

TESIS DOCTORAL

**Uso de arquitecturas MIC para la
aceleración de soluciones
numéricas en electromagnetismo**

Autor:

Juan Luis CAMPÓN
MOZO

Conforme, el Director:

Dr. Luis LANDESA
PORRAS

**** La conformidad del director/es de la tesis consta en el original en
papel de esta Tesis Doctoral*

Programa de Doctorado en Tecnología Aeroespacial:
Ingenierías Electromagnética, Electrónica,
Informática y Mecánica.

«Cada día sabemos más y entendemos menos.»

Albert Einstein

UNIVERSIDAD DE EXTREMADURA

Resumen

ESCUELA INTERNACIONAL DE POSTGRADO UEx

Programa de Doctorado en Tecnología Aeroespacial: Ingenierías
Electromagnética, Electrónica, Informática y Mecánica.

Uso de arquitecturas MIC para la aceleración de soluciones numéricas en electromagnetismo

por Juan Luis CAMPÓN MOZO

La mejora en la eficiencia de recursos computacionales para la resolución de problemas electromagnéticos es un tema complejo y de gran interés. La aparición en la última década de GPUs (*Graphics Processing Unit*) y tarjetas coprocesadoras Xeon Phi en las listas de los supercomputadores con mayor rendimiento, ha llevado a los investigadores a tratar de sacar el máximo provecho de estas nuevas tecnologías.

El objetivo principal de esta Tesis es mejorar la eficiencia del método MoM (*Method of Moments*) mediante la paralelización de algunos de sus algoritmos en procesadores con arquitectura Intel MIC (*Many Integrated Core*). Para ello, se realiza el modelado de un problema electromagnético mediante la metodología SIE-MoM (*Surface Integral Equation-Method of Moments*), y se desarrollan nuevos algoritmos para su ejecución en tarjetas coprocesadoras Intel Xeon Phi.

Los resultados obtenidos tras evaluar los tiempos de computación comparativamente entre las tarjetas Intel Xeon Phi y las CPUs Intel Xeon, indican que la arquitectura Intel MIC podría resultar adecuada en simulaciones electromagnéticas como complemento a CPUs.

UNIVERSIDAD DE EXTREMADURA

Abstract

ESCUELA INTERNACIONAL DE POSTGRADO UEx

Programa de Doctorado en Tecnología Aeroespacial: Ingenierías
Electromagnética, Electrónica, Informática y Mecánica.

Uso de arquitecturas MIC para la aceleración de soluciones numéricas en electromagnetismo

by Juan Luis CAMPÓN MOZO

Improving the efficiency of computational resources for solving electromagnetic problems is a complex subject of great interest. The growth of GPUs (*Graphics Processing Unit*) and Xeon Phi coprocessor boards on the lists of top-performing supercomputers over the past decade has led researchers to try to make the most of these new technologies.

The main objective of this Thesis is to improve the efficiency of the MoM method by parallelizing some of its algorithms on processors with Intel MIC (*Many Integrated Core*) architecture. For this purpose, the modeling of an electromagnetic problem is carried out using the SIE-MoM (*Surface Integral Equation-Method of Moments*) methodology, and new algorithms are developed for their execution on Intel Xeon Phi coprocessor cards.

The results obtained after evaluating computation time compared between Intel Xeon Phi cards and Intel Xeon CPUs, indicate that the Intel MIC architecture could be suitable in electromagnetic simulations as a complement to CPUs.

Lista de publicaciones

Publicaciones en revistas internacionales:

- J. L. Campon y Luis Landesa (2018). «Fast solution of electromagnetic scattering problems using Xeon Phi coprocessors». En: *The Journal of Supercomputing* 75, págs. 370-383

Contribuciones a congresos internacionales:

- L. Landesa, J. M. Taboada, J. L. Campon et al. (2017). «Successes and frustrations in the solution of large electromagnetic problems in supercomputers». En: *2017 International Applied Computational Electromagnetics Society Symposium - Italy (ACES)*. PONENCIA INVITADA. Págs. 1-2. DOI: [10.23919/ROPACES.2017.7916307](https://doi.org/10.23919/ROPACES.2017.7916307)

Agradecimientos

En primer lugar, me gustaría comenzar agradeciendo a Luis, el haberme brindado la confianza para realizar esta Tesis. Sin duda alguna, sin su conocimiento, su apoyo continuo y su vocación como docente, esta Tesis no habría sido posible.

No puede faltar el agradecimiento a mis padres, por su apoyo y por haberme regalado con su esfuerzo y sacrificio una educación y un modelo a seguir.

Por último, me gustaría agradecer a mi esposa y compañera de vida, por aguantarme en esos momentos complicados de desesperación y por mostrarme su apoyo incondicional en éste y en otros proyectos.

Gracias a todos.

Índice general

| | |
|---|------------|
| Resumen | v |
| Abstract | vii |
| Lista de publicaciones | ix |
| 1 Introducción | 1 |
| 1.1 Justificación y relevancia del estudio | 2 |
| 1.2 Propósito y objetivos | 3 |
| 1.3 Estructura de la memoria | 4 |
| 2 Metodología SIE-MoM | 7 |
| 2.1 Introducción | 8 |
| 2.2 Ecuaciones de Maxwell-Hertz-Heaviside | 9 |
| 2.2.1 Formulación | 9 |
| 2.2.2 Condiciones de contorno e iniciales | 12 |
| 2.3 Radiación electromagnética | 13 |
| 2.3.1 Formulación en volumen | 13 |
| 2.3.2 Operadores L_V y K_V | 15 |
| 2.3.3 Formulación en superficie | 16 |
| 2.4 Formulación integral de superficie | 19 |
| 2.5 Método de los Momentos | 25 |
| 2.5.1 Formulación | 25 |
| 2.5.2 Funciones base RWG | 28 |
| 2.5.3 Discretización de la SIE | 29 |
| 2.6 Extensión de la formulación a múltiples objetos | 33 |
| 3 Métodos iterativos | 39 |
| 3.1 Introducción y conceptos básicos | 40 |
| 3.2 Métodos estacionarios | 41 |
| 3.2.1 Convergencia | 43 |
| 3.3 Métodos de Krylov | 44 |

| | | |
|----------|---|-----------|
| 4 | Intel Many Integrated Core (MIC) | 47 |
| 4.1 | Evolución de las familias Xeon Phi | 48 |
| 4.2 | Arquitectura hardware Knights Corner (KNC) | 49 |
| 4.2.1 | Detalle de un core KNC | 50 |
| 4.2.2 | Jerarquía de cachés | 52 |
| 4.3 | Arquitectura software Knights Corner (KNC) | 53 |
| 4.3.1 | Manycore Platform Software Stack (MPSS) | 54 |
| 4.4 | Modelos de programación para coprocesadores Intel Xeon Phi | 55 |
| 4.5 | Paralelismo en coprocesadores Intel Xeon Phi | 56 |
| 4.5.1 | Paralelismo de datos | 56 |
| 4.5.2 | Paralelismo de tareas | 58 |
| 5 | Electromagnetismo computacional en arquitecturas manycore | 61 |
| 5.1 | Resultados con GPUs | 63 |
| 5.1.1 | FDTD | 63 |
| 5.1.2 | FEM | 64 |
| 5.1.3 | MoM | 66 |
| 5.2 | Resultados con Intel MIC | 69 |
| 5.2.1 | FDTD | 69 |
| 5.2.2 | FEM | 70 |
| 5.2.3 | MoM | 71 |
| 6 | Aceleración de MoM en arquitecturas Intel MIC | 73 |
| 6.1 | Especificaciones de la estación de trabajo | 75 |
| 6.2 | Paralelismo del código HEMCUVE en arquitecturas Intel MIC | 76 |
| 6.2.1 | Patrón de acceso a memoria y consideraciones sobre el bus PCIe | 76 |
| 6.2.2 | Optimizaciones escalares | 77 |
| 6.2.3 | Vectorización | 78 |
| 6.2.4 | Optimizaciones en código paralelo de memoria com- partida | 79 |
| 6.3 | Algoritmos mGMRES y mMVP | 80 |
| 6.4 | Geometría fija con un único objeto dispersor | 84 |
| 6.4.1 | Formulación SIE-MoM | 84 |
| 6.4.2 | Cálculo del emplazamiento óptimo de la fuente puntual Resultados | 87 89 |
| 6.4.3 | Cálculo de backscattering de una onda plana Resultados | 92 93 |
| 6.5 | Geometría variable con varios objetos dispersores | 94 |

| | | |
|----------|---|------------|
| 6.5.1 | Formulación SIE-MoM | 94 |
| 6.5.2 | Cálculo de la distribución óptima de un array de dis- persores | 97 |
| | Resultados | 98 |
| 7 | Conclusiones y líneas futuras | 101 |
| 7.1 | Conclusiones | 102 |
| 7.2 | Líneas futuras | 103 |
| | Bibliografía | 105 |

Índice de figuras

| | | |
|-----|---|----|
| 2.1 | Superficie de discontinuidad entre dos medios diferentes R_1 y R_2 | 12 |
| 2.2 | Geometría volumétrica del problema de radiación electromagnética | 15 |
| 2.3 | Geometría superficial del problema de radiación electromagnética | 17 |
| 2.4 | Detalle de la aproximación de r a S_δ | 18 |
| 2.5 | Geometría superficial con un dispersor y problema equivalente | 20 |
| 2.6 | Rao-Wilton-Glisson (RWG) basis function. | 28 |
| 2.7 | Geometría superficial del problema con múltiples objetos | 35 |
| 3.1 | Descomposición aditiva de la matriz A | 42 |
| 4.1 | Topología de un core KNC | 51 |
| 4.2 | Arquitectura Knights Corner | 53 |
| 4.3 | Modelos de programación sobre tarjetas coprocesadoras Intel Xeon Phi. | 56 |
| 6.1 | Extracto del informe de vectorización. | 79 |
| 6.2 | Paso de mensajes MPI entre CPU y Xeon Phi. | 81 |
| 6.3 | Detalle del cálculo masivo de los M MVPs en la Xeon Phi | 83 |
| 6.4 | Creación del objeto dispersor a partir de dos esferas. | 89 |
| 6.5 | Campo disperso por un objeto de oro ($\epsilon = -8 - j1,66$ a la frecuencia de simulación) en el plano $Z=0$. $\lambda=550\text{nm}$ | 90 |
| 6.6 | RCS de un objeto de oro ($\epsilon = -8 - j1,66$ a $\lambda=550\text{nm}$) iluminado con una onda plana. | 93 |
| 6.7 | Array de cubos ($\epsilon = 1 - j1000$) que maximizan el campo electromagnético. | 99 |

Índice de tablas

| | | |
|-----|---|----|
| 2.1 | Valores de coeficientes para distintas formulaciones. | 24 |
| 6.1 | Tiempo de ejecución del algoritmo mMVP en Xeon Phi | 91 |
| 6.2 | Comparativa de rendimiento entre Intel Xeon e Intel Xeon Phi | 91 |

Abreviaturas

| | |
|------------------|---|
| SIE-MoM | Surface Integral Equation-Method of Moments |
| FDTD | Finite Difference Time Domain |
| FEM | Finite Element Method |
| RAM | Random Access Memory |
| GMRES | Generalized Minimum Residual Method |
| CGS | Conjugate Gradient Squared |
| MVP | Matrix Vector Product |
| FMM | Fast Multipole Method |
| MLFMA | Multilevel Fast Multipole Algorithm |
| HEMCUVE | Hybrid ElectroMag. Code University of Vigo and Extremadura |
| N-EFIE | Normal Electric Field Integral Equation |
| T-EFIE | Tangential Electric Field Integral Equation |
| N-MFIE | Normal Magnetic Field Integral Equation |
| T-MFIE | Tangential Magnetic Field Integral Equation |
| CFIE | Combined Field Integral Equation |
| JCFIE | Electric current (J) Combined Field Integral Equation |
| MCFIE | Magnetic current (M) Combined Field Integral Equation |
| JMCFIE | Elec. and mag. current (J, M) Combined Field Integral Equation |
| PMCHWT | Poggio-Miller-Chang-Harrington-Wu-Tsai |
| CTF | Combined Tangential Formulation |
| CNF | Combined Normal Formulation |
| RWG | Rao Wilton Glisson |
| CBF | Charasteristic Basis Function |
| SOR | Successive Over Relaxation |
| CG | Conjugate Gradients |
| FOM | Full Orthogonalization Method |
| Bi-CG | Biconjugate Gradient Method |
| CGS | Conjugate Gradient Squared |
| Bi-CGSTAB | Biconjugate Gradient Stabilized |
| MVP | Matrix Vector Product |
| MIC | Many Integrated Core |

| | |
|---------------|--|
| GPU | Graphics Processing Unit |
| PCIe | Peripheral Component Interconnect express |
| GDDR | Graphics Double Data Rate |
| MCDRAM | MultiChannel Dynamic Random Access Memory |
| LGA | Land Grid Array |
| SFF | Small Form Factor |
| FLOPS | Floating Point Operations per Second |
| KNC | Knights Corner |
| KNL | Knights Landing |
| CPU | Central Processing Unit |
| VPU | Vector Processing Unit |
| SPU | Scalar Processing Unit |
| CRI | Core Ring Interconnect |
| DTD | Distributed Tag Directory |
| SIMD | Single Instruction; Multiple Data |
| SSE | Sreaming SIMD Extensions |
| IMCI | Initial Many Core Instructions |
| FMA | Fused Multiply Add |
| MESI | Modified Exclusive Shared Invalid |
| LRU | Least Recently Used |
| SCIF | Symmetric Communication Interface |
| API | Application Program Interface |
| MPI | Message Passing Interface |
| MPSS | Manycore Platform Software Stack |
| BIOS | Basic Input Output System |
| MMIO | Memory Mapped Input Output |
| TBB | Threading Building Blocks |
| MKL | Math Kernel Library |
| FFT | Fast Fourier Transform |
| CUDA | Compute Unified Device Architecture |
| PML | Perfectly Matched Layer |
| LAPACK | Linear Algebra Package |
| MAGMA | Matrix Algebra on GPU and Multicore Arquitectures |
| RCS | Radar Cross Section |
| PIC | Particle In Cell |
| PEC | Perfect Electric Conductor |
| HPC | High Performance Computing |
| hRWG | half Rao Wilton Glisson |

DDM **Domain Decomposition Methods**

Notación

| | |
|--|---|
| $A...Z, a...z$ | Escalares |
| $\vec{A}... \vec{Z}$ | Vectores |
| $\hat{A}... \hat{Z}, \hat{a}... \hat{z}$ | Vectores unitarios |
| ∇ | Gradiente |
| $\nabla \cdot$ | Divergencia |
| $\nabla \times$ | Rotacional |
| \vec{E} | Intensidad del campo eléctrico |
| \vec{H} | Intensidad del campo magnético |
| \vec{D} | Densidad de flujo eléctrico |
| \vec{B} | Densidad de flujo magnético |
| \vec{J} | Densidad de corriente eléctrica |
| \vec{M} | Densidad de corriente magnética |
| ρ_e | Densidad de carga eléctrica |
| ρ_m | Densidad de carga magnética |
| λ | Longitud de onda |
| ϵ | Permitividad eléctrica |
| μ | Permeabilidad magnética |
| η | Impedancia intrínseca del medio |
| β | Número de onda o constante de propagación |
| ω | Frecuencia angular |
| f | Frecuencia |
| t | Tiempo |
| j | Unidad de número imaginario |
| G | Función de Green en espacio libre |
| \int_S | Integral de superficie |
| $\langle \cdot \rangle$ | Producto escalar |
| \times | Producto vectorial |
| $\ \cdot\ $ | Norma vectorial |
| $[A]...[Z]$ | Matrices y vectores columna |
| $[A]^T$ | Traspuesto de $[A]$ |
| $[A]^*$ | Complejo conjugado de $[A]$ |

| | |
|----------------|---------------------------|
| $[A]^{-1}$ | Matriz inversa de $[A]$ |
| \mathbf{f}_n | Función base |
| \mathbf{w}_n | Función de ponderación |
| $O(\cdot)$ | Complejidad computacional |

Capítulo 1

Introducción

| | | |
|-----|--|---|
| 1.1 | Justificación y relevancia del estudio | 2 |
| 1.2 | Propósito y objetivos | 3 |
| 1.3 | Estructura de la memoria | 4 |

1.1. Justificación y relevancia del estudio

En la última década se ha generalizado el uso de arquitecturas manycore, en especial GPUs, como dispositivos de propósito general en computación de altas prestaciones. En Noviembre de 2019, 6 de los 10 supercomputadores con mayor rendimiento utilizaban GPUs o coprocesadores con arquitectura Intel MIC (*TOP 500 Supercomputers 2019*).

La evolución de las arquitecturas manycore optimizadas para computación paralela y la aparición de herramientas de programación, como CUDA (*Compute Unified Device Architecture*), han posibilitado que hoy día el uso de estas plataformas se haya generalizado en campos como el análisis de imágenes para biomedicina, robótica, simulaciones físicas en distintos ámbitos: mecánica de fluidos, astrofísica, transferencia de calor y energía, campos electromagnéticos, etc.

La arquitectura de procesadores Intel MIC, a pesar de presentar algunas ventajas con respecto a las GPUs, como la mejor adaptación a códigos existentes, no ha conseguido imponerse en el mercado y casi no existen referencias de su uso aplicado a la aceleración de problemas electromagnéticos.

Cualquier modelado electromagnético parte de una representación adecuada de la interacción del campo electromagnético con los diferentes objetos físicos y su entorno. Las ecuaciones de Maxwell son un compendio de ecuaciones que junto con las características constitutivas de los materiales y las condiciones de contorno del dominio en el que se plantea el problema electromagnético, consiguen explicar los fenómenos que surgen de la interacción de las ondas electromagnéticas con la materia.

El análisis de problemas de cálculo electromagnético se ha resuelto tradicionalmente mediante el uso de métodos numéricos. De entre los distintos métodos existentes destaca la metodología SIE-MoM, que conjuga ecuaciones integrales de superficie con el método de los momentos (R. F. Harrington, 1993). Dado que los sistemas de ecuaciones lineales que surgen de la aplicación de MoM dan lugar a matrices densas, es necesario reducir los tiempos de ejecución y el consumo de memoria en aras de obtener una mejora en la eficiencia de uso de los recursos computacionales. Los métodos iterativos clásicos como el método del residuo mínimo generalizado (GMRES, *Generalized Minimum Residual Method*) (Youcef Saad y Schultz, 1986; Kelley, 1995) o el

método del gradiente conjugado (CGS, *Conjugate Gradient Squared*) (Sonneveld, 1989; Kelley, 1995), etc. , junto a otros métodos más avanzados como los métodos rápidos basados en multipolos (FMM, *Fast Multipole Method*) (Coifman, Rokhlin y Wandzura, 1993) o los métodos basados en multipolos de tipo multinivel (MLFMA, *Multilevel Fast Multipole Algorithm*) (J. M. Song y Chew, 1995; J. Song, Lu y Chew, 1997) han conseguido reducir la complejidad computacional de las operaciones producto matriz-vector (MVP, *Matrix Vector Product*) en cada iteración.

En este ámbito y dada la experiencia previa del grupo de investigación en la solución de grandes problemas electromagnéticos (L. Landesa, J. M. Taboada, Campon et al., 2017), se ha decidido tratar de incorporar las capacidades de las nuevas arquitecturas manycore a los códigos existentes.

1.2. Propósito y objetivos

Se propone como línea básica de investigación en la presente Tesis, la mejora de la eficiencia computacional del método MoM mediante la paralelización masiva de algoritmos en tarjetas coprocesadoras con arquitectura Intel MIC. En concreto, los objetivos que se pretenden cumplir son:

- Desarrollo de un algoritmo capaz de realizar múltiples operaciones MVP de forma simultánea, aprovechando la gran capacidad de paralelización que soporta la arquitectura Intel MIC.
- Analizar problemas electromagnéticos complejos y comparar tiempos de ejecución en coprocesadores Intel Xeon Phi frente a procesadores Intel Xeon.

Dado que la arquitectura Intel MIC es relativamente novedosa, su aparición comercial data del año 2012, no existen muchas referencias en la bibliografía del uso de tarjetas coprocesadoras Intel Xeon Phi en la aceleración de métodos de electromagnetismo computacional. La mayoría de trabajos recogidos en la bibliografía relativos a la aceleración de la técnica de MoM sobre Intel Xeon Phi, han utilizado la factorización LU para resolver la ecuación matricial resultante.

La aportación principal de esta Tesis ha sido desarrollar un algoritmo que aprovecha la capacidad de la arquitectura Intel MIC para ejecutar código paralelo y aplicarlo en la resolución de los sistemas de ecuaciones matriciales

derivados de MoM. Concretamente, se ha desarrollado un algoritmo que, de forma paralela, realiza las distintas operaciones MVP requeridas en la resolución de un conjunto de problemas electromagnéticos, implementando el núcleo de las operaciones del método de los momentos en la arquitectura MIC.

1.3. Estructura de la memoria

La memoria de la Tesis se ha estructurado en siete capítulos. Este primer capítulo contiene una breve motivación del interés de este trabajo, además de los objetivos, las contribuciones aportadas y cómo está estructurada la memoria.

El segundo capítulo recoge el desarrollo matemático empleado en la metodología SIE-MoM. Se parte del análisis de la formulación integral de superficie para un objeto dispersor. Una vez planteadas las ecuaciones integrales que caracterizan el campo electromagnético objeto de estudio, se aplica la técnica de MoM para obtener un sistema de ecuaciones lineales mediante una discretización de las ecuaciones integrales previamente planteadas. Se completa el capítulo ampliando la formulación a múltiples objetos dispersores.

En el tercer capítulo se realiza una breve descripción de los métodos iterativos más representativos utilizados en la resolución de sistemas de ecuaciones lineales. Además, se referencia su uso en la resolución de problemas electromagnéticos.

La evolución y descripción de la arquitectura Intel MIC se presenta en el cuarto capítulo. Se incide en el análisis de la arquitectura hardware y software de la familia KNC, que será la utilizada en el desarrollo de la presente Tesis Doctoral. Por último, se analizarán las distintas aproximaciones a la hora de abordar el paralelismo, tanto a nivel de datos como de tareas.

El quinto capítulo presenta un compendio de las principales líneas de investigación existentes en torno al uso de arquitecturas manycore en la aceleración de problemas electromagnéticos. Se ha dividido el capítulo en dos secciones principales dedicadas a las dos tecnologías manycore más utilizadas en supercomputación, GPUs e Intel MIC. También se han clasificado los distintos trabajos referenciados en tres grandes grupos dependiendo de la técnica de partida utilizada, FDTD, FEM o MoM.

Los resultados obtenidos y los algoritmos desarrollados se presentan en el sexto capítulo. Tras realizar una descripción de las estrategias de optimización del código HEMCUVE en arquitecturas Intel MIC, se detallan los algoritmos implementados para conseguir la aceleración de los productos matriz-vector necesarios en la resolución de los sistemas de ecuaciones lineales de MoM. A continuación, se relacionan distintos problemas electromagnéticos en los que los algoritmos desarrollados son de aplicación, distinguiendo el caso de un único objeto dispersor o el de varios objetos dispersores. Tras realizar un análisis de los problemas electromagnéticos planteados, se discuten los resultados obtenidos comparativamente con la familia de procesadores Intel Xeon. Los resultados contenidos en este capítulo han sido seleccionados fundamentalmente para analizar el comportamiento de la arquitectura Intel MIC.

Finalmente, el séptimo y último capítulo ofrece al lector las conclusiones y líneas futuras de investigación que puedan dar continuidad al trabajo desarrollado.

Capítulo 2

Metodología SIE-MoM

| | | |
|-------|---|----|
| 2.1 | Introducción | 8 |
| 2.2 | Ecuaciones de Maxwell-Hertz-Heaviside | 9 |
| 2.2.1 | Formulación | 9 |
| 2.2.2 | Condiciones de contorno e iniciales | 12 |
| 2.3 | Radiación electromagnética | 13 |
| 2.3.1 | Formulación en volumen | 13 |
| 2.3.2 | Operadores L_V y K_V | 15 |
| 2.3.3 | Formulación en superficie | 16 |
| 2.4 | Formulación integral de superficie | 19 |
| 2.5 | Método de los Momentos | 25 |
| 2.5.1 | Formulación | 25 |
| 2.5.2 | Funciones base RWG | 28 |
| 2.5.3 | Discretización de la SIE | 29 |
| 2.6 | Extensión de la formulación a múltiples objetos | 33 |

2.1. Introducción

Las ecuaciones de Maxwell son un compendio de ecuaciones que junto con los parámetros constitutivos de los materiales (permitividad y conductividad eléctrica y permeabilidad magnética) consiguen explicar los fenómenos que surgen de la interacción de las ondas electromagnéticas con la materia.

La correcta definición de las condiciones de contorno del dominio en el que se plantea el problema electromagnético es indispensable para que se encuentre bien definido y tenga una única solución. En problemas causales también se hace necesario conocer las condiciones iniciales desde las que se parte, ya que los campos y flujos en un instante determinado dependen de los campos y flujos en instantes anteriores.

En la resolución de problemas electromagnéticos, por tanto, es imprescindible la correcta formulación de las ecuaciones de Maxwell y el planteamiento adecuado de las condiciones de contorno e iniciales del problema. En problemas sencillos con geometrías simples, como la dispersión producida por una esfera de material dieléctrico, es posible plantear una resolución analítica (Mie, 1908), pero a medida que la complejidad aumenta se hace necesario recurrir a métodos numéricos.

En las últimas décadas se ha producido un rápido incremento en la capacidad de cómputo de los ordenadores que ha permitido abordar problemas electromagnéticos cada vez más complejos mediante métodos numéricos.

Los métodos basados en ecuaciones integrales de superficie en conjunción con el método de los momentos (SIE-MoM, *Surface Integral Equation-Method of Moments*) (R. F. Harrington, 1993) han demostrado ser más eficientes que las aproximaciones volumétricas, ya sean en el dominio del tiempo, como puede ser el método de diferencias finitas en el dominio del tiempo (FDTD, *Finite Difference Time Domain*) (Hao, Nehl, Hafner et al., 2007) o en el dominio de la frecuencia, como el método de los elementos finitos (FEM, *Finite Element Method*) (J. Jin, 2014). La razón principal de esta mayor eficiencia reside en el hecho de que las incógnitas del problema quedan reducidas a densidades superficiales de corrientes eléctricas y magnéticas a través de las superficies frontera. Aun así, la transformación de las ecuaciones integrales en sistemas de ecuaciones lineales da como resultado un problema matricial con un elevado coste computacional, $O(N^3)$, y de consumo de memoria RAM, $O(N^2)$, siendo N el número de incógnitas.

Los métodos iterativos como el método del residuo mínimo generalizado, GMRES, o el método del gradiente conjugado, CGS, reducen el coste computacional del producto matriz-vector a $O(N^2)$. Otros métodos más avanzados basados en dividir la geometría en grupos y calcular las interacciones de forma diagonal mediante la expansión en multipolos, como los métodos rápidos basados en multipolos, FMM, o los métodos basados en multipolos de tipo multinivel, MLFMA, consiguen reducir aún más la complejidad computacional llegando hasta $O(N \log N)$.

2.2. Ecuaciones de Maxwell-Hertz-Heaviside

2.2.1. Formulación

James Clerk Maxwell publicó en 1864 el artículo titulado 'A Dynamical Theory of the Electromagnetic Field' (Maxwell, 1865) en el cual predecía que una onda electromagnética viaja en el vacío a la velocidad de la luz. Para ello, resumió y condensó de forma brillante los trabajos individuales de físicos como Gauss, Oersted, Ampere y Faraday entre otros, en la forma de veinte variables y veinte ecuaciones¹. Con el fin de generalizar y dar sentido a su teoría, Maxwell introdujo el concepto de corriente de desplazamiento que más tarde se demostró empíricamente.

Las contribuciones de Hertz y Heaviside (Sengupta y Sarkar, 2003), de forma independiente, llevaron a reformular las ecuaciones de Maxwell simplificándolas en las cuatro ecuaciones en notación vectorial que hoy día conocemos. Los experimentos de Hertz (H.Hertz, 1888), en los cuales generó, transmitió y recibió una onda electromagnética a una frecuencia determinada constituyen la primera demostración experimental de la hasta entonces cuestionada teoría de Maxwell.

Hoy en día, las conocidas como ecuaciones de Maxwell en su forma diferencial se resumen en:

¹En realidad las veinte ecuaciones pueden resumirse en ocho, ya que Maxwell dividió seis de las ecuaciones en sus tres componentes cartesianas.

$$\begin{aligned}
\nabla \times \vec{\mathcal{H}} &= \vec{\mathcal{J}} + \frac{\partial \vec{\mathcal{D}}}{\partial t} \\
\nabla \times \vec{\mathcal{E}} &= -\frac{\partial \vec{\mathcal{B}}}{\partial t} \\
\nabla \cdot \vec{\mathcal{D}} &= \rho_e \\
\nabla \cdot \vec{\mathcal{B}} &= 0
\end{aligned} \tag{2.1}$$

donde $\vec{\mathcal{H}}$ es la intensidad de campo magnético, $\vec{\mathcal{E}}$ es la intensidad de campo eléctrico, $\vec{\mathcal{D}}$ es la densidad de flujo eléctrico, $\vec{\mathcal{B}}$ es la densidad de flujo magnético, $\vec{\mathcal{J}}$ es la densidad de corriente eléctrica, $\frac{\partial \vec{\mathcal{D}}}{\partial t}$ y $\frac{\partial \vec{\mathcal{B}}}{\partial t}$ son las corrientes de desplazamiento y ρ_e la densidad volumétrica de carga libre.

Considerando un medio lineal, los campos $\vec{\mathcal{D}}$ y $\vec{\mathcal{E}}$ se relacionan a través de las propiedades dieléctricas de los materiales, $\vec{\mathcal{B}}$ y $\vec{\mathcal{H}}$ a través de sus propiedades magnéticas y $\vec{\mathcal{J}}$ y $\vec{\mathcal{E}}$ a través de la conductividad:

$$\begin{aligned}
\vec{\mathcal{D}} &= \varepsilon \vec{\mathcal{E}} \\
\vec{\mathcal{B}} &= \mu \vec{\mathcal{H}} \\
\vec{\mathcal{J}} &= \sigma \vec{\mathcal{E}}
\end{aligned} \tag{2.2}$$

siendo σ la conductividad, ε la permitividad eléctrica y μ la permeabilidad magnética. La permitividad eléctrica puede expresarse como $\varepsilon = \varepsilon_r \varepsilon_0$ siendo ε_r la permitividad relativa del medio y ε_0 la permitividad en el vacío. Lo mismo sucede con la permeabilidad magnética, $\mu = \mu_r \mu_0$.

Si expresamos las ecuaciones de Maxwell (2.1) mediante el uso de fasores, las derivadas en el dominio del tiempo se sustituyen por el producto $j\omega$ en el dominio de la frecuencia, dando lugar a las siguientes ecuaciones:

$$\begin{aligned}
\nabla \times \mathbf{H} &= \mathbf{J} + j\omega \mathbf{D} \\
\nabla \times \mathbf{E} &= -j\omega \mathbf{B} \\
\nabla \cdot \mathbf{D} &= \rho_e \\
\nabla \cdot \mathbf{B} &= 0
\end{aligned} \tag{2.3}$$

donde \mathbf{H} , \mathbf{E} , \mathbf{B} , \mathbf{D} y \mathbf{J} son los fasores de los campos vectoriales anteriormente descritos y ω la frecuencia angular. La variación temporal armónica $\exp(j\omega t)$

se omitirá por simplicidad a lo largo del capítulo.

En un medio homogéneo es posible sustituir las ecuaciones constitutivas (2.2) en (2.3) y obtener:

$$\begin{aligned}
 \nabla \times \mathbf{H} &= \mathbf{J} + j\omega\epsilon\mathbf{E} \\
 \nabla \times \mathbf{E} &= -j\omega\mu\mathbf{H} \\
 \nabla \cdot \mathbf{E} &= \frac{\rho_e}{\epsilon} \\
 \nabla \cdot \mathbf{H} &= 0
 \end{aligned}
 \tag{2.4}$$

Si se aplica la divergencia en la primera de las ecuaciones anteriores y además se sustituye $\nabla \cdot \mathbf{E}$, se obtiene una ecuación que relaciona la densidad de corriente eléctrica y la densidad de carga eléctrica, conocida como ecuación de continuidad:

$$\nabla \cdot \mathbf{J} = -j\omega\rho_e \tag{2.5}$$

Es de utilidad formular las ecuaciones de Maxwell en el hipotético caso de que existieran cargas magnéticas (aún no se han descubierto). Su existencia permitiría modelar de manera muy eficiente problemas en medios no permeables, en medios no homogéneos o en medios no isotrópicos.

Para ello, se añade al sistema de ecuaciones anterior una corriente magnética ficticia \mathbf{M} y una densidad de carga magnética ficticia ρ_m (Kolundžija y Djordjević, 2002). Ambas magnitudes se relacionan de forma idéntica a sus correspondientes magnitudes eléctricas y simulan matemáticamente las propiedades de estos medios:

$$\begin{aligned}
 \nabla \times \mathbf{H} &= \mathbf{J} + j\omega\epsilon\mathbf{E} \\
 \nabla \times \mathbf{E} &= -\mathbf{M} - j\omega\mu\mathbf{H} \\
 \nabla \cdot \mathbf{E} &= \frac{\rho_e}{\epsilon} \\
 \nabla \cdot \mathbf{H} &= \frac{\rho_m}{\mu}
 \end{aligned}
 \tag{2.6}$$

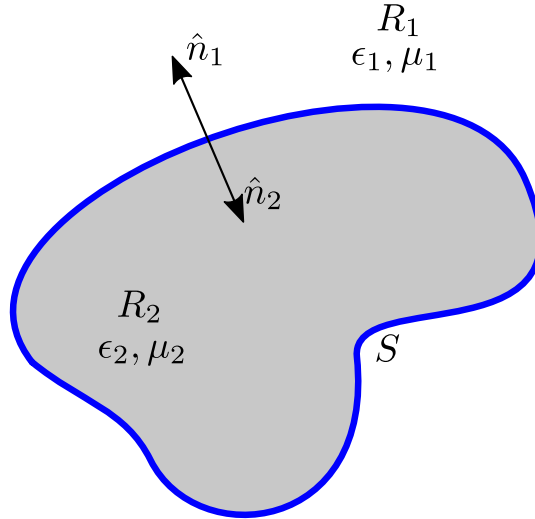


FIGURA 2.1: Superficie de discontinuidad entre dos medios diferentes R_1 y R_2

Del mismo modo en que obtuvimos la ecuación de continuidad (2.5), podemos obtener una ecuación que relaciona la densidad de corriente magnética y la densidad de carga magnética, conocida como ecuación de continuidad:

$$\nabla \cdot \mathbf{M} = -j\omega\rho_m \quad (2.7)$$

2.2.2. Condiciones de contorno e iniciales

Dado que las ecuaciones diferenciales no son válidas en las superficies de discontinuidad de dos medios diferentes, es necesario establecer unas condiciones de contorno que relacionen los valores que toman las magnitudes electromagnéticas en dos puntos infinitamente próximos ubicados a ambos lados de dicha superficie de discontinuidad.

Considérese un medio homogéneo (Región R_1) en el cual se introduce un objeto dieléctrico, también homogéneo (Región R_2), siendo los parámetros constitutivos ϵ_1, μ_1 y ϵ_2, μ_2 respectivamente, tal y como se observa en la figura 2.1. Defínase S como la superficie de discontinuidad de los dos medios y \hat{n}_i como el vector unitario normal a S apuntando hacia la región R_i .

Los campos en la superficie de discontinuidad están gobernados por las siguientes condiciones de contorno:

$$\begin{aligned}
\hat{n}_i \times (\mathbf{E}_i - \mathbf{E}_j) &= -\mathbf{M}_i \\
\hat{n}_i \times (\mathbf{H}_i - \mathbf{H}_j) &= \mathbf{J}_i \\
\hat{n}_i \cdot (\varepsilon_i \mathbf{E}_i - \varepsilon_j \mathbf{E}_j) &= \rho_e^s \\
\hat{n}_i \cdot (\mu_i \mathbf{H}_i - \mu_j \mathbf{H}_j) &= \rho_m^s
\end{aligned} \tag{2.8}$$

Para fijar las condiciones iniciales suele recurrirse a considerar el estado estacionario senoidal en el cual todas las magnitudes varían de la forma:

$$A(t) = \cos(\omega t + \psi)$$

siendo A una función armónica en el tiempo, ω la frecuencia angular y ψ la fase inicial.

El uso de fasores se introdujo en el capítulo a partir de las ecuaciones (2.3). A partir de este momento, las variaciones en el tiempo pueden analizarse realizando la transformada inversa de Fourier.

2.3. Radiación electromagnética

2.3.1. Formulación en volumen

La figura 2.2 muestra un problema básico de radiación electromagnética en un medio homogéneo e isótropo, de parámetros ε y μ , en el que se pretenden calcular los campos en cualquier punto del espacio a partir de las corrientes eléctricas y magnéticas, \mathbf{J} y \mathbf{M} . Definamos O como el origen de coordenadas, r como el vector de posición del punto de observación, r' el vector de posición del punto fuente, V la región del espacio en la que se ubica la corriente y la carga y R la distancia entre los puntos fuente y observación.

Consideremos las ecuaciones de Maxwell (2.6) formuladas para cualquier medio. La resolución matemática del problema se solventó añadiendo una corriente magnética ficticia \mathbf{M} y una densidad de carga magnética ficticia ρ_m . Si aplicamos el rotacional a ambos miembros de la segunda ecuación obtenemos:

$$\nabla \times \nabla \times \mathbf{E} = -j\omega\mu\nabla \times \mathbf{H} - \nabla \times \mathbf{M} \tag{2.9}$$

Si ahora sustituimos la primera ecuación de (2.6) en (2.9) y además utilizamos el número de onda $\beta = \omega \sqrt{\mu\epsilon}$:

$$\nabla \times \nabla \times \mathbf{E} = \beta^2 \mathbf{E} - j\omega\mu \mathbf{J} - \nabla \times \mathbf{M} \quad (2.10)$$

Aplicando la identidad vectorial $\nabla \times \nabla \times \mathbf{E} = \nabla(\nabla \cdot \mathbf{E}) - \nabla^2 \mathbf{E}$ a la ecuación anterior y sustituyendo la tercera ecuación de (2.6) resulta:

$$\nabla^2 \mathbf{E} + \beta^2 \mathbf{E} = j\omega\mu \mathbf{J} + \frac{\nabla \rho_e}{\epsilon} + \nabla \times \mathbf{M} \quad (2.11)$$

Si aplicamos la divergencia a la ecuación de continuidad (2.5) y sustituimos en (2.11) se obtiene:

$$\nabla^2 \mathbf{E} + \beta^2 \mathbf{E} = j\omega\mu \mathbf{J} - \frac{1}{j\omega\epsilon} \nabla(\nabla \cdot \mathbf{J}) + \nabla \times \mathbf{M} \quad (2.12)$$

Escribiendo la ecuación anterior en función de la impedancia intrínseca del medio $\eta = \sqrt{\frac{\mu}{\epsilon}}$ resulta:

$$\nabla^2 \mathbf{E} + \beta^2 \mathbf{E} = j\beta\eta[\mathbf{J} + \frac{1}{\beta^2} \nabla(\nabla \cdot \mathbf{J})] + \nabla \times \mathbf{M} \quad (2.13)$$

La solución a la ecuación de Helmholtz escalar en tres dimensiones,

$$\nabla^2 G(r, r') + \beta^2 G(r, r') = -\delta(r, r') \quad (2.14)$$

se obtiene mediante la siguiente función de Green en coordenadas cartesianas:

$$G(r, r') = \frac{e^{-j\beta|r-r'|}}{4\pi|r-r'|} \quad (2.15)$$

cuyo gradiente puede obtenerse derivando la ecuación anterior en función de $R = |r - r'|$:

$$\nabla G(r, r') = \left(-j\beta - \frac{1}{R} \right) \frac{e^{-j\beta R}}{4\pi R} \hat{R} \quad (2.16)$$

siendo $\hat{R} = \frac{r-r'}{|r-r'|}$, el vector unitario entre los puntos fuente y de observación.

Dado que las ecuaciones de Maxwell son lineales, \mathbf{J} y \mathbf{M} se pueden aproximar como una suma de fuentes puntuales distribuidas en la región del espacio V ,

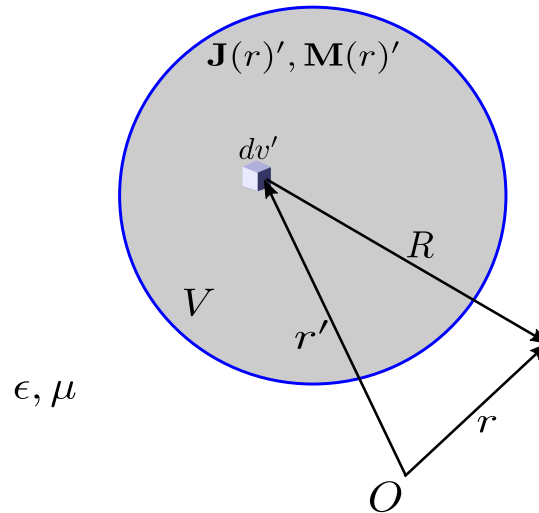


FIGURA 2.2: Geometría volumétrica del problema de radiación electromagnética

de forma que la ecuación (2.13) se transforma en una ecuación integral sobre el volumen V expresada en términos de la función de Green:

$$\begin{aligned} \mathbf{E}(r) = & -j\beta\eta \int_V \left[\mathbf{J}(r')G(r, r') + \frac{1}{\beta^2} \nabla' \cdot \mathbf{J}(r') \nabla G(r, r') \right] dv' \\ & + \int_V \mathbf{M}(r') \times \nabla G(r, r') dv' \end{aligned} \quad (2.17)$$

De modo similar es posible obtener:

$$\begin{aligned} \mathbf{H}(r) = & - \int_V \mathbf{J}(r') \times \nabla G(r, r') dv' \\ & - j\frac{\beta}{\eta} \int_V \left[\mathbf{M}(r')G(r, r') + \frac{1}{\beta^2} \nabla' \cdot \mathbf{M}(r') \nabla G(r, r') \right] dv' \end{aligned} \quad (2.18)$$

2.3.2. Operadores \mathbf{L}_V y \mathbf{K}_V

Las ecuaciones integrales (2.17) y (2.18) pueden reescribirse de forma compacta en función de los operadores \mathbf{L}_V y \mathbf{K}_V . Siguiendo esta nomenclatura los campos eléctricos y magnéticos se expresan como:

$$\begin{aligned}\mathbf{E} &= -\eta\mathbf{L}_V(\mathbf{J}) + \mathbf{K}_V(\mathbf{M}) \\ \mathbf{H} &= -\mathbf{K}_V(\mathbf{J}) - \frac{1}{\eta}\mathbf{L}_V(\mathbf{M})\end{aligned}\quad (2.19)$$

siendo:

$$\mathbf{L}_V(\mathbf{X}) = j\beta \int_V \left[\mathbf{X}(r')G(r, r') + \frac{1}{\beta^2} \nabla' \cdot \mathbf{X}(r') \nabla G(r, r') \right] dv' \quad (2.20)$$

$$\mathbf{K}_V(\mathbf{X}) = \int_V \mathbf{X}(r') \times \nabla G(r, r') dv' \quad (2.21)$$

2.3.3. Formulación en superficie

Si se parte de las ecuaciones (2.17) y (2.18) debido a corrientes volumétricas, es fácil extrapolar los resultados para obtener las ecuaciones producidas por corrientes superficiales (Kolundžija y Djordjević, 2002). Para ello, se sustituyen las densidades de corriente volumétrica eléctrica y magnética, \mathbf{J} y \mathbf{M} , por densidades de corriente superficiales, \mathbf{J}_S y \mathbf{M}_S . Del mismo modo, el elemento infinitesimal de volumen dv' es sustituido por el de superficie ds' y el operador ∇' deja de estar referido a volúmenes y pasa a referirse a superficies.

Considerando lo anterior, los operadores (2.20) y (2.21) aplicados a corrientes superficiales pueden expresarse como:

$$\mathbf{L}(\mathbf{X}_S) = j\beta \int_S \left[\mathbf{X}_S(r')G(r, r') + \frac{1}{\beta^2} \nabla' \cdot \mathbf{X}_S(r') \nabla G(r, r') \right] ds' \quad (2.22)$$

$$\mathbf{K}(\mathbf{X}_S) = \int_S \mathbf{X}_S(r') \times \nabla G(r, r') ds' \quad (2.23)$$

Sustituyendo (2.15) y (2.16) en las ecuaciones anteriores se obtiene:

$$\mathbf{L}(\mathbf{X}_S) = \frac{j\beta}{4\pi} \int_S \left[\frac{\mathbf{X}_S(r')}{R} - \frac{1}{\beta^2} \nabla' \cdot \mathbf{X}_S(r') \hat{R} \left[\frac{j\beta}{R} + \frac{1}{R^2} \right] \right] e^{-j\beta R} ds' \quad (2.24)$$

$$\mathbf{K}(\mathbf{X}_S) = \frac{-1}{4\pi} \int_S \left[\mathbf{X}_S(r') \times \hat{R} \left[\frac{j\beta}{R} + \frac{1}{R^2} \right] \right] e^{-j\beta R} ds' \quad (2.25)$$

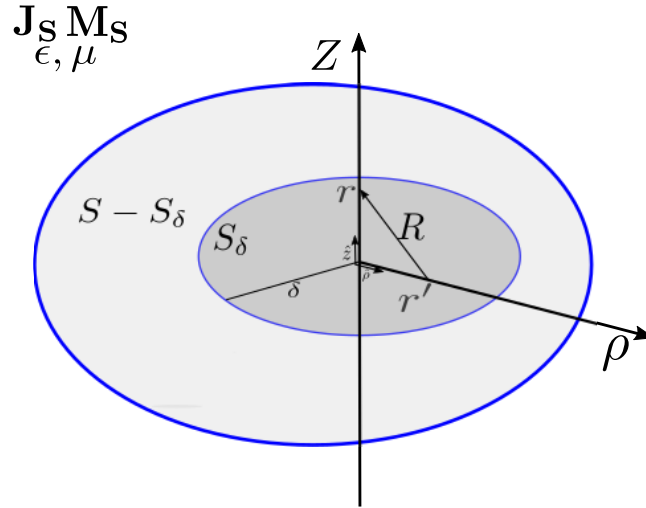


FIGURA 2.3: Geometría superficial del problema de radiación electromagnética

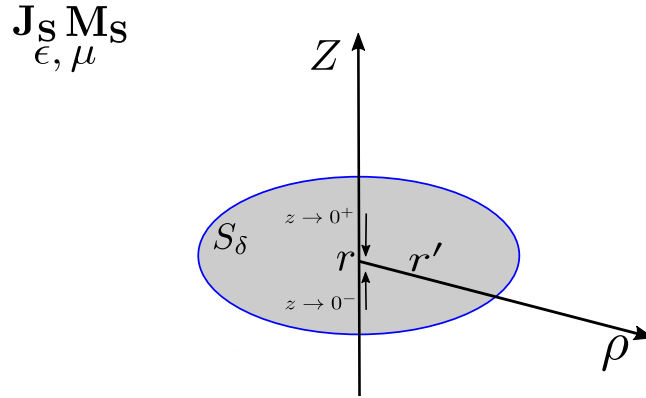
Los valores de las integrales de superficie de las expresiones de los operadores \mathbf{L} y \mathbf{K} dependen de la ubicación del punto de observación \mathbf{r} , donde se quieren calcular los campos. Si el punto está ubicado fuera de la superficie S , los operadores \mathbf{L} y \mathbf{K} toman los valores de las expresiones (2.24) y (2.25). En cambio, si el punto de observación está ubicado sobre la superficie S , existe una discontinuidad en los valores de dichas integrales dependiendo de la aproximación de \mathbf{r} a \mathbf{r}' .

Para clarificar este concepto, se divide la superficie S en dos partes. Por un lado, una pequeña superficie circular centrada en \mathbf{r} , de radio δ , denominada S_δ y por otro el resto de la superficie $S - S_\delta$ tal y como se observa en la figura 2.3. La evaluación de la integral de superficie $S - S_\delta$ mantiene intacto el valor de los operadores \mathbf{L} y \mathbf{K} . En cambio, si δ es lo suficientemente pequeño, los términos proporcionales a $\frac{1}{R}$ en las expresiones (2.24) y (2.25) son despreciables frente a los demás. Del mismo modo, las densidades de corriente y carga se mantienen constantes y $e^{-j\beta R} \approx 1$. De todo lo anterior, se obtiene:

$$\mathbf{L}_\delta(\mathbf{X}_S) = \frac{-j}{4\pi\beta} \nabla' \cdot \mathbf{X}_S(r') \int_{S_\delta} \frac{\hat{R}}{R^2} ds' \quad (2.26)$$

$$\mathbf{K}_\delta(\mathbf{X}_S) = \frac{-1}{4\pi} \mathbf{X}_S(r') \times \int_{S_\delta} \frac{\hat{R}}{R^2} ds' \quad (2.27)$$

Supóngase ahora un sistema de coordenadas cilíndrico centrado en \mathbf{r} en el

FIGURA 2.4: Detalle de la aproximación de r a S_δ .

que el eje Z es perpendicular a la superficie S_δ . En este sistema de coordenadas el módulo R se puede expresar como $R = \sqrt{\rho^2 + z^2}$ y el vector unitario \hat{R} como $\hat{R} = \frac{-\rho}{R}\hat{\rho} + \frac{z}{R}\hat{z}$ siendo z la altura y ρ la coordenada radial. La coordenada azimutal puede obviarse debido a la simetría existente en rotación. De esta forma, ds puede expresarse como $ds = 2\pi R dr$ con ρ variando entre 0 y δ .

La integral de las expresiones (2.26) y (2.27) puede escribirse como:

$$\int_{S_\delta} \frac{\hat{R}}{R^2} ds = 2\pi z \int_0^\delta \frac{dR}{R^2} \hat{z} = 2\pi \left(\frac{z}{|z|} - \frac{z}{\sqrt{\delta^2 + z^2}} \right) \hat{z} \quad (2.28)$$

Cuando z se aproxima a 0 desde la parte superior de la superficie S_δ la integral tiende a $2\pi\hat{z}$. En cambio si se aproxima a 0 desde la parte inferior de la superficie S_δ el valor tiende a $-2\pi\hat{z}$. La figura 2.4 muestra el detalle de la aproximación de r a la superficie S_δ .

Al sustituir estos valores en las integrales (2.26) y (2.27) y aplicar además la ecuación de continuidad (2.46) se obtiene:

$$\lim_{z \rightarrow \pm 0} \mathbf{L}_\delta(\mathbf{X}_S) = \mp \frac{1}{2} \frac{\rho_S}{\sqrt{\mu\epsilon}} \hat{z} \quad (2.29)$$

$$\lim_{z \rightarrow \pm 0} \mathbf{K}_\delta(\mathbf{X}_S) = \mp \frac{1}{2} \mathbf{X}_S \times \hat{z} \quad (2.30)$$

Retomando las ecuaciones (2.19), es posible expresar los campos en cualquier punto del espacio en función de las corrientes superficiales \mathbf{J}_S y \mathbf{M}_S como:

$$\begin{aligned} \mathbf{E} &= -\eta \mathbf{L}(\mathbf{J}_S) + \mathbf{K}(\mathbf{M}_S) \\ \mathbf{H} &= -\mathbf{K}(\mathbf{J}_S) - \frac{1}{\eta} \mathbf{L}(\mathbf{M}_S) \end{aligned} \quad (2.31)$$

siendo los operadores \mathbf{L} y \mathbf{K} :

$$\mathbf{L}(\mathbf{X}_S) = \begin{cases} \mathbf{L}_0(\mathbf{X}_S) & \forall z \neq 0 \\ \mathbf{L}_0(\mathbf{X}_S) - \frac{1}{2} \frac{\rho_S}{\sqrt{\mu\epsilon}} \hat{n} & z = 0^+ \\ \mathbf{L}_0(\mathbf{X}_S) & z = 0 \\ \mathbf{L}_0(\mathbf{X}_S) + \frac{1}{2} \frac{\rho_S}{\sqrt{\mu\epsilon}} \hat{n} & z = 0^- \end{cases}$$

$$\mathbf{K}(\mathbf{X}_S) = \begin{cases} \mathbf{K}_0(\mathbf{X}_S) & \forall z \neq 0 \\ \mathbf{K}_0(\mathbf{X}_S) - \frac{1}{2} \mathbf{X}_S \times \hat{n} & z = 0^+ \\ \mathbf{K}_0(\mathbf{X}_S) & z = 0 \\ \mathbf{K}_0(\mathbf{X}_S) + \frac{1}{2} \mathbf{X}_S \times \hat{n} & z = 0^- \end{cases}$$

y siendo \mathbf{L}_0 y \mathbf{K}_0 las expresiones (2.22) y (2.23) calculadas en el sentido del valor principal de Cauchy y \hat{n} el vector normal unitario a la superficie S .

2.4. Formulación integral de superficie

Partiendo del análisis realizado en la sección anterior para el caso básico de radiación electromagnética en un medio homogéneo e isótropo, introduciremos la formulación SIE para un único objeto dispersor de geometría arbitraria. Se asume una variación temporal armónica $\exp(j\omega t)$ que ha sido omitida en la formulación. El caso para múltiples dispersores se analizará en la sección 2.6.

Supongamos un caso simplificado de un medio homogéneo, (Región R_1), con permitividad ϵ_1 y permeabilidad μ_1 en el cual se introduce un objeto dieléctrico, también homogéneo, (Región R_2), con permitividad ϵ_2 y permeabilidad μ_2 como se observa en la figura (2.5). Definamos S como la superficie de discontinuidad de los dos medios, \hat{n}_2 como el vector unitario normal a S apuntando hacia la región R_2 y \hat{n}_1 como el vector unitario normal a S apuntando

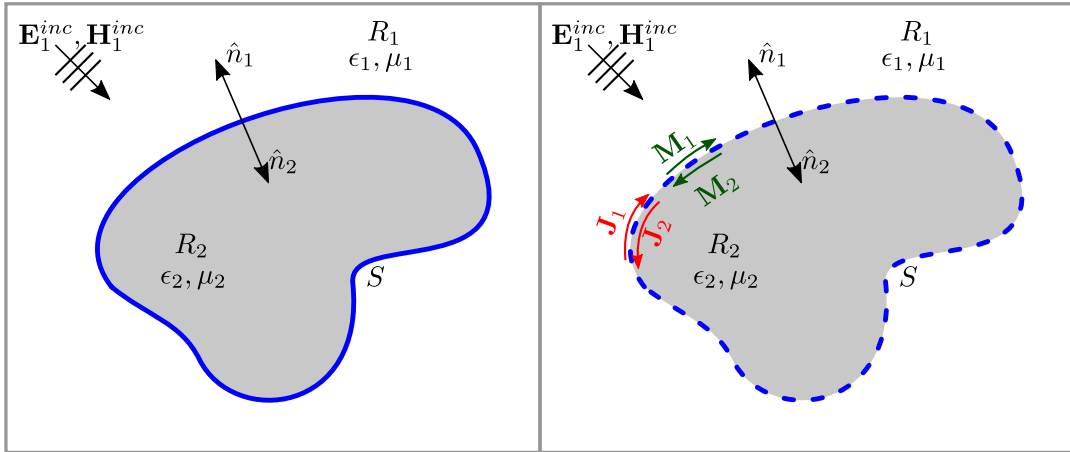


FIGURA 2.5: Geometría superficial con un dispersor y problema equivalente

hacia la región R_1 . Se define \mathbf{E}_i^{inc} y \mathbf{H}_i^{inc} como los campos eléctricos y magnéticos incidentes debido a las fuentes ubicadas en la región i . Los campos incidentes en los medios sin fuentes son nulos.

La resolución simplificada de estos problemas reside en sustituir los objetos dispersores por las densidades de corriente eléctrica y magnética equivalentes, \mathbf{J}_i y \mathbf{M}_i , aplicando el primer principio de equivalencia (R. F. Harrington, 2001) como se observa en la figura (2.5).

Se parte de dos campos incidentes denominados \mathbf{E}_1^{inc} y \mathbf{H}_1^{inc} . Las fuentes que originan dichos campos se ubican en la región 1 y en la superficie de discontinuidad que separa los dos medios en forma de corrientes eléctricas y magnéticas, \mathbf{J}_1 y \mathbf{M}_1 . Estas fuentes inducen en la superficie frontera unas nuevas corrientes, \mathbf{J}_2 y \mathbf{M}_2 que originan en ambas regiones los campos dispersos \mathbf{E}_i^{scat} y \mathbf{H}_i^{scat} . Los campos eléctricos y magnéticos totales, \mathbf{E}_i y \mathbf{H}_i , por tanto, se pueden escribir en función de los campos incidentes y los dispersos:

$$\begin{aligned}\mathbf{E}_i &= \mathbf{E}_i^{inc} + \mathbf{E}_i^{scat} \\ \mathbf{H}_i &= \mathbf{H}_i^{inc} + \mathbf{H}_i^{scat}\end{aligned}\tag{2.32}$$

El campo disperso debido al objeto puede calcularse a partir de las corrientes equivalentes tal y como vimos en (2.31):

$$\begin{aligned}\mathbf{E}_i^{scat} &= -\eta_i \mathbf{L}\{\mathbf{J}_i\} + \mathbf{K}\{\mathbf{M}_i\} \\ \mathbf{H}_i^{scat} &= -\mathbf{K}\{\mathbf{J}_i\} - \frac{1}{\eta_i} \mathbf{L}\{\mathbf{M}_i\}\end{aligned}\quad (2.33)$$

Aplicando el principio de equivalencia, el problema planteado puede descomponerse en un problema equivalente interior (los campos en R_2 se anulan) y otro exterior (los campos en R_1 se anulan) en los cuales las corrientes equivalentes son contrarias de forma que $\mathbf{J}_1 = -\mathbf{J}_2 \equiv \mathbf{J}$ y $\mathbf{M}_1 = -\mathbf{M}_2 \equiv \mathbf{M}$. Las corrientes equivalentes, aplicando las condiciones de contorno (2.8) de los campos en la superficie de discontinuidad son:

$$\begin{aligned}\text{N-EFIE}_i : \mathbf{M}_i &= -\hat{n}_i \times \mathbf{E}_i|_S \\ \text{N-MFIE}_i : \mathbf{J}_i &= \hat{n}_i \times \mathbf{H}_i|_S\end{aligned}\quad (2.34)$$

Las ecuaciones anteriores son conocidas como *Normal Electric Field Integral Equation* (N-EFIE) y *Normal Magnetic Field Integral Equation* (N-MFIE). En problemas de dieléctricos homogéneos se utilizan además las ecuaciones *Tangential Electric Field Integral Equation* (T-EFIE) y *Tangential Magnetic Field Integral Equation* (T-MFIE):

$$\begin{aligned}\text{T-EFIE}_i : \hat{n}_i \times \mathbf{M}_i &= -\hat{n}_i \times \hat{n}_i \times \mathbf{E}_i|_S = \mathbf{E}_{i,tan}|_S \\ \text{T-MFIE}_i : -\hat{n}_i \times \mathbf{J}_i &= -\hat{n}_i \times \hat{n}_i \times \mathbf{H}_i|_S = \mathbf{H}_{i,tan}|_S\end{aligned}\quad (2.35)$$

Las formulaciones normal y tangencial se obtienen dependiendo del método aplicado para proyectar los campos sobre la superficie (Kolundžija y Djordjević, 2002).

Sustituyendo (2.32) y (2.33) en (2.34) y (2.35) las ecuaciones integrales se pueden reescribir como:

$$\text{T-EFIE}_i : \quad \mathbf{E}_{i,tan}^{inc} = [\eta_i \mathbf{L}_0 \{ \mathbf{J}_i \} - \mathbf{K}_0 \{ \mathbf{M}_i \}]_{tan} + \frac{1}{2} \hat{n}_i \times \mathbf{M}_i \quad (2.36)$$

$$\text{T-MFIE}_i : \quad \mathbf{H}_{i,tan}^{inc} = \left[\mathbf{K}_0 \{ \mathbf{J}_i \} + \frac{1}{\eta_i} \mathbf{L}_0 \{ \mathbf{M}_i \} \right]_{tan} - \frac{1}{2} \hat{n}_i \times \mathbf{J}_i \quad (2.37)$$

$$\text{N-EFIE}_i : \quad \hat{n}_i \times \mathbf{E}_i^{inc} = \hat{n}_i \times [\eta_i \mathbf{L}_0 \{ \mathbf{J}_i \} - \mathbf{K}_0 \{ \mathbf{M}_i \}] - \frac{1}{2} \mathbf{M}_i \quad (2.38)$$

$$\text{N-MFIE}_i : \quad \hat{n}_i \times \mathbf{H}_i^{inc} = \hat{n}_i \times \left[\mathbf{K}_0 \{ \mathbf{J}_i \} + \frac{1}{\eta_i} \mathbf{L}_0 \{ \mathbf{M}_i \} \right] + \frac{1}{2} \mathbf{J}_i \quad (2.39)$$

Existen múltiples posibilidades de combinación para las ecuaciones anteriores. Debido a su estabilidad y precisión en escenarios problemáticos se plantean las siguientes formulaciones *Combined Field Integral Equation* (P Ylä-Oijala y Matti Taskinen, 2005b) (CFIE) combinando las ecuaciones que se encuentran en el mismo espacio de corrientes, \mathbf{J}_i o \mathbf{M}_i , para cada una de las regiones R_i :

$$\frac{a_i}{\eta_i} \text{T-EFIE}_i + b_i \text{N-MFIE}_i \quad (2.40)$$

$$-c_i \text{N-EFIE}_i + d_i \eta_i \text{T-MFIE}_i \quad (2.41)$$

donde η_i representa la impedancia intrínseca de cada región R_i ; T-EFIE, T-MFIE, N-EFIE y N-MFIE se corresponden con (2.36), (2.37), (2.38) y (2.39) respectivamente; a_i , b_i , c_i y d_i representan coeficientes complejos para cada región R_i .

La ecuación (2.40) se denomina *Electric current (J) Combined Field Integral Equation* (JCFIE) debido a que sus operadores mapean las corrientes \mathbf{J}_i y \mathbf{M}_i en \mathbf{J}_i . De modo similar, la ecuación (2.41) se denomina *Magnetic current (M) Combined Field Integral Equation* (MCFIE), ya que sus operadores realizan el mapeo de las corrientes \mathbf{J}_i y \mathbf{M}_i en \mathbf{M}_i .

Ahora bien, dado que dichas ecuaciones deben cumplirse para cada una de las regiones, es posible combinar a su vez las ecuaciones CFIE de cada una de las regiones anteriores en una única ecuación de la siguiente forma:²

²Nótese que se podía haber elegido cualquier combinación lineal de las ecuaciones (2.40) y (2.41)

$$\frac{a_1}{\eta_1} \text{T-EFIE}_1 - \frac{a_2}{\eta_2} \text{T-EFIE}_2 + b_1 \text{N-MFIE}_1 - b_2 \text{N-MFIE}_2 \quad (2.42)$$

$$-c_1 \text{N-EFIE}_1 + c_2 \text{N-EFIE}_2 + d_1 \eta_1 \text{T-MFIE}_1 - d_2 \eta_2 \text{T-MFIE}_2 \quad (2.43)$$

Sustituyendo (2.36), (2.37), (2.38) y (2.39) en (2.42), (2.43) y realizando operaciones se obtiene:

$$\begin{aligned} \text{JCFIE} : & a_1 \left(\mathbf{L}^1(\mathbf{J}) - \frac{1}{\eta_1} \mathbf{K}^1(\mathbf{M}) \right)_{tan} + a_2 \left(\mathbf{L}^2(\mathbf{J}) - \frac{1}{\eta_2} \mathbf{K}^2(\mathbf{M}) \right)_{tan} \\ & + b_1 \hat{n} \times \left(\mathbf{K}^1(\mathbf{J}) + \frac{1}{\eta_1} \mathbf{L}^1(\mathbf{M}) \right) - b_2 \hat{n} \times \left(\mathbf{K}^2(\mathbf{J}) + \frac{1}{\eta_2} \mathbf{L}^2(\mathbf{M}) \right) \\ & + \frac{1}{2} \left(\frac{a_1}{\eta_1} - \frac{a_2}{\eta_2} \right) \hat{n} \times \mathbf{M} + \frac{1}{2} (b_1 + b_2) \mathbf{J} \\ & = \frac{a_1}{\eta_1} (\mathbf{E}_1^{inc})_{tan} + \frac{a_2}{\eta_2} (\mathbf{E}_2^{inc})_{tan} + \hat{n} \times (b_1 \mathbf{H}_1^{inc} - b_2 \mathbf{H}_2^{inc}) \end{aligned} \quad (2.44)$$

$$\begin{aligned} \text{MCFIE} : & d_1 \left(\eta_1 \mathbf{K}^1(\mathbf{J}) + \mathbf{L}^1(\mathbf{M}) \right)_{tan} + d_2 \left(\eta_2 \mathbf{K}^2(\mathbf{J}) + \mathbf{L}^2(\mathbf{M}) \right)_{tan} \\ & - c_1 \hat{n} \times \left(\eta_1 \mathbf{L}^1(\mathbf{J}) - \mathbf{K}^1(\mathbf{M}) \right) - c_2 \hat{n} \times \left(-\eta_2 \mathbf{L}^2(\mathbf{J}) + \mathbf{K}^2(\mathbf{M}) \right) \\ & + \frac{1}{2} (-d_1 \eta_1 + d_2 \eta_2) \hat{n} \times (\mathbf{J}) + \frac{1}{2} (c_1 + c_2) (\mathbf{M}) \\ & = d_1 \eta_1 (\mathbf{H}_1^{inc})_{tan} - d_2 \eta_2 (\mathbf{H}_2^{inc})_{tan} - \hat{n} \times (c_1 \mathbf{E}_1^{inc} + c_2 \mathbf{E}_2^{inc}) \end{aligned} \quad (2.45)$$

donde $\hat{n}_1 = -\hat{n}_2 \equiv \hat{n}$, $\mathbf{J}_1 = -\mathbf{J}_2 \equiv \mathbf{J}$ y $\mathbf{M}_1 = -\mathbf{M}_2 \equiv \mathbf{M}$ ³.

Si se utiliza notación matricial, la combinación de las dos ecuaciones anteriores denominada, *Electric and Magnetic Current Combined Field Integral Equation* (JMCFIE) se expresa de la siguiente forma:

³Dado que las referencias a las regiones R_i , y por tanto a sus parámetros constitutivos (ϵ_i, μ_i) , desaparecen en las corrientes \mathbf{J} y \mathbf{M} , se hace necesario añadir dichas referencias a los operadores \mathbf{L}^i y \mathbf{K}^i

TABLA 2.1: Valores de coeficientes para distintas formulaciones.

| Formulación | a_i | b_i | c_i | d_i |
|-------------|----------|------------|-----------------|--------------------|
| PMCHWT | η_i | 1 | 1 | $\frac{1}{\eta_i}$ |
| CTF | 1 | 0 | 0 | 1 |
| CNF | 0 | 1 | 1 | 0 |
| MÜLLER | 0 | μ_{ri} | ϵ_{ri} | 0 |
| JMCFIE | 1 | 1 | 1 | 1 |

$$\begin{aligned}
& \left\{ \begin{array}{l} \left[\begin{array}{cc} (a_1 \mathbf{L}^1 + a_2 \mathbf{L}^2)_{tan} & \left(-\frac{a_1}{\eta_1} \mathbf{K}^1 - \frac{a_2}{\eta_2} \mathbf{K}^2 \right)_{tan} \\ (d_1 \eta_1 \mathbf{K}^1 + d_2 \eta_2 \mathbf{K}^2)_{tan} & (d_1 \mathbf{L}^1 + d_2 \mathbf{L}^2)_{tan} \end{array} \right] \\ + \left[\begin{array}{cc} \hat{n} \times (b_1 \mathbf{K}^1 - b_2 \mathbf{K}^2) & \hat{n} \times \left(\frac{b_1}{\eta_1} \mathbf{L}^1 - \frac{b_2}{\eta_2} \mathbf{L}^2 \right) \\ -\hat{n} \times (c_1 \eta_1 \mathbf{L}^1 - c_2 \eta_2 \mathbf{L}^2) & \hat{n} \times (c_1 \mathbf{K}^1 - c_2 \mathbf{K}^2) \end{array} \right] \\ + \frac{1}{2} \left[\begin{array}{cc} (b_1 + b_2) & \left(\frac{a_1}{\eta_1} - \frac{a_2}{\eta_2} \right) \hat{n} \times \\ (-d_1 \eta_1 + d_2 \eta_2) \hat{n} \times & (c_1 + c_2) \end{array} \right] \end{array} \right\} \begin{bmatrix} \mathbf{J} \\ \mathbf{M} \end{bmatrix} \\
= & \begin{bmatrix} \frac{a_1}{\eta_1} (\mathbf{E}_1^{inc})_{tan} - \frac{a_2}{\eta_2} (\mathbf{E}_2^{inc})_{tan} + \hat{n} \times (b_1 \mathbf{H}_1^{inc} + b_2 \mathbf{H}_2^{inc}) \\ d_1 \eta_1 (\mathbf{H}_1^{inc})_{tan} - d_2 \eta_2 (\mathbf{H}_2^{inc})_{tan} - \hat{n} \times (c_1 \mathbf{E}_1^{inc} + c_2 \mathbf{E}_2^{inc}) \end{bmatrix}
\end{aligned} \tag{2.46}$$

Los coeficientes a_i , b_i , c_i y d_i pueden tomar cualquier valor complejo para intentar obtener formulaciones válidas y estables. Dependiendo de la convergencia, precisión y los parámetros dieléctricos, existen determinados valores de los coeficientes que resultan idóneos. Las formulaciones más conocidas, que se encuentran ampliamente documentadas en la bibliografía aparecen en la tabla (2.1). Entre ellas, destacamos la formulación *Poggio-Miller-Chang-Harrington-Wu-Tsai* (PMCHWT) (Poggio y Miller, 1973; Chang y R. Harrington, 1977) y la *Combined Tangential Formulation* (CTF) (Pasi Ylä-Oijala, M Taskinen y Järvenpää, 2005) que combinan las ecuaciones tangenciales; *Combined Normal Formulation* (CNF) (Pasi Ylä-Oijala, M Taskinen y Järvenpää, 2005) y *Müller* (Müller y Higgins, 1960; P Ylä-Oijala y Matti Taskinen, 2005b) que combinan las ecuaciones normales y JMCFIE (P Ylä-Oijala y Matti Taskinen, 2005a) que combina ecuaciones tangenciales y normales, siendo la formulación más genérica, ampliamente utilizada por su precisión y buena convergencia.

2.5. Método de los Momentos

2.5.1. Formulación

MoM es una técnica introducida por Harrington (R. F. Harrington, 1993), que transforma ecuaciones integrales complejas en sistemas lineales que pueden resolverse de forma numérica. Es una de las técnicas numéricas más utilizadas en la actualidad en la resolución de ecuaciones integrales que aparecen en muchos problemas electromagnéticos de antenas y dispersión. Considérese el siguiente problema generalizado:

$$L(\mathbf{f}) = \mathbf{g} \quad (2.47)$$

donde L es un operador lineal, \mathbf{g} es la excitación o fuente, que se considera conocida y \mathbf{f} la respuesta o incógnita que debe ser determinada. En problemas electromagnéticos, L es un operador integro-diferencial, \mathbf{f} es la función incógnita y \mathbf{g} es una fuente de excitación conocida como por ejemplo una onda plana incidente.

El primer paso para resolver la ecuación (2.47) es expandir la función incógnita \mathbf{f} en una serie de funciones base conocidas, $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_n$, de la forma:

$$\mathbf{f} = \sum_n \alpha_n \mathbf{f}_n \quad (2.48)$$

donde α_n son coeficientes complejos desconocidos, y \mathbf{f}_n es un conjunto de funciones linealmente independientes, denominadas funciones de expansión o funciones base. Dado que las series de la ecuación (2.48) son infinitas, las funciones \mathbf{f}_n forman un conjunto completo de funciones base del espacio vectorial L_f . En la práctica, el sumatorio anterior se trunca obteniendo unas series finitas que dan una solución al problema aproximada.

$$\mathbf{f} \simeq \sum_{n=1}^N \alpha_n \mathbf{f}_n \quad (2.49)$$

Sustituyendo la ecuación (2.49) en (2.47) se obtiene:

$$\sum_{n=1}^N \alpha_n L(\mathbf{f}_n) \simeq \mathbf{g} \quad (2.50)$$

Dado que el operador L es lineal, se puede transformar la ecuación (2.50) en un sistema matricial de ecuaciones mediante la aplicación de la misma en una serie de m puntos o regiones sobre L_g . Para ello, se elige un conjunto adecuado de funciones de ponderación $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m$ y se realiza el producto escalar de dichas funciones con las funciones conocidas \mathbf{f}_n en el rango de L . Al sustituir en la ecuación (2.50) se obtiene un sistema de m ecuaciones lineales de la forma:

$$\sum_{n=1}^N \alpha_n \langle \mathbf{w}_m, L(\mathbf{f}_n) \rangle \simeq \langle \mathbf{w}_m, \mathbf{g} \rangle \quad (2.51)$$

con $m=1..M$. En general, se elige $M = N$ para que el sistema de ecuaciones tenga solución única.

En el MoM, el producto simétrico se define usualmente como

$$\langle \mathbf{f}, \mathbf{g} \rangle = \int_S \mathbf{f} \cdot \mathbf{g} ds \quad (2.52)$$

mientras que el producto escalar o interno se define como

$$\langle \mathbf{f}, \mathbf{g} \rangle = \int_S \mathbf{f} \cdot \mathbf{g}^* ds \quad (2.53)$$

donde el superíndice $*$ denota el complejo conjugado. En el caso de utilizar funciones reales, ambos productos coinciden.

El sistema de ecuaciones (2.51) se puede expresar en notación matricial de la forma

$$[Z] \cdot [I] = [V] \quad (2.54)$$

donde

$$[Z] = \begin{bmatrix} \langle \mathbf{w}_1, L(\mathbf{f}_1) \rangle & \langle \mathbf{w}_1, L(\mathbf{f}_2) \rangle & \cdots & \langle \mathbf{w}_1, L(\mathbf{f}_n) \rangle \\ \langle \mathbf{w}_2, L(\mathbf{f}_1) \rangle & \langle \mathbf{w}_2, L(\mathbf{f}_2) \rangle & \cdots & \langle \mathbf{w}_2, L(\mathbf{f}_n) \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{w}_m, L(\mathbf{f}_1) \rangle & \langle \mathbf{w}_m, L(\mathbf{f}_2) \rangle & \cdots & \langle \mathbf{w}_m, L(\mathbf{f}_n) \rangle \end{bmatrix} \quad (2.55)$$

$$[I] = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} \quad [V] = \begin{bmatrix} \langle \mathbf{w}_1, \mathbf{g} \rangle \\ \langle \mathbf{w}_2, \mathbf{g} \rangle \\ \vdots \\ \langle \mathbf{w}_m, \mathbf{g} \rangle \end{bmatrix} \quad (2.56)$$

El vector columna $[I]$ con las incógnitas del problema, que son los coeficientes de la expansión (2.49) de \mathbf{f} se obtiene resolviendo el sistema (2.54), de la forma

$$[I] = [Z^{-1}] \cdot [V] \quad (2.57)$$

donde Z^{-1} es la inversa de la matriz Z .

En problemas electromagnéticos, L es un operador integro-diferencial obtenido a partir de las ecuaciones de Maxwell y las condiciones de contorno, \mathbf{f} es la distribución de corriente inducida que se desea obtener y \mathbf{g} es un campo incidente conocido, la fuente de excitación del problema. Siguiendo la terminología de Harrington (R. F. Harrington, 1993) para este caso particular, la matriz Z se denomina matriz generalizada de impedancias o acoplos y el vector V se denomina vector de excitaciones.

La solución del sistema de ecuaciones matricial (2.54) se puede obtener mediante la matriz inversa o mediante métodos iterativos basados en factorizaciones. La elección adecuada del conjunto de funciones base \mathbf{f}_n y del conjunto de funciones de ponderación \mathbf{w}_m , es la clave para el adecuado comportamiento del método y afecta directamente a la precisión de la solución obtenida. Con el fin de que las soluciones obtenidas sean físicamente más creíbles y para que el sistema de ecuaciones resultante siempre cumpla el teorema de reacción, es preferible la utilización del producto simétrico en el procedimiento de ponderación, tal y como describe Wang (J. Wang, 1991).

En el caso concreto de utilizar el mismo conjunto de funciones base y de ponderación $\mathbf{w}_m \equiv \mathbf{f}_m$, se obtiene el denominado método de Galerkin. El uso del método de Galerkin, junto con un producto simétrico en el procedimiento de ponderación supone ventajas muy importantes en cuanto a los requisitos de cómputo y almacenamiento.

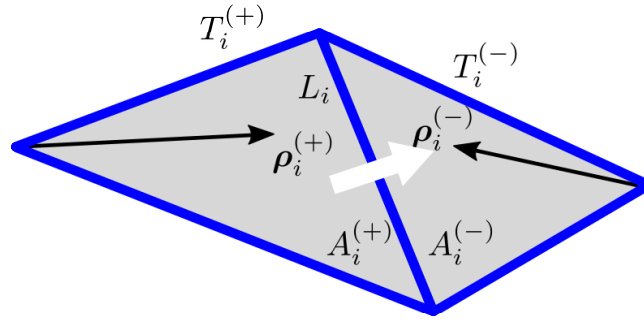


FIGURA 2.6: Rao-Wilton-Glisson (RWG) basis function.

2.5.2. Funciones base RWG

La elección adecuada de las funciones base y de ponderación es determinante para conseguir una precisión y eficiencia óptimas mediante MoM. A grandes rasgos, las funciones base modelan las densidades de corriente \mathbf{J} y \mathbf{M} sobre una discretización de la superficie. En el desarrollo de la Tesis se emplearon las funciones base de *Rao-Wilton-Glisson* (RWG) (M. Rao, R. Wilton y W. Glisson, 1982) tanto para la expansión de corrientes, como para el procedimiento de ponderación de las ecuaciones integrales (proc. de Galerkin). Aunque existen otras funciones base que pueden utilizarse, como las *Buffa-Christiansen functions* (BC) (Cools, Andriulli, De Zutter et al., 2011), las *RoofTop functions* (RT) (Andersson, 1993), etc., se ha decidido utilizar las bases RWG debido a su sencillez para implementarlas en código y a su versatilidad y precisión en el modelado de todo tipo de superficies por su mallado triangular.

Cada función base RWG es una función definida sobre cada par de triángulos $T_i^{(+)}$ y $T_i^{(-)}$, que comparten un lado de longitud L_i , tal y como se observa en la figura (2.6). Estas funciones base, f_i , se definen:

$$\mathbf{f}_i(\mathbf{r}') = \pm \frac{L_i}{2A_i^{(\pm)}} \vec{\rho}_i^{(\pm)}, \quad \mathbf{r}' \in T_i^{(\pm)} \quad (2.58)$$

donde $A_i^{(\pm)}$ y $\vec{\rho}_i^{(\pm)}$ son, respectivamente, el área y el vector de posición de los puntos fuente (\mathbf{r}') en el sistema local de coordenadas cilíndricas centrado en el vértice opuesto al lado común del triángulo $T_i^{(\pm)}$.

Al aplicar la divergencia a la función base RWG, se obtiene:

$$\nabla \cdot \mathbf{f}_i(\mathbf{r}') = \mp \frac{L_i}{A_i^{(\pm)}}, \quad \mathbf{r}' \in T_i^{(\pm)} \quad (2.59)$$

Dado que los valores de la divergencia en $T_i^{(+)}$ y $T_i^{(-)}$ son finitos, decimos que la función RWG es *div-conforming* (Z. Ergül y Gürel, 2006). Además, la divergencia de la corriente es proporcional a las densidades de carga a través de las ecuaciones de continuidad (2.5) y (2.7) de forma que la densidad de carga total en el par de triángulos que forman la base es nula. Por otra parte, la componente normal de la densidad de corriente en el borde exterior de la función no existe y por tanto no hay líneas de carga en la frontera de las funciones base.

2.5.3. Discretización de la SIE

Las ecuaciones (2.44) y (2.45) obtenidas a partir de la formulación integral de superficie, pueden discretizarse mediante el método de los momentos (R. F. Harrington, 1993). Las densidades de corriente \mathbf{J} y \mathbf{M} pueden aproximarse mediante expansión en una serie de funciones base conocidas $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_N$, de la forma:

$$\mathbf{J}(\mathbf{r}') = \sum_{i=1}^N J_i \mathbf{f}_i(\mathbf{r}') \quad (2.60)$$

$$\mathbf{M}(\mathbf{r}') = \sum_{i=1}^N M_i \mathbf{f}_i(\mathbf{r}') \quad (2.61)$$

donde J_i y M_i son coeficientes constantes complejos, y $\mathbf{f}_i(\mathbf{r}')$ es un conjunto de funciones linealmente independientes, denominadas funciones de expansión o funciones base.

De esta forma, las ecuaciones (2.44) y (2.45) pueden expresarse como:

$$\begin{aligned} \text{JCFIE} : & a_1 \sum_{i=1}^N \left(\mathbf{L}^1(\mathbf{f}_i) J_i - \frac{1}{\eta_1} \mathbf{K}^1(\mathbf{f}_i) M_i \right)_{tan} + a_2 \sum_{i=1}^N \left(\mathbf{L}^2(\mathbf{f}_i) J_i - \frac{1}{\eta_2} \mathbf{K}^2(\mathbf{f}_i) M_i \right)_{tan} \\ & + b_1 \sum_{i=1}^N \hat{n} \times \left(\mathbf{K}^1(\mathbf{f}_i) J_i + \frac{1}{\eta_1} \mathbf{L}^1(\mathbf{f}_i) M_i \right) - b_2 \sum_{i=1}^N \hat{n} \times \left(\mathbf{K}^2(\mathbf{f}_i) J_i + \frac{1}{\eta_2} \mathbf{L}^2(\mathbf{f}_i) M_i \right) \\ & + \sum_{i=1}^N \left(\frac{1}{2} \left(\frac{a_1}{\eta_1} - \frac{a_2}{\eta_2} \right) \hat{n} \times \mathbf{f}_i \right) M_i + \sum_{i=1}^N \left(\frac{1}{2} (b_1 + b_2) \mathbf{f}_i \right) J_i \\ & = \frac{a_1}{\eta_1} (\mathbf{E}_1^{inc})_{tan} + \frac{a_2}{\eta_2} (\mathbf{E}_2^{inc})_{tan} + \hat{n} \times (b_1 \mathbf{H}_1^{inc} - b_2 \mathbf{H}_2^{inc}) \end{aligned} \quad (2.62)$$

$$\begin{aligned}
\text{MCFIE} : & d_1 \sum_{i=1}^N (\eta_1 \mathbf{K}^1(\mathbf{f}_i) J_i + \mathbf{L}^1(\mathbf{f}_i) M_i)_{tan} + d_2 \sum_{i=1}^N (\eta_2 \mathbf{K}^2(\mathbf{f}_i) J_i + \mathbf{L}^2(\mathbf{f}_i) M_i)_{tan} \\
& - c_1 \sum_{i=1}^N \hat{n} \times (\eta_1 \mathbf{L}^1(\mathbf{f}_i) J_i - \mathbf{K}^1(\mathbf{f}_i) M_i) - c_2 \sum_{i=1}^N \hat{n} \times (-\eta_2 \mathbf{L}^2(\mathbf{f}_i) J_i + \mathbf{K}^2(\mathbf{f}_i) M_i) \\
& + \sum_{i=1}^N \left(\frac{1}{2} (-d_1 \eta_1 + d_2 \eta_2) \hat{n} \times \mathbf{f}_i \right) J_i + \sum_{i=1}^N \left(\frac{1}{2} (c_1 + c_2) \mathbf{f}_i \right) M_i \\
& = d_1 \eta_1 (\mathbf{H}_1^{inc})_{tan} - d_2 \eta_2 (\mathbf{H}_2^{inc})_{tan} - \hat{n} \times (c_1 \mathbf{E}_1^{inc} + c_2 \mathbf{E}_2^{inc})
\end{aligned} \tag{2.63}$$

Se define el producto simétrico mediante el operador definido en (2.52) como:

$$\langle \mathbf{f}, \mathbf{g} \rangle = \int_S \mathbf{f} \cdot \mathbf{g} dS'$$

Si utilizamos la notación genérica de función matemática

$$\mathcal{F}(\mathcal{X}) = \mathcal{Y}$$

para describir la ecuación integral lineal (2.46), es posible realizar el producto escalar en ambos miembros de la ecuación por una serie de funciones conocidas denominadas funciones de ponderación, w_m . Utilizando el procedimiento de ponderación de Galerkin (las funciones base son idénticas a las funciones de ponderación), $w_m = f_m$, sobre las funciones base se obtiene:

$$\langle \mathbf{f}_m, \mathcal{F}(\mathcal{X}) \rangle = \langle \mathbf{f}_m, \mathcal{Y} \rangle \tag{2.64}$$

En concreto, si aplicamos la ponderación con funciones \mathbf{f}_m , $m = 1, 2, \dots, N$ a las ecuaciones integrales (2.62) y (2.63), se obtiene el siguiente sistema de ecuaciones lineal de dimensiones $2N \times 2N$ formado por las N incógnitas de J y las N incógnitas de M :

$$\begin{aligned}
\text{JCFIE} : & a_1 \sum_{i=1}^N \langle \mathbf{f}_m, \mathbf{L}^1(\mathbf{f}_i) \rangle J_i - \frac{a_1}{\eta_1} \sum_{i=1}^N \langle \mathbf{f}_m, \mathbf{K}^1(\mathbf{f}_i) \rangle M_i + a_2 \sum_{i=1}^N \langle \mathbf{f}_m, \mathbf{L}^2(\mathbf{f}_i) \rangle J_i \\
& - \frac{a_2}{\eta_2} \sum_{i=1}^N \langle \mathbf{f}_m, \mathbf{K}^2(\mathbf{f}_i) \rangle M_i + b_1 \sum_{i=1}^N \langle \mathbf{f}_m, \hat{n} \times \mathbf{K}^1(\mathbf{f}_i) \rangle J_i + \frac{b_1}{\eta_1} \sum_{i=1}^N \langle \mathbf{f}_m, \hat{n} \times \mathbf{L}^1(\mathbf{f}_i) \rangle M_i \\
& - b_2 \sum_{i=1}^N \langle \mathbf{f}_m, \hat{n} \times \mathbf{K}^2(\mathbf{f}_i) \rangle J_i - \frac{b_2}{\eta_2} \sum_{i=1}^N \langle \mathbf{f}_m, \hat{n} \times \mathbf{L}^2(\mathbf{f}_i) \rangle M_i \\
& + \frac{1}{2} \left(\frac{a_1}{\eta_1} - \frac{a_2}{\eta_2} \right) \sum_{i=1}^N \langle \mathbf{f}_m, \hat{n} \times \mathbf{f}_i \rangle M_i + \frac{1}{2} (b_1 + b_2) \sum_{i=1}^N \langle \mathbf{f}_m, \mathbf{f}_i \rangle J_i \\
& = \langle \mathbf{f}_m, \frac{a_1}{\eta_1} (\mathbf{E}_1^{inc}) + \frac{a_2}{\eta_2} (\mathbf{E}_2^{inc}) + \hat{n} \times (b_1 \mathbf{H}_1^{inc} - b_2 \mathbf{H}_2^{inc}) \rangle
\end{aligned} \tag{2.65}$$

$$\begin{aligned}
\text{MCFIE} : & d_1 \eta_1 \sum_{i=1}^N \langle \mathbf{f}_m, \mathbf{K}^1(\mathbf{f}_i) \rangle J_i + d_1 \sum_{i=1}^N \langle \mathbf{f}_m, \mathbf{L}^1(\mathbf{f}_i) \rangle M_i + d_2 \eta_2 \sum_{i=1}^N \langle \mathbf{f}_m, \mathbf{K}^2(\mathbf{f}_i) \rangle J_i \\
& + d_2 \sum_{i=1}^N \langle \mathbf{f}_m, \mathbf{L}^2(\mathbf{f}_i) \rangle M_i - c_1 \eta_1 \sum_{i=1}^N \langle \mathbf{f}_m, \hat{n} \times \mathbf{L}^1(\mathbf{f}_i) \rangle J_i - c_1 \sum_{i=1}^N \langle \mathbf{f}_m, \hat{n} \times \mathbf{K}^1(\mathbf{f}_i) \rangle M_i \\
& + c_2 \eta_2 \sum_{i=1}^N \langle \mathbf{f}_m, \hat{n} \times \mathbf{L}^2(\mathbf{f}_i) \rangle J_i - c_2 \sum_{i=1}^N \langle \mathbf{f}_m, \hat{n} \times \mathbf{K}^2(\mathbf{f}_i) \rangle M_i \\
& + \frac{1}{2} ((-d_1 \eta_1 + d_2 \eta_2) \sum_{i=1}^N \langle \mathbf{f}_m, \hat{n} \times \mathbf{f}_i \rangle J_i + \frac{1}{2} (c_1 + c_2) \sum_{i=1}^N \langle \mathbf{f}_m, \mathbf{f}_i \rangle M_i \\
& = \langle \mathbf{f}_m, d_1 \eta_1 (\mathbf{H}_1^{inc})_{tan} - d_2 \eta_2 (\mathbf{H}_2^{inc})_{tan} - \hat{n} \times (c_1 \mathbf{E}_1^{inc} + c_2 \mathbf{E}_2^{inc}) \rangle
\end{aligned} \tag{2.66}$$

Las expresiones anteriores pueden reagruparse en función de los parámetros

$A_{mi}^r, B_{mi}^r, A_{mi}^{r'}, B_{mi}^{r'}, \tau_{mi}$ y τ_{mi}' :

$$\begin{aligned}
\text{JCFIE} : & a_1 \sum_{i=1}^N A_{mi}^1 J_i - \frac{a_1}{\eta_1} \sum_{i=1}^N B_{mi}^1 M_i + a_2 \sum_{i=1}^N A_{mi}^2 J_i - \frac{a_2}{\eta_2} \sum_{i=1}^N B_{mi}^2 M_i \\
& + b_1 \sum_{i=1}^N B_{mi}^{1'} J_i + \frac{b_1}{\eta_1} \sum_{i=1}^N A_{mi}^{1'} M_i - b_2 \sum_{i=1}^N B_{mi}^{2'} J_i - \frac{b_2}{\eta_2} \sum_{i=1}^N A_{mi}^{2'} M_i \\
& + \frac{1}{2} \left(\frac{a_1}{\eta_1} - \frac{a_2}{\eta_2} \right) \sum_{i=1}^N \tau_{mi} M_i + \frac{1}{2} (b_1 + b_2) \sum_{i=1}^N \tau_{mi}' J_i \\
& = \langle \mathbf{f}_m, \frac{a_1}{\eta_1} (\mathbf{E}_1^{inc}) + \frac{a_2}{\eta_2} (\mathbf{E}_2^{inc}) + \hat{n} \times (b_1 \mathbf{H}_1^{inc} - b_2 \mathbf{H}_2^{inc}) \rangle
\end{aligned} \tag{2.67}$$

$$\begin{aligned}
\text{MCFIE} : & d_1 \eta_1 \sum_{i=1}^N B_{mi}^1 J_i + d_1 \sum_{i=1}^N A_{mi}^1 M_i + d_2 \eta_2 \sum_{i=1}^N B_{mi}^2 J_i + d_2 \sum_{i=1}^N A_{mi}^2 M_i \\
& - c_1 \eta_1 \sum_{i=1}^N A_{mi}^{1'} J_i + c_1 \sum_{i=1}^N B_{mi}^{1'} M_i + c_2 \eta_2 \sum_{i=1}^N A_{mi}^{2'} J_i - c_2 \sum_{i=1}^N B_{mi}^{2'} M_i \\
& + \frac{1}{2} ((-d_1 \eta_1 + d_2 \eta_2) \sum_{i=1}^N \tau_{mi} J_i + \frac{1}{2} (c_1 + c_2) \sum_{i=1}^N \tau_{mi}' M_i) \\
& = \langle \mathbf{f}_m, d_1 \eta_1 (\mathbf{H}_1^{inc})_{tan} - d_2 \eta_2 (\mathbf{H}_2^{inc})_{tan} - \hat{n} \times (c_1 \mathbf{E}_1^{inc} + c_2 \mathbf{E}_2^{inc}) \rangle
\end{aligned} \tag{2.68}$$

donde:

$$\begin{aligned}
A_{mi}^r &= \langle \mathbf{f}_m, \mathbf{L}^r(\mathbf{f}_i) \rangle \\
B_{mi}^r &= \langle \mathbf{f}_m, \mathbf{K}^r(\mathbf{f}_i) \rangle \\
A_{mi}^{r'} &= \langle \mathbf{f}_m, \hat{n} \times \mathbf{L}^r(\mathbf{f}_i) \rangle \\
B_{mi}^{r'} &= \langle \mathbf{f}_m, \hat{n} \times \mathbf{K}^r(\mathbf{f}_i) \rangle \\
\tau_{mi} &= \langle \mathbf{f}_m, \hat{n} \times \mathbf{f}_i \rangle \\
\tau_{mi}' &= \langle \mathbf{f}_m, \mathbf{f}_i \rangle
\end{aligned}$$

Utilizando notación matricial, las dos ecuaciones anteriores pueden expresarse mediante un sistema lineal de ecuaciones:

$$\begin{bmatrix} \mathbf{Z}^{1J} & \mathbf{Z}^{1M} \\ \mathbf{Z}^{2J} & \mathbf{Z}^{2M} \end{bmatrix}_{2N \times 2N} \cdot \begin{bmatrix} \mathbf{J} \\ \mathbf{M} \end{bmatrix}_{2N \times 1} = \begin{bmatrix} \mathbf{V}^1 \\ \mathbf{V}^2 \end{bmatrix}_{2N \times 1} \tag{2.69}$$

en el que las cuatro submatrices de dimensiones $N \times N$, vienen dadas por las siguientes expresiones:

$$\begin{aligned}
\mathbf{Z}_{m,i}^{1J} &= a_1 A_{mi}^1 + a_2 A_{mi}^2 + b_1 B_{mi}^{1'} - b_2 B_{mi}^{2'} + \frac{1}{2}(b_1 + b_2) \tau'_{mi} \\
\mathbf{Z}_{m,i}^{1M} &= -\frac{a_1}{\eta_1} B_{mi}^1 - \frac{a_2}{\eta_2} B_{mi}^2 + \frac{b_1}{\eta_1} A_{mi}^{1'} - \frac{b_2}{\eta_2} A_{mi}^{2'} + \frac{1}{2} \left(\frac{a_1}{\eta_1} - \frac{a_2}{\eta_2} \right) \tau_{mi} \\
\mathbf{Z}_{m,i}^{2J} &= d_1 \eta_1 B_{mi}^1 + d_2 \eta_2 B_{mi}^2 - c_1 \eta_1 A_{mi}^{1'} + c_2 \eta_2 A_{mi}^{2'} + \frac{1}{2} (-d_1 \eta_1 + d_2 \eta_2) \tau_{mi} \\
\mathbf{Z}_{m,i}^{2M} &= d_1 A_{mi}^1 + d_2 A_{mi}^2 + c_1 B_{mi}^{1'} - c_2 B_{mi}^{2'} + \frac{1}{2} (c_1 + c_2) \tau'_{mi}
\end{aligned} \tag{2.70}$$

donde $\mathbf{i} = 1, 2, \dots, \mathbf{N}$ y $\mathbf{m} = 1, 2, \dots, \mathbf{N}$.

Los vectores \mathbf{J} y \mathbf{M} con las incógnitas del problema vienen definidos por:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \\ \vdots \\ \mathbf{J}_N \end{bmatrix} \quad \mathbf{M} = \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \vdots \\ \mathbf{M}_N \end{bmatrix}$$

y los vectores de excitaciones \mathbf{V}^1 y \mathbf{V}^2 se expresan:

$$\begin{aligned}
\mathbf{V}_{\mathbf{m}}^1 &= \langle \mathbf{f}_{\mathbf{m}}, \frac{a_1}{\eta_1} (\mathbf{E}_1^{inc}) + \frac{a_2}{\eta_2} (\mathbf{E}_2^{inc}) + \hat{n} \times (b_1 \mathbf{H}_1^{inc} - b_2 \mathbf{H}_2^{inc}) \rangle \\
\mathbf{V}_{\mathbf{m}}^2 &= \langle \mathbf{f}_{\mathbf{m}}, d_1 \eta_1 (\mathbf{H}_1^{inc})_{tan} - d_2 \eta_2 (\mathbf{H}_2^{inc})_{tan} - \hat{n} \times (c_1 \mathbf{E}_1^{inc} + c_2 \mathbf{E}_2^{inc}) \rangle
\end{aligned} \tag{2.71}$$

donde $\mathbf{m} = 1, 2, \dots, \mathbf{N}$.

2.6. Extensión de la formulación a múltiples objetos

En esta sección se realiza una ampliación del análisis efectuado en las secciones 2.4 y 2.5 para el caso de más de un objeto dispersor. Al igual que en el caso anterior se obvia la variación temporal armónica $\exp(j\omega t)$.

Se plantea un nuevo escenario, en el que se introducen un determinado número de objetos de composición y geometría arbitraria, (Regiones R_i), con permitividad ϵ_i y permeabilidad μ_i en un medio homogéneo, (Región R_1), con permitividad ϵ_1 y permeabilidad μ_1 , tal y como se observa en la figura

(2.7). Definamos S_{ij} como la interfaz de separación entre dos medios R_i y R_j , siendo R_j la región interna de la superficie y cumpliéndose siempre que $i < j$. Se define \hat{n}_{ij} como el vector unitario normal a S_{ij} apuntando hacia la región R_i , cumpliéndose que $\hat{n}_{ij} = -\hat{n}_{ji}$. Por último, se denomina \mathbf{E}_i^{inc} y \mathbf{H}_i^{inc} a los campos eléctricos y magnéticos incidentes debido a las fuentes ubicadas en la región i , teniendo presente que los campos incidentes en los medios sin fuentes son nulos.

El campo electromagnético total en la región R_i puede expresarse como la suma del campo incidente en la región R_i y los campos dispersos producidos por las corrientes generadas en la superficie que rodea a la región R_i :

$$\begin{aligned}\mathbf{E}_i &= \mathbf{E}_i^{inc} + \sum_{j'} \mathbf{E}_i^{scat}(\mathbf{J}_{ij'}, \mathbf{M}_{ij'}) \\ \mathbf{H}_i &= \mathbf{H}_i^{inc} + \sum_{j'} \mathbf{H}_i^{scat}(\mathbf{J}_{ij'}, \mathbf{M}_{ij'})\end{aligned}\tag{2.72}$$

donde j' hace referencia a cada una de las regiones adyacentes a la región R_i . Además, los campos dispersos pueden expresarse como:

$$\begin{aligned}\mathbf{E}_i^{scat}(\mathbf{J}_{ij'}, \mathbf{M}_{ij'}) &= -\eta_i \mathbf{L}\{\mathbf{J}_{ij'}\} + \mathbf{K}\{\mathbf{M}_{ij'}\} \\ \mathbf{H}_i^{scat}(\mathbf{J}_{ij'}, \mathbf{M}_{ij'}) &= -\mathbf{K}\{\mathbf{J}_{ij'}\} - \frac{1}{\eta_i} \mathbf{L}\{\mathbf{M}_{ij'}\}\end{aligned}\tag{2.73}$$

siendo η_i la impedancia intrínseca del medio y \mathbf{L} y \mathbf{K} los operadores definidos en (2.31).

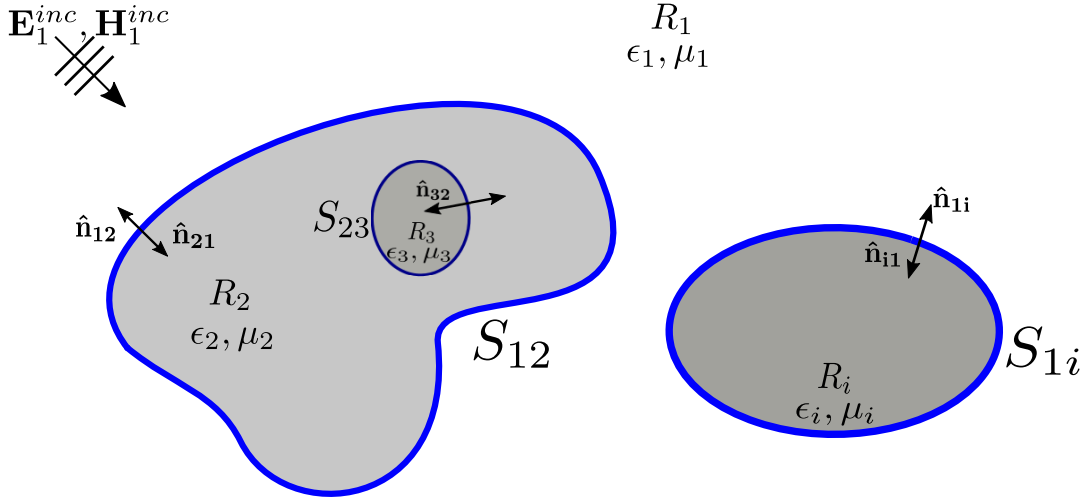


FIGURA 2.7: Geometría superficial del problema con múltiples objetos

Sustituyendo los campos totales en función de los campos incidentes y dispersos y aplicando las condiciones de contorno en las superficies de discontinuidad de forma similar al procedimiento seguido en (2.34) y (2.35), se obtiene:

$$\text{T-EFIE}_i : \quad \mathbf{E}_{i,tan}^{inc} = \hat{\mathbf{n}}_{ij} \times \mathbf{M}_{ij} + \sum_{j'} \left([\eta_i \mathbf{L}_0 \{ \mathbf{J}_{ij'} \} - \mathbf{K}_0 \{ \mathbf{M}_{ij'} \}]_{tan} - \frac{1}{2} \hat{\mathbf{n}}_{ij} \times \mathbf{M}_{ij'} \right) \quad (2.74)$$

$$\text{T-MFIE}_i : \quad \mathbf{H}_{i,tan}^{inc} = -\hat{\mathbf{n}}_{ij} \times \mathbf{J}_{ij} + \sum_{j'} \left(\left[\mathbf{K}_0 \{ \mathbf{J}_{ij'} \} + \frac{1}{\eta_i} \mathbf{L}_0 \{ \mathbf{M}_{ij'} \} \right]_{tan} + \frac{1}{2} \hat{\mathbf{n}}_{ij} \times \mathbf{J}_{ij'} \right) \quad (2.75)$$

$$\text{N-EFIE}_i : \quad \hat{\mathbf{n}}_{ij} \times \mathbf{E}_i^{inc} = -\mathbf{M}_{ij} + \hat{\mathbf{n}}_{ij} \times \sum_{j'} \left(\eta_i \mathbf{L}_0 \{ \mathbf{J}_{ij'} \} - \mathbf{K}_0 \{ \mathbf{M}_{ij'} \} - \frac{1}{2} \hat{\mathbf{n}}_{ij} \times \mathbf{M}_{ij'} \right) \quad (2.76)$$

$$\text{N-MFIE}_i : \quad \hat{\mathbf{n}}_{ij} \times \mathbf{H}_i^{inc} = \mathbf{J}_{ij} + \hat{\mathbf{n}}_{ij} \times \sum_{j'} \left(\mathbf{K}_0 \{ \mathbf{J}_{ij'} \} + \frac{1}{\eta_i} \mathbf{L}_0 \{ \mathbf{M}_{ij'} \} + \frac{1}{2} \hat{\mathbf{n}}_{ij} \times \mathbf{J}_{ij'} \right) \quad (2.77)$$

Las ecuaciones anteriores, pueden combinarse tal y como vimos en (2.40) y (2.41) para obtener las formulaciones JCFIE y MCFIE en cada una de las regiones R_i . Para combinar las ecuaciones CFIE de distintas regiones se utilizarán las ecuaciones (2.42) y (2.43). Si a continuación se aplica el procedimiento de discretización mediante el método de los momentos visto en la sección 2.5, se obtiene el siguiente sistema de ecuaciones lineal en notación matricial:

$$\begin{bmatrix} \mathbf{Z}_{ij,ij} & \mathbf{Z}_{ij,kl} & \dots & \mathbf{Z}_{ij,pq} \\ \mathbf{Z}_{kl,ij} & \mathbf{Z}_{kl,kl} & \dots & \mathbf{Z}_{kl,pq} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Z}_{pq,ij} & \mathbf{Z}_{pq,kl} & \dots & \mathbf{Z}_{pq,pq} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_{ij} \\ \mathbf{I}_{kl} \\ \vdots \\ \mathbf{I}_{pq} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{ij} \\ \mathbf{V}_{kl} \\ \vdots \\ \mathbf{V}_{pq} \end{bmatrix} \quad (2.78)$$

donde cada subbloque $\mathbf{Z}_{rs,tu}$ representa el acoplo de la superficie S_{tu} sobre la superficie S_{rs} . Cada uno de estos subbloques se expresan como:

$$\mathbf{Z}_{rs,tu} = \begin{bmatrix} \mathbf{Z}_{rs,tu}^{1J} & \mathbf{Z}_{rs,tu}^{1M} \\ \mathbf{Z}_{rs,tu}^{2J} & \mathbf{Z}_{rs,tu}^{2M} \end{bmatrix}$$

y los vectores \mathbf{I}_{rs} y \mathbf{V}_{rs} se definen como:

$$\mathbf{I}_{rs} = \begin{bmatrix} \mathbf{J}_{rs} \\ \mathbf{M}_{rs} \end{bmatrix} \quad \mathbf{V}_{rs} = \begin{bmatrix} \mathbf{V}_{rs}^1 \\ \mathbf{V}_{rs}^2 \end{bmatrix}$$

Los subbloques $\mathbf{Z}_{rs,tu}$ de la matriz de impedancias representan la interacción electromagnética entre las superficies S_{rs} y S_{tu} . Se pueden distinguir cinco casos diferenciados dependiendo de la relación existente entre las superficies.

El primer caso se produce cuando las superficies S_{rs} y S_{tu} son las mismas, y es conocido como autocoplo. Estos bloques se ubican en la diagonal de la matriz de impedancias y se definen como:

$$\begin{aligned} \mathbf{Z}_{rs,rs}^{1J}[m,i] &= a_r A_{mi}^r + a_s A_{mi}^s + b_r B_{mi}^{r'} - b_s B_{mi}^{s'} + \frac{1}{2}(b_r + b_s) \tau_{mi}' \\ \mathbf{Z}_{rs,rs}^{1M}[m,i] &= -\frac{a_r}{\eta_r} B_{mi}^r - \frac{a_s}{\eta_s} B_{mi}^s + \frac{b_r}{\eta_r} A_{mi}^{r'} - \frac{b_s}{\eta_s} A_{mi}^{s'} + \frac{1}{2} \left(\frac{a_r}{\eta_r} - \frac{a_s}{\eta_s} \right) \tau_{mi} \\ \mathbf{Z}_{rs,rs}^{2J}[m,i] &= d_r \eta_r B_{mi}^r + d_s \eta_s B_{mi}^s - c_r \eta_r A_{mi}^{r'} + c_s \eta_s A_{mi}^{s'} + \frac{1}{2} (-d_r \eta_r + d_s \eta_s) \tau_{mi} \\ \mathbf{Z}_{rs,rs}^{2M}[m,i] &= d_r A_{mi}^r + d_s A_{mi}^s + c_r B_{mi}^{r'} - c_s B_{mi}^{s'} + \frac{1}{2} (c_r + c_s) \tau_{mi}' \end{aligned} \quad (2.79)$$

Cuando la superficie S_{rs} recibe la excitación de la superficie S_{ru} a través del medio R_r (exterior a S_{rs} y a S_{ru}), el acoplo se determina:

$$\begin{aligned}
\mathbf{Z}_{rs,ru}^{1J}[m, i] &= a_r A_{mi}^r + b_r B_{mi}^{r'} \\
\mathbf{Z}_{rs,ru}^{1M}[m, i] &= -\frac{a_r}{\eta_r} B_{mi}^r + \frac{b_r}{\eta_r} A_{mi}^{r'} \\
\mathbf{Z}_{rs,ru}^{2J}[m, i] &= d_r \eta_r B_{mi}^{r'} - c_r \eta_r A_{mi}^{r'} \\
\mathbf{Z}_{rs,ru}^{2M}[m, i] &= d_r A_{mi}^r + c_r B_{mi}^{r'}
\end{aligned} \tag{2.80}$$

Si la superficie S_{rs} recibe la excitación de la superficie S_{st} a través del medio R_s (interior a S_{rs} y exterior a S_{st}), el acoplo se determina:

$$\begin{aligned}
\mathbf{Z}_{rs,st}^{1J}[m, i] &= -a_s A_{mi}^s + b_s B_{mi}^{s'} \\
\mathbf{Z}_{rs,st}^{1M}[m, i] &= \frac{a_s}{\eta_s} B_{mi}^s + \frac{b_s}{\eta_s} A_{mi}^{s'} \\
\mathbf{Z}_{rs,st}^{2J}[m, i] &= -d_s \eta_s B_{mi}^{s'} - c_s \eta_s A_{mi}^{s'} \\
\mathbf{Z}_{rs,st}^{2M}[m, i] &= -d_s A_{mi}^s + c_s B_{mi}^{s'}
\end{aligned} \tag{2.81}$$

En cambio, si la superficie S_{st} recibe la excitación de la superficie S_{rs} a través del medio R_s (exterior a S_{st} e interior a S_{rs}), el acoplo se determina:

$$\begin{aligned}
\mathbf{Z}_{st,rs}^{1J}[m, i] &= a_s A_{mi}^s + b_s B_{mi}^{s'} \\
\mathbf{Z}_{st,rs}^{1M}[m, i] &= -\frac{a_s}{\eta_s} B_{mi}^s + \frac{b_s}{\eta_s} A_{mi}^{s'} \\
\mathbf{Z}_{st,rs}^{2J}[m, i] &= d_s \eta_s B_{mi}^{s'} - c_s \eta_s A_{mi}^{s'} \\
\mathbf{Z}_{st,rs}^{2M}[m, i] &= d_s A_{mi}^s + c_s B_{mi}^{s'}
\end{aligned} \tag{2.82}$$

Por último, se puede dar el caso en el que las superficies no tengan ningún medio en común, ya sea el caso del acoplo entre las superficies S_{rs} y S_{tu} o bien entre las superficies S_{ps} y S_{rs} :

$$\begin{aligned}
\mathbf{Z}_{ps,rs}^{1J}[m, i] &= 0 \\
\mathbf{Z}_{ps,rs}^{1M}[m, i] &= 0 \\
\mathbf{Z}_{ps,rs}^{2J}[m, i] &= 0 \\
\mathbf{Z}_{ps,rs}^{2M}[m, i] &= 0
\end{aligned} \tag{2.83}$$

Los operadores A_{mi}^s , $A_{mi}^{/s}$, B_{mi}^s , $B_{mi}^{/s}$, τ_{mi} y τ_{mi}' se encuentran definidos en (2.68) y los coeficientes a_s , b_s , c_s y d_s que permiten seleccionar las formulaciones más extendidas en la comunidad científica aparecen en la tabla (2.1).

Se pueden encontrar otros detalles y otro estilo de formulación en las tesis anteriores del grupo realizadas por Javier Rivero (Rivero Campos, 2012) y Diego Martínez (Martínez Solís, 2014).

Capítulo 3

Métodos iterativos

| | | |
|-------|--|----|
| 3.1 | Introducción y conceptos básicos | 40 |
| 3.2 | Métodos estacionarios | 41 |
| 3.2.1 | Convergencia | 43 |
| 3.3 | Métodos de Krylov | 44 |

En el capítulo anterior se ha planteado un escenario genérico con múltiples objetos de composición y geometría arbitraria, en el que se formulan las ecuaciones integrales del campo electromagnético para cada una de las regiones que aparecen. Una vez planteado el problema, dichas ecuaciones integrales complejas son transformadas en sistemas de ecuaciones lineales mediante el método de los momentos. La discretización de la formulación integral conduce a un sistema de ecuaciones lineales en notación matricial (2.78) del tipo $Ax = b$ que debe ser resuelto, donde A representa la matriz de impedancias, x las densidades de corriente superficiales J y M (incógnitas del problema) y b el vector de excitaciones.

De forma general, la resolución de sistemas de ecuaciones del tipo $Ax = b$ se puede afrontar mediante dos tipos de métodos diferentes:

- **Métodos directos:** Obtienen la solución exacta en un número finito de iteraciones mediante la resolución matemática del sistema de ecuaciones. Presentan un elevado coste computacional, $O(N^3)$, y de consumo de memoria RAM, $O(N^2)$, siendo N el número de incógnitas. Además, si partimos de matrices dispersas (*Sparse Matrices 1973*) (matrices con un gran número de elementos a 0), las transformaciones que se aplican llevan a una matriz llena.
- **Métodos iterativos:** En cada iteración generan una aproximación, cada vez más exacta, de la solución final partiendo de una estimación inicial. Distinguimos entre métodos estacionarios y métodos basados en subespacios de Krylov. Reducen el coste computacional a $O(N^2)$.

Los métodos iterativos son más adecuados en la resolución de grandes sistemas de ecuaciones por el menor coste computacional y porque no modifican la matriz original, de forma que si se parte de matrices dispersas se pueden aplicar esquemas de almacenamiento optimizados.

Además, en la resolución de problemas electromagnéticos se utilizan métodos avanzados que dividen la geometría del problema en grupos y aproximan las interacciones entre dichos grupos, como por ejemplo FMM o MLF-MA, que consiguen reducir la complejidad computacional a $O(N \log N)$.

3.1. Introducción y conceptos básicos

Supongamos que se desea resolver el sistema de ecuaciones matricial

$$Ax = b \quad (3.1)$$

donde $A \in \mathbb{C}^n$ representa la matriz de impedancias, $x \in \mathbb{C}^n$ son las incógnitas del problema en forma de densidades de corriente superficiales y $b \in \mathbb{C}^n$ representa el vector de excitaciones.

Un método iterativo obtiene la solución aproximada del problema, x , en forma de una sucesión de vectores, x_1, x_2, \dots, x_k , partiendo de un vector inicial arbitrario x_0 (Y. Saad, 2003).

En cada iteración se obtiene un vector error, $e_k = x - x_k$, y un vector residuo, $r_k = b - Ax_k$. Dado que un método iterativo nunca ofrece una solución exacta con una sucesión finita, se debe establecer un criterio de parada sobre el residuo. Este criterio puede ser absoluto si $\|r_k\| < TOL$, o relativo si $\frac{\|r_k\|}{\|b\|} < TOL$, siendo $\|\cdot\|$ la norma vectorial y TOL el error máximo permitido.

Se dice que un método iterativo converge cuando $\lim_{k \rightarrow \infty} x_k = x$.

3.2. Métodos estacionarios

Dado el sistema de ecuaciones lineal $Ax = b$ se puede aplicar la partición $A = M - N$, siendo $M \neq A$ una matriz invertible. De esta forma, el sistema anterior se transforma en $Mx = Nx + b$.

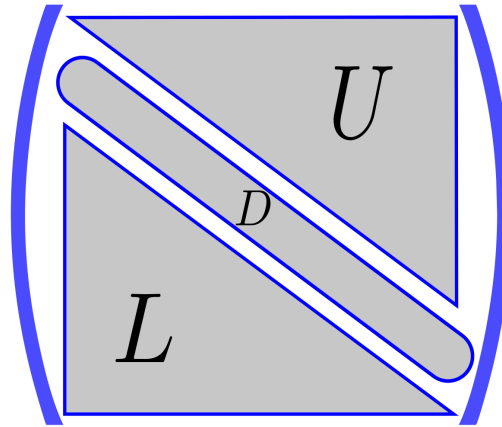
Si se considera la aproximación de la solución en la iteración $k + 1$, a partir de la aproximación en la iteración k , el sistema anterior puede reescribirse de la forma:

$$Mx_{k+1} = Nx_k + b \quad (3.2)$$

Despejando de la ecuación anterior, la solución aproximada en la iteración $k + 1$ se expresa:

$$x_{k+1} = Gx_k + f \quad (3.3)$$

siendo $G = M^{-1}N$ y $f = M^{-1}b$. Dado que la matriz de iteración G y el vector de iteración f son constantes en todo el proceso se dice que el método es *estacionario*.

FIGURA 3.1: Descomposición aditiva de la matriz A

Si aplicamos la igualdad $A = M - N$, la solución aproximada x_{k+1} se puede expresar de la siguiente forma en función del vector residuo:

$$x_{k+1} = x_k + M^{-1}(b - Ax_k) = x_k + M^{-1}r_k \quad (3.4)$$

Considérese la descomposición aditiva de la matriz A en $A = L + D + U$ siendo D la diagonal de A , L la parte estrictamente triangular inferior de A y U la parte estrictamente triangular superior de A tal y como aparece en la figura 3.1.

A continuación se presentan algunos métodos iterativos clásicos dependiendo de la elección de la matriz M (Ginestar, 2018-2019):

1. **Método de Jacobi.** Si se considera que la matriz M es igual a la diagonal de A , $M = D$, y se sustituye en (2.2) se obtiene:

$$x_{k+1} = x_k + D^{-1}(b - Ax_k) \quad (3.5)$$

que puede reescribirse como:

$$Dx_{k+1} = Dx_k + b - (L + D + U)x_k = b - Lx_k - Ux_k \quad (3.6)$$

2. **Método de Gauss-Seidel.** Si se considera que la matriz M es igual a la diagonal y la parte estrictamente triangular inferior de A , $M = D + L$, y se sustituye en (2.2) se obtiene:

$$x_{k+1} = x_k + (D + L)^{-1}(b - Ax_k) \quad (3.7)$$

que puede reescribirse como:

$$Dx_{k+1} = -Lx_{k+1} + (D + L)x_k + b - (L + D + U)x_k = b - Lx_{k+1} - Ux_k \quad (3.8)$$

3. **Método SOR, Successive Over Relaxation.** Consiste en una generalización de los métodos previos introduciendo un parámetro ω para acelerar la convergencia. Para ello se toma una descomposición de la matriz A de la forma:

$$\omega A = (D + \omega L) - (-\omega U + (1 - \omega)D) \quad (3.9)$$

que da lugar a la siguiente aproximación de la solución en la iteración $k + 1$:

$$x_{k+1} = (1 - \omega)x_k + \omega D^{-1} [b - Lx_{k+1} - Ux_k] \quad (3.10)$$

3.2.1. Convergencia

A continuación se clarificarán una serie de conceptos que nos resultarán útiles a la hora de analizar la convergencia de los métodos que acabamos de describir:

- Una matriz A de tamaño $n \times n$ es estrictamente diagonal dominante si verifica que sus elementos $|a_{ii}| > \sum_{j \neq i} |a_{ij}| \forall i$.
- Una matriz A de tamaño $n \times n$ es definida positiva si para todos los vectores no nulos $z \in \mathbb{C}^n$ se cumple que $z^T A z > 0 \forall z \neq 0$ siendo z^T el traspuesto del vector z .

La convergencia en el método de Jacobi se cumple para matrices diagonalmente dominantes. Se puede comprobar que el método de Gauss-Seidel es convergente además en matrices definidas positivas. En cuanto al método SOR, se cumple que en matrices diagonalmente dominantes converge si $0 < \omega \leq 1$ y en matrices definidas positivas converge si $0 < \omega < 2$.

3.3. Métodos de Krylov

Si partimos del sistema de ecuaciones matricial planteado en la introducción:

$$Ax = b \quad (3.11)$$

un método de proyección aproxima la solución x_m a partir de un subespacio $x_0 + K_m$ de dimensión m considerando la ortogonalidad del residuo $r_m = b - Ax_m$ con respecto a un nuevo subespacio L formado por m vectores linealmente independientes:

$$b - Ax \perp L \quad (3.12)$$

Se define el subespacio de Krylov de orden m , $K_m(A; v_0)$, generado por la matriz A y el vector v_0 , como el espacio expandido por $v_0, Av_0, \dots, A^{m-1}v_0$. A su vez, la secuencia de vectores v_0, \dots, v_{m-1} se considera base del espacio de Krylov generado por A si para cada m , los vectores v_0, \dots, v_{m-1} expanden el subespacio de Krylov $K_m(A; v_0)$.

Un método de proyección basado en subespacios de Krylov tiene como subespacio K_m el subespacio de Krylov $K_m(A; v_0)$ en el que el vector v_0 es el residuo r_0 .

Existen diferentes aproximaciones basadas en subespacios de Krylov dependiendo de la elección de cada uno de los subespacios L y K y de cómo se ha preconditionado el sistema (Youcef Saad, 1989; Vorst, 2003):

- Aproximación de Ritz-Galerkin ($L_m = K_m = K_m(A; r_0)$). En cada iteración se va construyendo la aproximación de la solución x_m , en la cual el residuo es ortogonal al subespacio de la forma:

$$b - Ax_m \perp K_m(A; r_0) \quad (3.13)$$

Los métodos de gradientes conjugados (CG, *Conjugate Gradients*) (Hestenes y Stiefel, 1952) y de ortogonalización completa (FOM, *Full Orthogonalization Method*) (Yousef Saad, 1981) son casos particulares de esta aproximación.

- Aproximación de residuos mínimos ($L_m = AK_m; K_m = K_m(A; r_0)$). En cada iteración se identifica la solución x_m , para la cual la norma euclídea $\|b - Ax_m\|_2$ es mínima sobre $K_m(A; r_0)$. El método más representativo de esta aproximación es método del residuo mínimo generalizado (GMRES, *Generalized Minimum Residual Method*) (Youcef Saad y Schultz, 1986; Kelley, 1995).
- Aproximación de Petrov-Galerkin ($L_m = K_m(A^T; r_0); K_m = K_m(A; r_0)$). De forma similar a la primera aproximación se trata de encontrar una solución x_m en la que el residuo $b - Ax_m$ sea ortogonal a un subespacio de dimensión m , en este caso $K_m(A^T; r_0)$:

$$b - Ax_m \perp K_m(A^T; r_0) \quad (3.14)$$

Una buena representación de este tipo de métodos es el método de gradientes biconjugados (Bi-CG, *Biconjugate Gradient Method*) (Fletcher, 1976).

- Aproximación de errores mínimos ($L_m = A^T K_m; K_m = K_m(A^T; r_0)$). En cada iteración se identifica la solución x_m en el subespacio $A^T K_m(A^T; r_0)$ para el cual la norma euclídea $\|x_m - x\|_2$ es mínima.

En la década de los 90 surgió el desarrollo de métodos híbridos para tratar de mejorar los ratios de convergencia vs. coste operacional de los métodos convencionales. Así, surgieron métodos como el método del gradiente conjugado cuadrado (CGS, *Conjugate Gradients Squared*) (Sonneveld, 1989; Kelley, 1995) o el método del gradiente biconjugado estabilizado (Bi-CGSTAB, *Bi-Conjugate Gradient Stabilized*) (Van der Vorst, 1992).

El uso de métodos iterativos en la resolución de problemas electromagnéticos utilizando la metodología SIE-MoM se encuentra ampliamente documentado en la bibliografía (Pasi Ylä-Oijala, M Taskinen y Järvenpää, 2005; Rivero, J. Taboada, Luis Landesa et al., 2010; M. Araújo, J. Taboada, Rivero et al., 2011; Ergul y Gurel, 2009).

Debido a la robustez y excelente convergencia de GMRES en las distintas formulaciones implementadas (Gomez-Sousa, Rubinos-Lopez y Martinez, 2015) y a la propia experiencia en cuanto a los resultados previos (L. Landesa, J. M. Taboada, Campon et al., 2017), se ha decidido paralelizar el producto matriz vector (MVP, *Matrix-Vector Product*) asociado a MoM en cada iteración del método GMRES sobre una tarjeta coprocesadora de la familia Intel Xeon Phi.

Para ello, se han desarrollado nuevos algoritmos que tratan de aprovechar al máximo la capacidad de cómputo de la Xeon Phi. Partiendo de un algoritmo GMRES modificado, se resolverán M problemas electromagnéticos realizando un único cálculo de los elementos de la matriz de impedancias. El detalle de los algoritmos se muestra en el capítulo 6.

Los resultados obtenidos en (Campon y Luis Landesa, 2018) demuestran que un sistema con tarjetas coprocesadoras Intel Xeon Phi es adecuado para el cálculo de problemas electromagnéticos usando un método iterativo de tipo GMRES.

Capítulo 4

Intel Many Integrated Core (MIC)

| | | |
|-------|--|----|
| 4.1 | Evolución de las familias Xeon Phi | 48 |
| 4.2 | Arquitectura hardware Knights Corner (KNC) | 49 |
| 4.2.1 | Detalle de un core KNC | 50 |
| 4.2.2 | Jerarquía de cachés | 52 |
| 4.3 | Arquitectura software Knights Corner (KNC) | 53 |
| 4.3.1 | Manycore Platform Software Stack (MPSS) | 54 |
| 4.4 | Modelos de programación para coprocesadores Intel Xeon Phi | 55 |
| 4.5 | Paralelismo en coprocesadores Intel Xeon Phi | 56 |
| 4.5.1 | Paralelismo de datos | 56 |
| 4.5.2 | Paralelismo de tareas | 58 |

Intel MIC es una arquitectura de multiprocesadores desarrollada y fabricada por Intel, que está optimizada para un alto grado de paralelismo, consiguiendo un rendimiento elevado y una alta eficiencia energética en aplicaciones masivamente paralelas.

Entre las características más relevantes de la arquitectura Intel MIC se distinguen:

- Su alta compatibilidad con aplicaciones previamente desarrolladas para entornos multinúcleo tradicionales de Intel, como Intel Xeon. Se beneficia de las mismas técnicas y herramientas empleadas en procesadores multinúcleo y permite programar utilizando lenguajes de alto nivel como OpenMP o Cilk Plus. Todo ello, implica una menor curva de aprendizaje frente a otras arquitecturas manycore como CUDA.
- Presenta un rendimiento óptimo en códigos altamente paralelizables. El rendimiento en códigos secuenciales frente a arquitecturas como Intel Xeon es muy pobre.
- Es la arquitectura de Intel con mayor eficiencia energética logrando picos de varios teraFLOPS con consumos en torno a 250-300 W.

En el presente capítulo se realizará un repaso de la arquitectura y evolución de los productos Intel MIC, poniendo el foco en la familia Knights Corner, cuyo modelo Intel Xeon Phi 7120P ha sido utilizado en el desarrollo de la Tesis Doctoral.

4.1. Evolución de las familias Xeon Phi

A lo largo de los años y desde su aparición comercial derivada del proyecto Larrabee, Intel ha desarrollado diferentes familias de productos Xeon Phi (Jackson, 2017; Furlinger, 2016):

- **Knights Ferry:** Familia de prototipado inicial derivada del fallido proyecto Larrabee (GPU de Intel) en desarrollo desde 2006. El producto final presentado en 2010 en tecnología de 45 nm consistía en una tarjeta PCIe de 32 cores a 1.2 Ghz, 2GB de memoria GDDR5, cachés L1 de 1 MB y L2 de 8 MB y consumo máximo de 300W.
- **Knights Corner (KNC):** Primera familia comercial de Intel MIC lanzada en 2012. Su desarrollo se basaba en tecnología de 22 nm en formato

de tarjeta coprocesadora PCIe 2.0 x16 o SFF 230-pin distinguiendo las series 3100, 5100 y 7100. El modelo más potente, el Intel Xeon Phi 7120P, equipado con 61 cores a 1.2Ghz, 16 GB de memoria GDDR5, cachés L1 de 32 KB para datos y 32 KB para instrucciones y L2 de 512 KB por core era capaz de obtener un rendimiento en el caso más favorable de 1.2 teraFLOPS con un consumo de 300W.

- **Knights Landing (KNL):** Segunda generación de tarjetas coprocesadoras Intel MIC lanzadas al mercado en 2016 en tecnología de 14 nm. Su principales diferencias con la generación anterior consistían por un lado en utilizar el socket LGA 3647 en detrimento del bus PCIe y en introducir un conjunto de instrucciones nuevo, el AVX-512. Su modelo más potente contenía 72 cores Silvermont (Atom), 16 GB de memoria RAM con cachés similares a la generación anterior pero con una arquitectura de memoria MCDRAM que supone un salto cualitativo bastante importante. El rendimiento máximo obtenido en el caso más favorable era de 3.4 teraFLOPS con un consumo de 260 W.
- **Knights Hill:** Se presentó como la tercera generación de tarjetas Xeon Phi, sucesora de Knights Landing, fabricada con tecnología de 10 nm. Su desarrollo se canceló en 2017 sin que saliera al mercado.
- **Knights Mill:** Variante especial de Knights Landing presentada en Diciembre de 2017 con mejoras específicas en el campo de la inteligencia artificial.

En 2013 el supercomputador Tihanne2 alcanzó el primer puesto del TOP500 equipando tarjetas coprocesadoras Intel Xeon Phi. Se mantuvo en la lista como el supercomputador más rápido hasta finales de 2015. Intel ha anunciado que la familia de productos Knights Landing dejará de producirse para centrarse en la nueva familia gráfica Intel Xe y fija la fecha para los últimos pedidos en Agosto de 2019.

4.2. Arquitectura hardware Knights Corner (KNC)

Los chips KNC, correspondientes a la familia comercial x100 de Intel, presentan las siguientes características (*Best Practice Guide Intel Xeon Phi v2.0 2017*; Vladimirov, Asai, Karpusenko et al., 2015; Jeffers y Reinders, 2013):

- Su fabricación se basa en tecnología de 22 nm con transistores Trigate 3D.
- Se presentan tres familias principales, la 3100 (57 cores a 1.1GHz y 6GB GDDR5), la 5100 (60 cores a 1.053Ghz y 8GB GDDR5) y la 7100 (61 cores a 1.238Ghz y 16GB GDDR5) con diferentes consumos de energía y anchos de banda de memoria.
- Cada tarjeta coprocesadora dispone de 8 controladores de memoria con dos canales Graphics DDR (GDDR) de 5.5 GB/s logrando anchos de banda de memoria de 352GB/s en la serie 7100.
- Los cores KNC están basados en arquitectura P54C y soportan instrucciones de 64 bits. Cada core dispone de una Vector Processing Unit (VPU) con registros SIMD de 512 bits y cachés L1 y L2. También presentan Hyperthreading con hasta 4 hilos por core.
- La interconexión de los cores, el bus PCIe 2.0 y los controladores de memoria GDDR5 se realiza a través de tres anillos bidireccionales Core Ring Interconnect (CRI) que conectan todos los componentes, tal y como se aprecia en la figura 4.2.
- También se dispone de un Distributed Tag Directory (DTD) para conseguir la coherencia caché global a todos los cores interconectando los Tags Directories (TD) de cada una de las cachés. Véase la figura 4.2.
- Cada tarjeta coprocesadora corre su propio sistema operativo (uOS) basado en Linux, que permite ejecutar código de forma nativa en la propia tarjeta. Emula la pila TCP/IP sobre el bus PCIe permitiendo la comunicación con el host dónde se aloja.

4.2.1. Detalle de un core KNC

En la figura 4.1 se observa la topología de un core KNC. Las instrucciones que se decodifican en cada core procedentes de la caché de instrucciones L1 pueden ser procesadas por una Scalar Processing Unit (SPU) o por una Vector Processing Unit (VPU) dependiendo del conjunto de instrucciones al que pertenezcan (Jeffers y Reinders, 2013).

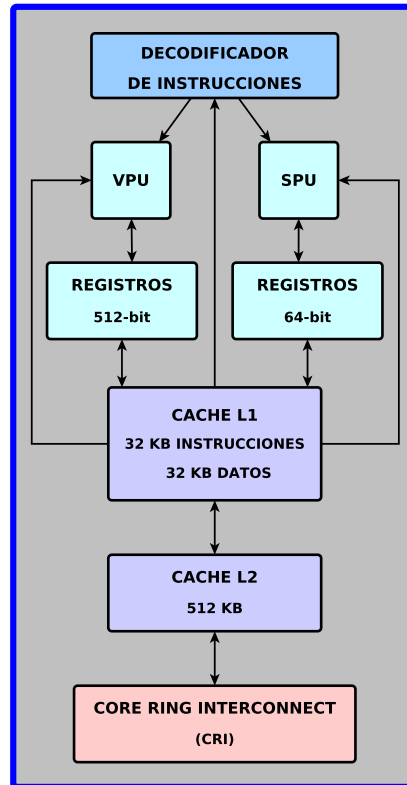


FIGURA 4.1: Topología de un core KNC

Los cores KNC no pueden procesar las extensiones SIMD MMX, SSE, SSE2, y AVX pero sí que pueden procesar las extensiones SIMD de 512 bits denominadas Intel Initial Many Core Instructions (IMCI) además de las instrucciones x86 de 64 bits.

La SPU se utiliza para mantener compatibilidad con la arquitectura x86, por ejemplo, para ejecutar el sistema Linux que corre en el coprocesador, pero no aporta rendimiento sobre los procesadores existentes, de hecho tan solo un 2% del área del chip KNC se utiliza para este propósito.

El uso de la VPU de cada core en las aplicaciones a paralelizar aporta la mejora de rendimiento teórica ideal que se comenta en las características técnicas del chip KNC y que llega a 1.2 Teraflops en la serie 7100. El conjunto de instrucciones IMCI utiliza registros de 512 bits que pueden alojar 8 elementos de doble precisión en coma flotante. Además, se introduce la instrucción Fused Multiply-Add (FMA) que permite realizar la multiplicación y la suma en el mismo ciclo de reloj.

Si se tienen en cuenta las consideraciones anteriores, el rendimiento máximo teórico en doble precisión de las tarjetas coprocesadoras KNC se calcula

(Vladimirov, Asai, Karpusenko et al., 2015):

$$\text{NumberOfCores} * \text{ClockFrequency} * \text{ElementsByRegistry} * 2(\text{FMA})\text{FLOPS}/\text{cycle}$$

En el caso del modelo que se ha utilizado en el desarrollo de la presente Tesis, Intel Xeon Phi 7120P, el rendimiento máximo teórico en doble precisión en el caso más favorable sería:

$$61 * 1,238\text{Ghz} * 8 * 2(\text{FMA}) = 1,208\text{GFLOPS}/\text{cycle}$$

4.2.2. Jerarquía de cachés

Cada core de la arquitectura KNC posee una jerarquía de memoria formada por una cache de instrucciones L1 de 32 KB, una caché de datos L1 de 32KB y una caché conjunta L2 para datos e instrucciones de 512KB tal y como se observa en la figura 4.1.

La correspondencia entre un bloque de memoria principal y uno de caché se realiza de manera asociativa por conjuntos de 8 vías y se mantiene la coherencia entre las distintas cachés locales a través del protocolo MESI con una política de reemplazo caché Least Recently Used (LRU) (Jeffers y Reinders, 2013).

Dado que la caché L2 es propia de cada core y es necesario mantener una coherencia a nivel global, se transfiere en paralelo la información del estado de cada bloque de caché entre los distintos TD de cada core. Cuando se produce un fallo de caché se genera una petición de dirección en el anillo CRI de direcciones y se revisa el DTD. Si el dato existe en alguna caché L2 de otro core se obtiene la dirección, el estado y la ubicación de la caché y se utiliza el anillo CRI de coherencia (Véase la figura 4.2). Si no existe, se realiza la petición a memoria principal y se actualizan los TD. Como característica adicional, el nivel L2 de caché soporta en su hardware la prebúsqueda de instrucciones y datos antes de que sean demandados por la caché.

La caché L2 distribuida entre los cores mediante el DTD hace que la cantidad de caché L2 total del chip KNC varíe entre los 512KB y 30,5MB (61 cores) dependiendo del código y datos que son compartidos entre los distintos cores.

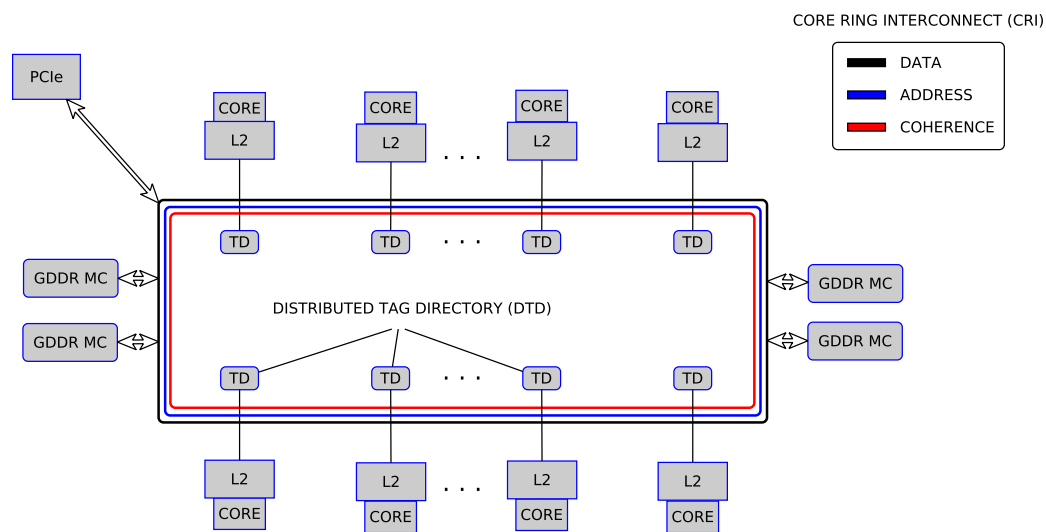


FIGURA 4.2: Arquitectura Knights Corner

4.3. Arquitectura software Knights Corner (KNC)

Las tarjetas coprocesadoras KNC se integran en hosts que corren sistemas operativos Windows o Linux y que dispongan de interfaces PCIe 2.0 x16. Esa cuestión se aborda con más detalle en el apartado 4.3.1.

A nivel de software toda la comunicación entre el host y la tarjeta coprocesadora se encuentra soportada por el driver Symmetric Communication Interface (SCIF). SCIF abstrae los detalles de la implementación hardware del bus PCIe proporcionando una Application Program Interface (API) cuyo objetivo principal es lograr el máximo rendimiento en la comunicación entre distintos nodos de computación, ya sean entre el host y la Xeon Phi o entre dos tarjetas coprocesadoras conectadas a distintos buses PCIe. La arquitectura de SCIF es independiente del sistema operativo y es la base de APIs más avanzadas como las que soportan Message Passing Interface (MPI), OpenCL y la pila TCP/IP.

Cuando se produce el arranque del sistema operativo anfitrión en el host, el firmware propio de cada tarjeta coprocesadora comienza el proceso de arranque local (bootstrap) en el que se inicializa el hardware, se produce la comunicación internodo mediante el driver SCIF y se arranca el sistema operativo de la tarjeta coprocesadora sobre el que descansan el resto de elementos de la pila de software. Este proceso se realiza en el primer arranque y en la recuperación tras un fallo catastrófico.

El sistema operativo de la Xeon Phi está basado en el código fuente del kernel de linux estándar con muy pocas modificaciones para adaptar aspectos muy concretos. El acceso al hardware propio de la Xeon Phi se realiza a través de un driver desarrollado para el propio sistema operativo. Además, se pueden extender algunas funcionalidades del kernel mediante la carga de módulos con modprobe.

4.3.1. Manycore Platform Software Stack (MPSS)

Para tener un nodo de computación operativo necesitamos un sistema anfitrión (CPU host) con una interfaz PCIe 2.0 x16 al que se conectará la tarjeta coprocesadora (Xeon Phi). Además, es necesario que la BIOS del host en cuestión tenga soporte para más de 4GB de entrada-salida mapeada en memoria (MMIO) y que cumpla con los requisitos de energía y refrigeración necesarios para alojar la Xeon Phi.

Una vez se han cumplido los requisitos hardware necesarios para tener un nodo de computación operativo, es necesario desplegar en el host la plataforma software necesaria para hacerlo funcionar. Este conjunto de software recibe el nombre de Intel Manycore Platform Software Stack (MPSS) y se ha certificado su uso en sistemas operativos linux de 64 bits con kernels posteriores a 2.6.34, en concreto las distribuciones Red Hat Enterprise Linux y Suse Linux Enterprise y en algunos sistemas operativos Microsoft de 64 bits. Véase (L. Q. Nguyen, 2014).

MPSS contiene los drivers SCIF, de gestión de energía, etc. y las librerías y herramientas necesarias para la gestión y diagnóstico de coprocesadores Intel Xeon Phi. Entre ellas y a destacar:

- *micctrl* se utiliza principalmente para aplicar y restablecer configuraciones sobre la Xeon Phi.
- *miccheck* realiza un conjunto de pruebas para verificar que tanto el funcionamiento como la configuración son correctos.
- *micinfo* se utiliza para obtener información del coprocesador y de las versiones de sistema operativo y drivers instalados.
- *micsmc* muestra información sobre distintos parámetros físicos de los coprocesadores: estadísticas del uso de cada core, memoria utilizada, temperatura, consumo de energía, etc.

- *micflash* es un comando que se utiliza para obtener información, leer y escribir sobre la memoria flash.
- *micrasd* es un demonio que se ejecuta en el host anfitrión y gestiona los logs procedentes de la tarjeta coprocesadora.
- *micnativeloadex* se utiliza para transferir aplicaciones y sus dependencias a la Xeon Phi y lanzar su ejecución desde el host anfitrión.

4.4. Modelos de programación para coprocesadores Intel Xeon Phi

La arquitectura estudiada en los apartados anteriores permite que tanto los datos como los ejecutables puedan ser intercambiados fácilmente entre el host y la Xeon Phi. Dependiendo de las necesidades de la aplicación, la ejecución puede comenzar en el host y pasar al coprocesador o comenzar en el coprocesador y pasar al host.

En la figura 4.3 aparece una visión conceptual de como el código de una aplicación puede lanzarse y ejecutarse de forma indistinta en el host y la Xeon Phi dando lugar a los siguientes modelos de programación:

- *Modelo nativo, ya sea en la CPU o en la Phi.* En este modelo la ejecución del programa se realiza íntegramente en el procesador o en la tarjeta coprocesadora, ignorando por completo al otro miembro del nodo de computación.
- *Modelo Offload.* Se conoce así el modelo en el cual el programa se lanza y comienza a ejecutarse en el procesador y llegado el momento, determinadas porciones del código altamente paralelizables pasan a ejecutarse en la tarjeta coprocesadora.
- *Modelo simétrico.* La aplicación se lanza y se ejecuta en ambos componentes del nodo de computación: procesador y tarjeta coprocesadora. La comunicación entre ambos se realiza normalmente a través de Message Passing Interface (MPI).

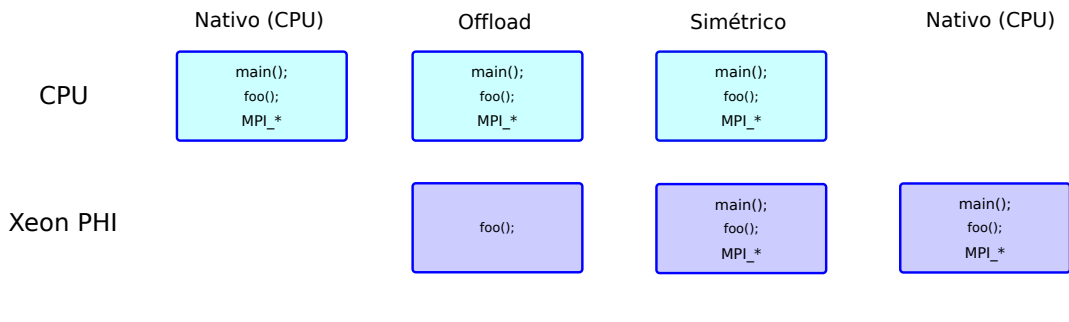


FIGURA 4.3: Modelos de programación sobre tarjetas coprocesadoras Intel Xeon Phi.

4.5. Paralelismo en coprocesadores Intel Xeon Phi

Como ha podido observarse al desgranar la arquitectura hardware y software KNC, las tarjetas coprocesadoras han sido diseñadas teniendo en cuenta su uso en entornos con un alto grado de paralelismo.

Las instrucciones IMCI con registros de 512 bits permiten que mediante la paralelización de datos, conocida como vectorización, se consigan realizar 8 operaciones matemáticas con datos en doble precisión en un ciclo de reloj. Además, gracias a la instrucción FMA, ese rendimiento puede llegar a doblarse si hay que realizar operaciones de suma y multiplicación sobre los mismos datos.

La arquitectura hardware de la Xeon Phi, permite que multitud de tareas puedan ser ejecutadas de forma simultánea por los distintos cores. Existen diferentes alternativas para conseguirlo como pueden ser el uso de directivas OpenMP, algoritmos basados en Intel Threading Building Blocks (TBB) o bucles Intel Cilk Plus.

Además, si la tarjeta coprocesadora forma parte de un nodo de computación, es posible el uso del paralelismo de memoria distribuida comunicando dicha tarjeta con el host anfitrión o con otras tarjetas coprocesadoras mediante paso de mensajes MPI.

4.5.1. Paralelismo de datos

El conjunto de instrucciones IMCI disponible en la arquitectura KNC permite disponer de 8 elementos de doble precisión en cada registro consiguiendo que en cada ciclo de reloj la VPU sea capaz de realizar operaciones con todos los elementos.

La vectorización complementa el paralelismo tradicional a nivel de tareas ya que a su vez cada una de las tareas paralelizadas pueden aprovechar la vectorización trabajando en paralelo con los diferentes conjuntos de datos contenidos en los registros.

Tal y como aparece en (Jeffers y Reinders, 2013), existen diferentes aproximaciones a la hora de abordar la vectorización:

- *Intel MKL* ([Intel Math Kernel Library Developer Reference 2019](#)). Librería matemática optimizada para proporcionar un alto rendimiento computacional en sistemas multiprocesador. Contiene funciones para álgebra lineal básica (operaciones vectoriales y matriciales), resolución de sistemas de ecuaciones lineales, transformadas de Fourier, estadística ... Si se utilizan las funciones que aparecen en la librería, la vectorización se realiza de forma automática sin la intervención del programador.
- *Autovectorización*. El compilador se ocupa de la vectorización. Funciona con bucles sencillos. Para activarlo es necesario habilitar el nivel de optimización con la directiva de compilación `-On`. El nivel de optimización por defecto es `-O2` que además de la autovectorización de algunos bucles, elimina código no necesario y realiza algunas otras optimizaciones. El nivel `-O3` realiza una autovectorización más agresiva fusionando bucles. Es recomendable probar ambos niveles de optimización porque aunque pudiera parecer que `-O3` va a proporcionar un mejor rendimiento que `-O2` en ocasiones no es así.
- *Directivas SIMD*. El uso de este tipo de directivas a partir de OpenMP 4.0 permiten al programador forzar la vectorización sobre un determinado bucle. Como se ha comentado, la autovectorización sólo funciona con bucles muy sencillos en los que no pueden existir los condicionales, debe tener un número de iteraciones fijo y conocido, no pueden existir dependencias en código entre distintas iteraciones, etc. Las directivas SIMD permiten al programador saltarse todas estas restricciones, pero requieren de un profundo conocimiento de la vectorización para evitar resultados erróneos.
- *Intel Cilk Plus*. Extensión de los lenguajes C y C++ que incorpora una notación muy simple para permitir el paralelismo a nivel de datos de los vectores de elementos que se definen en el código. La página web

oficial del producto lo ha discontinuado en 2018 y recomienda utilizar otros paradigmas como OpenMP. Véase [Intel Cilk Plus](#).

4.5.2. Paralelismo de tareas

Conseguir que distintas tareas se ejecuten de forma simultánea requiere que los distintos procesadores compartan el mismo espacio de direcciones en un sistema de memoria virtual único. La aplicación a paralelizar se segmenta en distintas partes que se ejecutan en ámbitos diferentes conocidos como hilos.

El programador a través de los distintos frameworks existentes tiene la responsabilidad de controlar el acceso de los distintos hilos a los datos comunes que residen en la memoria compartida. Existen diferentes frameworks que pueden utilizarse en la Intel Xeon Phi para conseguir el paralelismo a nivel de tareas:

- *OpenMP*. Se considera el estándar de facto a la hora de conseguir el paralelismo de tareas en memoria compartida. La especificación de la API de OpenMP (*OpenMP Application Program Interface 2013*) contiene un conjunto de directivas de compilación, librerías y variables de entorno para el paralelismo en memoria compartida en lenguajes C, C++ y Fortran. En términos generales, la ejecución del programa comienza con un único hilo que va ejecutando las instrucciones de manera secuencial hasta que se encuentra con una directiva reservada del tipo constructor (`#pragma omp parallel, for, task, sections...`). En ese momento se crea una región paralela con un grupo de hilos esclavos transformándose el hilo inicial en maestro de dicho grupo. Una vez la ejecución paralela finaliza, se sincronizan todos los hilos y se devuelve la ejecución secuencial al hilo maestro. Existen dos variables de entorno esenciales a la hora de controlar la ejecución en un entorno multiprocesador:
 - `OMP_NUM_THREADS`. Establece el número máximo de hilos que se pueden ejecutar en una región paralela.
 - `KMP_AFFINITY`. Determina el patrón de distribución de hilos entre los distintos cores. Si el valor que toma es `compact` los hilos se distribuirán llenando los cores empezando por el primero, si se elige `scatter` se irá ubicando un hilo en cada core hasta que se alcance al número total de cores, y después se continuará de nuevo en el

primer core. Por último, `balanced` es similar a `scatter` pero asegurándose que los distintos hilos que están en el mismo procesador tienen numeraciones de OpenMP próximas.

- *Intel TBB*. Es una librería para C++ que ofrece un amplio soporte para programación paralela en memoria compartida. Para poder utilizarla es necesario importar el espacio de nombres `tbb`. Entre las distintas funciones que se ofrecen se puede destacar la función `tbb::parallel_for()` que reemplaza un bucle secuencial por un bucle que se ejecuta en paralelo en los distintos procesadores. Dado que no hace uso de directivas de compilación el código desarrollado se porta con relativa facilidad. Además de la versión oficial de Intel existe una implementación en software libre con un gran soporte por parte de la comunidad. Véase [GitHub-intel/tbb](https://github.com/intel/tbb).
- *Intel Cilk Plus*. Es una extensión de los lenguajes C y C++ que ofrece soporte de paralelismo a nivel de datos y a nivel de tareas de una forma bastante sencilla. Basaba parte de su funcionalidad en la ofrecida por Intel TBB y se presentaba como una alternativa a OpenMP, estándar que finalmente se ha impuesto.

Capítulo 5

Electromagnetismo computacional en arquitecturas manycore

| | | |
|-------|------------------------------------|----|
| 5.1 | Resultados con GPUs | 63 |
| 5.1.1 | FDTD | 63 |
| 5.1.2 | FEM | 64 |
| 5.1.3 | MoM | 66 |
| 5.2 | Resultados con Intel MIC | 69 |
| 5.2.1 | FDTD | 69 |
| 5.2.2 | FEM | 70 |
| 5.2.3 | MoM | 71 |

La resolución de grandes problemas electromagnéticos en supercomputadoras ha sido tema de interés en las últimas décadas. El uso de la metodología SIE-MoM (R. F. Harrington, 1993) ha demostrado ser más eficiente que las técnicas volumétricas (Hao, Nehl, Hafner et al., 2007; J. Jin, 2014). A pesar de ello, la necesidad de reducir la complejidad computacional tanto a nivel de operaciones como de consumo de memoria condujo al desarrollo de técnicas como FMM (Coifman, Rokhlin y Wandzura, 1993) o MLFMA (J. M. Song y Chew, 1995; J. Song, Lu y Chew, 1997).

En la década anterior, diversos grupos de investigación, se propusieron resolver el mayor problema electromagnético hasta la fecha. Destacaron dos grupos, BiLCEM en Turquía, explotando el método MLFMA (Ö. Ergül, 2014) y HEMCUVE, grupo de electromagnetismo computacional de las Universidades de Vigo y Extremadura, aplicando la FFT a determinadas etapas de las técnicas FMM o MLFMA (J. Taboada, Luis Landesa, Obelleiro et al., 2010; Bertolo, M. G. Araújo, J. M. Taboada et al., 2011; J. Taboada, M. Araújo, Bertolo et al., 2010). En Agosto de 2010 el equipo de científicos de HEMCUVE consiguió resolver el mayor problema electromagnético de la historia con más de un billón de incógnitas (J. M. Taboada, L. Landesa, M. G. Araújo et al., 2011).

En la última década, se ha ido normalizando la presencia de GPUs y tarjetas coprocesadoras Intel Xeon Phi en las listas de los supercomputadores con mayor rendimiento. En 2013 el supercomputador Tihanne2 alcanzó el primer puesto del TOP500 equipando tarjetas coprocesadoras Intel Xeon Phi. Se mantuvo en la lista como el supercomputador más rápido hasta finales de 2015. En Noviembre de 2019, 6 de los 10 supercomputadores con mayor rendimiento utilizaban GPUs o coprocesadores Xeon Phi. Intel ha anunciado que abandonará la fabricación de la familia de productos Knights Landing en favor de las GPUs Intel Xe, lo que supone que la tendencia de la industria en cuanto a coprocesadores se refiere, se ha decantado por el uso de GPUs.

En los siguientes apartados se realizará un breve compendio de los distintos resultados obtenidos hasta la fecha en el empleo de tarjetas coprocesadoras en electromagnetismo computacional.

5.1. Resultados con GPUs

El uso de GPUs en el ámbito del electromagnetismo computacional ha despertado el interés de diversos grupos de investigación. Los primeros métodos desarrollados para su ejecución en GPUs fueron los basados en técnicas volumétricas por su mayor sencillez a la hora de ser implementados, como el FDTD (Krakiwsky, L. Turner y Okoniewski, 2004; Adams, Payne y Boppana, 2007) o el FEM (K. Liu, X. Wang, Y. Zhang et al., 2006). Más tarde, utilizando la técnica MoM (S. Peng y Z. Nie, 2008), se lograron aceleraciones de 20x en problemas de dispersión electromagnética. Estos primeros desarrollos utilizaron librerías gráficas como OpenGL o frameworks para habilitar el uso de la GPU como procesadores de propósito general (Buck, Foley, Horn et al., 2004).

La presentación en Noviembre de 2006 de la NVIDIA Geforce 8800 GTX, la primera GPU construida con la arquitectura unificada de dispositivos de cómputo (CUDA, *Compute Unified Device Architecture*) (Corporation, 2006), supuso un antes y un después en supercomputación debido a su alta eficiencia en cuanto a costes-rendimiento y a la posibilidad de utilizar una variante del lenguaje de programación C para desarrollar código de propósito general ejecutable en la GPU.

En los siguientes apartados se realizará un recorrido por distintas publicaciones que hacen uso de la potencia de cómputo de las GPUs en la resolución de problemas electromagnéticos.

5.1.1. FDTD

La técnica FDTD fue propuesta por Kane Yee en 1966 para la resolución de las ecuaciones de Maxwell en forma diferencial (Kane Yee, 1966). Se realiza una discretización del espacio mediante un mallado de dimensiones aproximadas de una fracción de la longitud de onda y se transforman las ecuaciones de Maxwell en un sistema de ecuaciones en diferencias finitas. Al escoger de forma correcta los puntos en los que se evalúan las componentes de los campos en estas ecuaciones, se satisfacen las condiciones de contorno para conductores ideales.

Uno de los primeros ejemplos del uso de GPUs en simulaciones electromagnéticas se encuentra recogido en (Krakiwsky, L. Turner y Okoniewski, 2004).

En él, se realiza la resolución de un problema de dispersión electromagnética en 2D a partir del método FDTD mediante la API OpenGL logrando aceleraciones de 10x. Otros trabajos que han obtenido muy buenos resultados mediante el método FDTD y algunos frameworks desarrollados sobre la API OpenGL aparecen referenciados en la bibliografía (Inman y Elsherbeni, 2006; Adams, Payne y Boppana, 2007).

Debido a su eficiencia y a la simplicidad de su formulación matemática, existen multitud de referencias bibliográficas respecto a aceleración mediante GPUs de la técnica FDTD utilizando CUDA. En (De Donno, A. Esposito, Tarricone et al., 2010) se realiza un estudio en 2D de un problema de interacción humana con antenas de estaciones base de radio consiguiendo aceleraciones pico de 42x. En (T. Nagaoka y S. Watanabe, 2010) se obtuvieron tasas de aceleración de 25x al analizar la interacción electromagnética en 3D sobre un cubo truncando el dominio mediante capas perfectamente acopladas (PML, *Perfectly Matched Layer*). En (Livesey, Stack, Costen et al., 2012) se realiza un análisis de las tres aproximaciones existentes para la aceleración de la FDTD tridimensional sobre GPUs analizando pros y contras de cada una de ellas.

La ejecución de la técnica FDTD en entornos heterogéneos distribuidos ha conseguido obtener ratios de aceleración en torno a 20x como se observa en (R. Jiang, S. Jiang, Y. Zhang et al., 2014; he, Tang, Xie et al., 2015), debido principalmente a la merma en rendimiento que supone la sobrecarga en la comunicación entre los nodos por el paso de mensajes MPI. Si bien es cierto, que recientemente en (Warren, Giannopoulos, Gray et al., 2018), se han obtenido tasas de aceleración de 30x en condiciones concretas y con desarrollos a medida para modelado de georradars mediante FDTD con el software de simulación electromagnética gprMax.

5.1.2. FEM

FEM es un método numérico utilizado para encontrar soluciones aproximadas de ecuaciones diferenciales parciales con condiciones de contorno. Es muy utilizado en el análisis de fenómenos físicos como pueden ser la mecánica de fluidos, problemas de transmisión de calor o campos electromagnéticos entre otros.

Los primeros trabajos en el desarrollo matemático del método, surgieron por la necesidad de resolver problemas de elasticidad y de análisis de estructuras en la década de los 40 en los campos de la ingeniería civil y la aeronáutica (Hrennikoff, 1941; Courant, 1943). En décadas posteriores la mayoría de trabajos propuestos sobre el método FEM se centraron en estas ramas de la ingeniería (Levy, 1953; M. J. Turner, R. W. Clough, Martin et al., 1956; R. Clough, 1960).

La aplicación de FEM en la resolución de problemas electromagnéticos surgió a finales de la década de los 60 para el análisis de los modos TE y TM en guías de onda metálicas (Ahmed y Daly, 1969; Silvester, 1969). Hoy en día, es un método muy utilizado en geometrías complicadas por la facilidad del modelado.

Uno de los primeros intentos de acelerar el método FEM en GPUs aparece en (K. Liu, X. Wang, Y. Zhang et al., 2006). Se consiguieron aceleraciones básicas, en torno a 2x, en la resolución de un problema de radiación electromagnética en 2D mediante la API OpenGL. En (Cecka, Lew y Darve, 2011), se describen las diferentes aproximaciones y estrategias utilizadas para el uso eficiente de la memoria cuando se aplica el método FEM en GPUs con la API CUDA.

Se han presentado numerosos trabajos centrados en la implementación de métodos iterativos basados en subespacios de Krylov para la resolución de sistemas de ecuaciones matriciales dispersos en GPUs, como los que se generan en la técnica FEM. Entre ellos a destacar, el método de los gradientes conjugados (Cevahir, Nukada y Matsuoka, 2009; Georgescu y Okuda, 2010), gradientes biconjugados (De Donno, A. Esposito, Monti et al., 2010; Ortega Lopez, Vázquez, Garzon et al., 2013) o el método del residuo mínimo generalizado (Bahí, Couturier y Khodja, 2011; Couturier y Domas, 2012). En esta misma línea se han desarrollado numerosos preconditionadores sobre GPUs para favorecer la convergencia de los métodos citados tal y como se recoge en (R. Li y Yousef Saad, 2013; Anzt, Gates, Dongarra et al., 2017).

Algunos trabajos específicos sobre la aceleración de FEM en GPUs aplicados al análisis electromagnético, se encuentran referenciados en la bibliografía. En (Meng, B. Nie, Wong et al., 2014) se presentan los principales problemas que ralentizan la aceleración GPU del método FEM en el análisis electromagnético y se proponen distintos enfoques para solventarlos. Se consiguen aceleraciones en torno a 20x en el análisis de distintos arrays de antenas. En (Fu, Lewis, Kirby et al., 2014) se realiza un análisis concreto de la ecuación de

Helmholtz con condiciones de frontera de Neumann logrando aceleraciones en torno a 50x en un problema acústico.

5.1.3. MoM

El método de los momentos es una técnica utilizada para transformar ecuaciones integrales en sistemas de ecuaciones lineales que pueden resolverse de forma numérica (R. F. Harrington, 1993).

La resolución de sistemas de ecuaciones lineales es parte fundamental en cualquier campo de la ingeniería. La discretización de la formulación integral del problema planteado en el capítulo 2 conduce a un sistema de ecuaciones lineales en notación matricial del tipo $Ax = b$ que debe ser resuelto.

Uno de los primeros ejemplos del uso de GPUs en la aceleración de MoM aplicado a un problema de dispersión electromagnética aparece en (S. Peng y Z. Nie, 2008). Se consiguieron tasas de aceleración de 20x con el hardware del que se disponía en su momento y poco antes de su publicación debido al uso de nuevas GPUs los autores afirman que lograron tasas en torno a 100x.

Una de las primeras referencias del uso de CUDA en la aceleración de MoM se utilizó para acelerar la ecuación integral de Helmholtz en 3D en el ámbito de la acústica (Takahashi y Hamada, 2009). Los autores lograron tasas de aceleración de 6-23x al resolver un problema de dispersión acústica utilizando GMRES frente a los cuatro cores de una CPU Xeon.

Tradicionalmente, se han proporcionado herramientas software para la resolución de sistemas de ecuaciones en clústeres de CPUs sobre sistemas de memoria compartida como pueden ser LAPACK (*LAPACK-Linear Algebra Package* 2013), MKL (*Intel Math Kernel Library Developer Reference* 2019) o ACML (*ACML-AMD Core Math Library* 2013). Con la normalización del uso de GPUs en problemas complejos en múltiples campos de la ingeniería, se han desarrollado librerías específicas para la resolución de sistemas de ecuaciones lineales mediante métodos directos como la factorización LU, QR o de Cholesky. Entre estas librerías destaca MAGMA de la Universidad de Tennessee (*MAGMA-Matrix Algebra on GPU and Multicore Architectures* 2017).

La mayoría de trabajos recogidos en la bibliografía relativos a la aceleración en GPUs de la técnica de MoM han utilizado la factorización LU para resolver la ecuación matricial resultante.

En (Lezar y Davidson, 2010a; Lezar y Davidson, 2010b; Lezar y Davidson, 2010c), los autores aceleraron en GPU la fase de creación de la matriz de impedancias de MoM, y una vez construida dicha matriz, analizaron diferentes aproximaciones para resolverla mediante factorización LU. Los resultados obtenidos fueron aplicados en el análisis electromagnético de la dispersión producida por una placa PEC consiguiendo tasas de aceleración en torno a 45x.

En 2014 se realizó una comparativa de la aceleración del método MoM en diferentes arquitecturas de GPUs: GT200, Fermi y Kepler (Noga, Topa, Danisz et al., 2014). Los autores desarrollaron códigos optimizados logrando acelerar las dos fases principales del método MoM, el cómputo de la matriz de impedancias Z y su posterior resolución mediante factorización LU. Se analizó un problema de radiación de un monopolo en 3D y se consiguieron aceleraciones de 15x respecto de la ejecución monohilo en una CPU. La arquitectura GPU que mejor se comportó fue la Fermi. En (Mrdakovic, Kostic, Olcan et al., 2017) los autores analizaron el efecto de acelerar la factorización LU sobre un clúster de GPUs llegando a la conclusión de que la eficiencia en paralelización se mantiene al incrementar el número de GPUs y el tamaño del problema.

Se han diseñado métodos para hacer frente a grandes problemas electromagnéticos eliminando las restricciones impuestas por la memoria RAM. En (Y. Chen, Lin, Y. Zhang et al., 2015) se aplica una técnica de volcado a disco para evitar el límite impuesto por la memoria RAM. Se han conseguido aceleraciones de 15x en el cálculo de la RCS bidimensional del modelado de un avión, respecto de la ejecución monohilo utilizando factorización LU.

Una de las técnicas avanzadas que se utilizan para reducir la complejidad de MoM es FMM (Coifman, Rokhlin y Wandzura, 1993). En esta técnica se realiza un muestreo de la geometría del problema y una posterior clasificación de las interacciones entre funciones base en dos grupos, interacciones cercanas y lejanas. Se realiza un cálculo equivalente al cómputo de las interacciones lejanas con un consumo de RAM y tiempo de CPU muy eficiente logrando reducir la complejidad total a $O(N^{3/2})$. Los primeros autores que implementaron la técnica FMM en GPUs fueron (Gumerov y Duraiswami, 2008), logrando aceleraciones de entre 30-60x en la resolución de la ecuación de Laplace en 3D. Existen diversos trabajos en cuanto a la aceleración de problemas electromagnéticos mediante la técnica FMM en clústeres de GPUs. De entre ellos, destacan: (Q. M. Nguyen, V. Dang, O. Kilic et al., 2013; V. Dang,

Q. Nguyen y O. Kilic, 2013; V. Dang, Tran y O. Kilic, 2016). En los anteriores artículos referenciados se llega a la conclusión de que la resolución de problemas de dispersión electromagnética en clústeres de GPUs tiene una buena escalabilidad a medida que se va aumentando el tamaño del problema. Se consiguen ratios de más de 70x y en (V. Dang, Tran y O. Kilic, 2016) se logra analizar un problema de más de 10 millones de incógnitas sobre un clúster de 13 nodos híbridos heterogéneos CPU-GPU equipados con GPUs Nvidia Tesla.

Otro de los algoritmos más utilizados para reducir la complejidad computacional de MoM es la versión multinivel de FMM, conocido como MLFMA (J. M. Song y Chew, 1995; J. Song, Lu y Chew, 1997). En (Xu, Ding, Fan et al., 2010) se utiliza una GPU como coprocesador para acelerar el cálculo del producto matriz-vector en la solución iterativa de un problema de dispersión electromagnética utilizando la técnica MLFMA. Se consiguieron aceleraciones de 8x respecto de una CPU con dos núcleos Intel Core 2 Duo. En (Guan, Yan y J. Jin, 2013) se calculó la dispersión electromagnética de diversos objetos en 3D sobre una GPU obteniendo tasas de aceleración de hasta 20x con respecto de la ejecución del algoritmo MLFMA en una CPU con 8 hilos de ejecución. Otros autores como N. Tran y O. Kilic, citados anteriormente por sus trabajos sobre la implementación de FMM sobre clústeres de GPUs en problemas de dispersión electromagnética han realizado avances similares sobre la técnica MLFMA. En (Tran y O. Kilic, 2016a; Tran y O. Kilic, 2016b) se analiza un problema de dispersión electromagnética sobre un clúster de GPUs de 12 nodos manteniendo la escalabilidad y consiguiendo tasas de aceleración de 37x sobre una CPU. Recientemente en (Hidayetoglu, Pearson, El Hajj et al., 2018), se ha resuelto un problema de dispersión inversa con 4 millones de incógnitas sobre un clúster de 4096 GPUs para la reconstrucción de imágenes tomográficas.

La aplicación de la técnica FFT a diversas etapas de FMM y MLFMA ha demostrado la capacidad de afrontar grandes problemas electromagnéticos en clústeres de CPUs (J. Taboada, Luis Landesa, Obelleiro et al., 2010; Bértolo, M. G. Araújo, J. M. Taboada et al., 2011; J. Taboada, M. Araújo, Bertolo et al., 2010; J. M. Taboada, L. Landesa, M. G. Araújo et al., 2011). No existen muchas referencias en la bibliografía respecto a la aplicación de la FFT en las técnicas descritas anteriormente en GPUs. En (V. Dang, Q. M. Nguyen y O. Kilic, 2014) se realiza una implementación del método FMM-FFT logrando aceleraciones superiores a 900x en un clúster de GPUs de 13 nodos respecto de la

ejecución monohilo sobre una CPU. Otro ejemplo de aplicación de la técnica FFT en FMM sobre una GPU aparece referenciado en (Cecka, 2017).

5.2. Resultados con Intel MIC

La aparición comercial de la arquitectura de multiprocesadores Intel MIC se produjo en 2012 con la familia KNC. Dado que es un producto relativamente novedoso no existen muchas referencias en la bibliografía del uso de tarjetas coprocesadoras Intel Xeon Phi en la aceleración de métodos de electromagnetismo computacional.

En los siguientes apartados se realiza un recorrido por algunos trabajos que han hecho uso de tarjetas Xeon Phi en la resolución de problemas electromagnéticos.

5.2.1. FDTD

El método de las diferencias finitas en el dominio del tiempo está basado en discretizaciones volumétricas y es una de las técnicas más utilizadas en electromagnetismo computacional.

Algunos trabajos en los que se hace uso de la arquitectura Intel MIC en la aceleración de problemas electromagnéticos mediante FDTD aparecen en (T. Nagaoka y S. Watanabe, 2013; Nakashima, 2015; Surmin, Bastrakov, Efimenko et al., 2016; Imai, Suzuki y Okubo, 2016).

En (T. Nagaoka y S. Watanabe, 2013) se presenta el uso de tarjetas coprocesadoras Intel Xeon Phi para acelerar la técnica FDTD en el análisis de la simulación bioelectromagnética sobre modelos humanos. Se lograron tasas de aceleración de 17x respecto de la ejecución monohilo de una CPU Intel Xeon E5-2680.

Otro ejemplo de uso de la arquitectura Intel MIC se muestra en el ámbito de la simulación de plasma mediante el método PIC (PIC, *Particle-In-Cell*). Un paso fundamental en la aplicación del método PIC consiste en la resolución de las ecuaciones de Maxwell en una malla del espacio mediante la técnica FDTD. Se han conseguido rendimientos en torno a 10x con respecto a ejecuciones monohilo sobre CPUs al utilizar toda la capacidad de cómputo de la

tarjeta coprocesadora (Nakashima, 2015; Surmin, Bastrakov, Efimenko et al., 2016).

En (Imai, Suzuki y Okubo, 2016) se realiza un estudio comparativo del rendimiento en paralelización de la técnica FDTD sobre CPUs, GPUs y Xeon Phi truncando el dominio mediante capas perfectamente acopladas (PML, *Perfectly Matched Layer*). Los resultados obtenidos indican que la paralelización en GPUs de la técnica FDTD dobla en rendimiento a la paralelización realizada sobre la arquitectura Intel MIC.

5.2.2. FEM

FEM es un método numérico que permite resolver ecuaciones diferenciales asociadas a un problema físico sobre geometrías complejas. Es ampliamente utilizado en el análisis de fenómenos físicos y en el ámbito de la ingeniería, ya sea en sistemas mecánicos, transferencia de calor y energía, campos electromagnéticos, ingeniería aeroespacial, robótica, biomedicina, etc.

No existen muchas referencias acerca de la aceleración del método FEM en tarjetas coprocesadoras Intel Xeon Phi. De hecho, no se han encontrado referencias respecto a la aceleración de problemas electromagnéticos.

En (Kruzel y Banas, 2014; Banas y Kruzel, 2014a; Banas y Kruzel, 2014b), los autores implementaron un algoritmo de integración numérica para acelerar una de las etapas de FEM mediante un kernel OpenCL parametrizado, que aporta la portabilidad entre diferentes arquitecturas. Se ejecutó el código en Intel MIC KNL y en GPUs Nvidia de las familias Kepler y Fermi demostrando la portabilidad del mismo y se comparó el rendimiento obtenido entre ambas arquitecturas.

Se han realizado adaptaciones del software de código abierto ParaFEM a los procesadores Intel Xeon Phi (Margetts, Mena, Hewitt et al., 2016). Se analizó un problema de ingeniería de materiales mediante la realización de pruebas de estrés a materiales elásticos logrando un buen escalado al utilizar todos los cores disponibles en la Xeon Phi. Otras referencias a problemas relacionados con ingeniería de materiales aparecen en (Tak y Park, 2017; Cai, G. Li y W. Liu, 2018). En (Cai, G. Li y W. Liu, 2018) se analizó una simulación en la que grandes rocas impactan sobre la estructura de un autobús. Al utilizar toda la potencia de cómputo disponible en la Xeon Phi, se consiguen aceleraciones de 17x respecto de la ejecución monohilo en CPU.

Otro ejemplo de uso del método FEM sobre la arquitectura Intel MIC aparece en (Weinberg, 2014) aplicado a la simulación de terremotos y ondas sísmicas.

5.2.3. MoM

El método de los momentos es una técnica numérica ampliamente utilizada en la resolución de sistemas de ecuaciones lineales. Una de las etapas de MoM con mayor consumo de tiempo de CPU y de memoria es la resolución de la ecuación matricial que se genera.

Como ya se explicó en el capítulo 3, en los métodos directos es necesario realizar una etapa de factorización previa a la etapa de resolución del sistema de ecuaciones lineales. La aparición de GPUs y tarjetas coprocesadoras con una gran capacidad de cómputo ha supuesto una oportunidad para abrir nuevos campos de investigación.

MAGMA (*MAGMA-Matrix Algebra on GPU and Multicore Architectures* 2017) es uno de los principales proyectos de referencia en la creación de librerías específicas para la resolución de sistemas de ecuaciones lineales en arquitecturas heterogéneas e híbridas. La librería MAGMA ha sido diseñada y elaborada por el equipo que desarrolló LAPACK y ofrece soporte para las factorizaciones LU, QR o de Cholesky.

La mayoría de trabajos recogidos en la bibliografía sobre la aceleración de MoM en tarjetas coprocesadoras Xeon Phi han hecho uso de las herramientas proporcionadas por MAGMA (Dongarra, Gates, A. Haidar et al., 2014; Dongarra, Gates, A. Haidar et al., 2015; A. Haidar, Tomov, Arturov et al., 2016). En (Dongarra, Gates, A. Haidar et al., 2015) y (A. Haidar, Tomov, Arturov et al., 2016) se ofrece una comparativa de rendimiento de la etapa de factorización LU, QR y Cholesky ofrecidas por la librería MAGMA en las familias de tarjetas coprocesadoras KNC y KNL de Intel Xeon Phi respecto a la ejecución de MKL en CPU. Si analizamos los resultados obtenidos se observa que se han logrado aceleraciones de 25-35x respecto de la ejecución monohilo en CPU en la familia KNC aumentando hasta tasas de 50x en la familia KNL. Si bien es cierto que los resultados anteriores están referidos exclusivamente a la etapa de factorización.

Se han obtenido resultados similares a los anteriormente citados con frameworks de desarrollo propios para acelerar la etapa de factorización LU (G. Zhang, Y. Chen, Y. Zhang et al., s.f.). En (Y. Chen, Y. Zhang, G. Zhang et al.,

2016) los autores utilizan la capacidad de cómputo de la Xeon Phi en una implementación híbrida MIC/CPU sobre un clúster de varios nodos. Las tarjetas coprocesadoras se utilizan exclusivamente para el cómputo de la etapa de factorización LU obteniendo aceleraciones en torno a 2x respecto de la ejecución en 2 CPUs Intel Xeon E5-2692 v2 sobre Intel MKL. Se comprobó la escalabilidad y el rendimiento del algoritmo implementado calculando la RCS de un modelo de avión.

Aunque no se han encontrado referencias en la bibliografía del uso de la FFT para acelerar alguna etapa de técnicas avanzadas como pueden ser FMM o MLFMA, existen referencias de la implementación del algoritmo FFT en 3D sobre la arquitectura Intel MIC (Y.-Q. Liu, Y. Li, Y.-Q. Zhang et al., 2014) logrando aceleraciones de 2.22x respecto de la implementación con Intel MKL sobre un procesado Intel Xeon E5-2680.

Capítulo 6

Aceleración de MoM en arquitecturas Intel MIC

| | | |
|-------|--|----|
| 6.1 | Especificaciones de la estación de trabajo | 75 |
| 6.2 | Paralelismo del código HEMCUVE en arquitecturas Intel MIC | 76 |
| 6.2.1 | Patrón de acceso a memoria y consideraciones sobre el bus PCIe | 76 |
| 6.2.2 | Optimizaciones escalares | 77 |
| 6.2.3 | Vectorización | 78 |
| 6.2.4 | Optimizaciones en código paralelo de memoria compartida | 79 |
| 6.3 | Algoritmos mGMRES y mMVP | 80 |
| 6.4 | Geometría fija con un único objeto dispersor | 84 |
| 6.4.1 | Formulación SIE-MoM | 84 |
| 6.4.2 | Cálculo del emplazamiento óptimo de la fuente puntual | 87 |
| | Resultados | 89 |
| 6.4.3 | Cálculo de backscattering de una onda plana | 92 |
| | Resultados | 93 |
| 6.5 | Geometría variable con varios objetos dispersores | 94 |
| 6.5.1 | Formulación SIE-MoM | 94 |
| 6.5.2 | Cálculo de la distribución óptima de un array de dispersores | 97 |

| | |
|--------------------------------------|----|
| Resultados | 98 |
|--------------------------------------|----|

A lo largo del presente capítulo se describe una herramienta que hace uso de la arquitectura Intel MIC para optimizar y acelerar el cómputo de los MVP necesarios en la resolución de los sistemas de ecuaciones lineales generados por MoM.

En la resolución de los sistemas de ecuaciones lineales se hace uso del método del residuo mínimo generalizado, GMRES, introducido en el capítulo 3.

Como framework de desarrollo se ha utilizado HEMCUVE, software privativo perteneciente al grupo de electromagnetismo computacional de las Universidades de Vigo y Extremadura, aportando los algoritmos necesarios adaptados a la arquitectura Intel MIC.

Tras presentar las características de la estación de trabajo utilizada, se exponen las distintas estrategias llevadas a cabo para optimizar el código sobre la arquitectura Intel MIC y los algoritmos desarrollados para lograr la aceleración de los productos matriz-vector sobre el coprocesador. Posteriormente, se analizan los problemas planteados (uno o varios objetos dispersores) y se muestran los resultados obtenidos.

6.1. Especificaciones de la estación de trabajo

La estación de trabajo que se ha utilizado es una cortesía de Intel al grupo de investigación en Mayo de 2014 que posteriormente ha sido modificada para incluir el coprocesador Intel Xeon Phi 7120P de la familia KNC.

Está formada por un procesador Intel Xeon CPU E5-2609 v2 @ 2.50GHz con 16 GB de RAM (en lo sucesivo *host*) que tiene 4 núcleos físicos y 16 hilos, con tres niveles de caché: L1 independiente para datos e instrucciones (128 KB cada uno), y niveles L2 y L3 compartidos de 1 MB y 10 MB respectivamente. Además, presenta un ancho de banda teórico de acceso a memoria de 42.6 GB/s y soporte de expansión para el bus PCIe.

El coprocesador Intel Xeon Phi 7120P con 61 núcleos (4 hilos por núcleo) @ 1.25GHz y 16GB de RAM se integra en el sistema mediante el bus PCIe 2.0 x16. Cada uno de los cores dispone de una VPU con registros SIMD de 512 bits y cachés L1 de 32 KB para datos y 32 KB para instrucciones y L2 compartida de 512KB. El rendimiento teórico pico es de 1.2 TFLOPS en doble precisión. El acceso a memoria ofrece una tasa de transferencia pico de 384 GB/s. La descripción detallada de la arquitectura KNC se presenta en el capítulo 4.

6.2. Paralelismo del código HEMCUVE en arquitecturas Intel MIC

Es indispensable que los algoritmos y aplicaciones que se pretenden ejecutar en el coprocesador sean apropiados y altamente eficientes para poder ser paralelizados de una forma óptima. El mero hecho de disponer de un coprocesador no indica que éstos sean mejor opción que los procesadores de propósito general.

Se parte del código altamente eficiente de HEMCUVE desarrollado en C++ y se pretende añadir al mismo un kernel altamente paralelizable en el que el coprocesador realice los productos matriz-vector Zx de manera paralela para varios vectores x , dado que se pretende resolver un problema con M partes izquierdas. Además, el coprocesador deberá calcular los elementos de Z a la vez que realiza la operación de producto matriz-vector.

El paralelismo en el coprocesador se consigue mediante OpenMP y el uso de las directivas `#pragma` que son capturadas por el compilador y permiten gestionar los distintos hilos de ejecución sobre memoria compartida.

La ejecución del código en la Xeon Phi en modo nativo requiere añadir al compilador `icc` de Intel la directiva de compilación `-mmic`. Aunque el simple hecho de compilar el código añadiendo la directiva anterior consigue ejecutables para la arquitectura Intel MIC, es necesario un proceso de optimización tanto en el planteamiento del problema como en su programación para conseguir resultados adecuados. Las siguientes estrategias de optimización de código han sido consultadas en (Jeffers y Reinders, 2013).

6.2.1. Patrón de acceso a memoria y consideraciones sobre el bus PCIe

El acceso a memoria en la Xeon Phi es rápido (384GB/s), casi tres veces más rápido que el ancho de banda en memoria de un sistema de doble socket Intel Xeon y unas nueve veces más rápido que el nodo donde se ejecuta. Es aconsejable para aprovechar esta característica que el acceso a memoria se realice en flujos. En el caso que nos ocupa, una vez calculado cada elemento de Z , se accede por referencia a las partes izquierdas que están ubicadas en posiciones contiguas de memoria favoreciendo el acceso rápido a los mismos. Otra práctica que se ha llevado a cabo en partes del código ha sido el intercambio

de bucles, externo por interno y viceversa ya que en ocasiones puede reducir los fallos de caché.

En cuanto al bus PCIe, es necesario tener en cuenta el tiempo de transferencia de los datos desde el host al coprocesador y viceversa. Dependiendo del tiempo de cómputo en el coprocesador puede que el uso no estuviera justificado. En el caso que se plantea, un producto matriz-vector en el que la complejidad es cuadrática, $O(N^2)$, el acceso a cada dato transferido se realizará n veces. Además, el cálculo de cada elemento de Z requiere de un gran número de operaciones aritméticas. Todo ello hace que la latencia de comunicación se considere despreciable.

6.2.2. Optimizaciones escalares

A continuación se exponen las optimizaciones escalares llevadas a cabo para mejorar el rendimiento de cada tarea paralela:

- *Uso de la palabra reservada const* - Siempre que una variable o los argumentos de una función no vayan a cambiar se han declarado con la palabra reservada `const`, logrando optimizaciones por parte del compilador al precalcular el valor de dichas variables.
- *Eliminación de operaciones redundantes* - Se ha prestado especial atención en no incluir en bucles internos operaciones que se efectúan de forma redundante en cada iteración. Estas operaciones se pasan a los bucles externos para evitar así cálculos innecesarios.
- *Precisión en funciones aritméticas* - La librería Intel Math define una versión acelerada de la mayoría de funciones matemáticas en el caso de que se utilicen datos en formato en coma flotante de simple precisión. Estas versiones aceleradas son iguales a las normales añadiéndoles el sufijo `-f`: `cosf`, `sinf`, etc.
- *Uso de exponenciales en base 2* - Se realizaron diversas pruebas sobre el código llegando a la conclusión de que una de las operaciones más costosas para la Xeon Phi era la $\exp(z)$. Tras un análisis exhaustivo, se observó que el uso principal de la función exponencial se producía en el cálculo de las integrales de la función de Green para un triángulo en los distintos puntos de observación. Se ha decidido sustituir la función $\exp(z)$ por el equivalente menos costoso en base 2 $\exp_{2f}(z*12e)$, siendo `12e` una constante precompilada de la forma `const float 12e =`

$\log_2(\exp(1.0f))$). La definición de esta constante se ha realizado fuera del bucle que realiza el cálculo de las integrales. Además, dado que z representa un número complejo se ha aplicado la siguiente conversión que resulta más efectiva en cuanto a tiempo de computación $\exp(z) = \exp_2(\text{real} * 12e) * (\cosf(\text{imag}) + j*\sinf(\text{imag}))$ y es realizable por la función `exp2`, que no soporta el tipo de datos de números complejos.

- *Opciones del compilador* - El compilador C++ de Intel puede representar las expresiones aritméticas en el código objeto de forma diferente según lo que se conoce como semántica de coma flotante. Se pueden aplicar diferentes directivas de compilación que deben ser utilizadas con precaución ya que pueden mejorar el rendimiento a costa de perder precisión. Se han realizado pruebas en el código con las opciones `-fp-model fast=2` y `fimf-precision=low` comprobando que el uso de la directiva `-fp-model fast=2` produce una reducción en el tiempo de cómputo en torno al 17% sin variar los resultados de salida, mientras que el uso de `fimf-precision=low` apenas produce mejora en rendimiento teniendo un impacto leve en los resultados. Se ha decidido utilizar la directiva `-fp-model fast=2` en nuestro kernel sobre la Xeon Phi.

6.2.3. Vectorización

Se ha utilizado la capacidad de autovectorización del compilador de Intel que ofrece muy buenos resultados sin tener que recurrir al uso de código en ensamblador ni a directivas SIMD. Para ello, se ha compilado el código con la directiva `-O3` que ofrece una vectorización más agresiva que `-O2` y se ha utilizado la directiva `-vec-report=3` para ver los resultados de la autovectorización.

La figura 6.1 muestra parte del informe de vectorización obtenido en el que se observa que algunos bucles han sido vectorizados y otros no, junto con una aclaración del motivo de la no vectorización. A modo de resumen, se han vectorizado un total de 444 bucles mientras que 522 no han sido candidatos a la autovectorización.

La alta dependencia e integración del kernel implementado con el framework de HEMCUVE y todos los desarrollos que se han ido realizando desde su aparición, han hecho imposible lograr una vectorización eficiente, por


```

../h_dielectricos_v106/m3derad_dielectric.h(774): (col. 31) remark: loop was not vectorized: not inner loop
../h_dielectricos_v106/m3derad_dielectric.h(783): (col. 12) remark: LOOP WAS VECTORIZED
../h_dielectricos_v106/m3derad_dielectric.h(783): (col. 12) remark: PEEL LOOP WAS VECTORIZED
../h_dielectricos_v106/m3derad_dielectric.h(783): (col. 12) remark: REMAINDER LOOP WAS VECTORIZED
../h_dielectricos_v106/m3derad_dielectric.h(784): (col. 12) remark: LOOP WAS VECTORIZED
../h_dielectricos_v106/m3derad_dielectric.h(784): (col. 12) remark: PEEL LOOP WAS VECTORIZED
../h_dielectricos_v106/m3derad_dielectric.h(784): (col. 12) remark: REMAINDER LOOP WAS VECTORIZED
../h_dielectricos_v106/m3derad_dielectric.h(791): (col. 31) remark: loop was not vectorized: unsupported loop structure
../h_dielectricos_v106/m3derad_dielectric.h(801): (col. 31) remark: loop was not vectorized: unsupported loop structure
../h_dielectricos_v106/m3derad_dielectric.h(787): (col. 5) remark: loop was not vectorized: not inner loop
../h_dielectricos_v106/m3derad_dielectric.h(751): (col. 3) remark: loop was not vectorized: nonstandard loop is not a vectorization candidate
../h_dielectricos_v106/m3derad_dielectric.h(769): (col. 5) remark: loop was not vectorized: nonstandard loop is not a vectorization candidate
../h_dielectricos_v106/m3derad_dielectric.h(567): (col. 21) remark: LOOP WAS VECTORIZED
../h_dielectricos_v106/m3derad_dielectric.h(567): (col. 25) remark: LOOP WAS VECTORIZED
../h_dielectricos_v106/m3derad_dielectric.h(567): (col. 29) remark: LOOP WAS VECTORIZED
../h_dielectricos_v106/m3derad_dielectric.h(567): (col. 33) remark: LOOP WAS VECTORIZED
../h_dielectricos_v106/m3derad_dielectric.h(597): (col. 10) remark: LOOP WAS VECTORIZED
../h_dielectricos_v106/m3derad_dielectric.h(598): (col. 27) remark: LOOP WAS VECTORIZED
../h_dielectricos_v106/m3derad_dielectric.h(585): (col. 3) remark: loop was not vectorized: not inner loop
../h_dielectricos_v106/./h_dielectricos_v106/m3dvector.h(69): (col. 21) remark: LOOP WAS VECTORIZED
../h_dielectricos_v106/./h_dielectricos_v106/m3dvector.h(69): (col. 21) remark: PEEL LOOP WAS VECTORIZED
../h_dielectricos_v106/./h_dielectricos_v106/m3dvector.h(69): (col. 21) remark: REMAINDER LOOP WAS VECTORIZED
../h_dielectricos_v106/m3dtriangle.h(953): (col. 28) remark: LOOP WAS VECTORIZED
../h_dielectricos_v106/m3dtriangle.h(766): (col. 28) remark: LOOP WAS VECTORIZED

```

FIGURA 6.1: Extracto del informe de vectorización.

ejemplo, muchos de los bucles presentan llamadas encadenadas a métodos de otras clases. Técnicas de optimización avanzadas como el alineado de datos y el relleno requerirían modificar la asignación y liberación de memoria de muchos tipos de datos personalizados heredados de HEMCUVE y estas mejoras no son el objetivo final de la Tesis.

6.2.4. Optimizaciones en código paralelo de memoria compartida

Tal y como se ha comentado, el paralelismo en la Xeon Phi se ha programado con las librerías de OpenMP. El paralelismo del core reside en que cada hilo de la Xeon Phi realiza el cálculo de cuatro elementos de la matriz Z . A su vez, cada uno de los elementos obtenidos será reaprovechado en cada hilo para obtener las corrientes eléctricas y magnéticas de las M partes izquierdas. El detalle del algoritmo utilizado se presenta en la siguiente sección.

Para paralelizar el código se ha añadido una directiva `#pragma omp parallel private(var1, var2, varn ...)` seguida de la directiva `#pragma omp for schedule(runtime)` para poder escoger en tiempo de ejecución la política de

planificación entre las tres existentes: `static`, `dynamic`, `guided`. Tras realizar diversas pruebas se llega a la conclusión de que la política de planificación que aporta las mejores tasas de rendimiento es la `static`, recomendada cuando todos los hilos tienen la misma carga de trabajo.

El uso de determinadas variables de entorno de OpenMP puede mejorar el rendimiento de las aplicaciones. Dependiendo de si el kernel a ejecutar va a realizar un uso intensivo de la memoria o de la capacidad de cómputo se pueden establecer diferentes valores de la variable `MIC_KMP_AFFINITY`. En el caso concreto analizado expuesto en la Tesis, sería recomendable que los hilos que trabajen con datos adyacentes en memoria estén ubicados en el mismo core para tratar de reducir al mínimo los fallos de caché. Se ha establecido el siguiente valor de la variable `export MIC_KMP_AFFINITY = compact, granularity = fine`, comprobando una leve mejora en el rendimiento.

6.3. Algoritmos mGMRES y mMVP

Uno de los objetivos principales de la Tesis es el desarrollo de un kernel altamente paralelizable sobre arquitecturas Intel MIC en el que el coprocesador realice los productos matriz-vector Zx necesarios en la técnica de MoM con M partes izquierdas. El cálculo de los elementos de Z se realiza de forma transparente a la CPU, aprovechando dicho cálculo para efectuar en bloque las distintas operaciones requeridas de cada una de las M simulaciones. Es importante recalcar que cada uno de los M problemas electromagnéticos o **tasks** planteados, puede representar una dirección de incidencia distinta o la posición de una fuente puntual en el caso de geometrías fijas, pero incluso puede aplicarse a que cada uno de los M elementos represente una geometría diferente. En las secciones 6.4.2, 6.4.3 y 6.5.2 se analizan con más detalle cada uno de los casos anteriores.

Se ha decidido aproximar la solución de los M sistemas de ecuaciones modelados con la metodología SIE-MoM, mediante un método iterativo basado en subespacios de Krylov, como es GMRES, debido a su excelente robustez y convergencia (Gomez-Sousa, Rubinos-Lopez y Martinez, 2015) y a la propia experiencia previa en el uso de arquitecturas manycore (L. Landesa, J. M. Taboada, Campon et al., 2017).

Dado que la matriz de impedancias Z es común a todas las **tasks**, se ha implementado un algoritmo iterativo basado en GMRES, mGMRES, que recibe

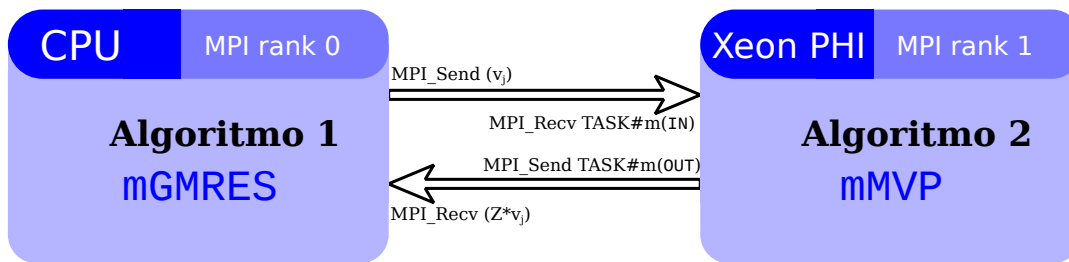


FIGURA 6.2: Paso de mensajes MPI entre CPU y Xeon Phi.

los vectores de excitación de los M tasks en una matriz y una vez ha concluido obtiene una aproximación de las densidades de corriente eléctricas y magnéticas de cada uno de los M problemas.

El nodo de computación utilizado está formado por un procesador Intel Xeon CPU E5-2609 v2 @ 2.50GHz con 16 GB de RAM (en lo sucesivo *host*) en el que se integra mediante el bus PCIe 2.0 x16 una tarjeta coprocesadora Intel Xeon Phi 7120P con 61 cores @ 1.25GHz y 16GB de RAM.

El modelo de programación empleado es el modelo simétrico, de modo que la aplicación correrá simultáneamente en ambos componentes del nodo de computación. En este punto, el paralelismo de memoria distribuida se conseguirá mediante el paradigma de paso de mensajes MPI soportado por la red TCP/IP simulada a través del driver SCIF.

El algoritmo mGMRES se ejecuta en CPU y cada vez que necesita realizar un producto matriz-vector (MVP) entre la matriz de impedancias Z , (A en el algoritmo 1) y los vectores generados por el algoritmo de Arnoldi en GMRES, (v_j en el algoritmo 1 y $TASK\#m(IN)$ en el algoritmo 2), realiza una transferencia vía MPI de los M vectores a la tarjeta coprocesadora.

En este momento, se realiza el cálculo de los elementos de la matriz Z en paralelo. Cada hilo realiza el cálculo de cuatro elementos de la matriz Z , que serán reutilizados en los MVPs de cada uno de los M tasks. Este proceso, recogido en el algoritmo 2, se realiza en paralelo en los hilos de la Xeon Phi. Además, cada hilo aprovecha en sus operaciones las ventajas ofrecidas por la librería Intel MKL.

Una vez el algoritmo mMVP finaliza, el conjunto de los M MVPs ($TASK\#m(OUT)$ en el algoritmo 2) se transfieren de vuelta a la CPU para que el algoritmo mGMRES pueda continuar con su iteración.

Algoritmo 1 {mGMRES} restarted GMRES(m_steps)

```

1: Inicio: Inicializar  $IN_0$ ,  $r_0 = f - AIN_0$  y  $v_1 = \frac{r_0}{\|r_0\|}$  {Llamada inicial a mMVP}
2: Repetir:
3: for  $j = 1$  to m_steps do
4:   for  $k = 1$  to M do
5:      $h_{(k)(i,j)} = (Av_j, v_i), i = 1, 2, \dots, j$ , {Llamada a mMVP}
6:      $\hat{v}_{(k)(j+1)} = Av_j - \sum_{i=1}^j h_{(k)(i,j)} v_{(k)(i)}$ 
7:      $h_{(k)(j+1,j)} = \|\hat{v}_{(k)(j+1)}\|$ , y
8:      $v_{(k)(j+1)} = \frac{\hat{v}_{(k)(j+1)}}{h_{(k)(j+1,j)}}$ 
9:     Construir la solución  $IN(k)$ 
10:    Reinicio:
11:     $r_m = f - Ax_m$ 
12:     $IN_0 = IN_m, v_1 = \frac{r_m}{\|r_m\|}$ 
13:   end for
14: end for

```

Algoritmo 2 {mMVP} massive MVP en Xeon Phi

Entrada: {Geometría del problema (bases RWG), matriz $TASK\#m(IN)$ }

Salida: {matriz $TASK\#m(OUT)$ }

```

1: # pragma omp parallel for
2: {Operaciones por cada hilo i}
3: for  $i = 1$  to  $N$  do
4:   Elegir columnas  $TASK\#m(OUT)_i, TASK\#m(OUT)_{i+N}$ 
5:   for  $j = 1$  to  $N$  do
6:     Calcular elementos  $Z_{i,j}, Z_{i,j+N}, Z_{i+N,j}, Z_{i+N,j+N}$ 
7:     Elegir columnas  $TASK\#m(IN)_j, TASK\#m(IN)_{j+N}$ 
8:     {* representa la sobrecarga de la rutina scal de BLAS}
9:      $TASK\#m(OUT)_i \leftarrow Z_{i,j} * TASK\#m(IN)_j + Z_{i,j+N} * TASK\#m(IN)_{j+N}$ 
10:     $TASK\#m(OUT)_{i+N} \leftarrow Z_{i+N,j} * TASK\#m(IN)_j + Z_{i+N,j+N} * TASK\#m(IN)_{j+N}$ 
11:   end for
12: end for

```

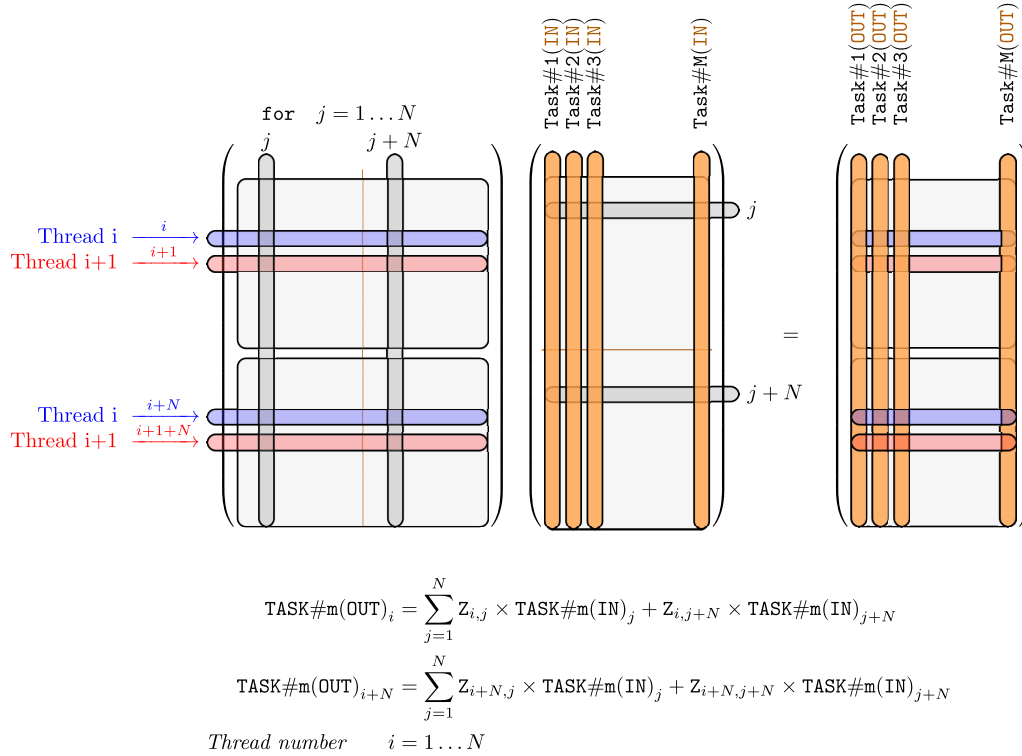


FIGURA 6.3: Detalle del cálculo masivo de los M MVPs en la Xeon Phi

La salida del algoritmo mGMRES representa el cálculo aproximado de las corrientes eléctricas y magnéticas, de cada uno de los M tasks (IN(k)) en el algoritmo 1).

El diagrama de transferencias MPI entre la CPU y la Xeon Phi se muestra en la figura 6.2.

En el desarrollo del algoritmo mGMRES se ha partido de la versión con reinicios presentada en (Youcef Saad y Schultz, 1986). Dado que se pretende abordar la resolución de M tasks de forma simultánea, es necesario modificar el algoritmo original añadiendo al mismo un bucle (línea 4 del algoritmo 1) que acumule los M MVPs en la matriz IN(k), teniendo en cuenta que los MVPs se realizarán de forma paralela en la tarjeta coprocesadora.

Siempre que sea necesaria una operación MVP en el desarrollo interno del algoritmo mGMRES se ejecutará en la tarjeta coprocesadora el algoritmo mMVP tras recibir los datos de la CPU (líneas 1 y 5 del algoritmo 1).

Cada hilo de la Xeon Phi, paralelizado utilizando OpenMP, calcula cuatro elementos de la matriz Z que son reutilizados por los M tasks. Se realiza el producto de todos los elementos j y $j + N$ de los M vectores TASK#m(IN) utilizando las rutinas incluidas en la librería MKL. De esta forma, se consigue

una mejora en eficiencia de cómputo y de accesos a memoria ya que las operaciones se realizan sólo N veces almacenando los resultados en memoria compartida. Al final, cada uno de los hilos realiza el cálculo de las corrientes eléctricas y magnéticas $\text{TASK\#m}(\text{OUT})_i$ e $\text{TASK\#m}(\text{OUT})_{i+N}$ de cada uno de los M tasks en dos puntos de observación. El esquema detallado del algoritmo 2 se puede observar en la figura 6.3.

6.4. Geometría fija con un único objeto dispersor

A continuación se presentan dos ejemplos diferentes que hacen uso del core implementado para resolver los M tasks que se plantean. En el primero de ellos, cada uno de los M problemas electromagnéticos representa la ubicación de una fuente puntual, mientras que en el segundo representa una dirección de incidencia diferente. Ambos problemas se plantean sobre una geometría fija formada por un único objeto dispersor.

Aunque el análisis teórico se puede consultar con más detalle en el capítulo 2, en un primer apartado se realiza una breve introducción del modelo matemático utilizado en el caso de un único objeto dispersor.

6.4.1. Formulación SIE-MoM

Supongamos un caso simplificado donde un objeto dispersor (Región R_2) con parámetros constitutivos ε_2 y μ_2 se introduce en un medio homogéneo (Región R_1) de parámetros constitutivos ε_1 y μ_1 . Definamos S como la superficie de discontinuidad de los dos medios y \hat{n}_i como el vector unitario normal a S apuntando hacia la región R_i .

Se parte de dos campos incidentes denominados $\mathbf{E}_1^{\text{inc}}$ y $\mathbf{H}_1^{\text{inc}}$. Las fuentes que originan dichos campos se ubican en la región 1 y en la superficie de discontinuidad que separa los dos medios en forma de corrientes eléctricas y magnéticas. Estas fuentes inducen sobre las superficies unas nuevas corrientes que generan los campos dispersos $\mathbf{E}_i^{\text{scat}}$ y $\mathbf{H}_i^{\text{scat}}$. Por tanto, los campos eléctricos y magnéticos totales se pueden escribir como:

$$\begin{aligned}\mathbf{E}_i &= \mathbf{E}_i^{inc} + \mathbf{E}_i^{scat} \\ \mathbf{H}_i &= \mathbf{H}_i^{inc} + \mathbf{H}_i^{scat}\end{aligned}\quad (6.1)$$

Para resolver el problema es necesario reemplazar el objeto dispersor por las densidades de corrientes superficiales eléctricas y magnéticas equivalentes, \mathbf{J}_i y \mathbf{M}_i :

$$\begin{aligned}\mathbf{E}_i^{scat} &= -\eta_i \mathbf{L}\{\mathbf{J}_i\} + \mathbf{K}\{\mathbf{M}_i\} \\ \mathbf{H}_i^{scat} &= -\mathbf{K}\{\mathbf{J}_i\} - \frac{1}{\eta_i} \mathbf{L}\{\mathbf{M}_i\}\end{aligned}\quad (6.2)$$

siendo η_i la impedancia intrínseca del medio R_i , y \mathbf{L} y \mathbf{K} los siguientes operadores integrales:

$$\mathbf{L}(\mathbf{X}_S) = j\beta \int_S \left[\mathbf{X}_S(r') G(r, r') + \frac{1}{\beta^2} \nabla' \cdot \mathbf{X}_S(r') \nabla G(r, r') \right] ds' \quad (6.3)$$

$$\mathbf{K}(\mathbf{X}_S) = \int_S \mathbf{X}_S(r') \times \nabla G(r, r') ds' + \frac{1}{2} \mathbf{X}_S \times \hat{n} \quad (6.4)$$

Si se aplica el principio de equivalencia y las condiciones de contorno en la superficie de discontinuidad, las corrientes equivalentes en cada una de las regiones pueden expresarse como:

$$\begin{aligned}\text{N-EFIE}_i : \mathbf{M}_i &= -\hat{n}_i \times \mathbf{E}_i|_S \\ \text{N-MFIE}_i : \mathbf{J}_i &= \hat{n}_i \times \mathbf{H}_i|_S\end{aligned}\quad (6.5)$$

Es posible utilizar otro método para proyectar los campos sobre la superficie de discontinuidad, obteniendo la denominada formulación tangencial:

$$\begin{aligned}\text{T-EFIE}_i : \hat{n}_i \times \mathbf{M}_i &= -\hat{n}_i \times \hat{n}_i \times \mathbf{E}_i|_S = \mathbf{E}_{i,tan}|_S \\ \text{T-MFIE}_i : -\hat{n}_i \times \mathbf{J}_i &= -\hat{n}_i \times \hat{n}_i \times \mathbf{H}_i|_S = \mathbf{H}_{i,tan}|_S\end{aligned}\quad (6.6)$$

En la resolución de problemas electromagnéticos, se hace necesario combinar las ecuaciones 6.5 y 6.6 que se encuentren en el mismo espacio de corrientes para cada una de las regiones. Se ha decidido utilizar las siguientes formulaciones combinadas por su gran estabilidad y precisión en problemas de dispersión (P Ylä-Oijala y Matti Taskinen, 2005b):

$$\frac{a_1}{\eta_1} \text{T-EFIE}_1 - \frac{a_2}{\eta_2} \text{T-EFIE}_2 + b_1 \text{N-MFIE}_1 - b_2 \text{N-MFIE}_2 \quad (6.7)$$

$$-c_1 \text{N-EFIE}_1 + c_2 \text{N-EFIE}_2 + d_1 \eta_1 \text{T-MFIE}_1 - d_2 \eta_2 \text{T-MFIE}_2 \quad (6.8)$$

Utilizando notación matricial, y escogiendo la formulación JMCFIE ($a_i, b_i, c_i, d_i = 1$), ampliamente utilizada por su precisión y buena convergencia (P Ylä-Oijala y Matti Taskinen, 2005a), la combinación de las ecuaciones anteriores puede expresarse como:

$$\begin{aligned} & \left\{ \begin{bmatrix} (\mathbf{L}^1 + \mathbf{L}^2)_{tan} & \left(-\frac{1}{\eta_1} \mathbf{K}^1 - \frac{1}{\eta_2} \mathbf{K}^2\right)_{tan} \\ (\eta_1 \mathbf{K}^1 + \eta_2 \mathbf{K}^2)_{tan} & (\mathbf{L}^1 + \mathbf{L}^2)_{tan} \end{bmatrix} \right. \\ & + \begin{bmatrix} \hat{n} \times (\mathbf{K}^1 - \mathbf{K}^2) & \hat{n} \times \left(\frac{1}{\eta_1} \mathbf{L}^1 - \frac{1}{\eta_2} \mathbf{L}^2\right) \\ -\hat{n} \times (\eta_1 \mathbf{L}^1 - \eta_2 \mathbf{L}^2) & \hat{n} \times (\mathbf{K}^1 - \mathbf{K}^2) \end{bmatrix} \\ & + \frac{1}{2} \begin{bmatrix} 2 & \left(\frac{1}{\eta_1} - \frac{1}{\eta_2}\right) \hat{n} \times \\ (-1\eta_1 + 1\eta_2) \hat{n} \times & 2 \end{bmatrix} \left. \right\} \begin{bmatrix} \mathbf{J} \\ \mathbf{M} \end{bmatrix} \\ & = \begin{bmatrix} \frac{1}{\eta_1} (\mathbf{E}_1^{inc})_{tan} - \frac{1}{\eta_2} (\mathbf{E}_2^{inc})_{tan} + \hat{n} \times (\mathbf{H}_1^{inc} + \mathbf{H}_2^{inc}) \\ \eta_1 (\mathbf{H}_1^{inc})_{tan} - \eta_2 (\mathbf{H}_2^{inc})_{tan} - \hat{n} \times (\mathbf{E}_1^{inc} + \mathbf{E}_2^{inc}) \end{bmatrix} \end{aligned} \quad (6.9)$$

Si se aplica el método MoM, la ecuación integral 6.9 se aproxima mediante el siguiente sistema de ecuaciones lineales de dimensiones $2N \times 2N$, siendo $2N$ el número de incógnitas del problema:

$$\begin{bmatrix} \mathbf{Z}^{1J} & \mathbf{Z}^{1M} \\ \mathbf{Z}^{2J} & \mathbf{Z}^{2M} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{J} \\ \mathbf{M} \end{bmatrix} = \begin{bmatrix} \mathbf{V}^1 \\ \mathbf{V}^2 \end{bmatrix} \quad (6.10)$$

en el que las cuatro submatrices de dimensiones $N \times N$, vienen dadas por las siguientes expresiones:

$$\begin{aligned}
\mathbf{Z}_{m,i}^{1J} &= A_{mi}^1 + A_{mi}^2 + B_{mi}^{1'} - B_{mi}^{2'} + \tau_{mi}' \\
\mathbf{Z}_{m,i}^{1M} &= \frac{1}{\eta_1} (A_{mi}^{1'} - B_{mi}^1) - \frac{1}{\eta_2} (A_{mi}^{2'} + B_{mi}^2) + \frac{1}{2} \left(\frac{1}{\eta_1} - \frac{1}{\eta_2} \right) \tau_{mi} \\
\mathbf{Z}_{m,i}^{2J} &= \eta_1 (B_{mi}^1 - A_{mi}^{1'}) + \eta_2 (B_{mi}^2 + A_{mi}^{2'}) + \frac{1}{2} (-\eta_1 + \eta_2) \tau_{mi} \\
\mathbf{Z}_{m,i}^{2M} &= A_{mi}^1 + A_{mi}^2 + B_{mi}^{1'} - B_{mi}^{2'} + \tau_{mi}'
\end{aligned} \tag{6.11}$$

donde $\mathbf{i} = 1, 2, \dots, N$ y $\mathbf{m} = 1, 2, \dots, N$ y representan los acoplamientos entre los puntos fuente y de observación \mathbf{r} y \mathbf{r}' .

Los vectores \mathbf{J} y \mathbf{M} con las incógnitas del problema vienen definidos por:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \\ \vdots \\ \mathbf{J}_N \end{bmatrix} \quad \mathbf{M} = \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \vdots \\ \mathbf{M}_N \end{bmatrix}$$

y los vectores de excitaciones \mathbf{V}^1 y \mathbf{V}^2 se expresan:

$$\begin{aligned}
\mathbf{V}_{\mathbf{m}}^1 &= \langle \mathbf{f}_{\mathbf{m}}, \frac{a_1}{\eta_1} (\mathbf{E}_1^{inc}) + \frac{a_2}{\eta_2} (\mathbf{E}_2^{inc}) + \hat{n} \times (b_1 \mathbf{H}_1^{inc} - b_2 \mathbf{H}_2^{inc}) \rangle \\
\mathbf{V}_{\mathbf{m}}^2 &= \langle \mathbf{f}_{\mathbf{m}}, d_1 \eta_1 (\mathbf{H}_1^{inc})_{tan} - d_2 \eta_2 (\mathbf{H}_2^{inc})_{tan} - \hat{n} \times (c_1 \mathbf{E}_1^{inc} + c_2 \mathbf{E}_2^{inc}) \rangle
\end{aligned} \tag{6.12}$$

donde $\mathbf{m} = 1, 2, \dots, N$.

6.4.2. Cálculo del emplazamiento óptimo de la fuente puntual

El problema abordado consiste en calcular el emplazamiento óptimo de una fuente puntual para que el campo dispersado por un objeto ubicado en un medio homogéneo sea máximo justamente en dicho punto. No se ha abordado el ejemplo como un problema de optimización, sino que se ha calculado el resultado de la dispersión en una nube densa de puntos, dado que las características del método desarrollado para la Xeon Phi permiten el cálculo masivo de dichas soluciones en la nube de puntos.

Para resolver el problema es necesario realizar M simulaciones diferentes según la ubicación de la fuente puntual. Para ello, se eligen M puntos alrededor del objeto dispersor en los que se ubica la fuente puntual y se realiza el cálculo del campo electromagnético disperso en los mismos emplazamientos en los que se ubican las fuentes. Este tipo de problemas, en el que los transmisores y los receptores se ubican en los mismos puntos, son muy útiles para obtener un mapa de superficie en el que observar el comportamiento del campo disperso y poder localizar los emplazamientos óptimos para evitar problemas o sacar partido de las localizaciones.

El cálculo del emplazamiento que maximiza el campo disperso requiere la resolución de tantos sistemas de ecuaciones lineales del tipo 6.10 como ubicaciones de la fuente puntual haya que evaluar. Dado que todos los M tasks se han planteado en el mismo escenario y a la misma frecuencia, la matriz Z es común a todos ellos.

Para aprovechar esta característica, se ha decidido abordar los M problemas de forma simultánea escogiendo el método GMRES para la resolución de los M sistemas de ecuaciones lineales:

1. Se realiza el cálculo de cada uno de los M vectores de excitación y se almacenan en una matriz de entrada.
2. Se adapta el método iterativo GMRES, para resolver los productos matriz-vector de los M problemas realizando un único cálculo de los elementos de la matriz Z (Algoritmos mGMRES y mMVP). Se ha desarrollado la versión con reinicios a pesar de su menor velocidad de convergencia para poder almacenar todos los vectores en memoria principal.
3. La solución de mGMRES contiene las densidades de corriente eléctricas y magnéticas aproximadas, \mathbf{J} y \mathbf{M} , de cada uno de los M problemas analizados. A partir de dichas corrientes, es posible realizar el cálculo del campo disperso y resolver el problema planteado.

La CPU (MPI rank 0) realizará el cálculo de los distintos vectores de excitación según la ubicación del transmisor, además de calcular el campo disperso y la ubicación óptima de la fuente puntual. En cambio, la tarjeta coprocesadora (MPI rank 1), se utilizará exclusivamente en el cálculo del producto matriz-vector (algoritmo mMVP) asociado a MoM en cada iteración del método mGMRES. Para lograr el paralelismo de datos en la Xeon Phi, se ha hecho uso de la librería Intel MKL y la autovectorización del compilador y para conseguir el paralelismo de tareas se ha recurrido al estándar OpenMP.

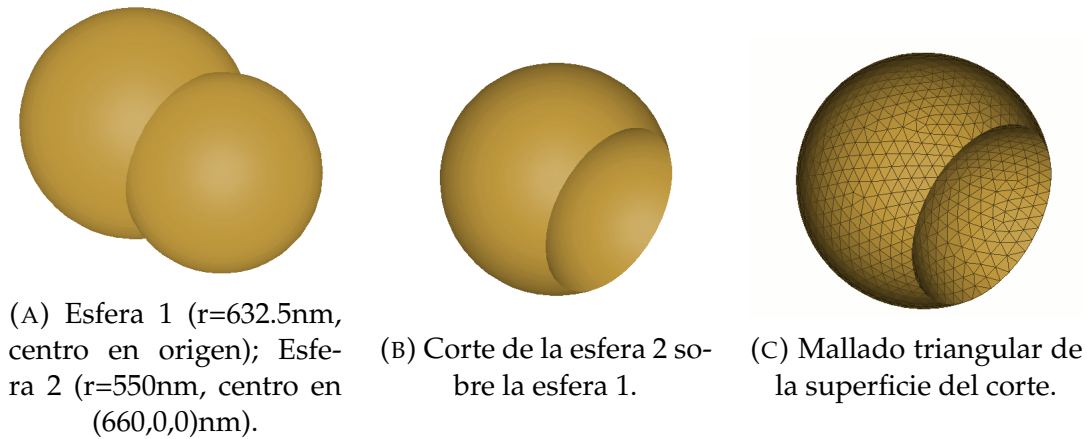


FIGURA 6.4: Creación del objeto dispersor a partir de dos esferas.

Resultados

Se considera un planteamiento en el que un objeto de oro es radiado por una fuente puntual de $\lambda = 550\text{nm}$ (A esta longitud de onda la permitividad relativa es $\varepsilon = -8,0 - j1,66$).

El proceso de creación del objeto dispersor aparece representado en la figura 6.4a. Se parte de dos esferas, una de ellas con radio 632.5 nm y centro en el origen de coordenadas y la otra con radio 550 nm y centro en $(660,0,0)$ nm. Al realizar el corte de una sobre la otra, se crea el objeto dispersor de la figura 6.4b. Dicho objeto es mallado con 2956 triángulos (Figura 6.4c), que producen a su vez 4434 funciones base RWG. Por tanto, el número de incógnitas considerando corrientes eléctricas y magnéticas, es $2N=8868$.

Se realiza un mallado del espacio alrededor del objeto dispersor en el plano $Z = 0$ con una resolución de 13706 puntos. En cada uno de estos puntos se ubicará una fuente puntual transmisora/receptora que conduce a uno de los M tasks. En la figura 6.5 se muestra el campo disperso en cada uno de los puntos de la malla a partir de 160 iteraciones del algoritmo mGMRES con restart de 22, $M = 13706$ y $2N = 19440$. Se han tenido en cuenta las simetrías del objeto dispersor para reducir el número de tasks a $M = 6853$.

Para mostrar la eficiencia del algoritmo mMVP, se compara el tiempo de ejecución en la tarjeta coprocesadora Xeon Phi con el tiempo de ejecución sobre el procesador Intel Xeon.

En la tabla 6.1, se muestra el tiempo de ejecución del algoritmo mMVP en la tarjeta coprocesadora, dependiendo del número de tasks (M), el número de

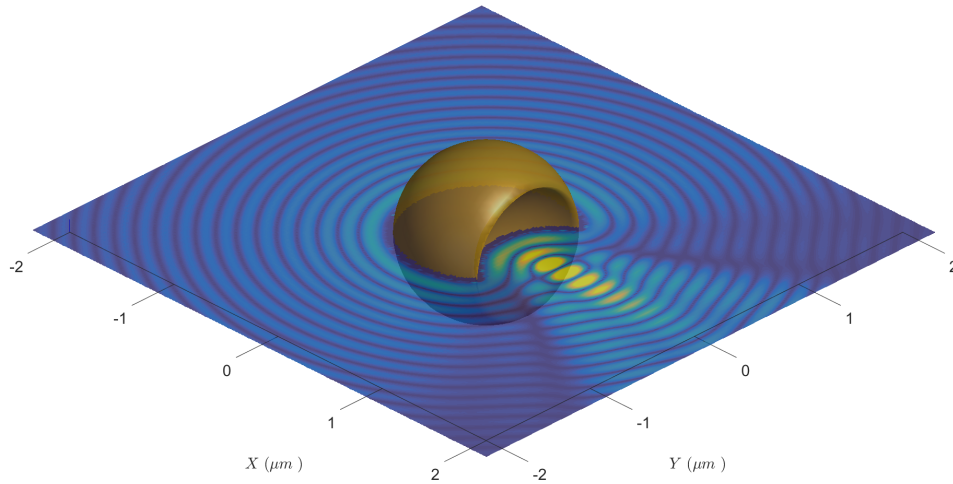


FIGURA 6.5: Campo disperso por un objeto de oro ($\epsilon = -8 - j1,66$ a la frecuencia de simulación) en el plano $Z=0$. $\lambda=550\text{nm}$

incógnitas ($2N$) y el número de hilos utilizados en la paralelización.

Además, la tabla 6.1 muestra el valor de eficiencia. Dado que el tiempo de ejecución se relaciona con el número de tasks de forma lineal y con el número de incógnitas de forma cuadrática, se puede definir la eficiencia:

$$\text{Eficiencia} = \frac{t_0 M N^2}{t M_0 N_0^2} \quad (6.13)$$

donde M_0 , N_0 y t_0 representan los valores tomados como referencia que se corresponden con el caso mejor obtenido en las distintas pruebas realizadas con la Xeon Phi. En el problema planteado, se ha tomado $M_0 = 3365$, $2N_0 = 3840$ y 240 hilos como la línea de base con 100% eficiencia y tiempo de ejecución $t_0 = 34,8$ s.

De los resultados obtenidos, merece la pena destacar el hecho de que si se escoge la máxima paralelización en la Xeon Phi (240 hilos), el aumento del número de incógnitas tiene un efecto despreciable en la eficiencia, siempre que exista suficiente memoria RAM tanto en el host como en la Xeon Phi.

También es objeto de apreciación que la variación en M es significativa. Las

TABLA 6.1: Tiempo de ejecución del algoritmo mMVP en Xeon Phi

| | M=96 2N=3840 | | | M=1184 2N=3840 | | | M ₀ =3365 2N ₀ =3840 | | | M=5455 2N=3840 | | |
|---------------|---------------|-----|------|-----------------|------|------|--|------|------|-----------------|------|------|
| Nhilos | 240 | 128 | 16 | 240 | 128 | 16 | 240 | 128 | 16 | 240 | 128 | 16 |
| Tiempo(s) | 18 | 27 | 134 | 22 | 34 | 160 | 34.8 | 48.4 | 221 | 63 | 63 | 303 |
| Eficiencia(%) | 5.5 | 3.7 | 0.7 | 55.7 | 36.0 | 7.7 | 100 | 71.9 | 15.7 | 89.5 | 89.5 | 18.6 |
| | M=96 2N=8868 | | | M=1184 2N=8868 | | | M=3365 2N=8868 | | | M=5455 2N=8868 | | |
| Nhilos | 240 | 128 | 16 | 240 | 128 | 16 | 240 | 128 | 16 | 240 | 128 | 16 |
| Tiempo(s) | 99 | 149 | 732 | 123 | 177 | 869 | 188.5 | 264 | 1178 | 335 | 325 | 1553 |
| Eficiencia(%) | 5.3 | 3.5 | 0.7 | 53.0 | 36.9 | 7.5 | 98.4 | 70.3 | 15.7 | 89.8 | 92.6 | 19.4 |
| | M=96 2N=19440 | | | M=1184 2N=19440 | | | M=3365 2N=19440 | | | M=5455 2N=19440 | | |
| Nhilos | 240 | 128 | 16 | 240 | 128 | 16 | 240 | 128 | 16 | 240 | 128 | 16 |
| Tiempo(s) | 460 | 683 | 3467 | 583 | 844 | 4104 | 901 | 1251 | 5681 | 1655 | 1618 | 7223 |
| Eficiencia(%) | 5.5 | 3.7 | 0.7 | 53.8 | 37.2 | 7.6 | 99.0 | 71.3 | 15.7 | 87.4 | 89.4 | 20.0 |

TABLA 6.2: Comparativa de rendimiento entre Intel Xeon e Intel Xeon Phi

| M=3365 2N=8868 | | | | M=3365 2N=19440 | | | |
|-----------------|-----|-----------------|-------|-----------------|------|-----------------|-----|
| Xeon E7-8867 v3 | | Xeon Phi 7120P | | Xeon E7-8867 v3 | | Xeon Phi 7120P | |
| Nhilos | 8 | Nhilos | 240 | Nhilos | 8 | Nhilos | 240 |
| T_Z (s) | 329 | T_{Z+MVP} (s) | 188.5 | T_Z (s) | 1634 | T_{Z+MVP} (s) | 901 |
| T_{MVP} (s) | 185 | T_{Total} (s) | 188.5 | T_{MVP} (s) | 848 | T_{Total} (s) | 901 |
| T_{Total} (s) | 514 | | | T_{Total} (s) | 2482 | | |

diferentes pruebas llevadas a cabo nos han mostrado una eficiencia máxima en torno a las 3000-3400 tasks.

La tabla 6.2 muestra una comparativa de rendimiento en cuanto a tiempo de ejecución de los nuevos algoritmos desarrollados para Xeon Phi, respecto de los tradicionales ejecutados sobre Intel Xeon. Para elaborar dicha comparativa, se ha resuelto el mismo problema en un sistema HPC que consta de cuatro procesadores Intel Xeon E7-8867 v3 a 2.50 GHz. Los cálculos han sido realizados utilizando el algoritmo tradicional sobre CPU que primero calcula la matriz de impedancias Z y después realiza las operaciones MVP (Rivero, J. Taboada, Luis Landesa et al., 2010).

La resolución del problema planteado en un procesador Intel Xeon E7-8867 v3 utilizando 8 hilos de paralelización con $M = 3365$ y $2N = 8868$ llevó un tiempo de 514s, mientras que en la Xeon Phi utilizando los nuevos algoritmos 188.5s. Profundizando un poco más, se observa que se han invertido 329s en el cálculo de la matriz de impedancias Z y 185s en el cálculo de los M MVPs utilizando la librería optimizada MKL. Es decir, se obtienen tiempos

de cómputo similares si comparamos el cálculo de los elementos de la matriz Z y la realización de los M MVPs en la Xeon Phi con el cálculo exclusivo de los M MVPs en la CPU Intel Xeon utilizando 8 hilos.

Suponiendo que la eficiencia en paralelización fuera en torno a 100 % en CPU, se ha logrado una aceleración de 22x respecto de la ejecución monohilo en el procesador Intel Xeon.

6.4.3. Cálculo de backscattering de una onda plana

A continuación se analiza un problema de backscattering en el que una onda plana incide sobre un objeto dispersor.

La resolución del problema requiere realizar M simulaciones diferentes dependiendo de la dirección de incidencia de la onda plana. Se eligen M direcciones de incidencia distintas sobre el espacio tridimensional y se calcula el campo electromagnético disperso en las mismas direcciones en que se incide. El cálculo del backscattering es básico en aplicaciones de radar y muy utilizado para conocer las pérdidas en cableados de fibra óptica.

El planteamiento utilizado es similar al del apartado 6.4.2, salvo que en el caso que nos ocupa cada simulación representa una dirección de incidencia distinta de una onda plana en lugar de una ubicación diferente de una fuente puntual. En resumen, se deben resolver tantos sistemas de ecuaciones del tipo 6.10 como direcciones de incidencia diferentes haya que evaluar con un planteamiento exacto al del apartado 6.4.2:

1. Se realiza el cálculo de cada uno de los M vectores de excitación y se almacenan en una matriz de entrada.
2. Se adapta el método iterativo GMRES, para resolver los productos matriz-vector de los M problemas realizando un único cálculo de los elementos de la matriz Z (Algoritmos mGMRES y mMVP). Se ha desarrollado la versión con reinicios.
3. La solución de mGMRES contiene las densidades de corriente eléctricas y magnéticas aproximadas, \mathbf{J} y \mathbf{M} , de cada uno de los M problemas analizados. A partir de dichas corrientes, es posible realizar el cálculo del campo disperso y resolver el problema planteado.

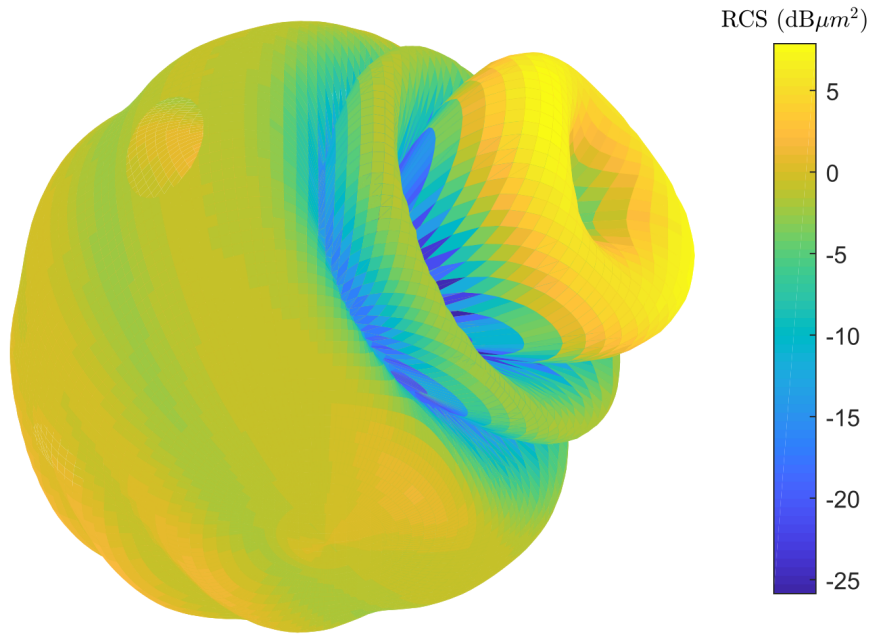


FIGURA 6.6: RCS de un objeto de oro ($\epsilon = -8 - j1,66$ a $\lambda=550nm$) iluminado con una onda plana.

Resultados

El objeto dispersor 6.4c es ahora iluminado por una onda plana cuya dirección de incidencia va variando según θ, ϕ con una longitud de onda de $\lambda = 550nm$. Para esta longitud de onda la constante dieléctrica relativa del oro es $\epsilon = -8 - j1,66$. El número de incógnitas considerando corrientes eléctricas y magnéticas, es $2N = 8868$.

En la figura 6.6 se muestra el RCS del objeto dispersor iluminado, según las distintas direcciones de incidencia de la onda plana a partir de 175 iteraciones del algoritmo mGMRES con restart de 25, $M = 12800$ y $2N = 19440$. Se han analizado un total de 12800 incidencias diferentes. A pesar de que el objeto presenta claramente simetría de revolución y se podría haber calculado un único plano (p.ej. $\phi = 0$ y variar sólo θ) se ha preferido realizar el cálculo en el espacio tridimensional para validar resultados. Se ha reducido el número de direcciones de incidencia a las presentes en un único cuadrante y después se han extrapolado los resultados. De esta forma, el número de tasks se ha reducido a $M = 3200$.

La figura 6.6 muestra el RCS del objeto dispersor 6.4c en un mallado de 12800 direcciones diferentes. Se puede observar claramente como el RCS permanece constante en la parte esférica del objeto dispersor y aumenta claramente en

torno a 5dB en las direcciones de incidencia en las que se ubica la hendidura. La ligera asimetría que se observa en la figura es debida a la discretización polar de las direcciones.

6.5. Geometría variable con varios objetos dispersores

A continuación, se plantea un escenario formado por diferentes objetos dispersores en el que cada uno de los M tasks representa una geometría distinta a evaluar.

En un primer apartado se plantea un resumen de la formulación empleada en el caso de varios objetos dispersores, teniendo presente que se puede consultar con más detalle en el capítulo 2. Posteriormente, se analiza un problema concreto y su adaptación al core implementado.

6.5.1. Formulación SIE-MoM

Se realizará un análisis partiendo del escenario planteado. Supongamos que en un medio homogéneo, (Región R_1), con permitividad ϵ_1 , se introduce un array de objetos dispersores, (Regiones R_i), con permitividad ϵ_i y permeabilidad μ_i . La interfaz de separación entre los medios R_1 y R_i se define como S_{1i} . También, se define \hat{n}_{i1} como el vector unitario normal a S_{1i} apuntando hacia la región R_i .

Se denomina \mathbf{E}_1^{inc} y \mathbf{H}_1^{inc} a los campos eléctricos y magnéticos incidentes debido a las fuentes ubicadas en la región 1, teniendo presente que los campos incidentes en los medios sin fuentes son nulos. Dado que estas fuentes inducen sobre las superficies de discontinuidad unas corrientes que generan los campos dispersos \mathbf{E}_i^{scat} y \mathbf{H}_i^{scat} , el campo electromagnético total en la región R_i puede expresarse como la suma del campo incidente en la región R_i y los campos dispersos producidos por las corrientes generadas en las distintas superficies:

$$\begin{aligned}\mathbf{E}_i &= \mathbf{E}_i^{inc} + \sum_{j'} \mathbf{E}_i^{scat}(\mathbf{J}_{ij'}, \mathbf{M}_{ij'}) \\ \mathbf{H}_i &= \mathbf{H}_i^{inc} + \sum_{j'} \mathbf{H}_i^{scat}(\mathbf{J}_{ij'}, \mathbf{M}_{ij'})\end{aligned}\quad (6.14)$$

donde j' hace referencia a cada una de las regiones adyacentes a la región R_i . Además, los campos dispersos pueden expresarse como:

$$\begin{aligned}\mathbf{E}_i^{scat}(\mathbf{J}_{ij'}, \mathbf{M}_{ij'}) &= -\eta_i \mathbf{L}\{\mathbf{J}_{ij'}\} + \mathbf{K}\{\mathbf{M}_{ij'}\} \\ \mathbf{H}_i^{scat}(\mathbf{J}_{ij'}, \mathbf{M}_{ij'}) &= -\mathbf{K}\{\mathbf{J}_{ij'}\} - \frac{1}{\eta_i} \mathbf{L}\{\mathbf{M}_{ij'}\}\end{aligned}\quad (6.15)$$

siendo η_i la impedancia intrínseca del medio y \mathbf{L} y \mathbf{K} los siguientes operadores integrales:

$$\mathbf{L}(\mathbf{X}_S) = j\beta \int_S \left[\mathbf{X}_S(r') G(r, r') + \frac{1}{\beta^2} \nabla' \cdot \mathbf{X}_S(r') \nabla G(r, r') \right] ds' \quad (6.16)$$

$$\mathbf{K}(\mathbf{X}_S) = \int_S \mathbf{X}_S(r') \times \nabla G(r, r') ds' + \frac{1}{2} \mathbf{X}_S \times \hat{n} \quad (6.17)$$

Aplicando el principio de equivalencia y las condiciones de contorno a las superficies de discontinuidad, las corrientes equivalentes en las superficies de discontinuidad se pueden expresar como:

$$\text{T-EFIE}_i : \quad \mathbf{E}_{i,tan}^{inc} = \hat{n}_{i1} \times \mathbf{M}_{i1} + \sum_{j'} \left([\eta_i \mathbf{L}_0\{\mathbf{J}_{ij'}\} - \mathbf{K}_0\{\mathbf{M}_{ij'}\}]_{tan} - \frac{1}{2} \hat{n}_{ij} \times \mathbf{M}_{ij'} \right) \quad (6.18)$$

$$\text{T-MFIE}_i : \quad \mathbf{H}_{i,tan}^{inc} = -\hat{n}_{i1} \times \mathbf{J}_{i1} + \sum_{j'} \left(\left[\mathbf{K}_0\{\mathbf{J}_{ij'}\} + \frac{1}{\eta_i} \mathbf{L}_0\{\mathbf{M}_{ij'}\} \right]_{tan} + \frac{1}{2} \hat{n}_{ij} \times \mathbf{J}_{ij'} \right) \quad (6.19)$$

$$\text{N-EFIE}_i : \quad \hat{n}_{i1} \times \mathbf{E}_i^{inc} = -\mathbf{M}_{i1} + \hat{n}_{i1} \times \sum_{j'} \left(\eta_i \mathbf{L}_0\{\mathbf{J}_{ij'}\} - \mathbf{K}_0\{\mathbf{M}_{ij'}\} - \frac{1}{2} \hat{n}_{ij} \times \mathbf{M}_{ij'} \right) \quad (6.20)$$

$$\text{N-MFIE}_i : \quad \hat{n}_{i1} \times \mathbf{H}_i^{inc} = \mathbf{J}_{i1} + \hat{n}_{i1} \times \sum_{j'} \left(\mathbf{K}_0\{\mathbf{J}_{ij'}\} + \frac{1}{\eta_i} \mathbf{L}_0\{\mathbf{M}_{ij'}\} + \frac{1}{2} \hat{n}_{ij} \times \mathbf{J}_{ij'} \right) \quad (6.21)$$

Combinando las ecuaciones anteriores que se encuentran en el mismo espacio de corrientes según la formulación JMC FIE y aplicando el método de los momentos, se obtiene el siguiente sistema de ecuaciones lineales en notación matricial:

$$\begin{bmatrix} \mathbf{Z}_{12,12} & \mathbf{Z}_{12,13} & \dots & \mathbf{Z}_{12,1r} \\ \mathbf{Z}_{13,12} & \mathbf{Z}_{13,13} & \dots & \mathbf{Z}_{13,1r} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Z}_{1r,12} & \mathbf{Z}_{1r,13} & \dots & \mathbf{Z}_{1r,1r} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_{12} \\ \mathbf{I}_{13} \\ \vdots \\ \mathbf{I}_{1r} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{12} \\ \mathbf{V}_{13} \\ \vdots \\ \mathbf{V}_{1r} \end{bmatrix} \quad (6.22)$$

donde cada subbloque $\mathbf{Z}_{1r,1s}$ representa el acoplo de la superficie S_{1s} sobre la superficie S_{1r} . Cada uno de estos subbloques se expresan como un sistema matricial de dimensiones $2N \times 2N$:

$$\mathbf{Z}_{1r,1s} = \begin{bmatrix} \mathbf{Z}_{1r,1s}^{1J} & \mathbf{Z}_{1r,1s}^{1M} \\ \mathbf{Z}_{1r,1s}^{2J} & \mathbf{Z}_{1r,1s}^{2M} \end{bmatrix}$$

y los vectores \mathbf{I}_{1r} y \mathbf{V}_{1r} se definen como:

$$\mathbf{I}_{1r} = \begin{bmatrix} \mathbf{J}_{1r} \\ \mathbf{M}_{1r} \end{bmatrix} \quad \mathbf{V}_{1r} = \begin{bmatrix} \mathbf{V}_{1r}^1 \\ \mathbf{V}_{1r}^2 \end{bmatrix}$$

Los subbloques $\mathbf{Z}_{1r,1s}$ de la matriz de impedancias representan la interacción electromagnética entre las superficies S_{1r} y S_{1s} . En el caso que nos ocupa se pueden distinguir dos casos diferenciados dependiendo de la relación existente entre las superficies.

El primer caso se produce cuando las superficies S_{1r} y S_{1s} son las mismas, y es conocido como autocoplo. Estos bloques se ubican en la diagonal de la matriz de impedancias y se definen como:

$$\begin{aligned} \mathbf{Z}_{1r,1r}^{1J}[m, i] &= A_{mi}^1 + A_{mi}^r + B_{mi}^{\prime 1} - B_{mi}^{\prime r} + \tau_{mi}' \\ \mathbf{Z}_{1r,1r}^{1M}[m, i] &= -\frac{1}{\eta_1} B_{mi}^1 - \frac{1}{\eta_r} B_{mi}^r + \frac{1}{\eta_1} A_{mi}^{\prime 1} - \frac{1}{\eta_r} A_{mi}^{\prime r} + \frac{1}{2} \left(\frac{1}{\eta_1} - \frac{1}{\eta_r} \right) \tau_{mi} \\ \mathbf{Z}_{1r,1r}^{2J}[m, i] &= \eta_1 B_{mi}^1 + \eta_r B_{mi}^r - \eta_1 A_{mi}^{\prime 1} + \eta_r A_{mi}^{\prime r} + \frac{1}{2} (-\eta_1 + \eta_r) \tau_{mi} \\ \mathbf{Z}_{1r,1r}^{2M}[m, i] &= A_{mi}^1 + A_{mi}^r + B_{mi}^{\prime 1} - B_{mi}^{\prime r} + \tau_{mi}' \end{aligned} \quad (6.23)$$

El otro caso posible en el problema planteado se produce cuando la superficie S_{1r} recibe la excitación de la superficie S_{1s} a través del medio R_1 . El acoplo en este caso se determina:

$$\begin{aligned}
 \mathbf{Z}_{1r,1s}^{1J}[m, i] &= A_{mi}^1 + B_{mi}^{1'} \\
 \mathbf{Z}_{1r,1s}^{1M}[m, i] &= -\frac{1}{\eta_1} B_{mi}^1 + \frac{1}{\eta_1} A_{mi}^{1'} \\
 \mathbf{Z}_{1r,1s}^{2J}[m, i] &= \eta_1 B_{mi}^1 - \eta_1 A_{mi}^{1'} \\
 \mathbf{Z}_{1r,1s}^{2M}[m, i] &= A_m^1 i + B^{1'} m i
 \end{aligned} \tag{6.24}$$

donde $\mathbf{i} = 1, 2, \dots, N$ y $\mathbf{m} = 1, 2, \dots, N$ y representan los acoplamientos entre los puntos fuente y de observación \mathbf{r} y \mathbf{r}' .

6.5.2. Cálculo de la distribución óptima de un array de dispersores

El problema consiste en calcular la distribución óptima de un array de $N \times N$ objetos dispersores, ubicados sobre un plano en el espacio tridimensional, que maximizan el campo electromagnético en un determinado punto de dicho espacio.

Se parte de una fuente puntual y un conjunto de elementos dispersores equiespaciados, ubicados sobre el plano de referencia. Se pretende conseguir la distribución óptima de los distintos elementos dispersores que maximice el campo electromagnético en un punto del espacio.

Para ello, se utiliza el framework de desarrollo HEMCUVE y la librería de algoritmos genéticos GALIB (*GALIB-A C++ Library of Genetic Algorithm Components* 2007). Los algoritmos desarrollados para conseguir acelerar los productos matriz-vector sobre tarjetas coprocesadoras Intel Xeon Phi son los que se han presentado en la sección 6.3.

Cada uno de los M individuos de la población del algoritmo genético se corresponde con una de las posibles distribuciones de los objetos dispersores en el plano de partida. Hay que tener en cuenta que la distribución en cuestión puede contener un número de dispersores entre 0 y $N \times N$ ubicados en las posiciones de partida sobre el plano original.

Cada uno de los M individuos de la población, que representa una posible distribución de objetos dispersores sobre el plano, está caracterizado por una matriz de impedancias Z del tipo 6.22, donde cada subbloque $Z_{1r,1s}$ representa el acoplo existente entre dos de los elementos dispersores. Teniendo en cuenta que cuando un objeto dispersor no existe, los subbloques de la matriz de impedancias Z afectados por dicho objeto son nulos, es posible aprovechar los algoritmos planteados en la Tesis para acelerar el problema planteado. Para ello, se parte de la falsa premisa de que todos los objetos dispersores están presentes en todos los individuos de la población y se fijan a 0 los elementos V_{1r} e I_{1r} correspondientes a los objetos dispersores que no existen. De este modo se logran simular múltiples geometrías con una única matriz Z común a todos los individuos.

De forma resumida el procedimiento es el siguiente:

- Se construye una geometría formada por la totalidad de los $N \times N$ elementos dispersores.
- En cada iteración del algoritmo genético se realiza el cálculo de cada uno de los M vectores de excitación. Cada uno de estos vectores se corresponde con cada uno de los individuos de la población y se almacenan en una matriz de entrada. Los elementos V_{1r} , correspondientes a los objetos dispersores que no existen en los distintos individuos de la población, son puestos a 0.
- Se resuelven los productos matriz-vector de los M problemas planteados realizando un único cálculo de los elementos de la matriz Z . Véanse los algoritmos 1 y 2.
- La solución de mGMRES contiene las densidades de corrientes eléctricas y magnéticas de los M problemas planteados, con la salvedad de que es necesario anular los elementos I_{1r} correspondientes a los objetos dispersores que no existen.

La descripción del nodo de computación utilizado y del uso de la coprocesadora Xeon Phi y la CPU han sido comentados en las secciones 6.1 y 6.3.

Resultados

A continuación se analiza un caso consistente en una geometría inicial formada por una agrupación de 10×10 cubos de material conductor con $\epsilon =$

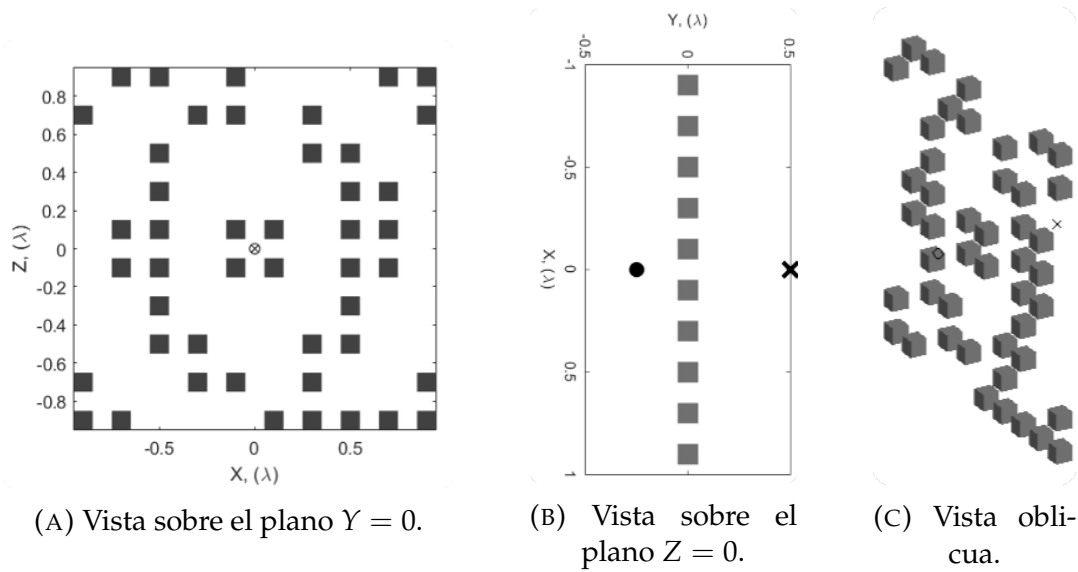


FIGURA 6.7: Array de cubos ($\epsilon = 1 - j1000$) que maximizan el campo electromagnético.

$1 - j1000$ dispuestos sobre el plano $Y = 0$. Cada uno de estos elementos tiene unas dimensiones $l = 0,1\lambda$ y se encuentran separados por una distancia entre centros de $0,2\lambda$, estando el centro de la geometría analizada ubicado en el origen de coordenadas. Para excitar el conjunto se utiliza una fuente puntual ubicada en $(0, -0,25, 0)$.

El objetivo del problema es maximizar el campo electromagnético en el punto $(0, 0,5, 0)$ mediante una distribución óptima de los distintos elementos de la geometría inicial. Para ello, se plantea un algoritmo genético con 200 generaciones y los siguientes parámetros: probabilidad de mutación, $pmut = 0,01$, probabilidad de cruce, $pcross = 0,9$ y número de individuos de la población, $M = 200$.

En cada generación del algoritmo genético se evalúa una población de 200 individuos mediante 175 iteraciones del algoritmo mGMRES con restart de 25, siendo M el número de individuos de la población y $2N = 3600$.

En la figura 6.7 se muestran desde distintas perspectivas la distribución de los cubos que maximizan el campo electromagnético. La ubicación de la fuente aparece representada por un círculo mientras que el punto de observación aparece representado por una cruz. Los resultados muestran una clara tendencia a la simetría que implican que la solución es bastante fiable teniendo en cuenta la gran complejidad del problema planteado.

Es importante resaltar que se ha evitado cualquier simplificación de simetrías para abordar un problema lo suficientemente complejo. Esto prueba la robustez del método implementado.

Capítulo 7

Conclusiones y líneas futuras

| | |
|------------------------------|-----|
| 7.1 Conclusiones | 102 |
| 7.2 Líneas futuras | 103 |

7.1. Conclusiones

La presente Tesis Doctoral se fundamenta en ofrecer un nuevo enfoque en el uso de la arquitectura Intel MIC en el ámbito del electromagnetismo computacional. Los precedentes en el uso de dicha arquitectura referenciados en la bibliografía para la aceleración de MoM, aprovechan la capacidad de cómputo proporcionada por la tarjeta coprocesadora en la etapa de factorización LU de los métodos iterativos estacionarios (Dongarra, Gates, A. Haidar et al., 2014; Dongarra, Gates, A. Haidar et al., 2015; A. Haidar, Tomov, Arturov et al., 2016; G. Zhang, Y. Chen, Y. Zhang et al., s.f.; Y. Chen, Y. Zhang, G. Zhang et al., 2016).

La aportación principal de esta Tesis reside en el hecho de que se ha utilizado la tarjeta coprocesadora de una forma novedosa para resolver un problema de dispersión electromagnética analizado mediante la metodología SIE-MoM utilizando métodos iterativos basados en subespacios de Krylov. Concretamente, se han desarrollado algoritmos propios que realizan las operaciones MVP necesarias en cada iteración de GMRES de forma paralela, demostrando que el uso de la arquitectura Intel MIC puede resultar muy adecuado en electromagnetismo computacional como complemento a CPUs (Campon y Luis Landesa, 2018). El capítulo 6 recoge una explicación exhaustiva de los algoritmos implementados y describe el modelo de programación utilizado, detallando la interrelación existente entre los distintos componentes del nodo de computación.

Además, se ha demostrado y comprobado el funcionamiento y aplicación de los algoritmos anteriormente desarrollados sobre dos tipos de escenarios: simulaciones con geometrías fijas para direcciones de incidencia distintas o ubicaciones distintas de una fuente puntual y simulaciones con geometrías variables.

Se han analizado varios problemas complejos de dispersión electromagnética con un objeto dispersor de oro radiado a frecuencias ópticas, concretamente $\lambda = 550nm$. A esta longitud de onda, el oro se comporta como un material plasmónico, cuya permitividad relativa es $\epsilon = -8,0 - j1,66$. Se ha realizado una simulación comparativa de tiempos de ejecución sobre la arquitectura Intel Xeon logrando una aceleración de 22x respecto de la ejecución monohilo.

También, se ha realizado una simulación para calcular la distribución óptima

de un array de objetos dispersores que maximizan el campo electromagnético en un determinado punto del espacio tridimensional. Para ello, se ha hecho uso de la librería de algoritmos genéticos GALIB y se ha planteado el problema de forma que se puedan simular múltiples geometrías con una única matriz de impedancias.

Con estas simulaciones se ha mostrado la robustez y versatilidad del core implementado y su aplicación a distintos escenarios electromagnéticos.

7.2. Líneas futuras

Como evolución del trabajo llevado a cabo en esta Tesis, la línea de desarrollo más directa sería la adaptación de los códigos existentes a métodos más avanzados que reduzcan aún más la complejidad computacional de MoM, como pueden ser FMM o MLFMA. A pesar de que el grupo de investigación no ha obtenido buenos resultados de escalabilidad del método MLFMA en la arquitectura Intel MIC (L. Landesa, J. M. Taboada, Campon et al., 2017), tal y como se ha desarrollado el método, realizando algunas adaptaciones se podría utilizar para el cálculo de las interacciones cercanas en el MLFMA, de modo que dichas interacciones cercanas se calcularían en la Xeon Phi mientras que en la CPU se realizarían el resto de operaciones.

Aunque la opción aparentemente más ventajosa para la Xeon Phi es el uso de bases de tipo RWG, en estos últimos años se está retomando la base half-RWG (hRWG) (Z. Peng, Lim y Lee, 2013), que no es más que la mitad de la base RWG (la parte correspondiente a un solo triángulo). Como la base de la programación que se ha implementado está asociada a triángulos, la parte de las integrales comunes con la RWG sería de fácil adaptación. Se estima que el resto de elementos no tendrían un coste de programación y adaptación a la Xeon Phi muy grande. Un aspecto fundamental por el cual se están retomando estas bases, son su utilidad en los métodos de descomposición de dominios (DDM, *Domain Decomposition Methods*) (Martín, Larios, Solís et al., 2020) en los que la arquitectura Xeon Phi puede tomar un fuerte protagonismo, pues se pueden mandar trozos de geometría a la Xeon Phi para su análisis en paralelo a otros trozos de geometría que sean analizados por otras estructuras de computación.

Intel ha anunciado que ha suspendido la producción de la familia de productos Intel Xeon Phi a finales de 2019 aunque continuará con los envíos hasta

mediados de 2020, en favor de su nueva arquitectura basada en GPUs Intel Xe. Partiendo de la base de que las GPUs se han impuesto como estándar de arquitecturas manycore, sería conveniente estudiar la posibilidad de adaptación de parte de los códigos existentes para aprovechar las capacidades de cómputo de este hardware. Como se ha analizado en el capítulo 5, parece que la aplicación en GPUs de técnicas avanzadas sobre MoM, como FMM o MLFMA, han presentado excelentes resultados en cuanto a escalabilidad y rendimiento y pueden constituir una opción a valorar.

Finalmente, otra posible línea de interés, en la que el grupo de investigación posee también una larga y acreditada experiencia sería la aplicación de la FFT a determinadas etapas de FMM o MLFMA. El desarrollo de nuevos algoritmos en GPUs supondría la capacidad de analizar grandes problemas electromagnéticos impensables hasta la fecha.

Bibliografía

- ACML-AMD Core Math Library (2013). URL: <https://web.archive.org/web/20141015020116/http://developer.amd.com/tools-and-sdks/cpu-development/amd-core-math-library-acml/>.
- Adams, S., J. Payne y R. Boppana (2007). «Finite Difference Time Domain (FDTD) Simulations Using Graphics Processors». En: *2007 DoD High Performance Computing Modernization Program Users Group Conference*, pág. 334.
- Ahmed, S. y P. Daly (1969). «Waveguide solutions by the finite-element method». En: *Radio and Electronic Engineer* 38 (4), 217-223(6). ISSN: 0033-7722.
- Andersson, T. (1993). «Moment-Method Calculations of Scattering by a Square Plate Using Singular Basis Functions and Multipole Expansions». En: *Journal of Electromagnetic Waves and Applications* 7.1, págs. 93-121. DOI: [10.1163/156939393X01092](https://doi.org/10.1163/156939393X01092).
- Anzt, Hartwig, Mark Gates, Jack Dongarra et al. (2017). «Preconditioned Krylov solvers on GPUs». En: *Parallel Computing* 68. Applications for the Heterogeneous Computing Era, págs. 32-44. ISSN: 0167-8191. DOI: <https://doi.org/10.1016/j.parco.2017.05.006>. URL: <http://www.sciencedirect.com/science/article/pii/S0167819117300777>.
- Araújo, Marta, Jose Taboada, Javier Rivero et al. (2011). «Comparison of Surface Integral Equations for Left-Handed Materials». En: *Progress in Electromagnetics Research-Pier* 118. DOI: [10.2528/PIER11031110](https://doi.org/10.2528/PIER11031110).
- Bahi, Jacques M., Raphaël Couturier y Lilia Ziane Khodja (2011). «Parallel GMRES implementation for solving sparse linear systems on GPU clusters». En: *SpringSim*.
- Banas, K. y F. Kruzel (2014a). «Comparison of Xeon Phi and Kepler GPU Performance for Finite Element Numerical Integration». En: *2014 IEEE Intl Conf on High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC,CSS,ICSS)*, págs. 145-148. DOI: [10.1109/HPCC.2014.27](https://doi.org/10.1109/HPCC.2014.27).
- (2014b). «OpenCL Performance Portability for Xeon Phi Coprocessor and NVIDIA GPUs: A Case Study of Finite Element Numerical Integration».

- En: *Euro-Par 2014: Parallel Processing Workshops*. Cham: Springer International Publishing, págs. 158-169. ISBN: 978-3-319-14313-2.
- Bértolo, J. M., M. G. Araújo, J. M. Taboada et al. (2011). «Extended near field preconditioner for the analysis of large problems using the nested-FMM-FFT algorithm». En: *Microwave and Optical Technology Letters* 53.2, pág. 430. DOI: [10.1002/mop.25709](https://doi.org/10.1002/mop.25709).
- Best Practice Guide Intel Xeon Phi v2.0* (2017). URL: <http://www.prace-ri.eu/> (visitado 04-08-2019).
- Buck, Ian, Tim Foley, Daniel Horn et al. (2004). «Brook for GPUs: Stream computing on graphics hardware». En: *ACM Trans. Graph.* 23, págs. 777-786. DOI: [10.1145/1186562.1015800](https://doi.org/10.1145/1186562.1015800).
- Cai, Yong, Guangyao Li y Wenyang Liu (2018). «Parallelized implementation of an explicit finite element method in many integrated core (MIC) architecture». En: *Advances in Engineering Software* 116, págs. 50-59. ISSN: 0965-9978. DOI: <https://doi.org/10.1016/j.advengsoft.2017.12.001>. URL: <http://www.sciencedirect.com/science/article/pii/S0965997817304593>.
- Campon, J. L. y Luis Landesa (2018). «Fast solution of electromagnetic scattering problems using Xeon Phi coprocessors». En: *The Journal of Supercomputing* 75, págs. 370-383.
- Cecka, Cris (2017). «Low Communication FMM-accelerated FFT on GPUs». En: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. SC '17. ACM*, 54:1-54:11. ISBN: 978-1-4503-5114-0. DOI: [10.1145/3126908.3126919](https://doi.org/10.1145/3126908.3126919). URL: <http://doi.acm.org/10.1145/3126908.3126919>.
- Cecka, Cris, Adrian J. Lew y E. Darve (2011). «Assembly of finite element methods on graphics processors». En: *International Journal for Numerical Methods in Engineering* 85.5, págs. 640-669. DOI: [10.1002/nme.2989](https://doi.org/10.1002/nme.2989).
- Cevahir, Ali, Akira Nukada y Satoshi Matsuoka (2009). «Fast Conjugate Gradients with Multiple GPUs». En: *Computational Science – ICCS 2009*. Berlin, Heidelberg: Springer Berlin Heidelberg, págs. 893-903. ISBN: 978-3-642-01970-8.
- Chang, Y. y R. Harrington (1977). «A surface formulation for characteristic modes of material bodies». En: *IEEE Transactions on Antennas and Propagation* 25.6, págs. 789-795. ISSN: 0018-926X. DOI: [10.1109/TAP.1977.1141685](https://doi.org/10.1109/TAP.1977.1141685).
- Chen, Y., Z. Lin, Y. Zhang et al. (2015). «Parallel out-of-core higher-order Method of Moments accelerated by Graphics Processing Units». En: *2015 IEEE*

- International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting*, págs. 1674-1675. DOI: [10.1109/APS.2015.7305226](https://doi.org/10.1109/APS.2015.7305226).
- Chen, Yan, Yu Zhang, Guanghui Zhang et al. (2016). «Hybrid MIC/CPU parallel implementation of MoM on MIC cluster for electromagnetic problems». En: *IEICE Transactions on Electronics E99.C*, págs. 735-743. DOI: [10.1587/transele.E99.C.735](https://doi.org/10.1587/transele.E99.C.735).
- Clough, R.W. (1960). *The Finite Element Method in Plane Stress Analysis*. American Society of Civil Engineers.
- Coifman, R., V. Rokhlin y S. Wandzura (1993). «The fast multipole method for the wave equation: a pedestrian prescription». En: *IEEE Antennas and Propagation Magazine* 35, págs. 7-12. DOI: [10.1109/74.250128](https://doi.org/10.1109/74.250128).
- Cools, K., F. P. Andriulli, D. De Zutter et al. (2011). «Accurate and Conforming Mixed Discretization of the MFIE». En: *IEEE Antennas and Wireless Propagation Letters* 10, págs. 528-531. ISSN: 1548 - 5757. DOI: [10.1109/LAWP.2011.2155022](https://doi.org/10.1109/LAWP.2011.2155022).
- Corporation, NVIDIA (2006). *NVIDIA GeForce 8800 GPU Architecture Overview*. TB-02787-001v0.9. URL: <http://web.cse.ohio-state.edu/~agrawal.28/788-su08/Papers/week2/GPU.pdf> (visitado 01-11-2019).
- Courant, Richard (1943). «Variational methods for the solution of problems of equilibrium and vibrations». En: *Bull. Amer. Math. Soc* 49.1, págs. 1-23.
- Couturier, Raphaël y Stéphane Domas (2012). «Sparse systems solving on GPUs with GMRES». En: *The Journal of Supercomputing* 59.3, págs. 1504-1516. ISSN: 1573-0484. DOI: [10.1007/s11227-011-0562-z](https://doi.org/10.1007/s11227-011-0562-z). URL: <https://doi.org/10.1007/s11227-011-0562-z>.
- Dang, V., Q. M. Nguyen y O. Kilic (2014). «GPU Cluster Implementation of FMM-FFT for Large-Scale Electromagnetic Problems». En: *IEEE Antennas and Wireless Propagation Letters* 13, págs. 1259-1262. ISSN: 1548-5757. DOI: [10.1109/LAWP.2014.2332972](https://doi.org/10.1109/LAWP.2014.2332972).
- Dang, V., N. Tran y O. Kilic (2016). «Scalable Fast Multipole Method for Large-Scale Electromagnetic Scattering Problems on Heterogeneous CPU-GPU Clusters». En: *IEEE Antennas and Wireless Propagation Letters* 15. ISSN: 1548 - 5757. DOI: [10.1109/LAWP.2016.2537779](https://doi.org/10.1109/LAWP.2016.2537779).
- Dang, Vinh, Quang Nguyen y Ozlem Kilic (2013). «Fast Multipole Method for Large-Scale Electromagnetic Scattering Problems on GPU Cluster and FPGA-Accelerated Platforms». En: *Applied Computational Electromagnetics Society Journal* 28, págs. 1187-1198.

- De Donno, D., A. Esposito, L. Tarricone et al. (2010). «Introduction to GPU Computing and CUDA Programming: A Case Study on FDTD [EM Programmer's Notebook]». En: *IEEE Antennas and Propagation Magazine* 52.3, págs. 116-122. DOI: [10.1109/MAP.2010.5586593](https://doi.org/10.1109/MAP.2010.5586593).
- De Donno, Danilo, Alessandra Esposito, Giuseppina Monti et al. (2010). «Iterative Solution of Linear Systems in Electromagnetics (And Not Only): Experiences with CUDA». En: vol. 6586, págs. 329-337. DOI: [10.1007/978-3-642-21878-1_41](https://doi.org/10.1007/978-3-642-21878-1_41).
- Dongarra, Jack, Mark Gates, Azzam Haidar et al. (2014). «Portable HPC Programming on Intel Many-Integrated-Core Hardware with MAGMA Port to Xeon Phi». En: *Parallel Processing and Applied Mathematics*. Ed. por Roman Wyrzykowski, Jack Dongarra, Konrad Karczewski et al. Berlin, Heidelberg: Springer Berlin Heidelberg, págs. 571-581. ISBN: 978-3-642-55224-3.
- (2015). «HPC Programming on Intel Many-integrated-core Hardware with MAGMA Port to Xeon Phi». En: *Sci. Program*. 2015. ISSN: 1058-9244. DOI: [10.1155/2015/502593](https://doi.org/10.1155/2015/502593). URL: <https://doi.org/10.1155/2015/502593>.
- Ergül, Özgür (2014). *The multilevel fast multipole algorithm (MLFMA) for solving large-scale computational electromagnetics problems*. Chichester, West Sussex: Wiley-IEEE Press. ISBN: 978-1-119-97741-4.
- Ergül, Özgür y Levent Gürel (2009). «Comparison of Integral-Equation Formulations for the Fast and Accurate Solution of Scattering Problems Involving Dielectric Objects with the Multilevel Fast Multipole Algorithm». En: *Antennas and Propagation, IEEE Transactions on* 57, págs. 176-187. DOI: [10.1109/TAP.2008.2009665](https://doi.org/10.1109/TAP.2008.2009665).
- Ergül, Zgür y Levent Gürel (2006). «Improving the accuracy of the magnetic field integral equation with the linear-linear basis functions». En: *Radio Science - RADIO SCI* 41. DOI: [10.1029/2005RS003307](https://doi.org/10.1029/2005RS003307).
- Fletcher, R (1976). «Conjugate Gradient Methods for Indefinite Systems». En: *Lecture Notes Math*. 506. DOI: [10.1007/BFb0080116](https://doi.org/10.1007/BFb0080116).
- Fu, Zhisong, T. James Lewis, Robert M. Kirby et al. (2014). «Architecting the finite element method pipeline for the GPU». En: *Journal of Computational and Applied Mathematics* 257, págs. 195-211. ISSN: 0377-0427. DOI: <https://doi.org/10.1016/j.cam.2013.09.001>. URL: <http://www.sciencedirect.com/science/article/pii/S0377042713004470>.
- Fürlinger, Karl (2016). *Intel Xeon Phi Overview: Hardware, Software, Programming (KNC Edition)*. URL: <http://www.sppexa.de/sppexa-activities/teaching-and-education.html> (visitado 01-08-2019).

- GALib-A C++ Library of Genetic Algorithm Components* (2007). URL: <http://lancet.mit.edu/ga/> (visitado 30-01-2020).
- Georgescu, Serban y Hiroshi Okuda (2010). «Conjugate Gradients on multiple GPUs». En: *International Journal for Numerical Methods in Fluids* 64, págs. 1254-1273. DOI: [10.1002/flid.2462](https://doi.org/10.1002/flid.2462).
- Ginestar, Damián (2018-2019). *Métodos iterativos*. [Web; accedido el 13-03-2019]. URL: [URL%7Bhttp://personales.upv.es/dginesta/docencia/posgrado/met_iterativos.pdf%7D](http://personales.upv.es/dginesta/docencia/posgrado/met_iterativos.pdf).
- Gomez-Sousa, Hipolito, Oscar Rubinos-Lopez y Jose Martinez (2015). «Comparison of iterative solvers for electromagnetic analysis of plasmonic nanostructures using multiple surface integral equation formulations». En: *Journal of Electromagnetic Waves and Applications*. DOI: [10.1080/09205071.2015.1120165](https://doi.org/10.1080/09205071.2015.1120165).
- Guan, J., S. Yan y J. Jin (2013). «An OpenMP-CUDA Implementation of Multilevel Fast Multipole Algorithm for Electromagnetic Simulation on Multi-GPU Computing Systems». En: *IEEE Transactions on Antennas and Propagation* 61.7, págs. 3607-3616. ISSN: 1558-2221. DOI: [10.1109/TAP.2013.2258882](https://doi.org/10.1109/TAP.2013.2258882).
- Gumerov, Nail A. y Ramani Duraiswami (2008). «Fast multipole methods on graphics processors». En: *Journal of Computational Physics* 227.18, pág. 8290. ISSN: 0021 - 9991. DOI: <https://doi.org/10.1016/j.jcp.2008.05.023>. URL: <http://www.sciencedirect.com/science/article/pii/S0021999108002921>.
- Haidar, A., S. Tomov, K. Arturov et al. (2016). «LU, QR, and Cholesky factorizations: Programming model, performance analysis and optimization techniques for the Intel Knights Landing Xeon Phi». En: *2016 IEEE High Performance Extreme Computing Conference (HPEC)*, págs. 1-7. DOI: [10.1109/HPEC.2016.7761591](https://doi.org/10.1109/HPEC.2016.7761591).
- Hao, Feng, Colleen L. Nehl, Jason H. Hafner et al. (2007). «Plasmon resonances of a gold nanostar». En: *Nano Lett.* 7.3, págs. 729-732. URL: <https://doi.org/10.1021/nl062969c>.
- Harrington, Roger F. (1993). *Field Computation by Moment Methods*. Wiley-IEEE Press. ISBN: 0780310144.
- (2001). *Time-Harmonic Electromagnetic Fields*. Wiley-IEEE Press.
- he, Bing, Long Tang, Jiang Xie et al. (2015). «Parallel Numerical Simulations of Three-Dimensional Electromagnetic Radiation with MPI-CUDA Paradigms». En: *Mathematical Problems in Engineering* 2015. DOI: [10.1155/2015/823426](https://doi.org/10.1155/2015/823426).

- Hestenes, M. R. y E. Stiefel (1952). «Methods of conjugate gradients for solving linear systems». En: *Journal of research of the National Bureau of Standards* 49, págs. 409-436.
- H.Hertz (1888). «On the Finite Velocity of Propagation of Electromagnetic Actions». En: *Electric Waves* 110.
- Hidayetoglu, M., C. Pearson, I. El Hajj et al. (2018). «A Fast and Massively-Parallel Inverse Solver for Multiple-Scattering Tomographic Image Reconstruction». En: *2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, págs. 64-74. DOI: [10.1109/IPDPS.2018.00017](https://doi.org/10.1109/IPDPS.2018.00017).
- Hrennikoff, Alexander (1941). «Solution of problems of elasticity by the framework method». En: *Journal of applied mechanics* 8.4, págs. 169-175.
- Imai, R., Y. Suzuki y K. Okubo (2016). «Performance comparison of the parallelized FDTD scheme with the PML implemented on GPU and MIC architectures». En: *2016 URSI Asia-Pacific Radio Science Conference (URSI AP-RASC)*, págs. 1572-1573. DOI: [10.1109/URSIAP-RASC.2016.7601292](https://doi.org/10.1109/URSIAP-RASC.2016.7601292).
- Inman, Matthew y Atef Elsherbeni (2006). «Programming video cards for computational electromagnetics applications». En: *Antennas and Propagation Magazine, IEEE* 47, págs. 71-78. DOI: [10.1109/MAP.2005.1608730](https://doi.org/10.1109/MAP.2005.1608730).
- Intel Math Kernel Library Developer Reference* (2019). URL: <https://software.intel.com/en-us/articles/mkl-reference-manual> (visitado 18-09-2019).
- Jackson, Adrian (2017). *Introduction to the Archer Knights Landing Cluster*. URL: <https://www.archer.ac.uk> (visitado 01-08-2019).
- Jeffers, J. y J. Reinders (2013). *Intel Xeon Phi Coprocessor High Performance Programming*. Elsevier Science. ISBN: 9780124104945.
- Jiang, Ronglin, Shugang Jiang, Yu Zhang et al. (2014). «GPU-Accelerated Parallel FDTD on Distributed Heterogeneous Platform». En: *International Journal of Antennas and Propagation* 2014, págs. 1-8. DOI: [10.1155/2014/321081](https://doi.org/10.1155/2014/321081).
- Jin, Jianming (2014). *The Finite Element Method in Electromagnetics*. 3rd. Wiley-IEEE Press. ISBN: 1118571363.
- Kane Yee (1966). «Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media». En: *IEEE Transactions on Antennas and Propagation* 14.3, págs. 302-307.
- Kelley, C. (1995). *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial y Applied Mathematics. DOI: [10.1137/1.9781611970944](https://doi.org/10.1137/1.9781611970944). eprint: <https://epubs.siam.org/doi/pdf/10.1137/1.9781611970944>. URL: <https://doi.org/10.1137/1.9781611970944>.

- Kolundžija, B.M. y A.R. Djordjević (2002). *Electromagnetic Modeling of Composite Metallic and Dielectric Structures*. Antennas and Propagation Library. Artech House. ISBN: 9780890063606. URL: <https://books.google.es/books?id=0x0vDwAAQBAJ>.
- Krakiwsky, S.E., L.E. Turner y M.M. Okoniewski (2004). «Acceleration of finite-difference time-domain (FDTD) using graphics processor units (GPU)». En: vol. 2, págs. V-265. ISBN: 0-7803-8251-X. DOI: [10.1109/ISCAS.2004.1329513](https://doi.org/10.1109/ISCAS.2004.1329513).
- Kruzel, F. y K. Banas (2014). «Finite element numerical integration on Xeon Phi coprocessor». En: *2014 Federated Conference on Computer Science and Information Systems*, págs. 603-612. DOI: [10.15439/2014F222](https://doi.org/10.15439/2014F222).
- Landesa, L., J. M. Taboada, J. L. Campon et al. (2017). «Successes and frustrations in the solution of large electromagnetic problems in supercomputers». En: *2017 International Applied Computational Electromagnetics Society Symposium - Italy (ACES)*. PONENCIA INVITADA. Págs. 1-2. DOI: [10.23919/ROPACES.2017.7916307](https://doi.org/10.23919/ROPACES.2017.7916307).
- LAPACK-Linear Algebra PACKage (2013). URL: <http://www.netlib.org/lapack/> (visitado 29-11-2019).
- Levy, Samuel (1953). «Structural Analysis and Influence Coefficients for Delta Wings». En: *Journal of the Aeronautical Sciences* 20.7, págs. 449-454. DOI: [10.2514/8.2690](https://doi.org/10.2514/8.2690).
- Lezar, E. y D. B. Davidson (2010a). «GPU acceleration of method of moments matrix assembly using Rao-Wilton-Glisson basis functions». En: *2010 International Conference on Electronics and Information Engineering*. Vol. 1. DOI: [10.1109/ICEIE.2010.5559837](https://doi.org/10.1109/ICEIE.2010.5559837).
- (2010b). «GPU-Accelerated Method of Moments by Example: Monostatic Scattering». En: *IEEE Antennas and Propagation Magazine* 52.6, págs. 120-135. DOI: [10.1109/MAP.2010.5723240](https://doi.org/10.1109/MAP.2010.5723240).
- (2010c). «GPU-based LU decomposition for large method of moments problems». En: *Electronics Letters* 46.17, págs. 1194-1196. DOI: [10.1049/el.2010.1680](https://doi.org/10.1049/el.2010.1680).
- Li, Ruipeng y Yousef Saad (2013). «GPU-accelerated preconditioned iterative linear solvers». En: *The Journal of Supercomputing* 63.2, págs. 443-466. ISSN: 1573-0484. DOI: [10.1007/s11227-012-0825-3](https://doi.org/10.1007/s11227-012-0825-3). URL: <https://doi.org/10.1007/s11227-012-0825-3>.
- Liu, K., X. Wang, Y. Zhang et al. (2006). «Acceleration of Time-Domain Finite Element Method (TD-FEM) Using Graphics Processor Units (GPU)».

- En: *2006 7th International Symposium on Antennas, Propagation EM Theory*, págs. 1-4.
- Liu, Yi-Qun, Yan Li, Yun-Quan Zhang et al. (2014). «Memory Efficient Two-Pass 3D FFT Algorithm for Intel® Xeon Phi™ Coprocessor». En: *Journal of Computer Science and Technology* 29.6, págs. 989-1002. ISSN: 1860-4749. DOI: [10.1007/s11390-014-1484-z](https://doi.org/10.1007/s11390-014-1484-z).
- Livesey, M., J. F. Stack, F. Costen et al. (2012). «Development of a CUDA Implementation of the 3D FDTD Method». En: *IEEE Antennas and Propagation Magazine* 54.5, págs. 186-195. DOI: [10.1109/MAP.2012.6348145](https://doi.org/10.1109/MAP.2012.6348145).
- M. Rao, Sadasiva, Donald R. Wilton y Allen W. Glisson (1982). «Electromagnetic scattering by surfaces of arbitrary shape». En: *Antennas and Propagation, IEEE Transactions on* 30, págs. 409-418. DOI: [10.1109/TAP.1982.1142818](https://doi.org/10.1109/TAP.1982.1142818).
- MAGMA-Matrix Algebra on GPU and Multicore Architectures (2017). URL: <https://icl.cs.utk.edu/magma/> (visitado 30-11-2019).
- Margetts, Lee, Jose Mena, W. Hewitt et al. (2016). «Parallel finite element analysis using the Intel Xeon Phi». En:
- Martín, V. F., D. Larios, D. M. Solís et al. (2020). «Tear-and-Interconnect Domain Decomposition Scheme for Solving Multiscale Composite Penetrable Objects». En: *IEEE Access* 8, págs. 107345-107352.
- Martínez Solís, Diego (2014). *Advanced methods on computational electromagnetics and their application to the study of plasmonics and metamaterials*.
- Maxwell, James Clerk (1865). «A dynamical theory of the electromagnetic field». En: *Philosophical Transactions of the Royal Society of London* 155. DOI: [10.1098/rstl.1865.0008](https://doi.org/10.1098/rstl.1865.0008). URL: <https://doi.org/10.1098/rstl.1865.0008>.
- Meng, H., B. Nie, S. Wong et al. (2014). «GPU accelerated finite-element computation for electromagnetic analysis». En: *IEEE Antennas and Propagation Magazine* 56.2, págs. 39-62. DOI: [10.1109/MAP.2014.6837065](https://doi.org/10.1109/MAP.2014.6837065).
- Mie, Gustav (1908). «Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen». En: *Annalen der Physik* 330.3, págs. 377-445. URL: <https://doi.org/10.1002/andp.19083300302>.
- Mrdakovic, B. L., M. M. Kostic, D. I. Olcan et al. (2017). «Acceleration of in-core LU-decomposition of dense MoM matrix by parallel usage of multiple GPUs». En: *2017 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS)*, págs. 1-4. DOI: [10.1109/COMCAS.2017.8244769](https://doi.org/10.1109/COMCAS.2017.8244769).

- Müller, C. y T.P. Higgins (1960). *Foundations of the Mathematical Theory of Electromagnetic Waves*. Springer. DOI: [10.1007/978-3-662-11773-6](https://doi.org/10.1007/978-3-662-11773-6).
- Nagaoka, T. y S. Watanabe (2010). «A GPU-based calculation using the three-dimensional FDTD method for electromagnetic field analysis». En: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, págs. 327-330. DOI: [10.1109/IEMBS.2010.5627705](https://doi.org/10.1109/IEMBS.2010.5627705).
- Nagaoka, Tomoaki y Soichi Watanabe (2013). «Efficient Three-Dimensional FDTD Computation with Emerging Many-Core Coprocessor for Bioelectromagnetic Simulation». En: *Transactions of Japanese Society for Medical and Biological Engineering* 51.Supplement. DOI: [10.11239/jsmbe.51.R-39](https://doi.org/10.11239/jsmbe.51.R-39).
- Nakashima, Hiroshi (2015). «Manycore challenge in particle-in-cell simulation: How to exploit 1 TFlops peak performance for simulation codes with irregular computation». En: *Computers 'I&' Electrical Engineering* 46. DOI: [10.1016/j.compeleceng.2015.03.010](https://doi.org/10.1016/j.compeleceng.2015.03.010).
- Nguyen, Loc Q. (2014). *Intel Manycore Platform Software Stack (Intel MPSS)*. URL: <https://software.intel.com/en-us/articles/intel-manycore-platform-software-stack-mpss> (visitado 04-09-2019).
- Nguyen, Q. M., V. Dang, O. Kilic et al. (2013). «Parallelizing Fast Multipole Method for Large-Scale Electromagnetic Problems Using GPU Clusters». En: *IEEE Antennas and Wireless Propagation Letters* 12, págs. 868-871. ISSN: 1548 - 5757. DOI: [10.1109/LAWP.2013.2271743](https://doi.org/10.1109/LAWP.2013.2271743).
- Noga, A., T. Topa, M. Danisz et al. (2014). «Accelerating method of moments by using modern GPU hardware». En: *2014 IEEE Fourth International Conference on Consumer Electronics Berlin (ICCE-Berlin)*, págs. 259-262. DOI: [10.1109/ICCE-Berlin.2014.7034283](https://doi.org/10.1109/ICCE-Berlin.2014.7034283).
- OpenMP Application Program Interface* (2013). URL: <https://www.openmp.org/specifications/> (visitado 21-09-2019).
- Ortega Lopez, Gloria, F. Vázquez, E M Garzon et al. (2013). «The BiConjugate gradient method on GPUs». En: *The Journal of Supercomputing* 64, págs. 49-58. DOI: [10.1007/s11227-012-0761-2](https://doi.org/10.1007/s11227-012-0761-2).
- Peng, S. y Z. Nie (2008). «Acceleration of the Method of Moments Calculations by Using Graphics Processing Units». En: *IEEE Transactions on Antennas and Propagation* 56.7, págs. 2130-2133. DOI: [10.1109/TAP.2008.924768](https://doi.org/10.1109/TAP.2008.924768).
- Peng, Z., K. Lim y J. Lee (2013). «A Discontinuous Galerkin Surface Integral Equation Method for Electromagnetic Wave Scattering From Nonpenetrable Targets». En: *IEEE Transactions on Antennas and Propagation* 61.7, págs. 3617-3628.

- Poggio, A. J. y E. K. Miller (1973). *Computer Techniques for Electromagnetics*. Pergamon Press.
- Rivero Campos, F. Javier (2012). *Electromagnetismo computacional avanzado para materiales artificiales y nanoestructuras plasmónicas*.
- Rivero, Javier, Jose Taboada, Luis Landesa et al. (2010). «Surface integral equation formulation for the analysis of left-handed metamaterials». En: *Optics express* 18, págs. 15876-86. DOI: [10.1364/OE.18.015876](https://doi.org/10.1364/OE.18.015876).
- Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. 2nd. Philadelphia, PA, USA: Society for Industrial y Applied Mathematics. ISBN: 0898715342.
- Saad, Youcef (1989). «Krylov Subspace Methods on Supercomputers». En: *SIAM J. Sci. Stat. Comput.* 10. DOI: [10.1137/0910073](https://doi.org/10.1137/0910073).
- Saad, Youcef y Martin H Schultz (1986). «GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems». En: *SIAM J. Sci. Stat. Comput.* 7.3, págs. 856-869. ISSN: 0196-5204. DOI: [10.1137/0907058](https://doi.org/10.1137/0907058). URL: <https://doi.org/10.1137/0907058>.
- Saad, Yousef (1981). «Krylov Subspace Methods for Solving Large Unsymmetric Linear Systems». En: *Mathematics of Computation - Math. Comput.* 37, pág. 105. DOI: [10.2307/2007504](https://doi.org/10.2307/2007504).
- Sengupta, D. L. y T. K. Nag Sarkar (2003). «Maxwell, Hertz, the Maxwellians, and the early history of electromagnetic waves». En: *IEEE Antennas and Propagation Magazine* 45, págs. 13-19.
- Silvester, P. (1969). «A General High-Order Finite-Element Analysis Program Waveguide». En: *IEEE Transactions on Microwave Theory and Techniques* 17.4, págs. 204-210. DOI: [10.1109/TMTT.1969.1126932](https://doi.org/10.1109/TMTT.1969.1126932).
- Song, J. M. y W. C. Chew (1995). «Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering». En: *Microwave and Optical Technology Letters* 10.1, págs. 14-19. DOI: [10.1002/mop.4650100107](https://doi.org/10.1002/mop.4650100107). URL: <https://doi.org/10.1002/mop.4650100107>.
- Song, J., C.-C. Lu y W. C. Chew (1997). «Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects». En: *IEEE Transactions on Antennas and Propagation* 45, págs. 1488-1493. DOI: [10.1109/8.633855](https://doi.org/10.1109/8.633855).
- Sonneveld, Peter (1989). «CGS, A Fast Lanczos-Type Solver for Nonsymmetric Linear Systems». En: *SIAM J. Sci. Stat. Comput.* 10.1. ISSN: 0196-5204. DOI: [10.1137/0910004](https://doi.org/10.1137/0910004).
- Sparse Matrices* (1973). Mathematics in Science and Engineering. Elsevier Science. ISBN: 9780080956084.

- Surmin, I.A., S.I. Bastrakov, E.S. Efimenko et al. (2016). «Particle-in-Cell laser-plasma simulation on Xeon Phi coprocessors». En: *Computer Physics Communications* 202, págs. 204-210. ISSN: 0010-4655. DOI: <https://doi.org/10.1016/j.cpc.2016.02.004>.
- Taboada, J. M., L. Landesa, M. G. Araújo et al. (2011). «Supercomputer solutions of extremely large problems in electromagnetics: From ten million to one billion unknowns». En: *Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP)*, págs. 3221-3225.
- Taboada, Jose, Marta Araújo, J.M. Bertolo et al. (2010). «MLFMA-FFT Parallel Algorithm for the Solution of Large-Scale Problems in Electromagnetics». En: *Progress in Electromagnetics Research-Pier* 105, págs. 15-30.
- Taboada, Jose, Luis Landesa, Fernando Obelleiro et al. (2010). «High Scalability FMM-FFT Electromagnetic Solver for Supercomputer Systems». En: *Antennas and Propagation Magazine, IEEE* 51, págs. 20-28. DOI: [10.1109/MAP.2009.5433091](https://doi.org/10.1109/MAP.2009.5433091).
- Tak, Moonho y Taehyo Park (2017). «Parallelized Simulation of a Finite Element Method in Many Integrated Core Architecture». En: *Journal of Engineering Materials and Technology* 139.2. ISSN: 0094-4289. DOI: [10.1115/1.4035326](https://doi.org/10.1115/1.4035326).
- Takahashi, Toru y Tsuyoshi Hamada (2009). «GPU-accelerated boundary element method for Helmholtz' equation in three dimensions». En: *International Journal for Numerical Methods in Engineering* 80.10, págs. 1295-1321. DOI: [10.1002/nme.2661](https://doi.org/10.1002/nme.2661).
- TOP 500 Supercomputers (2019). URL: <https://www.top500.org/lists/2019/11/> (visitado 20-01-2020).
- Tran, N. y O. Kilic (2016a). «Multilevel fast multipole algorithm for dielectric large-scale electromagnetics scattering problems using GPU clusters». En: *2016 IEEE International Symposium on Antennas and Propagation (APSURSI)*, págs. 755-756. DOI: [10.1109/APS.2016.7696086](https://doi.org/10.1109/APS.2016.7696086).
- (2016b). «Parallel Implementations of Multilevel Fast Multipole Algorithm on Graphical Processing Unit Cluster for Large-scale Electromagnetics Objects». En:
- Turner, M. J., R. W. Clough, H. C. Martin et al. (1956). «Stiffness and Deflection Analysis of Complex Structures». En: *Journal of the Aeronautical Sciences* 23.9, págs. 805-823. DOI: [10.2514/8.3664](https://doi.org/10.2514/8.3664).
- Van der Vorst, Henk (1992). «Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems». En:

- SIAM Journal on Scientific and Statistical Computing* 13, pág. 631. DOI: [10.1137/0913035](https://doi.org/10.1137/0913035).
- Vladimirov, A., R. Asai, V. Karpusenko et al. (2015). *Parallel Programming and Optimization with Intel Xeon Phi Coprocessors: Handbook on the Development and Optimization of Parallel Applications for Intel Xeon Processors and Intel Xeon Phi Coprocessors*. Colfax International. ISBN: 9780988523401.
- Vorst, Henk A. van der (2003). *Iterative Krylov Methods for Large Linear Systems*. Cambridge University Press. DOI: [10.1017/CB09780511615115](https://doi.org/10.1017/CB09780511615115).
- Wang, J.J.H. (1991). *Generalized moment methods in electromagnetics: formulation and computer solution of integral equations*. Wiley-interscience publication. J. Wiley. ISBN: 9780471514435.
- Warren, Craig, Antonios Giannopoulos, Alan Gray et al. (2018). «A CUDA-based GPU engine for gprMax: Open source FDTD electromagnetic simulation software». En: *Computer Physics Communications*. DOI: [10.1016/j.cpc.2018.11.007](https://doi.org/10.1016/j.cpc.2018.11.007).
- Weinberg, V. (2014). «Towards Porting a Real-World Seismological Application to the Intel MIC Architecture». En: DOI: [10.5281/zenodo.822684](https://doi.org/10.5281/zenodo.822684).
- Xu, Kan, Da Zhi Ding, Zheng Hong Fan et al. (2010). «Multilevel fast multipole algorithm enhanced by GPU parallel technique for electromagnetic scattering problems». En: *Microwave and Optical Technology Letters* 52.3, págs. 502-507. DOI: [10.1002/mop.24963](https://doi.org/10.1002/mop.24963). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/mop.24963>.
- Ylä-Oijala, P y Matti Taskinen (2005a). «Application of combined field Integral equation for electromagnetic scattering by dielectric and composite objects». En: *Antennas and Propagation, IEEE Transactions on* 53. DOI: [10.1109/TAP.2004.842640](https://doi.org/10.1109/TAP.2004.842640).
- (2005b). «Well-conditioned Muller formulation for electromagnetic scattering by dielectric objects». En: *Antennas and Propagation, IEEE Transactions on* 53, págs. 3316-3323. DOI: [10.1109/TAP.2005.856313](https://doi.org/10.1109/TAP.2005.856313).
- Ylä-Oijala, Pasi, M Taskinen y S Järvenpää (2005). «Surface integral equation formulations for solving electromagnetic scattering problems with iterative methods». En: *Radio Science - RADIO SCI* 40. DOI: [10.1029/2004RS003169](https://doi.org/10.1029/2004RS003169).
- Zhang, G., Y. Chen, Y. Zhang et al. «MIC accelerated LU decomposition for method of moments». En: *2015 IEEE International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting*. DOI: [10.1109/APS.2015.7304765](https://doi.org/10.1109/APS.2015.7304765).