–401, 2008.

[22] Ian Jeffrey. Finite-volume simulations of maxwell's equations on unstructured grids, 2011.

[23] J.-P. Berenger. A multiwire formalism for the fdtd method. *Electromagnetic Compatibility, IEEE Transactions on*, 42(3) :257 –264, aug 2000. ISSN 0018-9375.

[24] Gunnar Ledfelt. Hybrid time-domain methods and wire models for computational electromagnetics, 2001.

[25] F. Edelvik. A new technique for accurate and stable modeling of arbitrarily oriented thin wires in the FDTD method. *Electromagnetic Compatibility, IEEE Transactions on*, 45 :416–423, May 2003. ISSN 0018-9375,.

[26] C. Guiffaut, A. Reineix, and B. Pecqueux. New oblique thin wire formalism in the FDTD method with multiwire junctions. *Antennas and Propagation, IEEE Transactions on*, 60 :1458 – 1466, March 2012. ISSN 0018-926X.

[27] Edward F. Vance. *Coupling to shielded cables / Edward F. Vance.* Wiley, New York :, 1978. ISBN 0471041076. x, 183 p. : pp.

[28] C. R. Paul. *Analysis of multiconductor transmission lines.* 1994.

[29] TIBTECH innovations. Propriété des fibres métalliques. http://www.tibtech.com/fr/conductivite.php, 2011. [Online ; accessed 16-Feb-2012].

[30] E.F. Vance. Shielding effectiveness of braided wire shields. Research Report 172, Standford Research Institute, 1974.

[31] P. Degauque. *Compatibilité électromagnétique : des concepts de base aux applications.* Collection technique et scientifique des télécommunications. Hermès Science, 2007. ISBN 9782746216372.

[32] J.C. Clements, C.R. Paul, and A.T. Adams. Computation of the capacitance matrix for systems of dielectric-coated cylindrical conductors. *Electromagnetic Compatibility, IEEE Transactions on*, EMC-17(4) :238 –248, nov. 1975. ISSN 0018-9375.

[33] C.R. Paul and A.E. Feather. Computation of the transmission line inductance and capacitance matrices from the generalized capacitance matrix. *Electromagnetic Compatibility, IEEE Transactions on*, EMC-18(4) :175 –183, nov. 1976. ISSN 0018-9375.

[34] J.P. PARMANTIER, S. BERTUOL, X. FERRIERES, and B. MICHIELSEN. *Améliorations dans le logiciel CRIPTE : code LAPLACE, Création de tubes, Calcul en des positions intermédiaires sur des tubes.* Rapport ONERA no RTS 12/6767 PY, Février 1997.

[35] Kane Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *Antennas and Propagation, IEEE Transactions on*, 14 (3) :302 –307, may 1966. ISSN 0018-926X.

[36] J.A. Pereda, O. Garcia, A. Vegas, and A. Prieto. Numerical dispersion and stability analysis of the FDTD technique in lossy dielectrics. *Microwave and Guided Wave Letters, IEEE*, 8(7) :245 –247, jul 1998. ISSN 1051-8207.

[37] Karl S. Kunz and Raymond J. Luebbers. *The Finite Difference Time Domain Method for Electromagnetics.* CRC Press, Boca Raton, 1993.

[38] J. Charney, R. Fjörtoft, and J. Von Neumann. Numerical integration of the barotropic vorticity equation. *Tellus A*, 2(4), 2011.

[39] Christophe Guiffaut, Alain Reineix, P. Boudy, B. Pecqueux, and J.C. Joly. Incorporation d'impédances quelconques et de quadripôles passifs dans la méthode FDTD pour la modélisation des filtres de protection foudre. In *14ème Colloque International et Exposition sur la Compatibilité Electromagnétique et Journées Scientifiques d'URSI France - CEM 2008 - CNFRS/URSI.*

[40] Jean-Paul Vannel, Bernard Pecqueux, Christophe Guiffaut, Jean-Pierre Adam, and GIRAUDON Cyril. Modelling and simulation of wire protections using a time domain fdtd code (gorf3d). In *EUROEM 2012.*

[41] J.R. CARSON. *Wave propagation in overhead wires with earth return*, volume 5, pages 539–544. Bell System Technical Journal, October 1926.

[42] M.A. Mrozowski, M. ; Stuchly. Parameterization of media dispersive properties for FDTD. *Antennas and Propagation, IEEE Transactions on*, pages 1438–1439, sep 1997.

[43] E.C. Levi. Complex-curve fitting. *IRE Trans. on Automatic Control*, AC-4 :37–44, 1959.

[44] J.E. Jr. Dennis and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, Englewood Cliffs, 1983.

[45] A.V. Oppenheim and R.W. Schafer. *Digital Signal Processing.* Prentice-Hall, Englewood Cliffs, 1975. 56 pp.

[46] R.E. Miller, J.W. Thatcher, United States. Office of Naval Research. Mathematics Program, IBM World Trade Corporation, and International Business Machines Corporation. Mathematical Sciences Dept. *Complexity of Computer Computations : Proceedings.* IBM Research Symposia Series. Plenum Press, 1972. ISBN 9780306307072. LCCN 72085836.

[47] R.J. Kelley, D.F. ; Luebbers. Debye function expansions of empirical models of complex permittivity for use in FDTD solutions. *Antennas and Propagation Society International Symposium, IEEE*, 4 :372–375, june 2003.

[48] T.J. Luebbers R.J. Bucknell Univ. Lewisburg Kelley, D.F. Destan. Debye function expansions of complex permittivity using a hybrid particle swarm-least squares optimization approach. *Antennas and Propagation, IEEE Transactions on*, TAP-55 : 1999 – 2005, july 2007. ISSN 0018-926X.

[49] S. Twomey. *Introduction to the Mathmatics of Inversion in Remote Sensing and Indirect Measurements.* Elsevier Scientific, Amsterdam, The Nederlands, 1977.

[50] L.A. ; Vegas A. ; Prieto A. ; Dept. de Electron. Cantabria Univ. Santander Pereda, A. ; Vielva. Analyzing the stability of the FDTD technique by combining the von Neumann method with the Routh-Hurwitz criterion. *Microwave Theory and Techniques, IEEE*, 49(2) :377–381, Feb 2001. ISSN 0018-9480.

[51] J.-P. Berenger. A multiwire formalism for the FDTD method. *Electromagnetic Compatibility, IEEE Transactions on*, 42(3) :257 –264, aug 2000. ISSN 0018-9375.

[52] E.K Miller, A.J Poggio, and G.J Burke. An integro-differential equation technique for the time-domain analysis of thin wire structures. i. the numerical method. *Journal of Computational Physics*, Volume 12, Issue 1, May 1973.

[53] P. Degauque and J. Hamelin. *Compatibilité électromagnétique : bruits et pertubations radioeléctriques.* Collection technique et scientifigue des télécommunications. Dunod, 1990. ISBN 9782040188078.

[54] D.K. McLain M.A. Uman. Magnetic field of lightning return stroke. *J. Geophys. Res.*, page 6899–6910.

[55] C. Huygens. *Traité de la lumière.* Gauthier-Villars et Cie Editeurs, Paris, 1920.

[56] Viencent MOUYSSET. *Une méthode de sous-domaines pour la résolution des équations de Maxwell instationnaires en présence d'un ensemble non-connexe d'objets diffractant.* PhD thesis, Universite Toulouse III, December 2006.

[57] Torleif Martin. Modified fresnel coefficients for huygens' sources in fdtd. *Applied Computational Electromagnetics Society Journal.*, 17(1), Mar 2002.

[58] V. Cooray. Horizontal fields generated by return strokes. *Radio Science*, vol. 27 : 529–537, 1992.

[59] M. Rubinstein. An approximation formula for the calculation of the horizontal electric field from lightning at close, intermediate and long range. *IEEE Trans. On EMC*, vol. 38 :531, 1996.

[60] FreeCAD. Freecad - a parametric 3d cad modeler. http://sourceforge.net/apps/mediawiki/free-cad/index.php?title=Main_Page, 2013.

[61] IBM. Best practices for programming in c. http://www.ibm.com/developerworks/aix/library/au-hook_duttaC.html, 2013. [Online ; accessed 4-January-2013].

[62] A. Braquelaire and J.P. Braquelaire. *Méthodologie de la programmation en C : Bibliothèque standard, API POSIX*. Sciences sup. Dunod, 2000. ISBN 9782100047826.

[63] T. Ziadé. *Programmation Python : Conception et optimisation*. Eyrolles, 2009. ISBN 9782212852424.

[64] Matthias Felleisen. On the expressive power of programming languages. In *Science of Computer Programming*, pages 134–151. Springer-Verlag, 1990.

[65] T. Ziadé. *Python : Petit guide à l'usage du développeur agile*. InfoPro. Études, développement, intégration. Dunod, 2007. ISBN 9782100508839.

[66] Emmanuel Delahaye. Bonnes pratiques de codage en langage c. http://emmanuel-delahaye.developpez.com/tutoriels/c/bonnes-pratiques-codage-c/, 2009. [Online ; accessed 24-Feb-2012].

[67] Roedy Green. How to write unmaintainable code. http://thc.org/root/phun/unmaintain.html. [Online ; accessed 24-Feb-2012].

[68] Lafayette Ron Hubbill. Protégez votre emploi ! *Le Virus Informatique*, (22).

[69] Polyhedron sofware. http://www.polyhedron.com, 2012. [Online ; accessed 09-May-2012].

[70] K. Beck. *Test Driven Development : By Example*. The Addison-Wesley Signature Series. Addison-Wesley, 2003. ISBN 9780321146533. LCCN 02028037.

[71] Bruno Orsier. Tutoriel : Développement dirigé par les tests. http://bruno-orsier.developpez.com/tutoriels/TDD/pentaminos/, 2007. [Online ; accessed 24-Feb-2012].

[72] David Boissier. Développement dirigé par les tests : mise en pratique. http://dboissier.developpez.com/tutoriels/test-driven-development/, 2009. [Online ; accessed 24-Feb-2012].

[73] John Doe. Object-oriented scientific programming with fortran 90. In Jonathan Schaeffer, editor, *High Performance Computing Systems and Applications*, pages 47–58. Springer US, 1998.

[74] Some lesser-known truths about programming. http://automagical.rationalmind.net/2010/08/17/some-lesser-known-truths-about-programming/. [Online ; accessed 12-Nov-2012].

[75] La gestion de projet informatique. http://www.gestion-projet-informatique.vivre-aujourdhui.fr/. [Online ; accessed 21-Oct-2012].

[76] Wikipédia. Optimisation des performances des architectures multi-cœurs — wikipédia, l'encyclopédie libre. http://fr.wikipedia.org/w/index.php?title=Optimisation_des_performances_des_architectures_multi-c%C5%93urs&oldid=84354161, 2012. [En ligne ; Page disponible le 12-novembre-2012].

[77] Pypy. http://pypy.org/, 2012. [Online ; accessed 24-Feb-2012].

[78] The julia language. http://julialang.org. [Online ; accessed 09-May-2012].

[79] The computer language benchmarks game. http://shootout.alioth.debian.org/, 2012. [Online ; accessed 24-Feb-2012].

[80] Mythical man month 10 lines per developer day - how close on large projects ? http://stackoverflow.com/questions/966800/mythical-man-month-10-lines-per-developer-day-how-close-on-large-projects. [Online ; accessed 12-Nov-2012].

[81] Wikipedia. Pareto principle — wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Pareto_principle&oldid=531367625, 2013. [Online ; accessed 19-January-2013].

[82] K.R. Wadleigh and I.L. Crawford. *Software Optimization for High-Performance Computers*. HP Professional Series. Prentice Hall, 2000. ISBN 9780130170088. LCCN 00709512.

[83] Cython - c-extension for python. http://cython.org/.

[84] F2py : Fortran to python interface generator. http://cens.ioc.ee/projects/f2py2e.

[85] Neilen Marais. Driving and extending legacy codes using python. *IEEE Antennas and Propagation Magazine*, Vol. 49(No. 1), February 2007.

[86] Fwrap fortran for speed, python for comfort - python(cython(c(fortran))). http://fwrap.sourceforge.net/.

[87] Nvidia. What is gpu computing. http://www.nvidia.com/object/what-is-gpu-computing.html, 2013. [Online ; accessed 4-January-2013].

[88] Compute unified device architecture. http://www.nvidia.fr/object/cuda_home_new_fr.html.

[89] Opencl - the open standard for parallel programming of heterogeneous systems. http://www.khronos.org/opencl/.

[90] Wikipedia. Pipeline (computing) — wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Pipeline_(computing)&oldid=517237092, 2012. [Online ; accessed 18-November-2012].

[91] The message passing interface (mpi) standard. http://www.mcs.anl.gov/research/projects/mpi/.

[92] Zeromq - the intelligent transport layer. http://www.zeromq.org/.

[93] Openpalm. http://www.cerfacs.fr/globc/PALM_WEB/.

[94] Wikipédia. Labview — wikipédia, l'encyclopédie libre. http://fr.wikipedia.org/w/index.php?title=LabVIEW&oldid=84614620, 2012. [En ligne ; Page disponible le 20-février-2013].

[95] The message passing interface (mpi) standard. http://www.ni.com/labview/f/.

[96] Pierre Bonnet. Résolution des équation de maxwell instationnaires et harmoniques par une technique de volumes finies. application à des problèmes de compatibilité électromagnètique. Research Report, ONERA, 1998.

[97] Kurokawa. Power waves and the scattering matrix.

Notations

De manière générale, les quantités (inconnues), qui dépendent du temps et de l'espace, sont notées en minuscule (ex. i, v), à l'exception faite du champ électrique et magnétique, qui se note habituellement en lettre capitale $\boldsymbol{E}$ et $\boldsymbol{H}$. De plus, comme le nombre de l'exemple précédent, les quantités vectorielles sont en lettre grasse. Il est également d'habitude (dans la mesure du possible) de noter, en lettre capitale, les matrices et en minuscule les vecteurs (ex. $\boldsymbol{M} \cdot \boldsymbol{x} = \boldsymbol{y}$). Les composantes d'un vecteur sont généralement représentées par un indice (E_x), mais parfois le produit scalaire avec un vecteur unitaire est utilisé pour plus de clarté ($\boldsymbol{E} \cdot \boldsymbol{u_x}$). Les puissances ^{T} et $^{-1}$ représentent respectivement l'opérateur transposé et l'inverse de la quantité vectorielle. Nous avons également choisi la notation implicite des opérateurs vectoriels comme indiqué dans le tableau suivant.

En ce qui concerne la notation des inconnues dans un espace discrétisé, nous avons choisi d'utiliser une lettre capitale et une barre verticale pour séparer les indices de la discrétisation (ex. $v_1((n+1/2)\Delta t, p\Delta l) = V_1|_p^{n+1/2}$).

Dans un souci de concision, l'opérateur différentiel $\partial f/\partial x$ a été remplacé par $\partial_x f$.

Enfin, plusieurs symboles sont utilisés pour représenter une quantité duale ou assimilée à une autre. Ainsi, $\widetilde{x}$ est la représentation harmonique de la quantité x, $\hat{x}$ l'approximation de x.

$\mathbb{N}, \mathbb{N}^*$	ensemble des entiers naturels, des entiers strictement positifs
$\mathbb{R}, \mathbb{R}_+$	ensemble des réels et des réels positifs
$\mathbb{R}^d$	ensemble des vecteurs réels à d dimensions
$\mathrm{card}(\Omega)$	cardinal de l'ensemble Ω
$\cdot$	le produit scalaire
$\wedge$	le produit vectoriel
∇	l'opérateur Nabla (En trois dimensions cartésiennes $(\partial_x, \partial_y, \partial_z)^T$
$\boldsymbol{\nabla}\wedge$	l'opérateur rotationel
$\boldsymbol{\nabla}\cdot$	l'opérateur divergence
$\boldsymbol{\nabla} f$	le gradient de f
$\mathbf{1}$	Matrice identité
δ	Le symbole de Kronecker
$*$	Le produit de convolutions (usage : $(f * g)(t) = \int_{-\infty}^{+\infty} f(\tau - t) g(t) d\tau'$)
c_0	Vitesse de la lumière dans le vide ($= 299792458\mathrm{ms}^{-1}$)
μ_0	Perméabilité magnétique du vide ($= 4\pi - 7\mathrm{kgmA}^{-2}\mathrm{s}^{-2}$ ou Hm^{-1})
ϵ_0	Permittivité diélectrique du vide ($= 1/\mu_0 c_0^2 \approx 8.85418781710^{-12}\mathrm{A}^2\mathrm{s}^4\mathrm{kg}^{-1}\mathrm{m}^{-3}$ ou Fm^{-1})
Z_0	Impédance caractéristique du vide ($= \mu_0 c_0 \approx 376.730313461\mathrm{kgm}^2\mathrm{A}^{-1}\mathrm{s}^{-3}$)

Titre Stratégies d'hybridation FDTD/MTL de méthodes de simulation électromagnétique - Application à l'étude de grands systèmes complexes -

Résumé Dans ce mémoire, nous présentons une stratégie basée sur une approche hybride dans le domaine temporel, couplant une méthode de résolution des équations de Maxwell dans le domaine 3D (FDTD) avec une méthode de résolution des équations de ligne de transmission, afin de pouvoir simuler des problèmes électromagnétiques de grande échelle. Le mémoire donne les éléments d'hybridation pour deux cadres d'utilisation de cette approche : une approche multi-domaine et une approche multi-résolution ou d'échelle.

L'approche multi-domaine est une extension de la méthode FDTD 3D à plusieurs sous-domaines reliés par des structures filaires sur lesquelles on résout une équation de lignes de transmission par un formalisme FDTD 1D. La difficulté est d'abord d'avoir une définition implicite du champ électromagnétique dans la théorie des lignes de transmission, et d'autre part de prendre en compte les effets du sol sur les courants induits au niveau des lignes et sur les champs électromagnétiques.

L'approche multi-résolution ou d'échelle est conçue pour étendre les capacités de la méthode FDTD au traitement du routage de câbles complexes ayant une section plus petite que la taille de la cellule. Ce mémoire présente différentes techniques pour évaluer les paramètres de la ligne, basées sur la résolution d'un problème de Laplace 2D, ainsi qu'une méthode de couplage champs/câbles basée sur le courant de mode commun.

L'ensemble de ce travail nous a permis de proposer une méthode numérique efficace pour calculer les effets électromagnétiques induits par une source (type onde plane ou dipolaire) sur des sites de grande dimension, composés de plusieurs bâtiments reliés entre eux par un réseau de câbles. Dans ce cadre une application à la foudre a été réalisée.

Mots-clés Équation de Maxwell instationnaires, Hybridation forte, schéma différence finie, Foudre, Infrastructure terrestre, Ligne de transmission multi-conducteurs, Couplage champ/câbles, Modèle de sol

Title A time domain hybrid FDTD/MTL approach to study electromagnetic effects on interconnected ground installations

Abstract In this thesis, we present a strategy based on a hybrid approach in the time domain, by coupling 3D method (FDTD) with a multi-conductors transmission line (MTL) method, in order to simulate complex large scale electromagnetic problems. This report gives the theoretical and numerical elements for coupling these approaches for two kind of problems, which are the multi domains approach and the multi scale approach. The multiple domains approach is an extension of the classical FDTD method taking into account several 3D subdomains, interconnected by a wire network, on which a 1D transmission line formalism is used. The main issues are, on one hand to have an implicit expression of the electromagnetic field in the transmission line approach, and on the other hand to be able to take into account the ground effects on the induced currents, on the transmission line parameters and on the electromagnetic field. The multi scale approach is developed to extend the capabilities of FDTD to deal with complex cables routing. We assume that the cross section of the cables are smallest than the cell size, and in these problems, the 1D transmission line problem is physically included in the 3D global computational domain. The work done in this thesis leaded to a new field to transmission line coupling based on the common mode current, and an evaluation of the transmission. line parameters based on a Laplace equation resolution in 2D. In this work, we have elaborated and proposed efficient numerical strategies for the computation of electromagnetic induced effects on large and complex sites, composed of several interconnected distant buildings. An application to lightning problems have been done.

Keywords Unsteady Maxwell's Equations, Strong Hybridization, Finite Difference Scheme, Lightning, Ground Infrastructure, Multiconductor Transmission Line, Field-to-Wire Couplings, Soil Model