



TESIS DOCTORAL

Aplicación del Método de Descomposición de Dominios empleando soluciones asintóticas para problemas de radiación y dispersión electromagnética en plataformas complejas

David Larios Benítez

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

Conformidad de los directores:

Fdo. Dr. D. José Manuel Taboada Varela

Fdo. Dr. D. José Luis Rodríguez Rodríguez

Esta tesis cuenta con la autorización del director/a y codirector/a de la misma y de la Comisión Académica del programa. Dichas autorizaciones constan en el Servicio de la Escuela Internacional de Doctorado de la Universidad de Extremadura

2023

*“A veces la persona a la que nadie imagina capaz de nada
es la que hace cosas que nadie imagina.”*

- Alan Turing -

Agradecimientos institucionales

- Ministerio de Ciencia, Innovación y Universidades, proyecto EST21/00590.
- Ministerio de Economía y Competitividad y Fondos FEDER, proyectos MAT2014-58201-C2-1-R, MAT2014-58201-C2-2-R, TEC2017-85376-C2-X-R.
- Ministerio de Ciencia e Innovación, proyecto PID2020-116627RB-C2X.
- Junta de Extremadura y fondos FEDER y FSE, proyectos PD18032, IB13185, IB18073, GR15052, GR18055 y GR21072.



Unión Europea



Consejería de Economía e Infraestructuras



Consejería de Economía, Ciencia y Agenda Digital

Agradecimientos personales

En primer lugar, agradecer a mis padres y mi hermano todo el apoyo a lo largo de este tiempo. A ellos debo todo lo que soy y el haber llegado hasta este día. Tanto en el ámbito académico como en el resto de aspectos de mi vida. Sin ellos no habría sido posible. De igual modo a Laura, quien se ha vuelto parte imprescindible de la misma y me ha acompañado durante esta aventura. Un viaje plagado de mudanzas, una pandemia, una erupción volcánica, la guerra de Ucrania, la muerte de Isabel II, ... pero también un sin fin de buenos recuerdos.

En segundo lugar, no es posible entender esta Tesis sin el apoyo de todos mis compañeros. Gracias a Serna, Manu, Lucía, Dani, Alfonso, a aquellos que se fueron pero siguieron aquí, Jorge y Carlos. Y en especial a Víctor, que en palabras de Gandalf sería “un compañero que supera a todos los demás”. Quienes han compartido buenos y malos momentos, siempre dispuestos a echar una mano. En cuanto al resto de la compañía, no podemos olvidar a Fer, Fran, Víctor, las Claras, Ramiro, Luis y las 19 personas del grupo de Mérida. Sería igual de extenso citarlos, como agradecer uno a uno todo lo que les debo. Son las mejores personas que he podido conocer y de las que jamás me desprenderé.

En tercer lugar y por ir concluyendo, aunque siempre quedarán personas por mencionar. Agradecer a mis directores, a Tabo y a Banner, pero también a Luis, Inma, Obi, Marta y Yolanda que han estado durante todo este tiempo, la oportunidad de alcanzar esta meta y convencerme de que era posible. Su paciencia y ánimo han sido extraordinarios y los conocimientos sobre electromagnetismo un nuevo mundo. Muchas gracias.

A todos y todas mi infinito agradecimiento.

Resumen

El objetivo de esta Tesis se centra en dar solución a problemas de radiación y dispersión electromagnética empleando soluciones asintóticas basadas en la metodología de óptica física iterada (IPO). Para ello se resuelve de manera rigurosa y eficiente la ecuación integral del campo magnético (MFIE) mediante la utilización de modelos de fase progresiva sobre facetas triangulares planas para la representación de corrientes equivalentes superficiales. Este tipo de representación permite reducir de manera drástica el número de facetas necesarias en el mallado superficial, a la vez que se mantiene una elevada precisión en la predicción de los campos dispersados.

Se han incorporado además metodologías basadas en trazado de rayos sobre hardware específico (GPUs), basadas en OptiX, con el objetivo de acelerar la obtención de las matrices de visibilidades/ocultaciones desde la fuente y entre las facetas del propio modelo, poniendo el foco en el cálculo de la sección radar (RCS) monoestática de plataformas portantes complejas y de gran tamaño eléctrico. Estas herramientas se han complementado con la implementación de un software de detección de sectores ciegos (*blind sectors*) que permite la elaboración en tiempo real de diagramas de cobertura de antenas, sensores o cualquier tipo de dispositivos a bordo de las estructuras portantes. Este software también facilita la integración de la solución IPO como un *solver* adicional en el marco del método de descomposición de dominios (DDM), para la resolución rápida de dominios con geometrías suaves y de gran tamaño eléctrico.

El conjunto de aportaciones desarrolladas a lo largo de esta Tesis, conforma una alternativa viable y realista con la que abordar proyectos que requieran de simulaciones intensivas de elevada carga computacional, proporcionando además gran flexibilidad en el control de parámetros como la RCS y otros parámetros de compatibilidad electromagnética (EMC) desde las fases iniciales de diseño de una plataforma.

Abstract

The objective of this Thesis focuses on solving electromagnetic radiation and scattering problems using asymptotic solutions based on the iterative physical optics (IPO) methodology. For this, the magnetic field integral equation (MFIE) is solved rigorously and efficiently by applying progressive phase models on flat triangular facets for the equivalent surface currents. This drastically reduces the number of facets required in the surfaces meshing, while maintaining high degrees of accuracy in the prediction of the scattered fields.

Ray tracing methodologies based on OptiX and the use of specific hardware (GPUs) have also been addressed in this Thesis, with the aim of accelerating the calculation of the required visibility matrices from the source and between the model facets, keeping the focus on the calculation of the monostatic radar cross section (RCS) of complex large-scale platforms. These tools have been complemented with the implementation of a blind sectors detection software, which allows real-time confection of coverage patterns for antennas, sensors or any other kind of device under test on board the supporting platforms. This software also facilitates the integration of the IPO solution as an additional *solver* in the framework of the domain decomposition method (DDM), enabling the fast solution of domains with smooth surfaces and large electrical sizes.

The set of contributions developed throughout this Thesis gives rise to a viable and realistic alternative to address projects that require intensive simulations with a high computational cost, while providing great flexibility in the control of parameters such as RCS and other electromagnetic compatibility (EMC) parameters, from the early design stages of a platform.

Lista de publicaciones

Artículos Internacionales

- [1] D. M. Solís, V. F. Martín, M. G. Araújo, **D. Larios**, F. Obelleiro and J. M. Taboada, "Accurate EMC Engineering on Realistic Platforms Using an Integral Equation Domain Decomposition Approach", *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 4, pp. 3002-3015, Abril 2020, DOI: 10.1109/TAP.2019.2950862.
- [2] V. F. Martín, **D. Larios**, D. M. Solís, J. M. Taboada, L. Landesa and F. Obelleiro, "Tear-and-Interconnect Domain Decomposition Scheme for Solving Multiscale Composite Penetrable Objects", *IEEE Access*, vol. 8, pp. 107345-107352, 2020, DOI: 10.1109/ACCESS.2020.3000650.
- [3] S. Gullace, V. Montes-García, V. F. Martín, **D. Larios**, V. Girelli, F. Obelleiro, G. Calogero, S. Casalini and P. Samorì, "Universal Fabrication of Highly Efficient Plasmonic Thin-Films for Label-Free SERS Detection", *Small* 2021, vol. 17, pp. 2100755, DOI: 10.1002/smll.202100755.
- [4] X. Zhuo, M. Mychinko, W. Heyvaert, **D. Larios**, M. Obelleiro-Liz, J. M. Taboada, S. Bals, L. M. Liz-Marzán, "Morphological and Optical Transitions during Micelle-Seeded Chiral Growth on Gold Nanorods", *ACS Nano*, DOI: 10.1021/acsnano.2c08668.

Congresos nacionales e internacionales

- [1] V. F. Martín, **D. Larios**, J. M. Taboada, L. Landesa and D. M. Solís, "Fast Solution of Large Multi-Scale Electromagnetic Problems with MLFMA and Schwarz Domain Decomposition preconditioners," *XXXII Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2017)*, Cartagena (España), 6-8 Septiembre, 2017.
- [2] **D. Larios**, V. F. Martín, Y. Campos-Roca and J. M. Taboada, "Design of UWB Antennas for Communications in the VHF and UHF Frequency Bands," *XXXII Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2017)*, Cartagena (España), 6-8 Septiembre, 2017.
- [3] J. M. Taboada, D. M. Solís, V. F. Martín, **D. Larios**, F. Obelleiro and J. L. Rodríguez, "Surface Integral Equation-Domain Decomposition Scheme for Solving Multi-Scale Radiation and Scattering Problems," *International Conference on Electromagnetics in Advanced Applications (ICEAA 2017)*, Verona (Italia), 11-15 Septiembre, 2017. **Ponencia invitada.**
- [4] D. M. Solís, V. F. Martín, **D. Larios**, J. M. Taboada, J. L. Rodríguez and F. Obelleiro, "DDM-Assisted Analysis of Large Scale Radiating Structures," *XII European Conference on Antennas and Propagation (EuCAP 2018)*, Londres (Reino Unido), 9-13 Abril, 2018. **Ponencia invitada.**
- [5] **D. Larios**, V. F. Martín, F. Obelleiro, J. L. Rodríguez and J. M. Taboada, "Topside Electromagnetic Compatibility (EMC) of Communication Systems in the V/UHF Frequency Band," *XXXIII Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2018)*, Granada (España), 5-7 Septiembre, 2018.
- [6] V. F. Martín, **D. Larios**, J. M. Taboada, L. Landesa, F. Obelleiro and D. M. Solís, "Aceleración de Problemas Electromagnéticos Multiescala mediante el Método de Descomposición de Dominios," *XXXIII Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2018)*, Granada (España), 5-7 Septiembre, 2018.
- [7] V. F. Martín, D. M. Solís, **D. Larios**, J. M. Taboada, L. Landesa, J. L. Rodríguez and F. Obelleiro, "DDM-Assisted Analysis of Large Scale Composite Problems," *XIII European Conference on Antennas and Propagation (EuCAP 2019)*, Cracovia (Polonia), 31 Marzo-5 Abril, 2019. **Ponencia invitada.**
- [8] V. F. Martín, D. M. Solís, **D. Larios**, J. M. Taboada, J. L. Rodríguez and F. Obelleiro, "Solving Realistic Multiscale and Composite Problems using an Integral Equation Domain Decomposition Approach," *2019 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI 2019)*, Atlanta (USA), 7-12 Julio, 2019.
- [9] V. F. Martín, J. M. Taboada, D. M. Solís, **D. Larios** and F. Obelleiro, "Tear and Interconnect Domain Decomposition Analysis of Piecewise Penetrable Structures," *International Conference on Electromagnetics in Advanced Applications (ICEAA 2019)*, Granada (España), 9-13 Septiembre, 2019. **Ponencia invitada.**
- [10] J. M. Taboada, D. M. Solís, V. F. Martín, **D. Larios**, L. Landesa, J. L. Rodríguez and F. Obelleiro, "Fast Maxwell's Simulation of New Real-World Problems at the Nanoscale," *International Conference on Electromagnetics in Advanced Applications (ICEAA 2019)*, Granada (España), 9-13 Septiembre, 2019. **Ponencia invitada.**
- [11] **D. Larios**, M. G. Araújo, D. M. Solís, V. F. Martín, J. M. Taboada and F. Obelleiro, "Electromagnetic Analysis and Design of Radiating Systems on Board Real Platforms Via Domain Decomposition Method," *International Conference on Electromagnetics in Advanced Applications (ICEAA 2019)*, Granada (España), 9-13 Septiembre, 2019. **Ponencia invitada.**
- [12] V. F. Martín, D. M. Solís, **D. Larios**, J. M. Taboada, L. Landesa and F. Obelleiro, "Toward extremely scalable IE-DDM for distributed computing," *XIV European Conference on Antennas and Propagation (EuCAP 2020)*, Copenague (Dinamarca), Virtual, 15-20 Marzo, 2020. **Ponencia invitada.**

- [13] V. F. Martín, **D. Larios**, J. M. Taboada, L. Landesa and F. Obelleiro, "DG-JMCFIE Formulation for the Simulation of Composite Objects," *International Applied Computational Electromagnetics Society Symposium (ACES 2021)*, Virtual, 1-5 Agosto, 2021. **Ponencia invitada.**
- [14] V. F. Martín, **D. Larios**, D. M. Solís, L. Landesa, J. L. Rodríguez, F. Obelleiro and J. M. Taboada, "Distributed Hybrid MPI/OpenMP IE-DDM for Electromagnetic Modeling of Large Platforms," *International Conference on Electromagnetics in Advanced Applications (ICEAA 2021)*, Hawai (USA), Virtual, 9-13 Agosto, 2021. **Ponencia invitada.**
- [15] V. F. Martín, **D. Larios**, D. M. Solís, L. Landesa, J. L. Rodríguez, F. Obelleiro and J. M. Taboada, "An interior penalty DG-JMCFIE Formulation for Piecewise Homogeneous Objects with multimaterial junctions," *XXXVI Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2021)*, Vigo (España), Virtual, 20-24 Septiembre, 2021.
- [16] J. M. Taboada, V. F. Martín, D. M. Solís, **D. Larios**, L. Landesa and F. Obelleiro, "Fast Integral-Equation Simulation of Plasmonic and Photonic Problems at the Nanoscale," *XXXIV General Assembly and Scientific Symposium (GASS) of the International Union of Radio Science (Union Radio Scientifique Internationale-URSI)*, Roma (Italia), 28 Agosto-4 Septiembre, 2021. **Ponencia invitada.**
- [17] V. F. Martín, D. M. Solís, **D. Larios**, J. M. Taboada, J. L. Rodríguez and F. Obelleiro, "DG-JMCFIE Formulation for Composite Multimaterial Objects," *2021 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI 2021)*, Singapur (Singapur), 4-10 Diciembre, 2021.
- [18] V. F. Martín, **D. Larios**, L. Landesa, F. Obelleiro and J. M. Taboada, "Discontinuous Galerkin JMCFIE Formulation for Solving Multimaterial Composite Objects," *XVI European Conference on Antennas and Propagation (EuCAP 2022)*, Madrid (España), 27 Marzo-1 Abril, 2022. **Ponencia invitada.**
- [19] V. F. Martín, **D. Larios**, J. M. Taboada, L. Landesa and F. Obelleiro, "DG-JMCFIE-EFIE Formulation for Multi-material Complex Radiation Problems," *XIV Iberian Meeting on Computational Electromagnetics (EIEC 2022)*, Nuria (España), 24-27 Mayo, 2022. **Ponencia invitada.**
- [20] V. F. Martín, **D. Larios**, J. M. Taboada, L. Landesa and F. Obelleiro, "Discontinuous Galerkin JMCFIE-EFIE Formulation for Solving Multi-material Complex Antennas," *III Atlantic, Asia-Pacific Radio Science Conference (AT-AP-RASC 2022)*, Gran Canaria (España), 29 Mayo-3 Junio, 2022. **Ponencia invitada.**

Acrónimos

- ADAS** Advanced Driver Assistance Systems.
- API** Application Programming Interface.
- BVH** Bounding Volume Hierarchy.
- CAD** Computer Aided Design.
- CEM** Computational Electromagnetics.
- CPU** Central Processing Unit.
- CUDA** Compute Unified Device Architecture.
- DDM** Domain Decomposition Method.
- EFIE** Electric Field Integral Equation.
- EMC/EMI** Electromagnetic Compatibility and Interference.
- FaFFA** Fast Far Field Approximation.
- FDM** Finite Difference Method.
- FDTD** Finite Difference in the Time Domain.
- FEM** Finite Element Method.
- FMM** Fast Multipole Method.
- GLEW** OpenGL Extension Wrangler Library.
- GLFW** Graphics Library Framework.
- GLSL** OpenGL Shading Language.
- GLUT** OpenGL Utility Toolkit.
- GML** OpenGL Mathematics.
- GMRES** Generalized Minimal Residual Method.
- GO** Geometrical Optics.
- GPU** Graphics Processing Unit.
- GPUGPU** General Purpose Computing on Graphics Processing Units.
- GRE** Generalized Ray Expansion.
- GTD** Geometrical Theory of Diffraction.
- GUI** Graphical User Interface.

HEM Hybrid Expansion Method.

HFPM High Frequency Panel Method.

HH Horizontal transmit and Horizontal receive.

HPC High Performance Computing.

IBC Impedance Boundary Condition.

IE-AP Integral Equation-Asymptotic Phase.

ILDC Incremental Length Diffraction Coefficient.

ImGUI Immediate-mode Graphical User Interface.

IPO Iterative Physical Optics.

ISA Industry Standard Architecture.

J-CFIE J-Combined Field Integral Equation.

JMCFIE Electric and Magnetic Combined Field Integral Equation.

LPIT Linear Phase Interpolation over the Triangular domains.

M-CFIE M-Combined Field Integral Equation.

MFIE Magnetic Field Integral Equation.

MLFMA Multilevel Fast Multipole Method.

MLFMA-FFT Multilevel Fast Multipole Method-Fast Fourier Transform.

MM-PO Method of Moments—Physical Optics.

MoM Method of Moments.

MPI Message Passing Interface.

MVP Matrix Vector Product.

OS Operating System.

PEC Perfect Electric Conductor.

PO Physical Optics.

POHM Physical Optics Hybrid Method.

PTD Physical Theory of Diffraction.

PTX Parallel Thread Execution.

RAM Radar Absorbent Material.

RAM Random Access Memory.

RCS Radar Cross Section.

RT Ray Tracing.

RWG Rao-Wilton-Glisson.

SBR Shooting and Bouncing Ray.

SDK Software Development Kit.

SDL Simple DirectMedia Layer.

SFML Simple and Fast Multimedia Library.

SIE Surface Integral Equation.

SIE-DDM Surface Integral Equation-Domain Decomposition Method.

SIMT Single Instruction Multiple Threads.

TRBVH Treelet Restructuring Bounding Volume Hierarchy.

UAV Unmanned Aerial Vehicle.

UHF Ultra High Frequency.

UTD Uniform Theory of Diffraction.

VBO Vertex Buffer Objects.

VHF Very High Frequency.

VV Vertical transmit and Vertical receive.

Índice general

1. Introducción	1
1.1. Motivación de la Tesis	1
1.2. Antecedentes y estado del arte	2
1.3. Métodos numéricos de análisis electromagnético	4
1.3.1. Métodos full-wave	4
1.3.2. Métodos asintóticos	5
1.3.3. Métodos híbridos	6
1.4. Solución adoptada y aportaciones de esta Tesis	6
1.5. Estructura de la memoria de Tesis	7
2. Métodos de electromagnetismo computacional	9
2.1. Ecuación integral superficial (SIE)	9
2.2. Método de los momentos (MoM)	12
2.2.1. Procedimiento de expansión	13
2.2.2. Procedimiento de ponderación	14
2.2.3. Funciones base	15
2.2.4. Ejemplo de aplicación	16
2.3. Multilevel fast multipole algorithm (MLFMA)	17
2.3.1. Ejemplo de aplicación	19
2.4. Método de descomposición de dominios (DDM)	20
2.4.1. Condiciones de transmisión	22
2.4.2. Acoplo por radiación	23
2.4.3. Ejemplo de aplicación	24
3. Desarrollo y aplicación de la ecuación integral para el campo magnético (MFIE) iterada	27
3.1. Resolución de la MFIE iterada	28
3.1.1. Óptica física (PO)	28
3.1.1.1. Funciones base	30
3.1.1.2. Ejemplos de aplicación	33
3.1.2. Óptica física iterada (IPO)	35
3.1.2.1. Mallado adaptativo	39
3.1.2.2. Matrices de visibilidades	40
3.1.2.3. Detalles de implementación	41
3.1.2.4. Ejemplo de aplicación	43
3.2. Trazado de rayos: NVIDIA OptiX API	47
3.2.1. Aplicaciones del trazado de rayos	48
3.2.2. Estructura de la arquitectura OptiX	49
3.2.3. Programas definidos por el usuario	50
3.2.3.1. Ray Generation	51
3.2.3.2. Bounding Box	53
3.2.3.3. Intersection	54
3.2.3.4. Closest Hit	55
3.2.3.5. Any Hit	57

3.2.3.6. Miss	58
3.2.3.7. Exception	58
3.2.4. OpenGL	58
3.2.5. Módulo de visibilidades de IPO	62
4. Resultados numéricos	66
4.1. Sección equivalente radar (RCS)	66
4.2. Detección de sectores ciegos (Blind Sectors)	72
4.2.1. Estudio de diagramas de cobertura óptica	73
4.2.1.1. Configuración y posicionamiento	74
4.2.1.2. Representación de los diagramas de cobertura óptica y sectores ciegos	76
4.2.2. Versiones beta	81
4.3. Método híbrido DDM-IPO	83
5. Conclusiones	88
5.1. Conclusiones	88
5.2. Líneas futuras	89

Índice de figuras

1.1. Distribución de la parte real de la densidad de corriente superficial obtenida por el método de descomposición de dominios para un avión F22, excitando un pod de 11 antenas de hélice con absorbente, analizadas por MoM y acopladas mediante MLFMA a 1 GHz.	2
1.2. Esquema de la distribución paralelizada y deslocalizada del software de simulación electromagnética M^3 -HEMCUVE mediante DDM.	7
2.1. Representación de un problema equivalente para una geometría arbitraria incidida por una onda plana.	10
2.2. Parámetros geométricos de una función base superficial RWG.	15
2.3. Diseño de una antena de patrón espiral sobre una cavidad PEC con grosor, rellena de un absorbente dieléctrico.	16
2.4. Excitación delta-gap en la unión entre los dos patrones espirales suspendidos en el aire.	17
2.5. Parte real de la distribución de corrientes eléctricas superficiales.	17
2.6. División geométrica mediante <i>octrees</i>	17
2.7. Estructura multinivel del MLFMA.	18
2.8. Vehículo volador no tripulado escogido para la estimación de RCS monoestática mediante MLFMA a 3 GHz.	20
2.9. Resultados de RCS monoestática obtenidos por MLFMA para un UAV a 3 GHz.	20
2.10. Ejemplo de una configuración DDM. Vista detallada de los dominios que pertenecen al pod, D1 a D4, y antenas, D5 a D16, con sus respectivos <i>solvers</i> y formulaciones.	21
2.11. Ejemplo de división geométrica en 18 subdominios (conformados por dominios solapados [D1, D2, D3] y dominios flotantes [D4 a D18]), con sus respectivos <i>flaps</i> internos y externos, para un avión F-22.	23
2.12. División de la superestructura en subdominios. Incluyendo la disposición de las antenas del mástil de popa y sus subdominios a la izquierda y la disposición de las antenas del mástil de proa y sus subdominios a la derecha. Mostrando los <i>flaps</i> correspondientes para los dominios en contacto (solapados).	25
2.13. a) Parte real de la densidad de corrientes eléctrica superficial sobre el buque analizado a 300 MHz por el esquema SIE-DDM b) Acoplo mutuo (S_{12}) entre antenas, para el caso de la antena transmisora #3 en la banda de V/UHF.	26
3.1. Representación de un problema equivalente para una geometría arbitraria como ejemplo de desarrollo de la formulación PO.	29
3.2. Parámetros geométricos asociados a una faceta triangular LPIT.	31
3.3. a) Funciones base RWG b) Funciones base LPIT. Las flechas representan la magnitud y dirección de la densidad de corriente superficial y el degradado la variación de fase.	33
3.4. Comparativa de resultados obtenidos de la simulación MLFMA y PO para un triedro de 50λ de arista a 300 MHz.	34
3.5. Comparativa de medidas y resultados MLFMA y PO de RCS monoestática para un Slicy a 8 GHz.	35
3.6. Representación de un problema equivalente para una geometría arbitraria como ejemplo de desarrollo de la formulación IPO, para el caso de incidencia directa o de 1º orden.	36

3.7. Representación de un problema equivalente, para una geometría arbitraria, como ejemplo de desarrollo de la formulación IPO, para el caso de una contribución de orden k	37
3.8. Esquema geométrico del criterio de subdivisión.	39
3.9. Relación de visibilidades $iVis$ representadas en rojo, para una faceta de observación representada en amarillo.	41
3.10. Contorno de sombra sobre áreas conflictivas debido al mallado.	41
3.11. Triedro de 50λ de arista a 300 MHz.	43
3.12. Mallado adaptativo para $h = 1$ y a) $\zeta = 1/2$, b) $\zeta = 2/3$, c) $\zeta = 1$ y d) $\zeta = 2$	43
3.13. RCS monoestática de un triedro de 50λ a 300 MHz resuelto por el algoritmo IPO naive (a) malla uniforme (b) malla adaptativa.	44
3.14. RCS monoestática de un triedro 50λ a 300 MHz resuelto por el algoritmo IPO-GO (a) malla uniforme (b) malla adaptativa.	44
3.15. Geometría de referencia (Slicy).	45
3.16. Comparativa de medidas y solución MLFMA e IPO de RCS monoestática de un slicy a) a 8GHz b) a 14 GHz.	45
3.17. RCS monoestática de un Slicy a 60GHz resuelto por el algoritmo IPO naive (a) malla uniforme (b) malla adaptativa.	46
3.18. RCS monoestática de un Slicy a 60GHz resuelto por el algoritmo IPO GO (a) malla uniforme (b) malla adaptativa.	46
3.19. Comparación entre la arquitectura CPU vs GPU.	48
3.20. Esquema gráfico del trazado de rayos a través de la arquitectura OptiX.	49
3.21. Esquema de procesos de OptiX.	50
3.22. Esquema gráfico del trazado de rayos a través de los distintos sistemas de coordenadas de la arquitectura OptiX.	52
3.23. Esquema gráfico del trazado de rayos en la elección de los contornos de aceleración de la arquitectura OptiX.	53
3.24. Representación del conjunto de funcionalidades gráficas de la librería ImGUI de OpenGL.	62
3.25. Parametrización de las direcciones de incidencia para la determinación de las matrices de visibilidades IPO ($iLit$) en aplicaciones de RCS monoestática.	63
3.26. Comparativa de la obtención de la matriz $iLit$ para 4 direcciones de incidencia, considerando mismo posicionamiento fuente y observador, en la parte superior, y fijando el punto observador y variando el punto fuente, en la parte inferior.	63
3.27. Comparativa del trazado de rayo basado en la representación coherente de la escena y la solución eficiente para únicamente determinar la visibilidad de las facetas.	64
3.28. Representación de los resultados obtenidos para la implementación eficiente del módulo de visibilidades IPO de la matriz $iVis$ a la izquierda e $iLit$ a la derecha, para el caso de un Eurocopter EC665 Tigre.	65
4.1. Ejemplo de objetivo de ámbito militar real, correspondiente al patrullero Tabarca (P-28) de la Armada Española.	67
4.2. Comparativa entre el mallado uniforme y el mallado adaptativo del buque Tabarca para $h = 0,6$ y $\zeta = 2$	67
4.3. Representación en detalle de las diferencias del mallado uniforme con respecto al adaptativo para el buque Tabarca.	68
4.4. Iluminación para una dirección de incidencia sobre el buque Tabarca.	69
4.5. Esquema del modelo de 4 rayos para contemplar la presencia del plano de mar en la predicción del cálculo de RCS.	69
4.6. Resultados numéricos de la RCS monoestática del buque Tabarca.	70
4.7. Definición de sectores angulares.	71
4.8. Portada del software de predicción de sectores ciegos.	72
4.9. Sistema radar móvil escogido para el estudio de los diagramas de cobertura óptica y sectores ciegos.	74
4.10. Desplazamientos y rotaciones del punto de observación (cámara) disponibles, mediante las funcionalidades del ratón.	75

4.11. Comparativa de los resultados obtenidos al redimensionar la ventana de representación de referencia (imagen superior izquierda), realizando un escalado (imagen superior central), un aumento del rango acimutal (imagen inferior izquierda) y un zoom (derecha), mediante un desplazamiento diagonal, horizontal y vertical respectivamente.	75
4.12. Representación del efecto <i>ojo de pez</i> a la derecha, de la antena dispuesta en la representación de la izquierda. En un ejemplo de línea de visión de una antena directiva en aplicaciones radar.	76
4.13. Representación de los 360° en acimut (efecto ojo de pez) y de -45° a 90° en elevación, a la izquierda en coordenadas cartesianas y a la derecha en coordenadas polares.	76
4.14. Representación de la cuadrícula y numeración de los ejes tanto en la representación cartesiana a izquierdas como en la representación polares a derechas. Se muestran las diferentes configuraciones conmutables entre sí de arriba a abajo.	77
4.15. Representación de superposiciones en distintas configuraciones. Iluminación individualizada en los 4 primeros casos y simultánea en el último.	78
4.16. Representación en color, por defecto (rojo), de los diferentes acabados disponibles en función de los aspectos de iluminación y reflexión u absorción del material.	79
4.17. Representación simultánea de diversos puntos de iluminación, con los que identificar los ocultamientos e interferencia entre sistemas.	80
4.18. Representación de las facetas visibles para una dirección de incidencia (método de depuración de la matriz <i>iLit</i>).	80
4.19. Representación de la reflexión recursiva de un solo nivel.	81
4.20. Representación gráfica de la versión beta del software de detección de sectores ciegos.	82
4.21. Animación del plano de mar basada en el ejemplo <i>OptiX Ocean</i> provisto por NVIDIA.	82
4.22. Configuración de los dispersores tenidos en cuenta para el esquema DDM-IPO.	83
4.23. Contribuciones de RCS de los diferentes elementos del problema enmascaradas.	84
4.24. Contribuciones de RCS de los diferentes elementos del problema.	84
4.25. Contribuciones de RCS de los diferentes elementos del problema.	85
4.26. Contribuciones de RCS de los diferentes elementos del problema enmascaradas.	85
4.27. RCS monoestática del buque Tabarca obtenidas por IPO-GO, DDM-IPO y las medidas.	86
4.28. RCS monoestática del buque Tabarca obtenidas por IPO-GO, DDM-IPO y las medidas con un promediado angular de 2°	86
4.29. Mediana de la RCS monoestática del buque Tabarca obtenidas por IPO-GO, DDM-IPO y las medidas representada en sectores angulares de 10°	87

Índice de tablas

2.1. Parámetros de simulación del buque analizado por DDM.	24
3.1. Resultados de simulación del estudio paramétrico de un triedro de 50λ a 300 MHz mediante IPO	44
3.2. Resultados de simulación del estudio paramétrico de un Slicy mediante IPO.	46
4.1. Estadísticos para los resultados de RCS monoestática obtenidos por IPO-GO y las medidas.	71
4.2. Salida de terminal del software de sectores ciegos para dos casos de estructuras complejas.	73
4.3. Número de incógnitas y tiempo de cómputo de los diferentes dominios del esquema DDM-IPO.	84
4.4. Estadísticos para los resultados de RCS monoestática obtenidos por IPO-GO, DDM-IPO y las medidas.	87
4.5. Estadísticos para los resultados de RCS monoestática obtenidos por IPO-GO, DDM-IPO y las medidas, con promediado angular de 2°	87

Índice de algoritmos

1. Algoritmo base (IPO-Naive).	42
2. Algoritmo acelerado (IPO-GO).	42
3. Pseudocódigo del esquema de procesos de OptiX.	50

Índice de códigos

3.1. Código del Ray Generation Program	52
3.2. Código del Bounding Box Program	54
3.3. Código del Intersection Program	55
3.4. Código del Closest Hit Program	56
3.5. Código del Any Hit Program	57
3.6. Código del Miss Program	58
3.7. Código del Exception Program	58
3.8. Representación gráfica mediante OpenGL, basado en la librería GLUT.	60

CAPÍTULO 1 --- Introducción

Contenido

1.1. Motivación de la Tesis	1
1.2. Antecedentes y estado del arte	2
1.3. Métodos numéricos de análisis electromagnético	4
1.3.1. Métodos full-wave	4
1.3.2. Métodos asintóticos	5
1.3.3. Métodos híbridos	6
1.4. Solución adoptada y aportaciones de esta Tesis	6
1.5. Estructura de la memoria de Tesis	7

1.1. Motivación de la Tesis

La estimación de la sección equivalente radar o RCS (*Radar Cross Section*) puede definirse como la probabilidad de detección y seguimiento de objetivos y blancos de interés, en cualquier tipo de escenario, de manera efectiva. Esto resulta de especial interés a la hora de evaluar parámetros fundamentales en el funcionamiento de un radar, y se ha convertido en un requisito indispensable durante las primeras etapas del proceso de diseño y fabricación de vehículos, buques y aviones, especialmente en el ámbito militar. Esta reciente y acuciante demanda ha devenido en una prioridad incluso en el caso de plataformas ya construidas, donde el cálculo de la RCS es requerido de cara a la realización de modificaciones que permitan reducir su firma radar para su adecuación en operaciones de vigilancia, así como en la planificación de contramedidas para la defensa frente amenazas. Esta necesidad no ha quedado únicamente relegada al ámbito militar, sino que se está extendiendo con interés creciente al ámbito civil. En parte, esto es debido al auge de los nuevos procedimientos de asistencia a la conducción (*Advanced Driver Assistance Systems, ADAS*) para vehículos autónomos, aunque existen muchas otras aplicaciones que demandan este tipo de cálculos.

Desde hace tiempo, la solución a estos y otros problemas de naturaleza electromagnética gira en torno al empleo de técnicas numéricas rigurosas (*full-wave*) de análisis electromagnético. Sin embargo, el elevado orden de magnitud de las frecuencias de operación de los equipos radar (típicamente hasta banda X, entre 8 y 12 GHz, aunque son cada vez más habituales los radares en bandas superiores, como la banda Ku, de 12 a 18 GHz), transforma los objetivos y blancos de interés, usualmente compuestos por geometrías complejas con características multiescala, en cuerpos extraordinariamente grandes en términos de tamaño eléctrico. Esto sucede por ejemplo en el cálculo de la RCS de plataformas aeronáuticas de gran complejidad como el F22 mostrado en la figura 1.1, cuyo tamaño eléctrico requiere la resolución de sistemas de ecuaciones del orden de centenares de millones de incógnitas a frecuencias radar mediante los métodos convencionales *full-wave* (que exigen entre 150 y 300 incógnitas por longitud de onda cuadrada). Lo cual es por el momento únicamente asumible mediante el uso de computadores de altas prestaciones (*High Performance Computing, HPC*), requiriendo además grandes cantidades de recursos y tiempos de computación elevados.

Las metodologías *full-wave* junto con los últimos avances en arquitecturas de computación dispo-

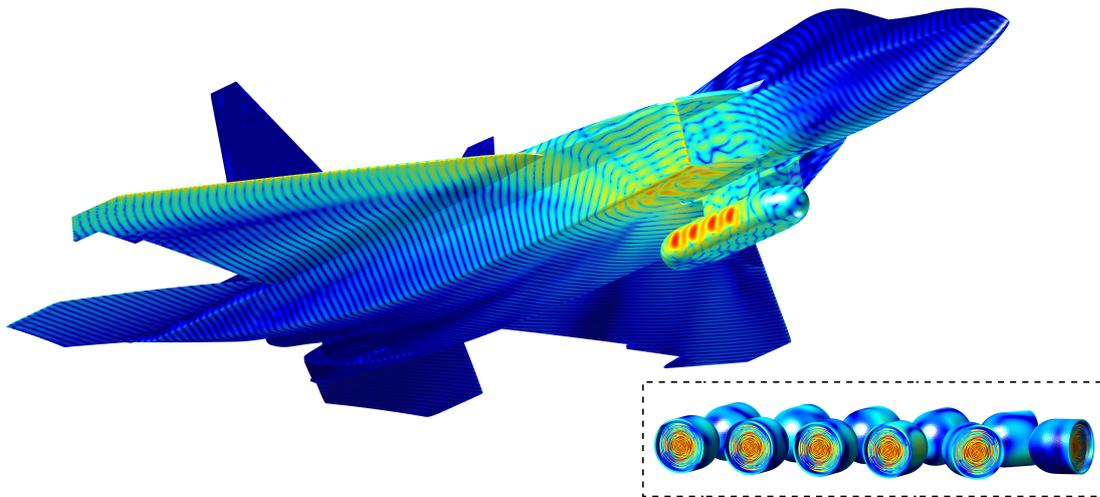


Figura 1.1: Distribución de la parte real de la densidad de corriente superficial obtenida por el método de descomposición de dominios para un avión F22, excitando un pod de 11 antenas de hélice con absorbente, analizadas por MoM y acopladas mediante MLFMA a 1 GHz.

nibles hoy en día son capaces de abordar problemas de gran tamaño como el anteriormente descrito. No obstante, esto no significa que dichas soluciones sean posibles con los recursos disponibles y, especialmente, en los tiempos de cálculo requeridos. Este es el caso de problemas de gran interés como el del cálculo de la RCS monoestática de aviones o buques en banda X, cuya resolución en servidores de cálculo de altas prestaciones requiere del orden de varias semanas de cálculo, si no meses. En estos caso, si bien estrictamente es posible su resolución, ésta no resulta una alternativa viable en el desempeño de proyectos que exigen tiempos reducidos y flexibilidad a la hora de evaluar diferentes diseños. Esto pone de manifiesto al existencia de una demanda de nuevas técnicas que permitan dar solución a estas necesidades en tiempos de cómputo razonables manteniendo a la vez una precisión suficiente.

1.2. Antecedentes y estado del arte

La literatura en el contexto de las principales aportaciones de esta Tesis, en materia de cálculo de la RCS mediante óptica física (*Physical Optics*, PO) y métodos derivados, como la óptica física iterada (*Iterative Physical Optics*, IPO), así como en el empleo de técnicas que incorporen unidades de procesamiento gráfico (*Graphics Processing Unit*, GPU), resulta extensa. El empleo de métodos asintóticos de alta frecuencia ha sido por mucho tiempo la única alternativa viable para la resolución de problemas electromagnéticos de esta envergadura. No obstante, el continuo desarrollo del hardware y, en especial, los recientes avances en trazado de rayos (*Ray Tracing*, RT) mediante el uso de GPUs, entre otros, han provocado que estos métodos hayan sufrido una importante transformación que afecta a su usabilidad y precisión. Entre estos cambios cabe destacar el empleo de nuevas técnicas y algoritmos de cómputo para realizar las tareas de trazado de rayos, que presentaban hasta hace no mucho tiempo un elevado coste computacional. Estos cálculos requieren de la detección de colisiones, determinación de visibilidades, propagaciones, así como de diversos aspectos gráficos, todos ellos de gran aplicación en el ámbito de la óptica física a problemas de elevada complejidad en cuanto a tamaño y detalles geométricos. De este modo, estas metodologías se han convertido de nuevo en una alternativa rápida y cada vez más precisa con la que abordar nuevos retos computacionales que, como decíamos anteriormente, quedan en la actualidad fuera del alcance de las técnicas *full-wave* en programas de desarrollo que exigen un elevado grado de interactividad en la etapa de diseño. Por todos estos motivos, las aportaciones en este ámbito resultan abundantes y novedosas. Se mencionan algunas de ellas a continuación, que se han

considerado más relevantes para la realización de esta Tesis Doctoral.

En primer lugar citar las recientes aportaciones al código RECOTA (*Response of COmplex Targets*), de la empresa Boeing Aerospace. Basado en la óptica física y la teoría de la difracción (*Physical Theory of Diffraction*, PTD) para modelar las contribuciones de primer orden, junto a un algoritmo de identificación de diedros que permite evaluar la respuesta a múltiples direcciones de incidencia, se aplican técnicas de *back-face culling* y *ray tracing* para determinar las ocultaciones u oclusiones de las facetas visibles por parte de otras facetas del modelo. De esta manera obtuvieron buenos resultados para un misil genérico con el que contrastaron medidas [1].

Otra implementación relevante es el programa GRECO [2] (*GRaphical Electromagnetic COmputing*), que fue en su momento la primera implementación que incorporaba GPUs en determinados cálculos de la RCS que requerían de elevados tiempos de cómputo, mejorándolos significativamente. Este hecho lo consolidó como un ejemplo temprano de lo que posteriormente se denominaría computación de propósito general en GPUs (*General Purpose Computing on Graphics Processing Units*, GPUGPU). Esta implementación se aprovecha del *z-buffer* y el *frame buffer* para realizar las ocultaciones desde el punto de vista del radar y obtener contribuciones relativas a cada punto (píxel). Esta metodología incorpora una formulación sencilla de PO a la que se incorporan las condiciones de impedancia superficial (*Impedance Boundary Condition*, IBC) con las que caracterizar materiales absorbentes a frecuencias radar (*Radar Absorbent Material*, RAM), así como (*Incremental Length Diffraction Coefficient*, ILDC) [3]. No obstante, otras versiones del mismo código incluyen una formulación híbrida de óptica geométrica (*Incremental Length Diffraction Coefficient*, GO) y PO para modelar una doble reflexión [4].

Xpatch[®] [5] es un conjunto de códigos para la predicción de la RCS basada en la técnica de trazado y reflexión de rayos (*Shooting and Bouncing Ray*, SBR), incluyendo efectos de difracción. Esta solución no solo obtiene la estimación de la RCS del blanco completo, sino que posee herramientas capaces de estimar el centro de radiación y la imagen radar. Xpatch[®] admite hibridación con métodos *full-wave*, aunque sólo para pequeños elementos o detalles geométricos.

Finalmente, y directamente relacionado con esta Tesis, cabe mencionar rcsASET [6], cuya solución está basada en la óptica física iterada para objetivos complejos. La metodología IPO fue inicialmente propuesta para problemas de dispersión en cavidades [7] y posteriormente extendida a geometrías arbitrarias [8]. Aunque IPO es un algoritmo basado en el método asintótico de alta frecuencia PO, puede observarse también desde la óptica de una solución iterada de la ecuación integral para el campo magnético (*Magnetic Field Integral Equation*, MFIE), equiparable por tanto, bajo ciertas premisas, a una metodología *full-wave*. Tal y como se desarrolla en [6], esta metodología es capaz de superar las limitaciones debidas a las múltiples reflexiones y a los límites abruptos en las regiones de sombra de las formulaciones asintóticas clásicas, mediante la aplicación de múltiples radiaciones difusas (a diferencia de [5], donde los efectos de orden superior están basados en el trazado de rayos). Los resultados obtenidos por rcsASET para blancos canónicos son buenos; no obstante, en el caso de blancos realistas se aprecia cierta pérdida de precisión con respecto a las medidas, lo que puede ser en parte atribuido a las aproximaciones consideradas. Particularmente, este código hace uso de la aproximación rápida de campo lejano (*Fast Far Field Approximation*, FaFFA) [9] para acelerar los cálculos, que resultan inconsistentes con la aplicación estricta de las reglas de sombreado inherentes al método. Además, la necesidad de muestrear las corrientes según el criterio de Nyquist conlleva un aumento en el número de incógnitas a más de 2 millones para un modelo de un tanque genérico en banda X, por lo que los tiempos de cómputo resultan prohibitivos en cualquier caso prohibitivos para la resolución de determinados problemas de interés práctico para la industria.

1.3. Métodos numéricos de análisis electromagnético

Históricamente, los avances científicos en el campo del electromagnetismo computacional (*Computational Electromagnetics, CEM*) se remontan un buen número de años atrás, con la resolución analítica de las primeras ecuaciones integrales y diferenciales obtenidas a partir de las ecuaciones de Maxwell para objetos canónicos. Ante la imposibilidad de resolver dichas ecuaciones de manera analítica para objetos más complejos, se desarrollaron técnicas numéricas capaces de evaluar las ecuaciones de manera computacional para dar solución a problemas electromagnéticos arbitrarios. Aquella ardua tarea sigue siendo una constante hoy en día, donde los esfuerzos se centran en resolver problemas cada vez mayores y más realistas con los recursos computacionales disponibles.

Existen básicamente dos grandes categorías de métodos numéricos en electromagnetismo computacional en función de las estrategias escogidas para la resolución de las ecuaciones de Maxwell, los métodos de onda completa o *full-wave* (también llamados de baja frecuencia) y los métodos asintóticos, o de alta frecuencia.

No obstante, a día de hoy sigue sin existir una solución genérica que se demuestre óptima para abordar los diferentes problemas electromagnéticos que demandan este tipo de soluciones. Esto se debe a los diversos factores que intervienen a la hora de escoger la solución adecuada o el método que mejor se adapte a las necesidades de cada problema. Sin olvidar la necesidad de adaptar la implementación de dichos métodos a los requerimientos hardware y software de los HPCs actuales. Todo ello ha hecho que surgiesen soluciones que tratan de combinar las fortalezas de diversas metodologías, en lo que ha dado en denominarse métodos híbridos. Actualmente, esta categoría permite abordar una gran cantidad de soluciones que no son fácilmente alcanzables mediante métodos *full-wave* o asintóticos puros, ya sea por tiempo de computación excesivo o por falta de precisión.

A continuación citaremos algunos de los métodos más relevantes en base a las categorías anteriormente citadas, describiendo brevemente las ventajas e inconvenientes de dichas metodologías a fin de concluir finalmente el motivo de la solución adoptada.

1.3.1. Métodos full-wave

Originalmente surgieron como soluciones a pequeños problemas electromagnéticos, capaces de modelar antenas de hilo desarrollando matemáticamente la ecuación integral para el campo eléctrico (*Electric Field Integral Equation, EFIE*) como hizo [10]. Limitados por los procedimientos computacionales de la época, quedaron relegados a una cuestión más bien teórica ante la imposibilidad de resolver las ecuaciones para objetos medianamente grandes. Hasta que en 1967, Harrington formuló por primera vez el método de los momentos (*Method of Moments, MoM*) [11], lo cual dio lugar a sucesivas variantes del método y de formulaciones como las ecuaciones integrales de superficie (*Surface Integral Equation, SIE*). Desde entonces los métodos de baja frecuencia son denominados comúnmente métodos de onda completa o *full-wave*, debido a la resolución del conjunto de las ecuaciones de Maxwell sin asumir simplificaciones o aproximaciones.

Actualmente, los avances en los métodos SIE basados en MoM han permitido el estudio de numerosos problemas reales con cada vez mayor tamaño eléctrico y complejidad. Esto se ha logrado en parte gracias al desarrollo de algoritmos de aceleración capaces de superar las limitaciones del MoM, cuyo coste computacional es de $O(N^3)$, con N el número de funciones base (incógnitas). Alguno de los algoritmos de aceleración más destacados son el método multipolar rápido (*Fast Multipole Method, FMM*) [12, 13], su versión multinivel, el *Multilevel Fast Multipole Algorithm* (MLFMA) [14-20] o su versión altamente escalable basada en la transformada rápida de Fourier (*Multilevel Fast Multipole Method-Fast Fourier Transform, MLFMA-FFT*) [21-24], este último con un coste de tan solo $O(N \log N)$.

No obstante, existen alternativas a las SIE, como puede ser el empleo de técnicas basadas en ecuaciones diferenciales. Algunos ejemplos son el método de elementos finitos (*Finite Element Method*, FEM) [25], el método de diferencias finitas (*Finite Difference Method*, FDM) [26] o el método de diferencias finitas en el dominio del tiempo (*Finite Difference in the Time Domain*, FDTD) [27], etc. Aunque mucho más sencillos en cuanto a implementación, al estar basados en la discretización de los campos, en lugar de emplear corrientes equivalentes, su principal inconveniente es que requieren de discretizaciones volumétricas en las que el número de incógnitas crece exponencialmente con el tamaño de los objetos y el dominio de cálculo, haciendo inviable o muy dificultosa la caracterización de sistemas radiantes o problemas de dispersión. Aun así, estas aproximaciones volumétricas se han vuelto populares en ciertos campos de la física, como en comunicaciones ópticas o la caracterización de materiales avanzados, probablemente debido a la disponibilidad de softwares comerciales como COMSOL [28] o CST[29]. No obstante, estas soluciones limitan el tamaño de los problemas a resolver a unas pocas longitudes de onda.

1.3.2. Métodos asintóticos

Inicialmente fueron diseñados para modelar los distintos mecanismos de dispersión, reflexión y refracción electromagnética, para lo cual se basan en aproximaciones asintóticas de las ecuaciones de Maxwell para altas frecuencias (longitudes de onda muy pequeñas en comparación con el tamaño de los dispersores y sus detalles geométricos). Estos métodos aportan soluciones rápidas, aunque de precisión limitada, que resultan especialmente atractivas en el caso de geometrías geoméricamente sencillas y de gran tamaño eléctrico, o bien para problemas tan complejos en los que esta estimación es la única aproximación viable frente a la incapacidad de ser abordados por los métodos *full-wave*. Un caso típico es la predicción de la RCS monoestática de grandes estructuras.

Las aproximaciones asintóticas se pueden dividir a su vez en dos grupos bien diferenciados: los métodos basados en el trazado de rayos, y los métodos basados en la estimación de la distribución de corrientes inducidas. Las primeras soluciones comenzaron con la óptica geométrica [30], cuya precisión se encontraba muchas veces limitada al ignorar los campos acoplados que se daban en el interior de cavidades debido a las múltiples reflexiones, así como los efectos de difracción en bordes y discontinuidades geométricas. Como solución a estos problemas, surgieron técnicas como el trazado por haz gaussiano, que fueron incorporadas en el método de expansión de rayos generalizado (*Generalized Ray Expansion*, GRE) [31-33], o la teoría geométrica de la difracción (*Geometrical Theory of Diffraction*, GTD) [34]. En el caso de la GTD, ésta fue extendida mediante la teoría uniforme de la difracción (*Uniform Theory of Diffraction*, UTD) [35], solventando los valores de campo singulares en la frontera de las regiones de sombra entre onda incidente y reflejada. No obstante, todos estos métodos seguían resultando poco apropiados para problemas con mecanismos de dispersión más complejos, como las cáusticas, o adolecían de problemas de divergencia debido a la necesidad de un trazado de rayos masivo. Los métodos basados en la estimación de la distribución de corrientes equivalentes sobre la superficie modelan el comportamiento electromagnético de los campos radiados en espacio libre por dichas corrientes de manera más precisa. Métodos asintóticos como el PO [30] obtienen la distribución de corrientes a partir de la MFIE, o en el caso de su variante IPO [7], de manera recursiva a partir de la excitación inicial dada por la aproximación de corrientes de Kirchhoff (también utilizadas por el método GRE). No obstante, estos métodos precisan de técnicas que permitan determinar las visibilidades u ocultaciones entre facetas, lo cual puede conllevar un elevado coste computacional si no se utilizan las herramientas adecuadas. Entre las diferentes alternativas para llevar a cabo estas ocultaciones, las utilizadas más habituales se basan en algoritmos de descomposición en árboles cuaternarios (*quad-tee*) [36] o la determinación mediante *z-buffering* [37].

1.3.3. Métodos híbridos

Una de las primeras incorporaciones a esta categoría fue el *High Frequency Panel Method* (HFPM) [38]. Este método daba solución a un reducido conjunto de problemas, constituidos a base de cilindros 2D, considerando funciones base constantes definidas a trozos. Aunque limitado, este método sentó las bases para otros como la ecuación integral asintótica de fase (*Integral Equation-Asymptotic Phase*, IE-AP) [39], que extendía mediante ecuaciones integrales de superficie la idea iniciada por el HFPM a superficies conductoras 2D arbitrarias para problemas de dispersión electromagnética. Fue cuestión de tiempo abordar problemas 3D mediante la combinación de algunas de las técnicas anteriormente descritas. Este fue el caso del método híbrido de óptica física (*Physical Optics Hybrid Method*, POHM) [40], combinando el MoM con el PO. De igual modo que sus predecesores 2D, el POHM dio lugar a diversas variantes de entre las que destaca el método híbrido EFIE-MFIE (*Hybrid Expansion Method*, HEM) [41]. Estas técnicas fueron sucesivamente mejoradas hasta alcanzar resultados superiores a los métodos de alta frecuencia puros, pero todavía distantes a las soluciones *full-wave* de los métodos de baja frecuencia en el caso de geometrías complejas. alguna de las más recientes, como la implementación del método híbrido del método de los momentos-óptica física (*Method of Moments—Physical Optics*, MM-PO), desarrollada en una de las tesis de este grupo [42], logró resultados excepcionales en el cálculo de parámetros precisos como el acoplo mutuo entre antenas a bordo estructuras más o menos complejas. No obstante, persisten las limitaciones a la hora de modelar determinadas regiones próximas a los elementos radiantes, debido al elevado coste computacional que conlleva el uso del MoM.

En la actualidad, estrategias conceptualmente similares a los métodos híbridos han adquirido relevancia debido a sus buenos resultados en la resolución de problemas electromagnéticos de gran tamaño. Destaca principalmente el método de descomposición de dominios (*Domain Decomposition Method*, DDM), introducido por primera vez en [43], que puede interpretarse como un preconditionador de la matriz del sistema de ecuaciones del problema, aunque conceptualmente guarda grandes semejanzas con algunos métodos de hibridación comentados previamente. El DDM se basa en la descomposición del problema completo en problemas de menor tamaño resueltos localmente de manera independiente, y posteriormente unificados mediante el cálculo de los acoplamientos mutuos entre subdominios. Esta idea permite independizar claramente los dominios que requieren de tratamientos específicos. Así, en las últimas décadas surgieron múltiples implementaciones de esquemas DDM basados en elementos finitos con excelentes resultados [44-48]. El método fue adquiriendo relevancia, presentándose implementaciones basadas en la ecuación integral, SIE (*Surface Integral Equation-Domain Decomposition Method*, SIE-DDM), con resultados realmente destacables [49, 50].

1.4. Solución adoptada y aportaciones de esta Tesis

En base a los antecedentes citados anteriormente, el objetivo de esta Tesis se centra en dar solución a problemas de radiación y dispersión electromagnéticos empleando soluciones asintóticas basadas en la metodología de IPO. Para aplicar dicha metodología, se resolverá de manera rigurosa la MFIE iterada, incorporando técnicas novedosas de trazado de rayos sobre hardware específico. El objetivo final consiste en la obtención de resultados rápidos y precisos en el cálculo de la RCS monoestática de plataformas portantes complejas de gran tamaño, para las que las técnicas *full-wave* disponibles hoy en día, como el MLFMA o el DDM, resultan insuficientes o impracticables (por sus elevados tiempos de cómputo) de cara a la ejecución sistemática y reiterada de estudios pormenorizados desde las fases de diseño. El objetivo planteado es alcanzable gracias al empleo de IPO junto con funciones de interpolación lineal de fase sobre subdominios triangulares planos (*Linear Phase Interpolation over the Triangular domains*, LPIT) [51, 52], que permiten reducir drásticamente la resolución del mallado (número de facetas) necesaria para la resolución del problema. También será determinante el desarrollo de rutinas de trazado de rayos que permitan acelerar la obtención de las diferentes matrices de visibilidades/ocultaciones necesarias en la aplicación de IPO. El conjunto de aportaciones y metodologías *quasi-full-wave* desarrolladas a lo largo de esta Tesis, conformarán una alternativa viable con la que abordar proyectos que requieran de

simulaciones intensivas, proporcionando además gran flexibilidad de cara a la realización de cambios y optimizaciones durante las fases de diseño. Todo ello al tiempo que se garantizan precisiones suficientes para respaldar los diseños alcanzados.

De manera adicional, dadas las ventajas evidentes que ofrece la metodología *multisolver* basada en DDM a la hora de abordar con gran precisión y eficiencia problemas complejos de gran tamaño eléctrico, se ha optado por incorporar las técnicas desarrolladas en esta Tesis, basadas en IPO, como un módulo (o *solver*) independiente dentro del esquema DDM. El propósito será combinar la resolución de dominios de mayor complejidad y pequeño o mediano tamaño mediante técnicas *full-wave*, con la fuerte compresión de las regiones de tamaño eléctrico extremadamente grandes y de superficies más suaves, sobre las que se aplicará el *solver* IPO. Lo anterior permitirá abordar de manera más eficiente problemas de gran tamaño eléctrico, mediante la selección inteligente de dominios geométricos que permitan aislar las características multiescala del resto de superficies eléctricamente grandes. Esta implementación requerirá de un software de detección de sectores ciegos (*blind sectors*), que también ha sido desarrollado en esta Tesis, para determinar las visibilidades entre las facetas iluminadas por las regiones IPO y los dominios *full-wave*.

En definitiva, las aportaciones mencionadas lograrán extender el rango de aplicaciones de las metodologías puramente *full-wave* más avanzadas, habilitando la solución de una amplia gama de problemas EM que a día de hoy resultan inabordables dado su excesivo coste computacional.

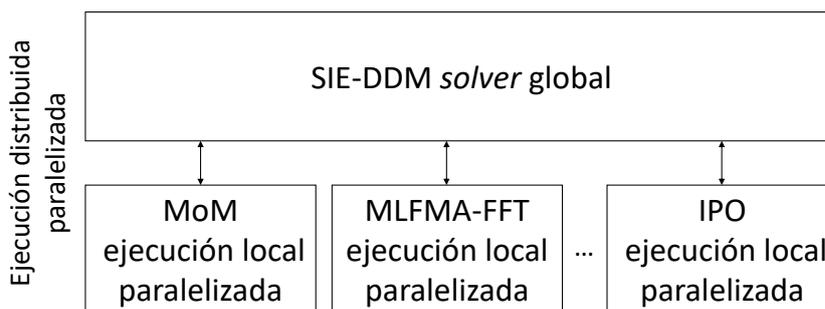


Figura 1.2: Esquema de la distribución paralelizada y deslocalizada del software de simulación electromagnética M^3 -HEMCUVE mediante DDM.

Los métodos y códigos desarrollados en esta Tesis parten de la base del programa de análisis electromagnético (*full-wave*) M^3 -HEMCUVE [53] (*Hybrid Electro Magnetic Code Universities of Vigo and Extremadura*), diseñado e implementado por el equipo de EM3Works de la Universidad de Vigo y Extremadura como un software de simulación electromagnética de propósito general. Este software utiliza la metodología distribuida *multisolver* basada en el esquema SIE-DDM [54] y los estándares de paralelización MPI (*Message Passing Interface*) y openMP, los cuales permiten distribuir la ejecución en diferentes servidores, así como paralelizar la resolución de los diferentes *solvers* de manera local, en base a los recursos de cómputo de cada máquina, tal y como muestra el esquema de la figura 1.2.

1.5. Estructura de la memoria de Tesis

Esta Tesis ha sido estructurada en cinco capítulos principales y diversas secciones, la cual comienza por este primer capítulo en el que se expone la motivación de esta Tesis y se contextualizan los antecedentes y el estado del arte. Se introducen los diferentes métodos numéricos de análisis electromagnéticos distinguiendo entre clases y se deduce la solución adoptada y las aportaciones y objetivos de esta Tesis.

En el segundo capítulo se detallan los métodos de simulación electromagnética *full-wave* (MoM,

MLFMA y DDM) implementados por el grupo de investigación en el que se ha realizado esta Tesis. Se desarrolla la formulación SIE a partir de las ecuaciones de Maxwell y las condiciones de contorno que unificarán por completo el resto de metodologías. Por último, se presentan ejemplos de aplicación para cada uno de ellos, siendo desarrollados a lo largo de esta Tesis y en conjunto a los proyectos y colaboraciones que tuvieron lugar durante la misma.

En el tercer capítulo se desarrolla de igual forma el método basado en IPO en que se centra esta Tesis, junto con las aportaciones novedosas que se incluyen a esta metodología en materia de trazado de rayos y desarrollo de herramientas destinadas a la detección de blancos y sectores ciegos (*blind sectors*).

En el cuarto capítulo se dedica un apartado propio a resultados numéricos y ejemplos de aplicación, entre las que se destacan tres principales aportaciones: la aplicación de la metodología IPO al cálculo de la RCS monoestática de un buque real, la obtención de diagramas de cobertura óptica y detección de blancos mediante técnicas de trazado de rayos y la incorporación de la metodología IPO al esquema SIE-DDM.

Por último, se incluye un capítulo de conclusiones a fin de resumir las principales aportaciones de esta Tesis, los retos más destacables, las limitaciones a la hora de llevar a cabo dichas aportaciones y las posibles líneas futuras que permitirían continuar el desarrollo de la solución implementada, a fin de continuar mejorando su carácter multipropósito en términos de tiempo, precisión y recursos.

CAPÍTULO 2 **Métodos de electromagnetismo computacional**

Contenido

2.1. Ecuación integral superficial (SIE)	9
2.2. Método de los momentos (MoM)	12
2.2.1. Procedimiento de expansión	13
2.2.2. Procedimiento de ponderación	14
2.2.3. Funciones base	15
2.2.4. Ejemplo de aplicación	16
2.3. Multilevel fast multipole algorithm (MLFMA)	17
2.3.1. Ejemplo de aplicación	19
2.4. Método de descomposición de dominios (DDM)	20
2.4.1. Condiciones de transmisión	22
2.4.2. Acoplo por radiación	23
2.4.3. Ejemplo de aplicación	24

En este capítulo abordaremos algunos de los métodos *full-wave* más comunes en el ámbito CEM, como son el MoM, el MLFMA y el DDM. Estos métodos constituyen el software de simulación electromagnética M3 del grupo de investigación, del que se ha partido para el desarrollo de los métodos incluidos en esta Tesis. Plantearemos brevemente en este capítulo cada uno de estos métodos y mencionaremos algunas de las características propias de su implementación en M3. Para una visión más detallada de los mismos se remite a las tesis doctorales [42] en el caso del MoM, en [55] en el caso del MLFMA y [56] en el caso del DDM.

Para comenzar se dedicará en primer lugar una sección a la revisión de la formulación de las SIEs, que constituyen el nexo común entre los diferentes *solvers* y permiten unificar la descripción del problema completo bajo una misma formulación, de manera más sencilla y compacta. Esto reportará una descripción coherente y modular, lo cual en última instancia facilitará la incorporación de nuevas metodologías.

2.1. Ecuación integral superficial (SIE)

En esta sección desarrollaremos brevemente la formulación integral superficial para un problema arbitrario, partiendo de las ecuaciones de Maxwell en su forma integral y las condiciones de contorno. En concreto nos centraremos en la ecuación integral combinada de campo para las corrientes eléctricas y magnéticas (*Electric and Magnetic Combined Field Integral Equation*, JMCFIE) [57], válida para objetos

combinando materiales conductores y dieléctricos homogéneos u homogéneos a trozos. Esta es la formulación que se ha utilizado en el desarrollo de esta Tesis.

Consideremos un objeto homogéneo a trozos cuyas regiones de diferentes materiales se denotan por R_i , con $i = 1 \dots M$, siendo M el número de regiones, tal como se muestra en la figura (2.1). Se asume de aquí en adelante una variación temporal armónica de tipo $e^{j\omega t}$, que será omitida de la formulación por comodidad. Las propiedades de los materiales de cada región quedan definidas por la permitividad dieléctrica compleja $\varepsilon_i = \varepsilon_{r_i} \varepsilon_0$ y la permeabilidad magnética, también compleja, $\mu_i = \mu_{r_i} \mu_0$, donde ε_{r_i} y μ_{r_i} son las constantes de permitividad y permeabilidad relativas de la región R_i , respectivamente, mientras que ε_0 y μ_0 son las constantes en el vacío. Denotaremos con S_{ij} (o S_{ji}) la superficie (o interfaz) entre las regiones R_i y R_j . \hat{n}_{ij} es el vector unitario normal a la superficie S_{ij} y que apunta hacia R_i , donde $\hat{n}_{ij} = -\hat{n}_{ji}$. En la región R_i , las ecuaciones de campo eléctrico (EFIE) y campo magnético (MFIE) pueden formularse de dos maneras distintas, dependiendo de la proyección realizada de los campos sobre la superficie. De tal modo, definimos las corrientes eléctricas y magnéticas equivalentes $\mathbf{J}_{ij}(\mathbf{r}')$ y $\mathbf{M}_{ij}(\mathbf{r}')$ sobre las interfaz S_{ij} que envuelve R_i , las cuales están relacionadas con los campos en esta región respectivamente mediante:

$$\mathbf{J}_{ij}(\mathbf{r}') = \hat{n}_{ij} \times \mathbf{H}_i(\mathbf{r}'), \quad (2.1)$$

$$\mathbf{M}_{ij}(\mathbf{r}') = -\hat{n}_{ij} \times \mathbf{E}_i(\mathbf{r}'). \quad (2.2)$$

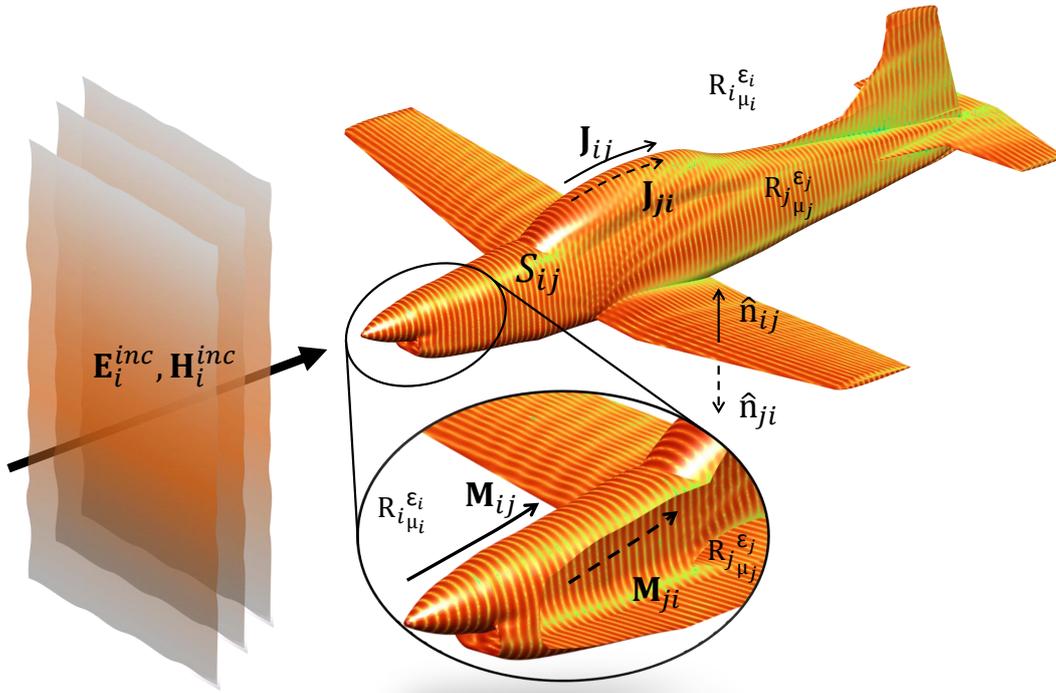


Figura 2.1: Representación de un problema equivalente para una geometría arbitraria incidente por una onda plana.

Según el principio de equivalencia aplicado al conjunto de campos eléctricos y magnéticos totales $\mathbf{E}_i(\mathbf{r}')$ y $\mathbf{H}_i(\mathbf{r}')$ en R_i , obtenemos las ecuaciones tangenciales (T) EFIE y MFIE como:

$$\text{T-EFIE}_i: \sum_{j' \in Gr_i} (\eta_i \mathcal{L}_i(\mathbf{J}_{ij'}) - \mathcal{K}_i(\mathbf{M}_{ij'}))_{tan} + \frac{1}{2} \hat{n}_{ij} \times \mathbf{M}_{ij}(\mathbf{r}) = (\mathbf{E}_i^{inc}(\mathbf{r}))_{tan}, \quad \mathbf{r} \in S_{ij} \forall j \in Gr_i \quad (2.3)$$

$$\text{T-MFIE}_i: \sum_{j' \in Gr_i} (\mathcal{K}_i(\mathbf{J}_{ij'}) + \frac{1}{\eta_i} \mathcal{L}_i(\mathbf{M}_{ij'}))_{tan} - \frac{1}{2} \hat{n}_{ij} \times \mathbf{J}_{ij}(\mathbf{r}) = (\mathbf{H}_i^{inc}(\mathbf{r}))_{tan}, \quad \mathbf{r} \in S_{ij} \forall j \in Gr_i. \quad (2.4)$$

De la misma forma, las ecuaciones rotadas o normales (N) en S_{ij} en R_i se pueden escribir como:

$$\text{N-EFIE}_i : \hat{n}_{ij} \times \sum_{j' \in Gr_i} (\eta_i \mathcal{L}_i(\mathbf{J}_{ij'}) - \mathcal{X}_i(\mathbf{M}_{ij'})) - \frac{1}{2} \mathbf{M}_{ij}(\mathbf{r}) = \hat{n}_{ij} \times \mathbf{E}_i^{inc}(\mathbf{r}), \quad \mathbf{r} \in S_{ij} \forall j \in Gr_i, \quad (2.5)$$

$$\text{N-MFIE}_i : \hat{n}_{ij} \times \sum_{j' \in Gr_i} (\mathcal{X}_i(\mathbf{J}_{ij'}) + \frac{1}{\eta_i} \mathcal{L}_i(\mathbf{M}_{ij'})) + \frac{1}{2} \mathbf{J}_{ij}(\mathbf{r}) = \hat{n}_{ij} \times \mathbf{H}_i^{inc}(\mathbf{r}), \quad \mathbf{r} \in S_{ij} \forall j \in Gr_i, \quad (2.6)$$

donde Gr_i denota el conjunto de índices de las regiones que envuelven la región R_i , η_i es la impedancia intrínseca en el medio R_i , $\mathbf{E}_i^{inc}(\mathbf{r})$ y $\mathbf{H}_i^{inc}(\mathbf{r})$ son los campos incidentes debido a las fuentes ubicadas dentro de R_i y los operadores integro-diferenciales \mathcal{L} y \mathcal{X} son definidos como:

$$\mathcal{L}_i(\mathbf{X}_i) = jk_i \int_S \mathbf{X}_i(\mathbf{r}') G_i(\mathbf{r}, \mathbf{r}') dS' - \frac{1}{jk_i} \nabla \int \nabla' \cdot \mathbf{X}_i(\mathbf{r}') G_i(\mathbf{r}, \mathbf{r}') dS', \quad (2.7)$$

$$\mathcal{X}_i(\mathbf{X}_i) = PV \int_S \mathbf{X}_i(\mathbf{r}') \times \nabla G_i(\mathbf{r}, \mathbf{r}') dS', \quad (2.8)$$

con \mathbf{r} los puntos de observación que se aproximan a S_{ij} desde el interior de la región R_i y $\mathbf{r}' \in S'_{ij}$ los puntos fuente en los límites que envuelven la región R_i . ∇' denota la divergencia respecto a los puntos fuente. PV denota el valor principal de la integral (2.8), k_i es el número de onda en R_i , y definimos:

$$G_i(\mathbf{r}, \mathbf{r}') = \frac{e^{-jk_i|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|}, \quad (2.9)$$

como la función homogénea de Green en la región R_i . El conjunto de ecuaciones de (2.3) a (2.6) aplican a todas las interfaces $S_{ij'}$ que rodean a R_i (incluido S_{ij}). Llegados a este punto, podemos aplicar el procedimiento mencionado en [57], obteniendo un conjunto de formulaciones SIE estables y testadas. A partir de las cuales, podemos combinar dichas ecuaciones integrales para deducir dos nuevas ecuaciones para el campo combinado en S_{ij} : una para las corrientes eléctricas equivalentes (*J-Combined Field Integral Equation*, J-CFIE) y otra para las corrientes magnéticas equivalentes (*M-Combined Field Integral Equation*, M-CFIE), definidas respectivamente como:

$$\text{J-CFIE}_i : a_i \frac{1}{\eta_i} \text{T-EFIE}_i + b_i \text{N-MFIE}_i, \quad (2.10)$$

$$\text{M-CFIE}_i : -c_i \text{N-EFIE}_i + d_i \eta_i \text{T-MFIE}_i, \quad (2.11)$$

donde a_i , b_i , c_i y d_i son coeficientes complejos de ponderación. Ambas ecuaciones son muy utilizadas para determinar las corrientes eléctricas y magnéticas, ya que están definidas en el subespacio vectorial al que pertenecen éstas. Las corrientes eléctricas están bien ponderadas en la J-CFIE, mientras que las corrientes magnéticas están bien ponderadas en la M-CFIE. El último paso es combinar las anteriores ecuaciones para las regiones R_i y R_j , a fin de obtener una única ecuación de cada tipo para cada superficie S_{ij} de la forma:

$$\text{J-CFIE}_{ij} : \text{J-CFIE}_i + \text{J-CFIE}_j, \quad (2.12)$$

$$\text{M-CFIE}_{ij} : -\text{M-CFIE}_i + \text{M-CFIE}_j, \quad (2.13)$$

definiendo de este modo las ecuaciones integrales combinadas de campo de la formulación JMCFIE.

En el caso de un conductor eléctrico perfecto (*Perfect Electric Conductor*, PEC), rodeado o en contacto con múltiples objetos penetrables, podemos simplificar la formulación JMCFIE eliminando la ecuación (2.11) y escogiendo correctamente los parámetros de (2.10), derivando de este modo la CFIE convencional para PEC:

$$\text{CFIE}_i : \alpha \text{T-EFIE}_i + (1 - \alpha) \text{N-MFIE}_i, \quad (2.14)$$

donde $0 < \alpha < 1$ pondera la contribución de las ecuaciones T-EFIE y N-MFIE. Dado que el campo eléctrico tangencial se desvanece en superficies PEC, las corrientes magnéticas equivalentes también se desvanecen en las interfaces y uniones que involucran una o más superficies PEC. Por tanto, la EFIE resulta apropiada para evaluar superficies abiertas PEC mediante el ajuste $\alpha = 1$. Por otra parte, en caso de estar en presencia de diferentes materiales en contacto, deberemos incluir explícitamente en la formulación las uniones entre superficies de tres o más regiones dieléctricas para hacer cumplir las condiciones de contorno.

2.2. Método de los momentos (MoM)

Una vez formuladas las SIEs que modelarán analíticamente el problema electromagnético estaremos en condiciones de abordar el problema numéricamente. Para ello comenzaremos por el método por excelencia en el ámbito CEM, el método de los momentos. En esta sección tan solo resumiremos brevemente los aspectos más relevantes de este método a fin de introducir la terminología necesaria para desarrollar las aportaciones recogidas en esta Tesis.

La formulación SIE desarrollada en este apartado corresponde a la T-EFIE (comúnmente denominada sencillamente EFIE), válida para superficies abiertas y cerradas, y que a efectos de desarrollo matemático permite obtener un sistema de ecuaciones determinado a partir de la denominada matriz de impedancias. En el caso de la MFIE y CFIE la formulación es igualmente aplicable, tal como se puede ver por ejemplo en [42].

Para el desarrollo del método partiremos definiendo un operador lineal genérico $\mathcal{L}(\cdot)$ como:

$$\mathcal{L}(\mathbf{f}) = \mathbf{g}, \quad (2.15)$$

donde $\mathcal{L}(\cdot)$ define una transformación del espacio vectorial \mathbf{f} al espacio vectorial \mathbf{g} . Estas últimas son funciones vectoriales complejas pertenecientes al rango de $\mathcal{L}(\cdot)$, donde \mathbf{f} es la función vectorial desconocida del problema y \mathbf{g} una función conocida. Dichas funciones se definen sobre dominios en el espacio, líneas, superficies o volúmenes. En este caso y para el objetivo de esta Tesis tan solo se desarrollará para superficies.

Proseguimos con el procedimiento para la función \mathbf{f} que llamaremos de *expansión* y que desarrollaremos algo más en el subapartado 2.2.1 para no interrumpir la formulación del método. El cual consiste en expandir las funciones desconocidas \mathbf{f} en una serie de funciones conocidas $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_n$, multiplicadas por una serie de coeficientes constantes desconocidos $\alpha_1, \alpha_2, \dots, \alpha_n$, como se muestra en:

$$\mathbf{f} = \sum_{n=1}^{\infty} \alpha_n \mathbf{f}_n \quad \Rightarrow \quad \mathbf{f} \simeq \sum_{n=1}^N \alpha_n \mathbf{f}_n. \quad (2.16)$$

La función \mathbf{f} modela la distribución de corrientes aproximada sobre un dominio del espacio, que debe ser truncada en una serie de N funciones bases infinitesimales para ser computacionalmente resuelta. Estas nuevas funciones base \mathbf{f}_n pertenecen al espacio vectorial de $\mathcal{L}(\cdot)$ y caracterizan las diversas corrientes inducidas sobre la superficie en una región infinitesimal. En el desarrollo de esta Tesis y en la implementación de esta metodología utilizaremos las comúnmente conocidas funciones base de *Rao-Wilton-Glisson* (RWG) [58], algo más extendidas en el subapartado 2.2.3.

A continuación, incluiremos la aproximación de la ecuación (2.16) en (2.15) y aplicaremos la propiedad lineal del operador $\mathcal{L}(\cdot)$ para obtener la expresión:

$$\mathbf{g} \simeq \sum_{n=1}^N \alpha_n \mathcal{L}(\mathbf{f}_n). \quad (2.17)$$

Proseguiremos aplicando un proceso de *ponderación*, extendido en el subapartado 2.2.2, mediante un producto escalar interno denotado por $\langle \cdot, \cdot \rangle$, de una serie de M funciones $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n$ perteneciente al rango de $\mathcal{L}(\cdot)$, para obtener un sistema de M ecuaciones lineales como el descrito en:

$$\langle \mathbf{w}_m, \mathbf{g} \rangle \simeq \sum_{n=1}^N \alpha_n \langle \mathbf{w}_m, \mathcal{L}(\mathbf{f}_n) \rangle. \quad (2.18)$$

Llegados a este punto, es posible utilizar el método de Galerkin para simplificar elementos, considerando las mismas funciones de expansión y ponderado ($\mathbf{f}_n \equiv \mathbf{w}_n$), utilizando las RWGs. De igual manera, igualamos N y M para obtener una única solución del sistema de ecuaciones. Estas consideración suponen múltiples ventajas en términos de computo, almacenamiento y tiempo.

Una vez obtenida la ecuación (2.18) procedemos a desarrollar el característico producto matriz vector (*Matrix Vector Product*, MVP) propio del método de los momentos, junto al desglose de los elementos que lo componen, dispuestos en:

$$\mathbf{Z} \cdot \mathbf{I} = \mathbf{V}, \quad (2.19)$$

$$\mathbf{Z} = \begin{bmatrix} \langle \mathbf{w}_1, \mathcal{L}(\mathbf{f}_1) \rangle & \langle \mathbf{w}_1, \mathcal{L}(\mathbf{f}_2) \rangle & \cdots & \langle \mathbf{w}_1, \mathcal{L}(\mathbf{f}_N) \rangle \\ \langle \mathbf{w}_2, \mathcal{L}(\mathbf{f}_1) \rangle & \langle \mathbf{w}_2, \mathcal{L}(\mathbf{f}_2) \rangle & \cdots & \langle \mathbf{w}_2, \mathcal{L}(\mathbf{f}_N) \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{w}_N, \mathcal{L}(\mathbf{f}_1) \rangle & \langle \mathbf{w}_N, \mathcal{L}(\mathbf{f}_2) \rangle & \cdots & \langle \mathbf{w}_N, \mathcal{L}(\mathbf{f}_N) \rangle \end{bmatrix},$$

$$\mathbf{I} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} \langle \mathbf{w}_1, \mathbf{g} \rangle \\ \langle \mathbf{w}_2, \mathbf{g} \rangle \\ \vdots \\ \langle \mathbf{w}_N, \mathbf{g} \rangle \end{bmatrix}.$$

Detallado de manera más extensa en [42], el vector columna \mathbf{I} corresponde al conjunto de variables (incógnitas) del sistema, es decir, el vector de coeficientes de la distribución de corrientes electromagnéticas de la función \mathbf{f} . A su vez, la matriz \mathbf{Z} se define como la matriz de impedancias generalizada o de acoplos del sistema. \mathbf{V} es el vector de excitaciones que contiene los coeficientes de la función \mathbf{g} de campo incidente. De esta manera, se define un sistema de ecuaciones compatible determinado que contempla la formulación SIE junto con las condiciones de contorno del problema.

Este sistema es posible de resolver mediante diferentes técnicas como invertir la matriz, métodos iterativos, ... En el caso del MoM, el procedimiento numérico se basa en la inversión de la matriz de impedancias o del sistema \mathbf{Z} , factorizando de la forma:

$$\mathbf{I} = \mathbf{Z}^{-1} \cdot \mathbf{V}, \quad (2.20)$$

despejando el vector de incógnitas del sistema matricial y evaluando por completo las ecuaciones de Maxwell en un proceso *full-wave*.

2.2.1. Procedimiento de expansión

El procedimiento de expansión, descrito a continuación de manera genérica, consiste en descomponer la densidad de corriente superficial \mathbf{J} en un sumatorio de funciones denominadas funciones base \mathbf{f}_n , conocidas y que modelan adecuadamente la distribución de corriente y la distribución de carga asociada, multiplicadas por coeficientes complejos desconocidos. Las funciones empleadas se detallan en el subapartado 2.2.3. De esta forma, se tiene la siguiente aproximación:

$$\mathbf{J}(\mathbf{r}) \simeq \sum_{n=1}^N I_n \mathbf{f}_n(\mathbf{r}), \quad (2.21)$$

donde N es el número total de funciones base e I_n los coeficientes incógnita del problema. Teniendo en cuenta el desarrollo en [42], la expresión (2.21) puede llegar a escribirse en términos de la componente tangencial del campo eléctrico incidente \mathbf{E}_{tan}^i y el operador \mathcal{L} tal que:

$$-\mathbf{E}_{tan}^i \simeq \sum_{n=1}^N I_n \mathcal{L}(\mathbf{f}_n)_{tan}. \quad (2.22)$$

El operador \mathcal{L} descrito en (2.7), puede a su vez ser descrito como:

$$\mathcal{L} = -j\omega\mathbf{A} - \nabla\Phi, \quad (2.23)$$

donde \mathbf{A} y Φ son el potencial vector y el potencial escalar respectivamente, descritos como:

$$\mathbf{A}(\mathbf{r}) = \frac{\mu}{4\pi} \int_S \mathbf{J} G(\mathbf{r}, \mathbf{r}') dS', \quad (2.24)$$

$$\Phi(\mathbf{r}) = j \frac{1}{4\pi\omega\epsilon} \int_S \nabla'_s \cdot \mathbf{J} G(\mathbf{r}, \mathbf{r}') dS', \quad (2.25)$$

donde ∇'_s representa la divergencia superficial con respecto al dominio de integración (fuente) y $G(\mathbf{r}, \mathbf{r}')$ la función de Green en espacio libre, que describimos en (2.9). Esto permite reescribir la expresión (2.22) como:

$$\mathbf{E}_{tan}^i \simeq \sum_{n=1}^N I_n (j\omega \mathbf{A}_n + \nabla \Phi_n)_{tan}, \quad (2.26)$$

donde \mathbf{A}_n y Φ_n ahora representarán a los potenciales vector y escalar parciales, correspondientes a la n -ésima base superficial, tal y como se describe respectivamente en:

$$\mathbf{A}_n(\mathbf{r}) = \frac{\mu}{4\pi} \int_{S_n} \mathbf{f}_n G(\mathbf{r}, \mathbf{r}') dS', \quad (2.27)$$

$$\Phi_n(\mathbf{r}) = j \frac{1}{4\pi\omega\epsilon} \int_{S_n} \nabla'_s \cdot \mathbf{f}_n G(\mathbf{r}, \mathbf{r}') dS'. \quad (2.28)$$

2.2.2. Procedimiento de ponderación

Para calcular los coeficientes de la expansión de corrientes, la ecuación (2.24) ha de ser evaluada en N regiones sobre la superficie S . Para ello, se definen un conjunto de funciones de ponderación \mathbf{w}_m , con $m = 1, \dots, N$, en el rango del operador $\mathcal{L}(\cdot)$ a través del producto simétrico:

$$\langle \mathbf{f}, \mathbf{w} \rangle = \int_S \mathbf{f} \cdot \mathbf{w} dS. \quad (2.29)$$

La definición de este producto simétrico permite obtener un sistema de ecuaciones de N ecuaciones lineales de la forma:

$$\langle \mathbf{E}^i, \mathbf{w}_m \rangle \simeq \sum_{n=1}^N I_n \langle -\mathcal{L}(\mathbf{f}_n), \mathbf{w}_m \rangle, \quad (2.30)$$

el cual puede ser desarrollado como:

$$\langle \mathbf{E}^i, \mathbf{w}_m \rangle \simeq \sum_{n=1}^N I_n (j\omega A_{mn} + \Phi_{mn}), \quad (2.31)$$

donde, por la definición del producto simétrico:

$$A_{mn} = \int_{S_m} \mathbf{A}_n \cdot \mathbf{w}_m dS, \quad (2.32)$$

$$\Phi_{mn} = \int_{S_m} \nabla \Phi_n \cdot \mathbf{w}_m dS. \quad (2.33)$$

Este sistema de ecuaciones lineales puede expresarse en la forma matricial de (2.19) típica del MoM, donde los elementos de la matriz de impedancias \mathbf{Z} vienen dados por:

$$Z_{mn} = \langle -\mathcal{L}(\mathbf{f}_n), \mathbf{w}_m \rangle = j\omega A_{mn} + \Phi_{mn}, \quad (2.34)$$

y donde el vector \mathbf{I} contiene los coeficientes de la expansión de la corriente, I_n . El vector de excitaciones \mathbf{V} viene dado por:

$$\mathbf{V} = \langle \mathbf{E}^i, \mathbf{w}_m \rangle. \quad (2.35)$$

Una vez se obtenga el vector de incógnitas \mathbf{I} , se dispondrá de una aproximación de la densidad de corriente sobre la superficie que podrá emplearse en el cálculo de cualquier otro parámetro de interés como diagramas de radiación, impedancias de entrada, sección radar, etc.

2.2.3. Funciones base

En esta sección se describen brevemente las funciones base (y test en el caso de aplicar el esquema de ponderación de Galerkin [11]) más ampliamente utilizadas en la comunidad CEM, dada su versatilidad y propiedades que las hacen especialmente adecuadas para el MoM en electromagnetismo. Se trata de las funciones base RWG, definidas sobre parejas de triángulos conectados mediante un lado en común. Estas bases se definen sobre mallados conformados, en los que se garantiza la conectividad entre facetas triangulares adyacentes, lo que en última instancia garantizará la continuidad de las corrientes equivalentes entre elementos. Estas corrientes eléctricas y magnéticas superficiales fluyen de un triángulo a otro a través de su lado común según:

$$\mathbf{f}_n = \begin{cases} \frac{l_n}{2A_n^+} \rho_n^+, & \text{si } \mathbf{r} \text{ en } T_n^+, \\ \frac{l_n}{2A_n^-} \rho_n^-, & \text{si } \mathbf{r} \text{ en } T_n^-, \\ 0, & \text{en otro caso.} \end{cases} \quad (2.36)$$

Como podemos ver en la figura 2.2, las funciones base RWG están definidas sobre una superficie en una región del espacio compuesta por dos triángulos T_n^+ y T_n^- de áreas A_n^+ y A_n^- respectivamente. ρ_n^\pm es el vector de posición de los puntos de la base con respecto al sistema local de coordenadas cilíndricas, centrado en el vértice libre del triángulo T_n^\pm , y l_n es la longitud del lado común de los dos triángulos que forman la base, siendo el flujo de la corriente del triángulo T_n^+ a T_n^- .

Del desarrollo planteado en [55], basado en [42], se deducen las siguientes propiedades relacionadas con la ecuación (2.36) para la representación de la densidad de corrientes.

- La componente normal se anula en todo el borde exterior de la función, de modo que no se acumulan líneas de carga en su frontera.
- La componente normal en el lado común es constante y de valor unitario en todo el lado, debido a los factores de normalización $l_n/2A_n^\pm$, con lo que se garantiza que tampoco existe acumulación de carga en el lado común a ambos triángulos.
- La divergencia de la función base (directamente relacionada con la densidad de carga) se puede obtener analíticamente según:

$$\nabla_s \cdot \mathbf{f}_n^\pm(\mathbf{r}) = \pm \frac{l_n}{A_n^\pm}, \quad \text{si } \mathbf{r} \text{ en } T_n^\pm. \quad (2.37)$$

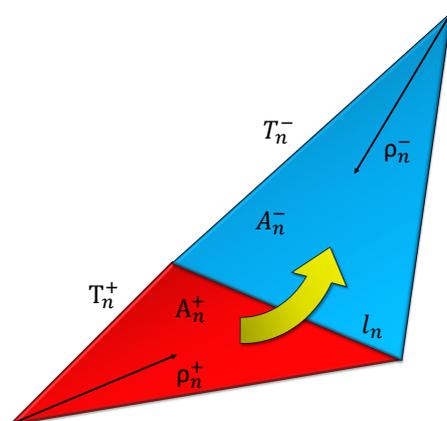


Figura 2.2: Parámetros geométricos de una función base superficial RWG.

2.2.4. Ejemplo de aplicación

En este apartado, tras haber detallado a grandes rasgos los pormenores de esta metodología, trataremos una de las principales aplicaciones de este método en materia de diseño y modelado de antenas. Este método es especialmente utilizado en este tipo de aplicaciones debido a su precisión analizando aspectos fundamentales como el cálculo de la impedancia de entrada o el acoplo mutuo entre antenas, incluyendo la presencia de diferentes materiales y caracterizando los procesos físicos que suceden en el entorno próximo al punto de alimentación. Esto permite obtener resultados muy precisos teniendo en cuenta los pequeños detalles susceptibles de alterar parámetros tan sensibles como los mencionados anteriormente.

El ejemplo propuesto en la figura 2.3 consiste en la caracterización de una antena de patrón espiral sobre una cavidad PEC con grosor, rellena de un absorbente dieléctrico. Este ejemplo plantea un caso completo del modelado de antenas que involucren diferentes materiales en contactos, elementos radiantes suspendidos y una configuración SIE que contemple diversas formulaciones analizadas por MoM. En este análisis de la distribución de corrientes superficiales equivalentes en espacio libre se han considerado dos medios: el medio radiado de permitividad y permeabilidad relativa $\epsilon_r = \mu_r = 1$ (aire) y el medio absorbente con pérdidas dadas por su constante dieléctrica relativa, $\epsilon_r = \mu_r = 4,97 - 10i$.

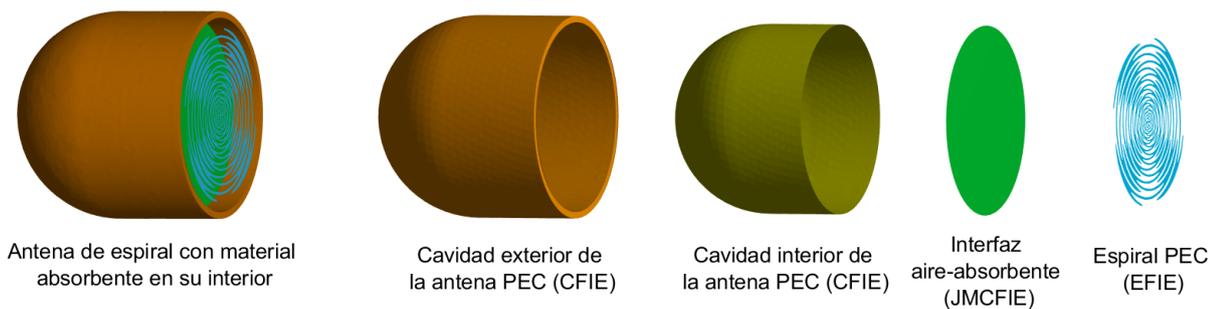


Figura 2.3: Diseño de una antena de patrón espiral sobre una cavidad PEC con grosor, rellena de un absorbente dieléctrico.

La antena diseñada se encuentra optimizada para operar en la banda C de comunicaciones, en torno a 6 GHz. Típicamente utilizada en aplicaciones de comunicaciones satelitales o radares meteorológicos. Esta antena tiene unas dimensiones totales de $0.2 \times 0.2 \times 0.2$ m equivalentes a 4λ a la frecuencia de operación. La implementación del código permite la configuración o asignación de diversas formulaciones dispuestas en la figura 2.3, para la cual se ha escogido la configuración más apropiada en bases a las características del problema. En el caso de la cavidad PEC con grosor, resulta apropiada la formulación CFIE ($\alpha = 0,5$). En el caso de la interfaz dieléctrica, aplicaremos la formulación JMCFIE, basada en corrientes equivalentes tanto eléctricas como magnéticas. Por último, la formulación EFIE ($\alpha = 1$) se aplica para la espiral constituida de material PEC infinitamente delgado (superficies abiertas), mostrada en detalle en la figura 2.4.

El resultado numérico del análisis electromagnético por MoM postprocesado y dispuesto en la figura 2.5 demuestra la capacidad del método para resolver sistemas complejos en presencia de factores multiescala de manera precisa. Esta figura representa la parte real de la distribución de corrientes eléctricas sobre la superficie en el medio *aire*. El patrón intrincado de forma sinuosa de la antena ha requerido de una discretización de $\lambda/100$ con respecto a $\lambda/10$ utilizado en el resto de la antena para modelar los pequeños detalles del dibujo, incrementando el número de grados de libertad totales del sistema hasta alcanzar las 41758 incógnitas. Esta simulación requirió de un total de 5.55 minutos para resolver el problema completo. La máquina que se utilizó para el análisis de este problema fue una AMD EPYC 7763 64-Core Processor.

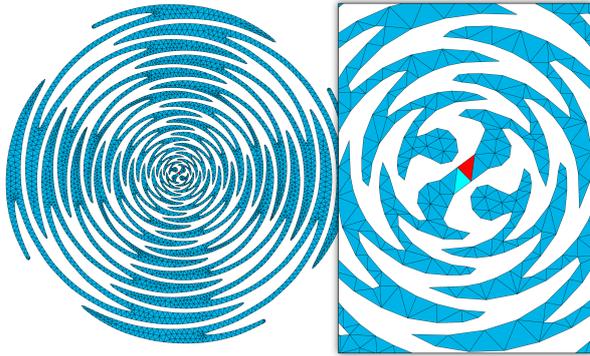


Figura 2.4: Excitación delta-gap en la unión entre los dos patrones espirales suspendidos en el aire.

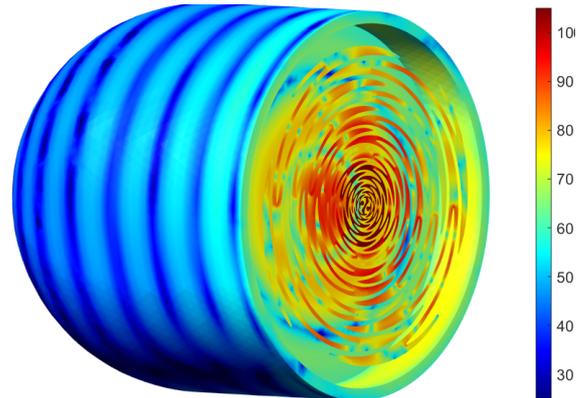


Figura 2.5: Parte real de la distribución de corrientes eléctricas superficiales.

2.3. Multilevel fast multipole algorithm (MLFMA)

El *multilevel fast multipole algorithm* surge como un algoritmo de aceleración proporcionando una alternativa viable a la resolución de problemas de elevado tamaño eléctrico y el consecuente coste computacional que implican los métodos directos como el MoM. Su origen se remonta al FMM, introducido por Rokhlin [59] en el campo de la astronomía y posteriormente adaptado al campo del electromagnetismo en [60]. Se trata de un algoritmo basado en la expansión multipolar de la función de Green en espacio libre y que requiere de una estructura jerárquica que divide la geometría total en conjuntos de $O(N^{1/2})$ funciones bases por grupo o celda.

La estructura jerárquica del FMM divide la geometría en celdas mediante *octrees* [61]. Se trata de celdas cúbicas cuyo objetivo reside en simplificar la interacción entre funciones base/test, tal y como muestra la figura 2.6 para un ejemplo con un sólo nivel.

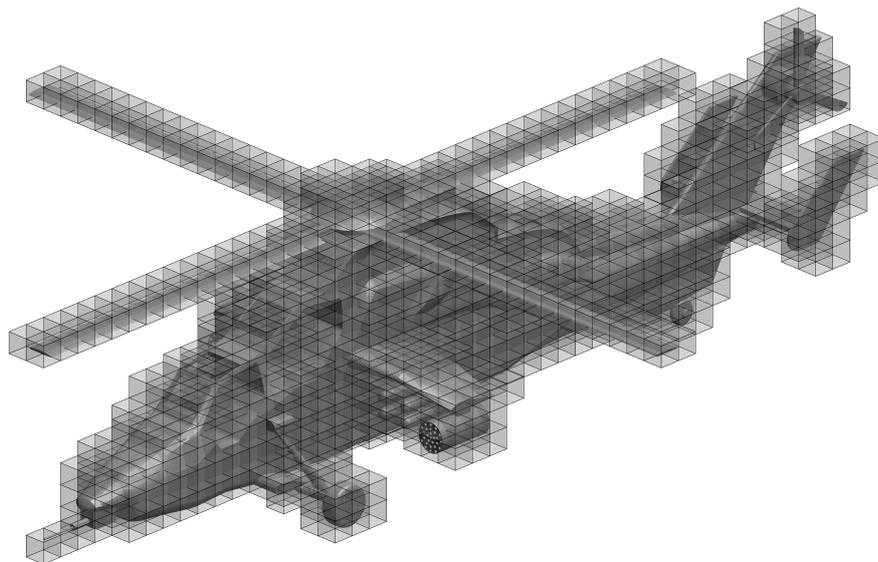


Figura 2.6: División geométrica mediante *octrees*.

En el caso multinivel, la descomposición se organiza mediante una estructura en forma de árbol, como muestra la figura 2.7, en la que los niveles superiores se construyen de forma recursiva en

base a las relaciones jerárquicas entre celdas que establece el entrelazado de bits. Cada celda queda inequívocamente identificada mediante la representación binaria obtenida a partir de los centros de cada una de las N funciones bases. Esta estrategia de división multinivel y asignación de funciones base a celdas resulta altamente eficiente, evitando además la formación de grupos vacíos.

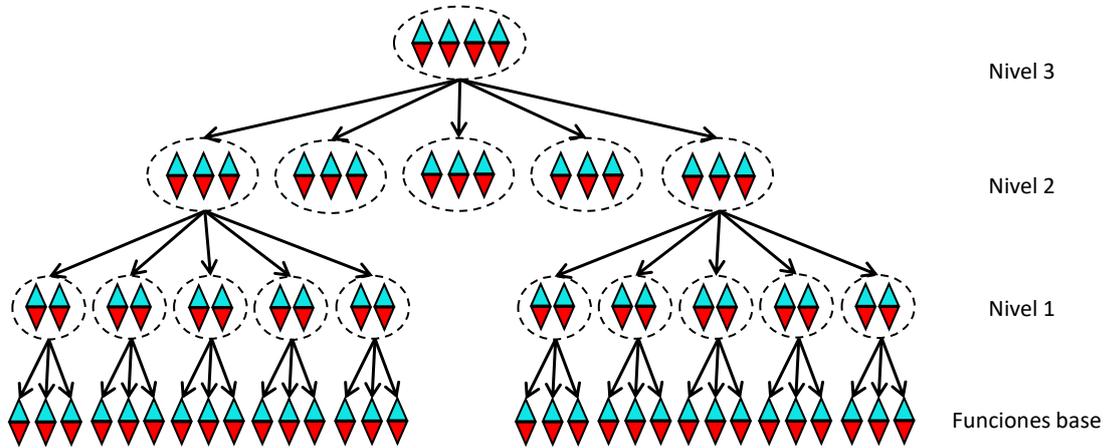


Figura 2.7: Estructura multinivel del MLFMA.

Las interacciones entre funciones base, requeridas en el MoM, se realizan entonces a nivel de celda mediante una secuencia de procesos de radiación y recepción. Se obtiene así una importante reducción del número de cálculos necesarios, resultando finalmente en un coste computacional de $O(N \log N)$ [55]. Se describe brevemente a continuación la secuencia fundamental de operaciones, basada en la aplicación de expansiones multipolares de la función de Green así como una descomposición de la radiación en ondas planas:

- (i) Proceso de *agregación*: se traslada la radiación de las bases de cada grupo al centro del mismo.
- (ii) *Traslación*: que calcula la interacción entre centros de grupos.
- (iii) *Desagregación*: se traslada la radiación recibida en cada grupo observación, proveniente de la traslación de radiaciones de todos los grupos fuente, hacia las bases del mismo.

Este procedimiento permite por tanto comprimir (en tiempo y memoria) los MVP requeridos en la resolución iterativa del MoM. Se utiliza para ello en esta Tesis un método Krylov, concretamente el método GMRES *Generalized Minimal Residual* (GMRES) [62]. De esta manera, el MVP se puede calcular para un grupo o celda de observación p como:

$$\sum_{n=1}^N Z_{mn} I_n = \sum_{q \in B_p} \sum_{n \in G_q} Z_{mn} I_n + \left(\frac{-jk}{4\pi} \right)^2 \int_{S^2} \mathbf{v}_{mp} \sum_{q \in B_p} \alpha_{pq}(k, \mathbf{r}_{pq}) \sum_{n \in G_q} \mathbf{v}_{qn}(\hat{k}) I_n d^2 \hat{k}, \quad m \in G_p, \quad (2.38)$$

donde B_p es el conjunto de índices de los grupos cercanos (adyacentes) al grupo p , G_p es el conjunto de índices correspondientes a las funciones de test del grupo p , G_q es el conjunto de índices correspondientes a las funciones base del grupo q , y S^2 es la esfera de Ewald o esfera unidad. En la ecuación (2.38), las interacciones próximas entre las funciones base y de test, pertenecientes a grupos adyacentes, son calculadas utilizando directamente el procedimiento de MoM. Por el contrario, las contribuciones de grupos lejanos (no adyacentes) son evaluadas utilizando el FMM según los pasos mencionados anteriormente: (i) *agregando* la radiación de las funciones bases de cada grupo q al centro de su respectivo grupo, de la forma:

$$\mathbf{v}_{qn}(\hat{k}) = \int_{\Delta_n} (\bar{\mathbf{I}} - \hat{k} \hat{k}) \cdot \mathbf{f}_n(\mathbf{r}) e^{-j\mathbf{k} \cdot (\mathbf{r}_q - \mathbf{r})} dS, \quad (2.39)$$

donde \mathbf{r}_q es el centro del q -ésimo grupo e $\bar{\mathbf{I}}$ es el operador diádico identidad; (II) *trasladando* las radiaciones agregadas entre los diferentes grupos mediante el operador de traslación:

$$\alpha_{pq}(k, \mathbf{r}_{pq}) = \sum_{l=0}^L (-j)^l (2l+1) h_l^{(2)}(kr_{pq}) P_l(\hat{\mathbf{k}} \cdot \hat{\mathbf{r}}_{mn}), \quad (2.40)$$

donde $h_l^{(2)}$ es la función esférica de Hankel de segunda especie, P_l es el polinomio de Legendre de grado 1, y L es el número de términos de la expansión multipolar [17]; y (III) *desagregando* la radiación recibida en el grupo p debido a la radiación de las demás celdas sobre las funciones de test del grupo receptor como:

$$\mathbf{V}_{mp}(\hat{\mathbf{k}}) = \alpha \int_{\Delta_m} (\bar{\mathbf{I}} - \hat{\mathbf{k}}\hat{\mathbf{k}}) \cdot \mathbf{f}_m(\mathbf{r}) e^{-j\mathbf{k} \cdot (\mathbf{r} - \mathbf{r}_p)} dS - (1 - \alpha) \hat{\mathbf{k}} \times \int_{\Delta_m} \mathbf{f}_m \times \hat{\mathbf{n}}_m(\mathbf{r}) e^{-j\mathbf{k} \cdot (\mathbf{r} - \mathbf{r}_p)} dS, \quad (2.41)$$

Con este algoritmo, el coste computacional se reduce de $O(N_{it}N^2)$ a $O(N_{it}N^{1.5})$, con N_{it} el número de iteraciones requeridas para obtener el error residual deseado. Aplicando su versión multinivel este coste puede reducirse aún más hasta alcanzar $O(N_{it}N \log N)$, para lo cual se debe incluir además traslaciones exponenciales verticales (entre niveles diferentes), así como interpolación e interpolación adjunta (o antinterpolación) de los campos entre diferentes niveles, todo ello partiendo de una descomposición geométrica multinivel mediante *octrees*.

Para aprovechar la capacidad de ejecución distribuida *multicore* de los equipos de memoria compartida o distribuida se puede aplicar la variante del método conocida como MLFMA-FFT. Esta extensión implementa la traslación entre los grupos del nivel más grueso de la descomposición *octree* mediante la realización de una convolución circular tridimensional de las muestras de la esfera de Ewald. Esta operación se puede implementar de manera muy eficiente en el dominio transformado, aplicando para ello transformadas rápidas de Fourier. De este modo el MLFMA-FFT evita la comunicación excesiva entre procesos distribuidos, repartiendo además de manera equitativa la carga entre los diferentes procesos, dando lugar a una implementación paralela altamente escalable.

2.3.1. Ejemplo de aplicación

En este apartado mostraremos un ejemplo de aplicación del MLFMA. En él trataremos de mostrar la capacidad del método para abordar problemas de mayor tamaño eléctrico en tiempos reducidos y con un nivel de precisión suficiente para problemas de radiación y dispersión electromagnética.

El objeto de este estudio es un vehículo volador no tripulado (*Unmanned Aerial Vehicle*, UAV), también conocidos como drones, basado en el modelo militar de la fuerza aérea estadounidense General Atomics MQ-1 Predator. A efecto de la simulación electromagnética, esta estructura será considerada completamente PEC, tal y como muestra la figura 2.8. Se ha analizado mediante MLFMA aplicando la formulación CFIE, al tratarse de un objeto PEC cerrado. Las dimensiones son de aproximadamente $75 \times 145 \times 20 \lambda$ con respecto a la frecuencia de operación considerada, de 3 GHz (banda S). La resolución ha implicado un total de 1.6 millones de incógnitas aplicando una resolución de mallado de aproximadamente $\lambda/10$.

La complejidad del problema no queda únicamente determinada por las dimensiones del objeto, sino por la caracterización del problema completo, que en este caso se trata de una RCS monoestática en polarización HH (*Horizontal transmit and Horizontal receive*). Se ha escogido un barrido en acimut de 0° a 180° (aprovechando la simetría del problema) con una resolución angular de 0.5° , y una única elevación de 0° . Esta configuración requiere de un total de 361 simulaciones independientes para el cálculo de las corrientes equivalentes superficiales. A partir de estas corrientes se calcula el campo dispersado, y de éste la RCS en cada dirección.

Los resultados numéricos se muestran en la figura 2.9. Concretamente esta figura contiene la parte real de la distribución de corrientes eléctricas superficiales para la dirección de incidencia 0° , a la vez se

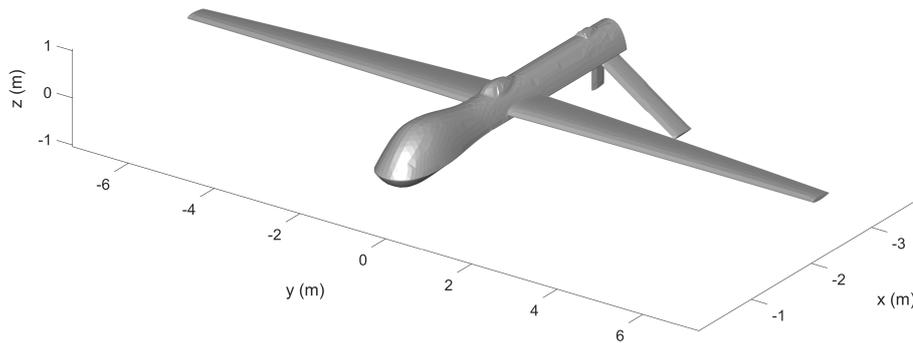


Figura 2.8: Vehículo volador no tripulado escogido para la estimación de RCS monoestática mediante MLFMA a 3 GHz.

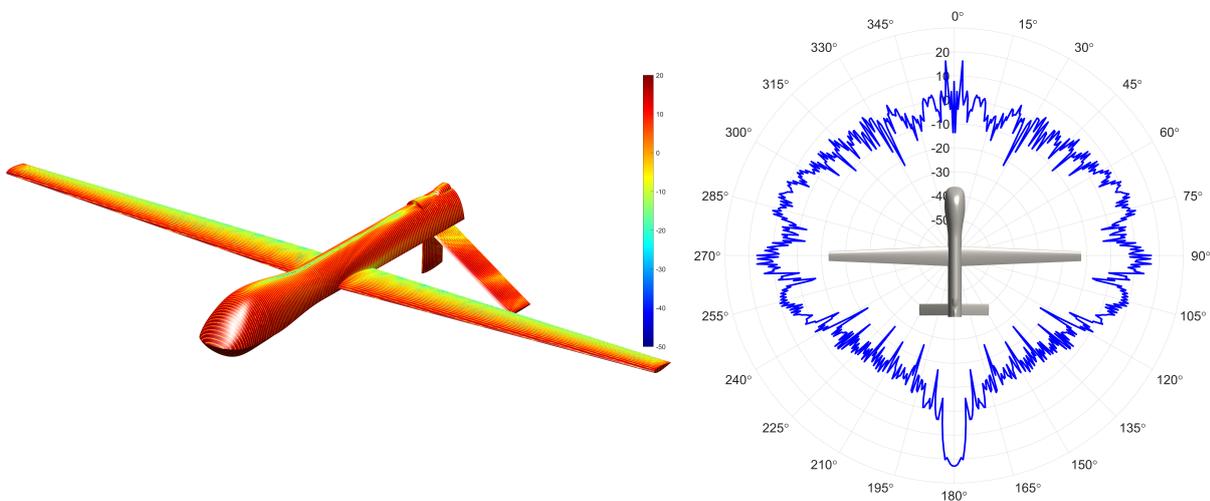


Figura 2.9: Resultados de RCS monoestática obtenidos por MLFMA para un UAV a 3 GHz.

muestra la RCS HH monoestática. El tiempo total requerido para la simulación de las 361 direcciones de incidencia fue de 18.45 horas. Se ha considerado un error residual de 10^{-3} para detener el algoritmo GMRES, más que suficiente para el cálculo en aplicaciones de dispersión como es el caso de la RCS. La máquina utilizada fue un AMD EPYC 7763 64-Core Processor.

2.4. Método de descomposición de dominios (DDM)

El método de descomposición de dominios permite abordar la simulación electromagnética de ciertas estructuras cuyo sistema está mal condicionado o cuyo residuo converge de manera muy lenta. Esto se debe a que el método actúa como un preconditionador de la matriz del sistema. Para ello, el esquema DDM divide el problema completo en problemas de menor tamaño a los que llamaremos *dominios* o *subdominios*. Dichos subdominios están normalmente escogidos aprovechando las particularidades y divisiones naturales de la geometría, independizando partes del problema global que impliquen factores físicos determinantes, como regiones susceptibles de fuertes acoplos entre sí, diversos materiales, o factores de escala muy diferentes.

A la hora de diseñar un esquema de DDM de manera eficiente, es importante tener en cuenta tres aspectos técnicos. El primero es definir una formulación unificada para el problema completo, que

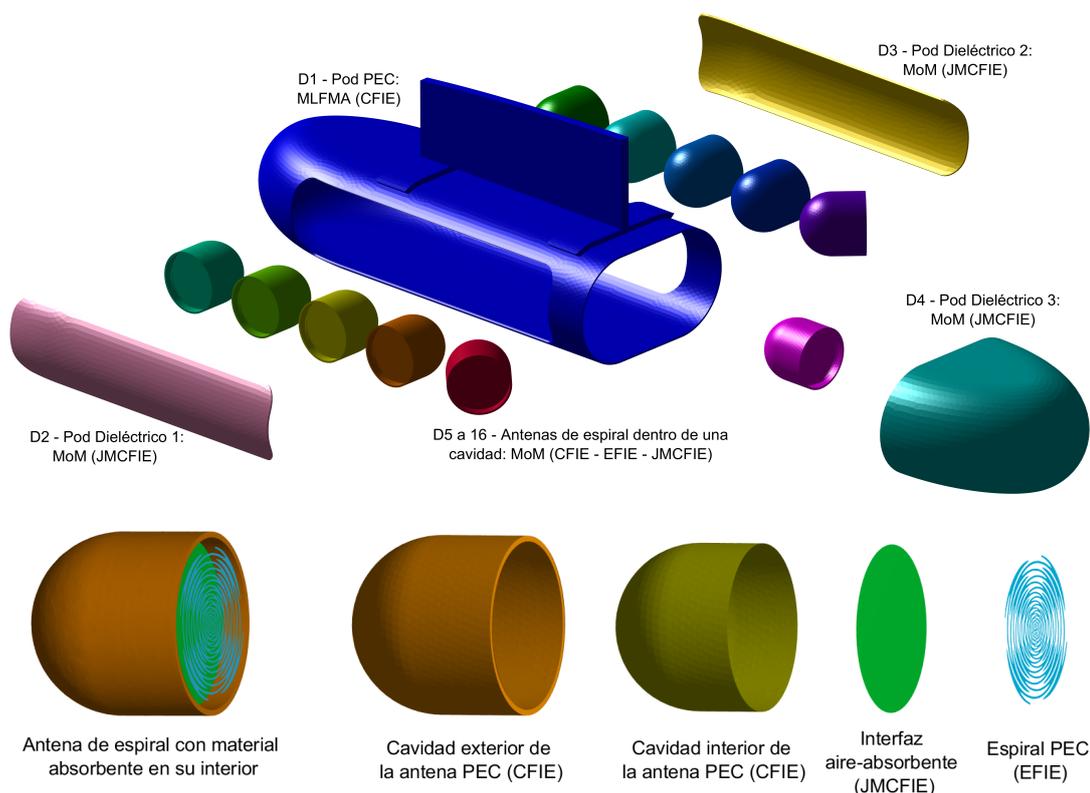


Figura 2.10: Ejemplo de una configuración DDM. Vista detallada de los dominios que pertenecen al pod, D1 a D4, y antenas, D5 a D16, con sus respectivos *solvers* y formulaciones.

permita acoplar globalmente los diferentes *solvers* independientes aplicados a los diferentes problemas en un único sistema de ecuaciones bien condicionado y preciso. En segundo lugar, deben elegirse de manera apropiada las formulaciones de los *solvers* y los preconditionadores que mejor se ajusten a las características propias de cada subdominios, atendiendo a las condiciones de transmisión entre ellos. De esta manera, es posible aislar problemas que implican factores de escala entre sí muy diferentes manteniendo un buen condicionamiento del sistema matricial global. Y en tercer lugar, es preciso incluir algoritmos de aceleración eficientes y precisos tanto en la resolución individual de cada *solver* como en el cálculo del acoplo mutuo entre subdominios. La paralelización de subrutinas para la viabilidad del método resulta además un factor clave. Este esquema trata de ser reflejado en la figura 2.10, en la que se muestra un ejemplo de configuración adecuada de los *solvers* escogidos y su formulación a la hora de ser resueltos individualmente.

Una vez tenido esto en cuenta, procedemos al desarrollo del método a partir del proceso de preconditionamiento. El condicionamiento de un sistema matricial viene determinado por su factor de condicionamiento K [63], para los que pequeños valores de K mejoran el condicionamiento y permiten alcanzar la estabilidad del sistema. Dicho condicionamiento es intrínseco a cada problema y dependiente del sistema de ecuaciones, por lo que el preconditionamiento se basa en la transformación de los sistemas de ecuaciones en sistemas equivalentes de idéntico resultado. Este proceso puede ser aplicado a sistemas matriciales como el proporcionado por el MoM, multiplicando ambos miembros de la igualdad por la matriz de preconditionamiento \mathbf{M} de la siguiente manera:

$$\mathbf{Z} \cdot \mathbf{I} = \mathbf{V} \quad \Rightarrow \quad \mathbf{M} \cdot \mathbf{Z} \cdot \mathbf{I} = \mathbf{M} \cdot \mathbf{V}, \quad (2.42)$$

donde \mathbf{Z} es la matriz de impedancias, \mathbf{V} es el vector de términos independientes o vector de excitaciones e \mathbf{I} es el vector de incógnitas o vector de corrientes que contiene los coeficientes complejos de corrientes eléctricas y magnéticas equivalentes. Se trata entonces de encontrar una matriz \mathbf{M} que pueda cumplir la

condición:

$$K(\mathbf{M} \cdot \mathbf{Z}) < K(\mathbf{Z}), \quad (2.43)$$

En este caso, la matriz de preconditionamiento aplicada es $\mathbf{M} = \mathbf{Z}^{-1}$. Para aplicar este preconditionador en la práctica, se realiza una partición geométrica en base a las características propias del problema, como materiales, divisiones naturales de las superficies, etc., como ya se ha comentado anteriormente. Cada elemento puede ser independizado como un dominio en base a los criterios escogidos a fin de obtener un conjunto de subdominios no solapados. No obstante, estos dominios podrán estar en contacto, por lo que surge la necesidad de aplicar condiciones de transmisión que garanticen la continuidad de las corrientes equivalentes entre dominios.

Por tanto, podemos considerar un total de N dominios constituidos por la división geometría escogida, para la cual cada dominio i constituye un elemento de la matriz de impedancia global \mathbf{Z} a partir de las matrices de impedancia de cada bloque \mathbf{Z}_i . Estas submatrices corresponden al acoplo de cada dominio i consigo mismo en ausencia del resto. Por consiguiente, la matriz de preconditionamiento del sistema \mathbf{M} se define como la matriz diagonal de las inversas de las matrices de impedancia de cada dominio \mathbf{Z}_i^{-1} , de la forma:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{1i} & \cdots & \mathbf{Z}_{1N} \\ \mathbf{Z}_{i1} & \mathbf{Z}_{ii} & \cdots & \mathbf{Z}_{iN} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Z}_{N1} & \mathbf{Z}_{Ni} & \cdots & \mathbf{Z}_{NN} \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} \mathbf{Z}_{11}^{-1} & 0 & \cdots & 0 \\ 0 & \mathbf{Z}_{ii}^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{Z}_{NN}^{-1} \end{bmatrix}. \quad (2.44)$$

Cada bloque diagonal \mathbf{Z}_i puede obtenerse a partir de la matriz de impedancias original en base a una matriz de restricción \mathbf{R}_i propia de cada dominio, de dimensión $n_i \times N$ con n_i la dimensión del bloque \mathbf{Z}_i . Esta relación puede ser descrita como:

$$\mathbf{Z}_i = \mathbf{R}_i \cdot \mathbf{Z} \cdot \mathbf{R}_i^T. \quad (2.45)$$

Las matrices de restricción permiten a su vez relacionar el vector de incógnitas global \mathbf{I} con el subvector de incógnitas correspondiente a cada dominio \mathbf{I}_i , de la siguiente manera $\mathbf{I}_i = \mathbf{R}_i \cdot \mathbf{I}$. De este modo, por medio de la transpuesta \mathbf{R}_i^T y la inversa de cada matriz de impedancia de cada dominio \mathbf{Z}_i^{-1} se puede definir la matriz de preconditionamiento \mathbf{M} como:

$$\mathbf{M} = \sum_{i=1}^N \mathbf{R}_i^T \cdot \mathbf{Z}_i^{-1} \cdot \mathbf{R}_i. \quad (2.46)$$

A través de lo dispuesto en la ecuación (2.46), a continuación podrá resolverse cada inversa de la matriz de impedancia asociada a cada dominio como un subdominio independiente mediante el método más apropiado, típicamente MoM en los pequeños y MLFMA en los grandes.

Este preconditionador mejora significativamente el condicionamiento del sistema global, que puede ser resuelto mediante el algoritmo iterativo externo GMRES con mejor convergencia. En este proceso calculamos el MVP derecho de la igualdad (2.42) entre la matriz de preconditionamiento \mathbf{M} y el vector de términos independientes \mathbf{V} una sola vez. Posteriormente, para cada iteración del GMRES externo se realiza el MVP derecho, del lado izquierdo de la igualdad, mediante uno de los métodos anteriormente descritos, acoplado la solución de los diferentes subdominios entre sí. A continuación, resolvemos los sistemas para cada dominio mediante el método escogido. Una vez resueltos los subproblemas en cada subdominio, sus soluciones son unificadas dando lugar a la solución completa de corrientes en la iteración global.

2.4.1. Condiciones de transmisión

Para lograr una buena convergencia con el preconditionador anterior es fundamental imponer el cumplimiento de las llamadas *condiciones de transmisión* entre subdominios que están en contacto.

Entre las diferentes estrategias disponibles, aplicaremos la formulación de [64]. Esta formulación consiste en extender los subdominios, mediante solapas (o *flaps*) definidos sobre el subdominio o subdominios adyacentes de al menos un cuarto de la longitud de onda de grosor. Para facilitar esta tarea, los subdominios pueden ser definidos de manera apropiada y en el orden de secuenciación necesario en el modelo CAD (*Computer Aided Design*), ya que suelen ser generados específicamente para facilitar la aplicación del DDM.

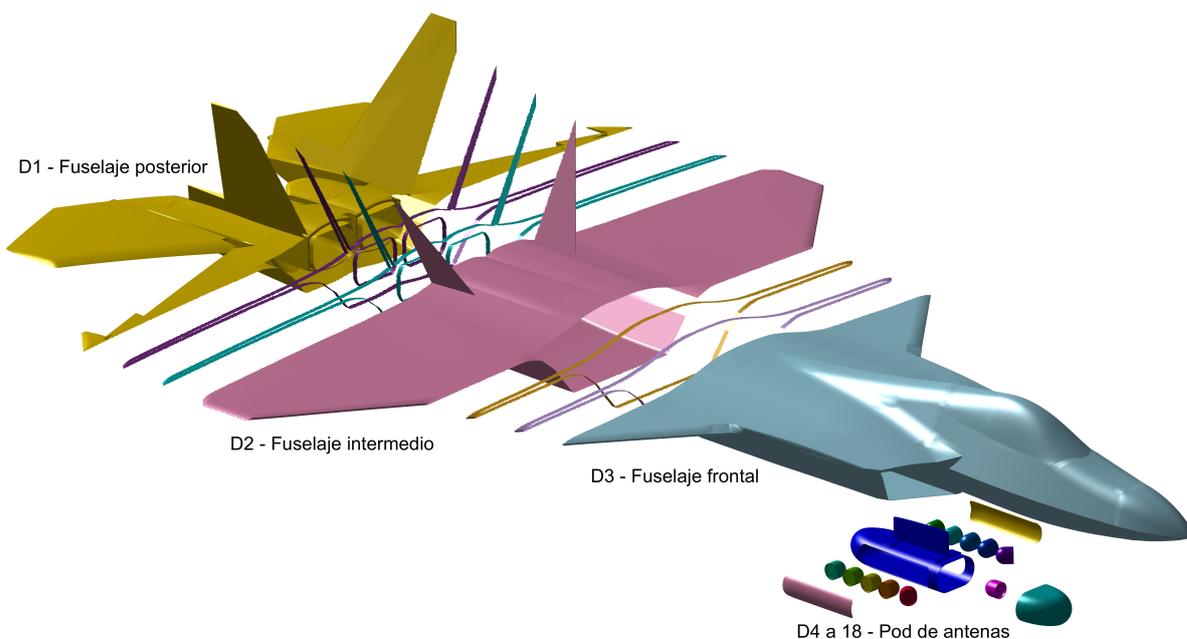


Figura 2.11: Ejemplo de división geométrica en 18 subdominios (conformados por dominios solapados [D1, D2, D3] y dominios flotantes [D4 a D18]), con sus respectivos *flaps* internos y externos, para un avión F-22.

Cada subdominio puede ser por tanto un modelo CAD independiente compuesto por dos componentes: primero un *dominio reducido*, que puede estar compuesto por uno o más superficies, y segundo, los *flap internos* que delimitan los contornos en contacto con otros dominios. Estos *flaps internos* constituyen a su vez los *flaps externos* del dominio o dominios colindantes. La unión entre el *dominio reducido* y los *flaps internos* constituyen el *dominio real o restringido*, mientras que la unión entre el *dominio restringido* y los *flaps externos* da lugar al denominado *dominio aumentado*, que es el que finalmente se resuelve en cada problema local del preconditionador.

La figura 2.11 muestra un ejemplo de un F22 dividido en tres dominios solapados con sus respectivos *flaps internos*, junto al anterior ejemplo de la figura 2.10 que mostraba dominios flotantes. Los componentes de cada dominios son ensamblados y mallados en conjunto, obteniendo un mallado conformado, al igual que los anteriores métodos, permitiendo la compatibilidad entre *solvers*. La imposición de mallados conformados garantiza el flujo de la corriente a través de los subdominios en contacto.

2.4.2. Acoplo por radiación

En el desarrollo de la formulación del método mencionábamos la posibilidad de realizar el cálculo del MVP global a través del GMRES externo mediante cualquiera de los métodos desarrollados en esta Tesis (con especial mención al MLFMA por ser el idóneo en muchos casos). No obstante, es posible aplicar otros métodos. Esto es así debido a la naturaleza multisolver del DDM, que permite independizar por completo los cálculos entre dominios y simplificar los procesos de ejecución y distribución de subdominios entre nodos de computación. Para ello, se ha realizado una implementación del DDM que permite realizar

los acoplamientos entre diferentes subdominios mediante radiación. Esta variante está basada en el cálculo del campo cercano y su proyección entre dominios.

Por tanto desarrollaremos los procesos de acoplo mutuo por radiación una vez definidos el *dominio reducido, restringido, aumentado* y los *flaps internos* y *externos* del apartado anterior. Para ello, partiremos de las corrientes obtenidas por la iteración global más reciente del GMRES externo, calculando iterativamente la radiación de cada dominio reducido sobre el resto de los dominios, excluyéndose a si mismo. Este cálculo se realizará mediante el método anteriormente citado. A continuación, acoplamos la radiación al resto de dominios de manera equivalente a los procesos de agregación, traslación y desagregación del MLFMA. Este proceso se realizaría de manera equivalente para los dominios no solapados o flotantes. No obstante, el tratamiento de las *junctions* implica de una corrección para condicionar el flujo de la corriente sobre el contorno. Para ello, calculamos la radiación de los *flaps internos* de cada dominios sobre el resto de dominios, excluyendo la radiación del *flap interno* sobre su *flap externo* correspondiente y acoplándolo sobre cada dominio. El acoplo de la radiación del *flap interno* sobre el *flap externo* la realizaremos mediante MLFMA, ya que debido a la proximidad entre ambos ha sido probado su efectividad con resultados más precisos y la ausencia de singularidades. Por último, llevaremos a cabo el autoacoplo (consigo mismo) del mismo modo que el acoplo entre el *flap interno* y *externo* en base a la misma casuística. Esta implementación de DDM con acoplamientos globales mediante radiación permitirá la inclusión de regiones IPO que se ha realizado en esta Tesis.

2.4.3. Ejemplo de aplicación

En este apartado mostraremos cómo la implementación del DDM proporciona una solución precisa de las ecuaciones de Maxwell. Para ello nos centraremos en el cálculo de parámetros de antenas muy sensibles como la impedancia de entrada y el acoplo mutuo, involucrando decenas de antenas y diversos factores multiescala. El objetivo de este apartado será concluir que el esquema SIE-DDM constituye una solución apropiada en el contexto de problemas de compatibilidad e interferencia electromagnética (*Electromagnetic Compatibility and Interference, EMC/EMI*) de gran tamaño. Este ejemplo a su vez forma parte de los resultados obtenidos para una de las publicaciones indexadas que se realizaron durante el transcurso de esta Tesis [65].

Las dimensiones del modelo realista (aunque ficticio) del buque son aproximadamente 140 m de largo, 20 m de ancho y 40 m de alto (140 x 20 x 40 en términos de longitud de onda, λ , para la frecuencia intermedia de análisis, 300 MHz). Son necesarias un total de 13.8 millones incógnitas para modelar el problema completo. Para el que el basto factor multiescala es evidente, al fluctuar entre rangos de $\lambda/10$ en superficies suaves de la geometría a $\lambda/1639$ en regiones pertenecientes a las antenas, con detalles finos (especialmente en los alrededores de los puntos de alimentación).

Tabla 2.1: Parámetros de simulación del buque analizado por DDM.

Dominios	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8
Incógnitas	64 575	480 887	992 569	64 314	439 633	698 046	2 771 909	2 820 127
Memoria (GB)	1.4	9.6	19.2	1.3	8.8	13.4	50.5	51.7
Tiempo inicialización (s)	8.28	38.66	73.34	8.6	35.66	8.851.01	13193.66	196.89
Tiempo ejecución (s)	9.22	71.7	74.6	9.08	63.66	51.28	459.97	264.29
Dominios	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}
Incógnitas	2 881 272	2 770 883	12 309	12 309	12 309	12 309	6 804	6 804
Memoria (GB)	52.9	51.9	5.5	5.5	5.5	5.5	1.4	1.4
Tiempo inicialización (s)	200.91	193.66	92.9	92.9	92.9	92.9	40.8	40.8
Tiempo ejecución (s)	299.12	459.97	0.23	0.23	0.23	0.23	0.11	0.11
Dominios	D_{17}	D_{18}	D_{19}	D_{20}	D_{21}	D_{22}	D_{23}	
Incógnitas	4 320	4 320	2 498	2 498	2 498	2 498	2 384	
Memoria (GB)	0.6	0.6	0.2	0.2	0.2	0.2	0.1	
Tiempo inicialización (s)	18.43	18.43	3.72	3.72	3.72	3.72	3.38	
Tiempo ejecución (s)	0.05	0.05	0.03	0.03	0.03	0.03	0.02	

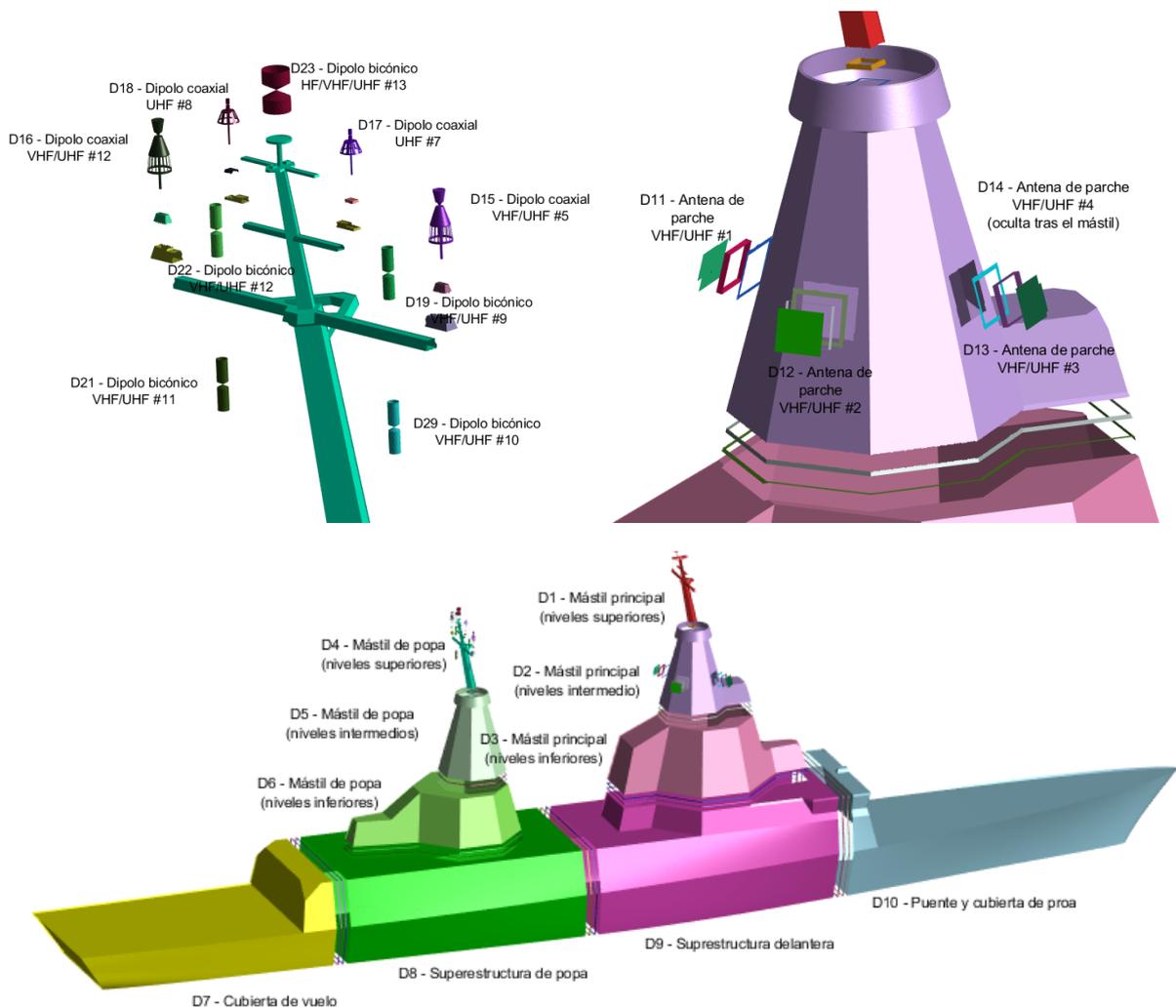


Figura 2.12: División de la superestructura en subdominios. Incluyendo la disposición de las antenas del mástil de popa y sus subdominios a la izquierda y la disposición de las antenas del mástil de proa y sus subdominios a la derecha. Mostrando los *flaps* correspondientes para los dominios en contacto (solapados).

En consecuencia, el problema se ha descompuesto en los 23 subdominios dispuestos en la figura 2.12. De entre los cuales, $D_1 - D_{10}$ corresponden a partes estructurales del buque que serán resueltas de manera independiente mediante MLFMA-FFT y los 13 subdominios restantes ($D_{11} - D_{23}$) corresponden a antenas a bordo del buque que serán analizadas mediante MoM. Donde $D_{11} - D_{14}$ y $D_{17} - D_{18}$ son las antenas #1 - #4 y #7 - #8 de los sistemas S_1 y S_3 respectivamente (en la banda de *Very High Frequency*, VHF) y $D_{15} - D_{16}$ y $D_{19} - D_{23}$ son las antenas #5 - #6 y #9 - #13 de los sistemas S_2 y S_4 respectivamente (en la banda de *Very and Ultra High Frequency*, V/UHF).

En este estudio se ha realizado un barrido en frecuencia de 100 MHz a 550 MHz en la principal banda de operación de los sistemas involucrados (V/UHF). El problema completo es considerado PEC, asignando la formulación EFIE ($\alpha = 1$) para las superficies abiertas pertenecientes a las antenas y CFIE ($\alpha = 0,5$) para el resto de superficies.

Los resultados obtenidos muestran una convergencia extraordinaria para el esquema SIE-DDM, al cual le toman 4 iteraciones externas de Krylov y 1.8 horas de simulación en llegar a un error residual de $5 \cdot 10^{-8}$. Este resultado contrasta enormemente con las 25 mil iteraciones y 60 horas que requeriría el MLFMA en obtener un residuo de $2 \cdot 10^{-3}$. Por lo que este es un buen ejemplo de cómo mallados de gran tamaño, que contemplan varias excitaciones locales y diversos factores multiescala, ralentizan la

convergencia del problema cuando se utiliza MLFMA.

En la figura 2.13 (a) podemos observar la distribución de corrientes equivalente sobre el buque a 300 MHz como prueba de los resultados obtenidos por medio de la formulación SIE para métodos *full-wave*. A partir de estas corrientes se obtiene el acoplamiento mutuo S_{12} entre antenas de los sistemas a bordo simulados. para el caso de la antena transmisora #3 (fiigura 2.13 (b)).

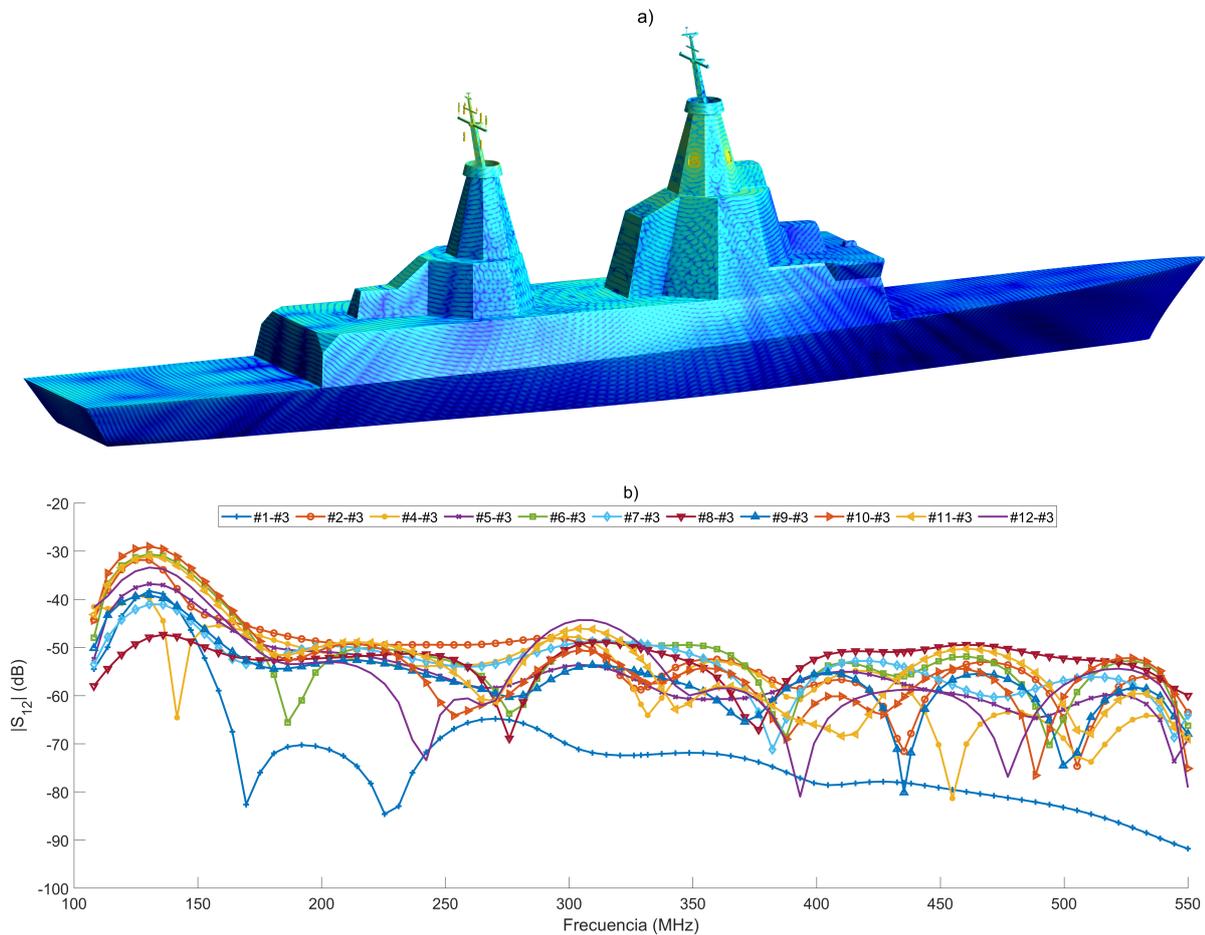


Figura 2.13: a) Parte real de la densidad de corrientes eléctrica superficial sobre el buque analizado a 300 MHz por el esquema SIE-DDM b) Acoplo mutuo (S_{12}) entre antenas, para el caso de la antena transmisora #3 en la banda de V/UHF.

Algunos datos estadísticos correspondientes a este ejemplo pueden observarse en la tabla 2.1, donde se muestra la información para cada dominio aumentado: número de incógnitas, memoria consumida, tiempo de inicialización (consumido una sola vez para cada frecuencia) y tiempo de ejecución calculado como el promedio de los tiempos de resolución a lo largo de todas las iteraciones externas. Dichas simulaciones fueron ejecutadas en un cluster con $4 \times$ Xeon E7-8867v3@2.5 GHz (4×16 núcleos = 64 núcleos) y 1 TB de memoria RAM (*Random Access Memory*).

CAPÍTULO 3 **Desarrollo y aplicación de la ecuación integral para el campo magnético (MFIE) iterada**

Contenido

3.1. Resolución de la MFIE iterada	28
3.1.1. Óptica física (PO)	28
3.1.1.1. Funciones base	30
3.1.1.2. Ejemplos de aplicación	33
3.1.2. Óptica física iterada (IPO)	35
3.1.2.1. Mallado adaptativo	39
3.1.2.2. Matrices de visibilidades	40
3.1.2.3. Detalles de implementación	41
3.1.2.4. Ejemplo de aplicación	43
3.2. Trazado de rayos: NVIDIA OptiX API	47
3.2.1. Aplicaciones del trazado de rayos	48
3.2.2. Estructura de la arquitectura OptiX	49
3.2.3. Programas definidos por el usuario	50
3.2.3.1. Ray Generation	51
3.2.3.2. Bounding Box	53
3.2.3.3. Intersection	54
3.2.3.4. Closest Hit	55
3.2.3.5. Any Hit	57
3.2.3.6. Miss	58
3.2.3.7. Exception	58
3.2.4. OpenGL	58
3.2.5. Módulo de visibilidades de IPO	62

Este capítulo se ha dividido en dos bloques relacionados con las dos principales aportaciones de esta Tesis: la implementación de un método basado en IPO mediante la resolución iterada de la MFIE, y la incorporación de trazado de rayos sobre GPU para la obtención de matrices de visibilidad y sectores ciegos.

La primera aportación de calado que se recoge en esta Tesis se basa en la implementación del método IPO mediante la resolución iterada y rigurosa de la MFIE. Comenzaremos incluyendo la formulación del método de PO así como la definición de las funciones base LPIT, que serán las que se apliquen en esta Tesis. Posteriormente el método será extendido a su versión iterada para la inclusión de múltiples reflexiones en la estructura. A lo largo del desarrollo de este capítulo se pondrá de relieve la necesidad de modelar correctamente las reflexiones en diedros, triedros, cavidades y en general elementos dispersores con geometría compleja, cuyas contribuciones resultan significativas en el cálculo de la RCS global. Se describirán las aportaciones novedosas en materia de preprocesado geométrico, mallado adaptativo y

obtención de las matrices de visibilidades ($iLit$ e $iVis$), necesarias para el correcto funcionamiento del método. Adicionalmente, se presentará una aceleración basada en la aplicación de GO en la tercera reflexión, que permitirá mejorar los tiempos sin pérdida aparente de precisión. Finalmente, se pondrá a prueba la eficiencia del método mediante ejemplos canónicos y geometrías arbitrarias que demuestren la capacidad del método.

En lo que respecta a la obtención de las matrices de visibilidades mediante trazado de rayos sobre GPU, introduciremos brevemente la terminología usual en *ray tracing*, contextualizando las aplicaciones y el entorno que propiciaron el auge de esta tecnología. En este contexto, abordaremos la arquitectura OptiX, empleada en esta Tesis, y profundizaremos en algunos de sus aspectos fundamentales. Nos centraremos especialmente en los procesos que facilitan la elaboración de las herramientas requeridas en los trabajos de esta Tesis. Concluiremos esta sección pormenorizando aspectos relevantes como la interfaz gráfica de usuario mediante OpenGL y sus librerías, así como la implementación del módulo de visibilidades de IPO.

3.1. Resolución de la MFIE iterada

Como ya introdujimos anteriormente en el capítulo 1, la óptica física constituye uno de los métodos asintóticos o de alta frecuencia basado en la estimación de la distribución de corrientes equivalentes sobre superficies. Esta técnica proporciona resultados rápidos y de una precisión controlada para la caracterización electromagnética de problemas de dispersión en estructuras con geometrías sencillas y gran tamaño eléctrico. No obstante, al igual que otros métodos asintóticos como pueden ser GO, GTD o UTD, la óptica física ignora contribuciones relevantes en aplicaciones como el cálculo de la RCS, especialmente en el caso de detalles geométricos que impliquen la formación de diedros, triedros o cavidades, al no tener en cuenta las múltiples reflexiones en dichos elementos.

La solución propuesta en esta Tesis surge de la necesidad de implementar un método capaz de obtener resultados muy rápidos y suficientemente precisos sobre grandes estructuras en los que las contribuciones de los detalles geométricos previamente mencionados puedan ser tenidos en cuenta de forma eficiente y con suficiente precisión. Comenzaremos para ello detallando la formulación del método de óptica física, que se recoge en la siguiente sección.

3.1.1. Óptica física (PO)

La formulación del método de óptica física parte del desarrollo de la MFIE para transformar un problema de campo radiado por unas fuentes en espacio libre en un problema equivalente de corrientes equivalentes inducidas sobre las superficies. Esta equivalencia permite evaluar las corrientes sobre la superficie y acto seguido, estimar la contribución de dichas corrientes equivalentes en diferentes puntos del espacio. Así pues, el procedimiento a seguir en la metodología PO se puede resumir de la siguiente manera:

- Propagar el campo magnético en espacio libre hacia las superficies a partir de las corrientes equivalentes superficiales de las fuentes, o partiendo propiamente de un campo incidente hacia dicha región.
- Evaluar el campo magnético incidente sobre las facetas visibles que componen la geometría de PO con el fin de inducir y obtener la densidad de corriente eléctrica equivalente asociada a las superficies PO. Dicho proceso requerirá del previo cálculo de las facetas visibles a la incidencia directa de las fuentes, de las que solo se deducirá la corriente eléctrica que aportará contribución de campo a los puntos de observación.

- Por último, realizar la operación inversa: propagar el campo magnético en espacio libre hacia los puntos de observación a partir de las corrientes eléctricas superficiales inducidas en las zonas visibles de las superficies PO. Con ello, se logra caracterizar la contribución directa por reflexión de una fuente sobre geometrías de gran tamaño eléctrico.

A continuación, partiremos de la notación dispuesta en la figura (3.1) para el caso de una o más fuentes impresas y puntos de observación localizados en \mathbf{r} , asumiendo una dirección de incidencia directa sobre la superficie del objeto. El término $e^{j\omega t}$ se omitirá a efectos de formulación para facilitar su desarrollo y los superíndices i y s denotarán el carácter incidente o reflejado (dispersado o *scattered*) respectivamente del campo eléctrico \mathbf{E} y magnético \mathbf{H} . En esta notación, \mathbf{J} y \mathbf{M} son las densidades de corriente eléctrica y magnéticas de las fuentes impresas respectivamente. \mathbf{J}^{PO} es la densidad de corriente eléctrica inducida en la superficie S de la región PO. $\hat{\mathbf{n}}$ es el vector normal a la misma superficie S , donde S^{lit} corresponde a las regiones visibles o iluminadas (representadas en rojo), y S^{shd} corresponde a las regiones de sombra u ocultas (representadas en azul), cuya contribución a la dispersión resulta en principio poco relevante y por tanto puede ser obviada. Por último, \mathbf{r}' denota los puntos sobre la superficie PO. En nuestro caso los baricentros (o centroides) de las facetas triangulares planas del mallado aplicado para representar las superficies.

El cálculo de las regiones de visibilidad y ocultación será desarrollado pormenorizadamente en secciones posteriores. No obstante, para el desarrollo de la formulación en esta sección, consideraremos que disponemos de las facetas visibles para cada dirección de incidencia, almacenadas por columnas en la matriz de visibilidades que denominaremos $iLit$, definida como una matriz de dimensión $N \times M$, siendo N el número de facetas y M el número de direcciones de incidencia requeridas por la aplicación. Se trata de una matriz booleana en la que se indican mediante ceros las facetas no visibles, y mediante unos las facetas visibles desde la fuente.

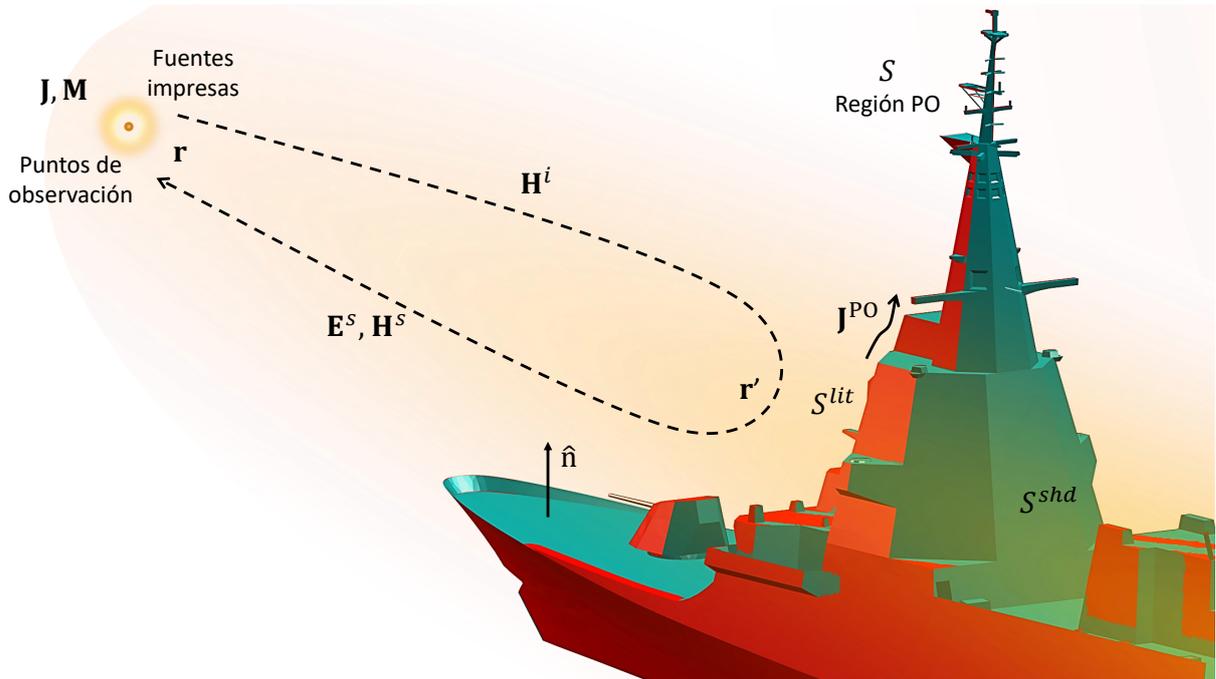


Figura 3.1: Representación de un problema equivalente para una geometría arbitraria como ejemplo de desarrollo de la formulación PO.

La formulación PO comienza evaluando el campo \mathbf{H}^i radiado por \mathbf{J}^i sobre la región S , sustituyendo las corrientes impresas generadas por las fuentes en \mathbf{r} por campos radiados en espacio libre dirigidos

hacia la región PO situada en \mathbf{r}' como:

$$\mathbf{H}^i(\mathbf{r}') = \oint_S \mathbf{J}(\mathbf{r}) \times \nabla G(\mathbf{r}' - \mathbf{r}) dS, \quad (3.1)$$

donde f denota el valor principal de la integral, $\nabla G(\mathbf{r}' - \mathbf{r})$ es el gradiente de la función de Green en espacio libre y $R = |\mathbf{r}' - \mathbf{r}|$ es la distancia entre la fuente y la región PO y β la constante de propagación en el medio. Al desarrollar la ecuación (3.1) por completo en base a (2.9) obtenemos:

$$\mathbf{H}^i(\mathbf{r}') = \oint_S \left(j\beta + \frac{1}{R} \right) \hat{R} \times \mathbf{J}(\mathbf{r}) \frac{e^{-j\beta R}}{4\pi R} dS. \quad (3.2)$$

El siguiente paso es calcular las corrientes inducidas $\mathbf{J}^{PO}(\mathbf{r}')$ sobre la superficie S , la cual únicamente toma valores distintos de cero sobre las facetas visibles desde la fuente S^{lit} . El conjunto de facetas que componen esta porción de superficie son identificadas por medio de la matriz $iLit$. Esta relación puede escribirse como:

$$\mathbf{J}^{PO}(\mathbf{r}') = \begin{cases} 2\hat{n} \times \mathbf{H}^i(\mathbf{r}'), & \mathbf{r}' \in S^{lit}, \\ 0, & \mathbf{r}' \in S^{shd}. \end{cases} \quad (3.3)$$

Como podemos observar, la principal ventaja de este método con respecto a los métodos *full-wave* como MoM, MLFMA o DDM es la sencillez con la que evaluar las corrientes eléctricas equivalentes inducidas sobre la superficie, evitando así almacenar e invertir matrices de gran tamaño. No obstante, esto es a costa de eliminar las múltiples interacciones dentro de la propia estructura, con lo que dependiendo del tipo de geometría se reducirá consecuentemente la precisión del resultado obtenido.

Por último, se puede seguidamente radiar el campo \mathbf{H}^s a partir de las corrientes \mathbf{J}^{PO} , como:

$$\mathbf{H}^s(\mathbf{r}) = \oint_S \mathbf{J}^{PO}(\mathbf{r}') \times \nabla G(\mathbf{r} - \mathbf{r}') dS'. \quad (3.4)$$

En el caso de la figura 3.1, los puntos de observación coinciden con la dirección de las fuentes impresas, tal como sería el caso en el cálculo de una RCS monoestática. No obstante, los puntos de observación pueden estar localizados en cualquier región del espacio.

3.1.1.1. Funciones base

Las funciones LPIT surgieron de la necesidad de desarrollar nuevas funciones base capaces de independizar el tamaño de las facetas que modelan la geometría de la longitud de onda del problema. De esta forma, su tamaño queda únicamente condicionado por la fidelidad geométrica a la hora de representar las superficies. A continuación, describimos el procedimiento para evaluar la radiación mediante el esquema LPIT.

Para ello consideraremos un triángulo arbitrario T , como el descrito en la figura 3.2. A partir de estos parámetros, podemos calcular la radiación producida por esta faceta, resolviendo:

$$\mathbf{I}_c = \int_T \mathbf{J}^{PO} \frac{e^{-j\beta R}}{R} dS. \quad (3.5)$$

donde \mathbf{J}^{PO} es la densidad de corriente inducida en la faceta triangular obtenida por PO mediante la ecuación (3.3), y $R = |\mathbf{r} - \mathbf{r}'|$ es la distancia entre el punto de observación \mathbf{r} y el punto fuente \mathbf{r}' . En base a la figura 3.2, si el punto de observación está lo suficientemente alejado de la faceta, entonces R puede ser aproximado como $R \simeq R_g - \mathbf{R}_s \cdot \hat{R}_g$ para el término de fase de la integral (3.5), y como $R \simeq R_g$ para la amplitud, con $\mathbf{R}_g = \mathbf{r} - \mathbf{r}_g$ y $\mathbf{R}_s = \mathbf{r}' - \mathbf{r}_g$, siendo \mathbf{r}_g el vector de posición del baricentro o centroide del triángulo. Por tanto, la integral (3.5) puede expresarse como:

$$\mathbf{I}_c = \frac{1}{R} e^{-j\beta R_g} e^{-j\beta \mathbf{r}_g \cdot \hat{R}_g} \mathbf{I}'_c, \quad (3.6)$$

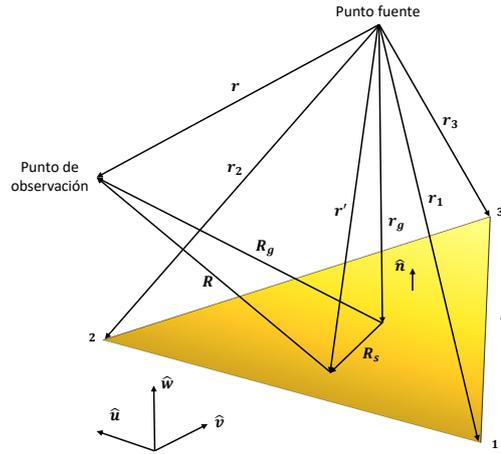


Figura 3.2: Parámetros geométricos asociados a una faceta triangular LPIT.

donde:

$$\mathbf{I}'_c = \int_T \mathbf{J}^{PO} \cdot e^{-j\beta \mathbf{r}' \cdot \hat{\mathbf{R}}_g} dS. \quad (3.7)$$

Para resolver la integral \mathbf{I}'_c , es conveniente incorporar un sistema de coordenadas local $\zeta(u, v, w)$, como muestra la figura 3.2. En este contexto, el triángulo T se encuentra en el plano $w = 0$ y, por lo tanto, solo se necesitan dos variables de superficie en la integración. Por tanto, \mathbf{I}'_c puede reescribirse como:

$$\mathbf{I}'_c = I'_{c,u} \hat{u} + I'_{c,v} \hat{v}. \quad (3.8)$$

con:

$$I'_{c,u} = \int_T J_u e^{j(\beta/R_g)(u'R_{g,u} + v'R_{g,v})} dv' du', \quad (3.9)$$

$$I'_{c,v} = \int_T J_v e^{j(\beta/R_g)(u'R_{g,u} + v'R_{g,v})} dv' du'. \quad (3.10)$$

Para evaluar la integral (3.9) y (3.10), asumimos una constante para la magnitud de la densidad de corriente en la faceta, y una aproximación lineal mediante un polinomio de primer grado para la fase:

$$J_u(u, v) = A_u e^{-j(B_{u,u}u + B_{u,v}v + B_{u,0})}, \quad (3.11)$$

$$J_v(u, v) = A_v e^{-j(B_{v,u}u + B_{v,v}v + B_{v,0})}. \quad (3.12)$$

Haciendo uso de esta aproximación, las integrales (3.9) y (3.10) puede ser reescritas como sigue:

$$I'_{c,\alpha} = \int_T A_\alpha e^{j[(k_u + B_{\alpha,u})u' + (k_v + B_{\alpha,v})v' + B_{\alpha,0}]} dv' du', \quad (3.13)$$

donde $\alpha = u, v$, $k_u = \beta(R_{g,u}/R_g)$, y $k_v = \beta(R_{g,v}/R_g)$. La integral (3.13) puede ser resuelta realizando una transformación a un nuevo sistema de coordenadas local $\zeta'(p, q)$ definido en la faceta (denominado *coordenadas naturales* del triángulo). La matriz de transformación es:

$$T_1 = \begin{bmatrix} r_{3,u} - r_{1,u} & r_{3,v} - r_{1,v} & 0 \\ r_{2,u} - r_{1,u} & r_{2,v} - r_{1,v} & 0 \\ r_{1,u} & r_{1,v} & 1 \end{bmatrix}, \quad (3.14)$$

y la integral (3.13) puede reescribirse como:

$$I'_{c,\alpha} = |\det(T_1)| \int_0^1 \int_0^{1-p} C_0 e^{j(D_p p + D_q q + D_0)} dq dp, \quad (3.15)$$

donde $|\det(T_i)| = 2\Delta$ con Δ el área del triángulo. La siguiente relación viene dada por los nuevos coeficientes y el valor del integrando en los tres vértices:

$$\begin{aligned} C_0 &= |J_{\alpha,0}| & D_p &= \phi_{\alpha,3} - \phi_{\alpha,1} + k_p, \\ D_0 &= \phi_{\alpha,1} + k_0 & D_q &= \phi_{\alpha,2} - \phi_{\alpha,1} + k_q, \end{aligned} \quad (3.16)$$

con:

$$\begin{bmatrix} k_p \\ k_q \\ k_0 \end{bmatrix} = T_1 \begin{bmatrix} k_u \\ k_v \\ 0 \end{bmatrix}, \quad (3.17)$$

y donde $|J_{\alpha,i}|$ y $\phi_{\alpha,i}$ son la magnitud y la fase de la componente α de la densidad de corriente en el i -ésimo vértice del triángulo. La densidad de corriente en el vértice $J_{\alpha,i} = |J_{\alpha,i}| e^{-j\phi_{\alpha,i}}$ puede obtenerse del valor en el centroide como:

$$J_{\alpha,i} = J_{\alpha,0} e^{-j\beta(\mathbf{r}_i - \mathbf{r}_g)\hat{r}_v}. \quad (3.18)$$

$J_{\alpha,0} = |J_{\alpha,0}| e^{-j\phi_{\alpha,0}}$ es el valor de la componente α de la densidad de corriente en el centroide del triángulo, y \mathbf{r}_v el vector de propagación del campo incidente expresado en coordenadas del sistema local (u, v, w) . Para evitar la ambigüedad al calcular la corriente en los tres vértices del triángulo, todos los cálculos convienen expresarlos en base a la notación de módulo y fase, en vez de usar la notación de parte real e imaginaria.

Por tanto, la integral (3.15) puede ser resuelta analíticamente tal que:

$$I'_{c,\alpha} = \frac{j2\Delta C_0 e^{jD_0}}{D_p D_q (D_q - D_p)} \left[jD_p (e^{jD_q} - 1) - jD_q (e^{jD_p} - 1) \right]. \quad (3.19)$$

La expresión anterior contiene singularidades, y no puede ser evaluada directamente cuando $|D_p|$, $|D_q|$ o $|D_q - D_p|$ se aproximan a cero. Para tratar estas casuísticas especiales, se ha desarrollado un procedimiento que consiste en expandir de manera apropiada el término (3.19) en una serie de Taylor. De manera similar, se realiza este procedimiento en [66], pero con una diferencia principal, en nuestra formulación, la expansión en serie se aplica al resultado de la integral en lugar de expandir el integrando y luego integramos los términos resultantes. En este contexto, consideramos tres casos en cuenta.

- Caso $|D_p| \approx 0$: El término e^{jD_p} es expandido en una serie de Taylor sobre jD_p :

$$e^{jD_p} = \sum_{n=0}^{\infty} \frac{(jD_p)^n}{n!}. \quad (3.20)$$

Reemplazando en (3.19), obtenemos:

$$I'_{c,\alpha} = \frac{2\Delta C_0 e^{jD_0}}{jD_q (jD_q - jD_p)} \times \left\{ e^{jD_q} - jD_q \left[\sum_{n=1}^{\infty} \frac{(jD_p)^{n-1}}{n!} \right] - 1 \right\}. \quad (3.21)$$

- Caso $|D_q| \approx 0$: El término e^{jD_q} es expandido en una serie de Taylor sobre jD_p y sustituido en (3.19):

$$I'_{c,\alpha} = \frac{2\Delta C_0 e^{jD_0}}{jD_p (jD_p - jD_q)} \times \left\{ e^{jD_p} - jD_p \left[\sum_{n=1}^{\infty} \frac{(jD_q)^{n-1}}{n!} \right] - 1 \right\}. \quad (3.22)$$

- Caso $|D_p|, |D_q| \approx 0$: Ambos términos e^{jD_p} y e^{jD_q} son expandidos en una serie de Taylor sobre jD_p y jD_q , respectivamente, y después sustituidos en (3.19):

$$I'_{c,\alpha} = \frac{2\Delta C_0 e^{jD_0}}{(jD_q - jD_p)} \sum_{n=1}^{\infty} \frac{(jD_q)^n - (jD_p)^n}{(n+1)!}. \quad (3.23)$$

- Caso $|D_q - D_p| \approx 0$: La expresión (3.19) puede ser reescrita como:

$$I'_{c,\alpha} = \frac{j2\Delta C_0 e^{jD_0}}{D_p D_q (D_q - jD_p)} \times \left[jD_p e^{jD_p} e^{j(D_q - D_p)} - jD_q e^{jD_p} + jD_q - jD_p \right]. \quad (3.24)$$

Expandiendo el termino $e^{j(D_q - D_p)}$ en una serie de Taylor de los parámetros $D_q - D_p$, obtenemos:

$$e^{j(D_q - D_p)} = \sum_{n=0}^{\infty} \frac{[j(D_q - D_p)]^n}{n!}, \quad (3.25)$$

y reemplazando en (3.24) llegamos a:

$$I'_{c,\alpha} = \frac{2\Delta C_0 e^{jD_0}}{jD_p jD_q} \times \left[1 - e^{jD_p} + jD_p e^{jD_p} \sum_{n=1}^{\infty} \frac{[j(D_q - D_p)]^{n-1}}{n!} \right]. \quad (3.26)$$

Por tanto, el procedimiento de expansión de las funciones base LPIT consiste en aproximar la amplitud de cada componente de la densidad de corriente por un distribución constante en cada triángulo y la fase mediante una interpolación poligonal de primer orden, como trata de representar la figura 3.3. La principal ventaja de esta aproximación reside en contemplar la variación de fase como parte de la formulación, permitiendo aumentar el tamaño de triángulo o faceta significativamente sin perder precisión. Esta solución logra independizar la discretización de la geometría de la longitud de onda del problema. Adicionalmente, esta formulación considera dos direcciones vectoriales sobre cada triángulo con que poder representar cualquier dirección de la corriente. Por este motivo, las funciones base LPIT no permite garantizar la continuidad de la corriente entre triángulos adyacentes e implica la presencia de cargas ficticias en los contornos de las facetas. No obstante, en el caso de PO no es necesario obtener la divergencia para evaluar la integral de radiación (que puede ser resuelta analíticamente), por lo que no supone un problema.

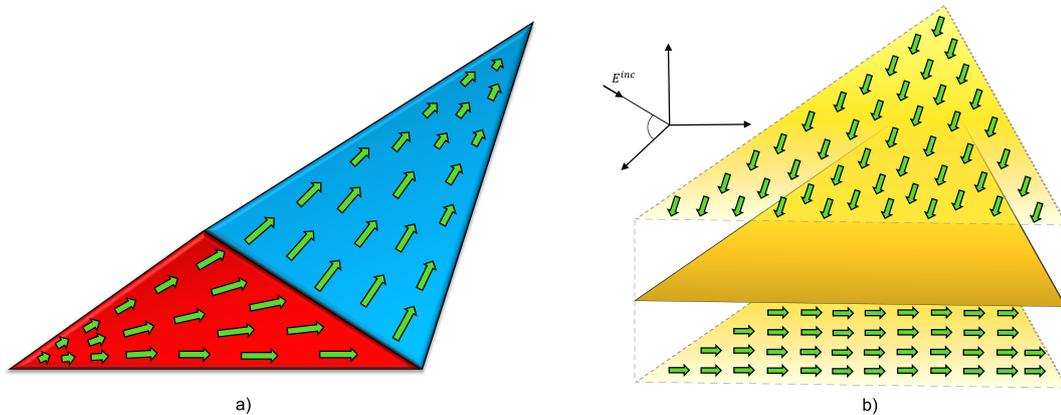


Figura 3.3: a) Funciones base RWG b) Funciones base LPIT. Las flechas representan la magnitud y dirección de la densidad de corriente superficial y el degradado la variación de fase.

3.1.1.2. Ejemplos de aplicación

Se exponen a continuación un par de ejemplos con el objetivo de mostrar la precisión del método de PO y poner de relieve sus limitaciones, que tratarán de ser solventadas en la siguientes secciones mediante la versión iterada. Para ello, los resultados obtenidos mediante PO serán contrastados con las referencias obtenidas mediante MLFMA. Las pruebas se centrarán en el cálculo de la RCS monoestática para estas geometrías de referencia.

El primer ejemplo consiste en el cálculo de la RCS monoestática de un triedro de material PEC. Se trata de un ejemplo de referencia ampliamente documentado que permite evaluar las capacidades del método de cálculo para caracterizar las múltiples interacciones que se producen entre sus caras debido a las reflexiones múltiples. Esta geometría supone un claro ejemplo con el que mostrar la importancia de modelar correctamente las reflexiones múltiples a la hora de obtener resultados precisos en la predicción de la RCS.

Se considera por tanto un triedro de $50 \times 50 \times 50 \lambda$ de arista, tal y como muestra la figura 3.4, que será analizado a la frecuencia de 300 MHz. Dadas sus dimensiones, el problema EM resultante supone un coste computacional considerable utilizando técnicas *full-wave*, en las que como es bien sabido se requieren discretizaciones del orden de $\lambda/10$ ($h = 0,1$). En nuestro caso, se ha aplicado MLFMA para obtener el resultado de referencia, implicando un total de 1.3 millones de incógnitas.

Para la resolución mediante PO se han considerado dos mallados uniformes. El primero con $h = 50$ y el segundo con $h = 5$, dando lugar a 3 y 348 facetas respectivamente. Se calculará la RCS en un plano con $\theta = 54,7^\circ$ y un margen angular acimutal de $0^\circ \leq \phi \leq 90^\circ$. La resolución angular es de $\Delta\phi = 1^\circ$ en el caso de la solución de referencia MLFMA (91 direcciones) y $\Delta\phi = 0,25^\circ$ (361 direcciones) para la caracterización mediante PO. Se considera polarización horizontal tanto para la onda incidente como para la dispersada (polarización HH).

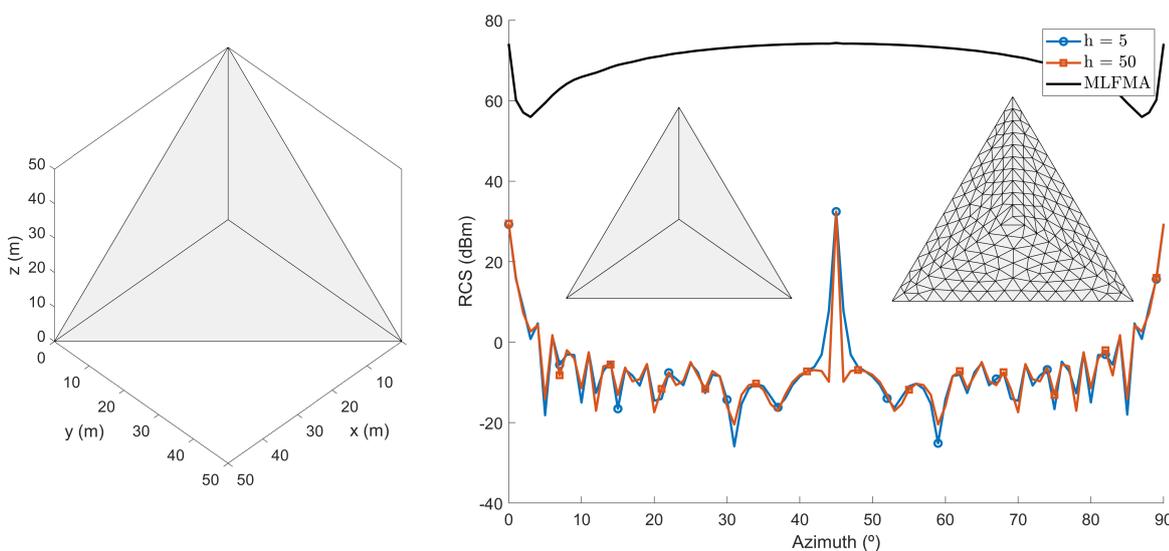


Figura 3.4: Comparativa de resultados obtenidos de la simulación MLFMA y PO para un triedro de 50λ de arista a 300 MHz.

Los resultados provistos por la figura 3.4 reflejan la incapacidad del método PO para determinar correctamente las múltiples interacciones, reportando unos 80 dB de diferencia con respecto a la referencia MLFMA, aproximadamente. No obstante, este error se da en menor medida para las direcciones de incidencia próximas a la perpendicular de las facetas del triedro ($\phi = 0^\circ$, $\phi = 45^\circ$ y $\phi = 90^\circ$), cuya reflexión directa es devuelta en mayor medida a los puntos de observación, siendo menores las reflexiones múltiples en el interior de la geometría. Aun así, la solución resulta inexacta incluso en el caso del mallado uniforme más denso, de 5λ .

No obstante, cabe destacar la notable diferencia de tiempos de cómputo entre el MLFMA y PO. El tiempo total requerido para el caso MLFMA ha sido de 40.86 h (para 91 direcciones) frente a los 0.97 s de la solución PO para las 361 direcciones.

Ha quedado demostrado que la solución PO resulta poco apropiada para modelar cavidades o elementos dispersores dependientes del acoplamiento mutuo entre facetas debido a la reflexión, modelando únicamente la contribución directa reflejada proveniente de las facetas perpendiculares a la dirección

de incidencia. Por tanto, el siguiente ejemplo trata de representar un problema arbitrario para una geometría de referencia compuesta por diferentes formas canónicas, denominada *Slicy* [67]. Esta geometría contempla un escenario genérico, cuyos elementos dispersores y concavidades suponen una contribución secundaria, aunque importante, y no un condicionante tan estricto como en el ejemplo anterior. Se trata a priori de un ejemplo más favorable para la estimación de RCS mediante PO.

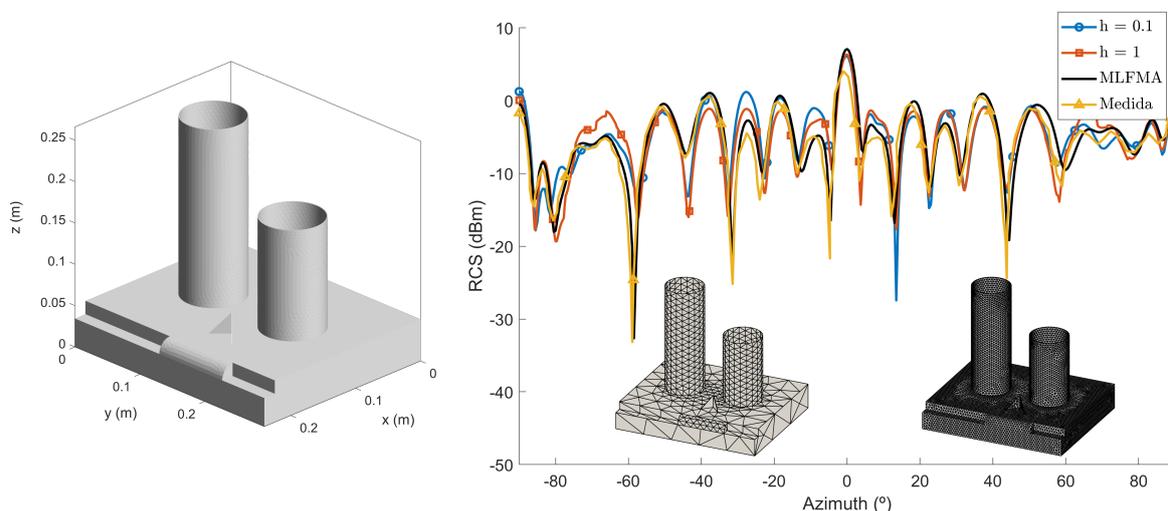


Figura 3.5: Comparativa de medidas y resultados MLFMA y PO de RCS monoestática para un Slicy a 8 GHz.

La geometría Slicy se detalla en la figura 3.5. Posee una dimensiones de 24.06 x 29.2 x 26.6 cm, y se ha analizado a la frecuencia de 8 GHz para obtener el cálculo de una RCS monoestática. El número total de incógnitas para una densidad de mallado de $\lambda/10$ ($h = 0,1$), como el requerido para el caso MLFMA, es de 137 340 incógnitas. Una cifra bastante más elevada que la proporcionada para los mallados uniformes con $h = 1$ y $h = 0,1$ escogidos para PO, de 1 721 y 91 560 facetas respectivamente. En este caso, el mallado ha tratado de ajustarse a los parámetros descritos, manteniendo la curvatura de las superficies. Es importante mencionar que las técnicas basadas en PO no precisan de mallados conformados entre las diferentes partes de un modelo, lo que permite minimizar el número de facetas necesario. La RCS monoestática se calcula para un barrido angular comprendido entre $-90^\circ \leq \phi \leq 90^\circ$, con una resolución de $\Delta\phi = 0,5^\circ$ y un total de 361 direcciones, con $\theta = 90^\circ$. La polarización escogida es en este caso vertical en transmisión y recepción, es decir, VV (*Vertical transmit and Vertical receive*).

Los resultados dispuestos en la figura 3.5 muestran una predicción más precisa que en la geometría anterior, en mejor sintonía por tanto con la simulación MLFMA y las medidas de referencia [67]. La disposición de la geometría con respecto a los puntos de observación y la contribución menos significativa de los dispersores con respecto a las contribuciones directas, permiten que el comportamiento electromagnético de la estructura pueda ser caracterizado con mayor precisión mediante el uso de PO. En este ejemplo, los tiempos requeridos de computo fueron de 2.54 h para el MLFMA, y de 6.44 s para el mallado $h = 0,1$.

3.1.2. Óptica física iterada (IPO)

La formulación del método de óptica física iterada, basada en el desarrollo original de [7], parte de las corrientes superficiales equivalentes deducidas de la aplicación de PO, lo que denominaremos en adelante solución de *primer orden*. Partiendo de esta solución con la contribución directa se establecerá un proceso recursivo que permitirá el cálculo de reflexiones de orden superior en el interior de la estructura y su contribución en las direcciones de observación, convergiendo de este modo a una solución cada vez más precisa en la que se incluirán mecanismos más allá de las contribuciones

directas.

En el caso concreto de la implementación elaborada en esta Tesis, se parte de la hipótesis de que la estimación de RCS de grandes superestructuras está conformada por varias reflexiones y diversos mecanismos de dispersión, como la formación de ondas progresivas o de superficie o las reflexiones múltiples en diedros, triedros y cavidades. Para los propósitos considerados en esta Tesis se ha considerado suficiente para representar la mayor parte de dichas contribuciones significativas la aplicación de reflexiones hasta tercer orden. No obstante, la extensión de la solución propuesta a órdenes superiores es inmediata, a expensas de la correspondiente penalización en tiempos de cómputo.

El método de óptica física iterada parte de la MFIE, que es resuelta de manera iterativa mediante la aplicación recursiva de la metodología de PO. Para ello, se aplica una descomposición jerárquica en el que las facetas iluminadas en el primer orden por la onda incidente, a las que denominaremos facetas *madre*, se convierten a su vez en fuentes radiantes que iluminan a las facetas denominadas *hijas*, que son aquellas facetas visibles desde la faceta madre. El procedimiento se repite de manera recursiva iluminando desde las facetas hijas a las facetas *nietas*, que proporcionan el tercer orden de interacciones. El proceso podría repetirse de manera recursiva hasta alcanzar un orden genérico K , aunque como ya se ha mencionado con anterioridad, se ha considerado que orden tres es suficiente para alcanzar las cotas de precisión que han sido requeridas en esta Tesis.

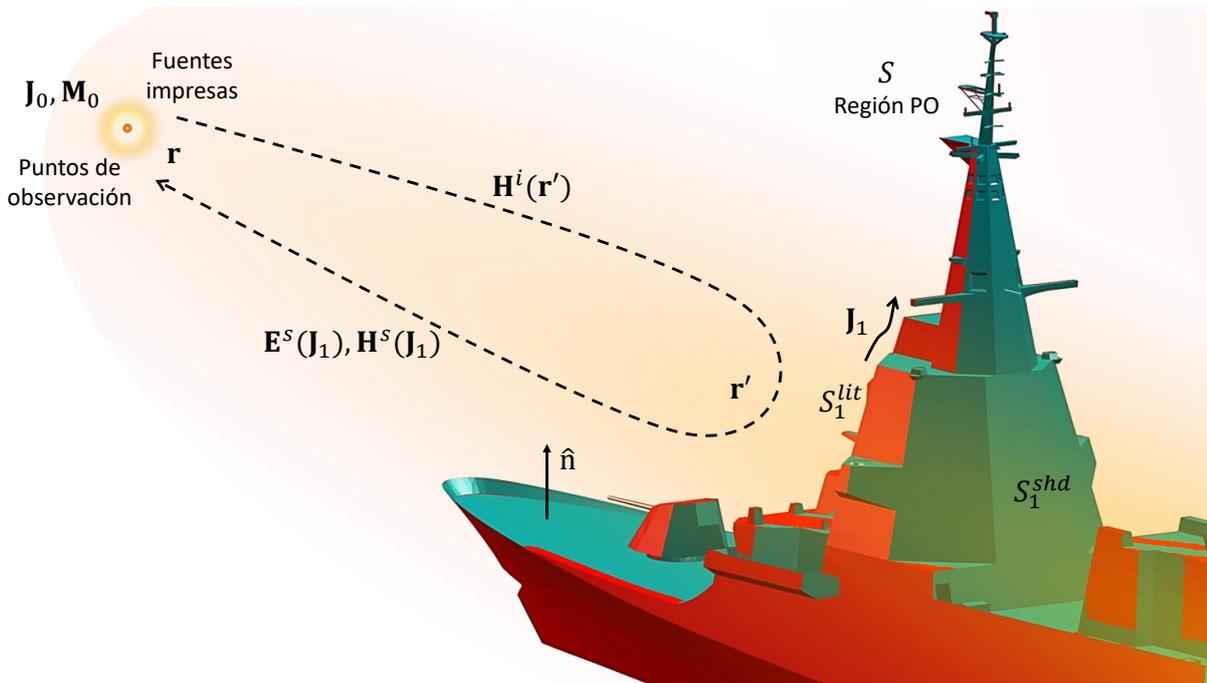


Figura 3.6: Representación de un problema equivalente para una geometría arbitraria como ejemplo de desarrollo de la formulación IPO, para el caso de incidencia directa o de 1º orden.

El proceso se detalla en la figura 3.6, que muestra el ejemplo de la figura 3.1 expresado ahora en términos de la formulación IPO. Se muestra en esta figura la reflexión directa o de 1º orden. Las facetas iluminadas por la fuente (onda incidente) dispersan un campo electromagnético calculado a partir de las corrientes J_1 inducidas sobre la superficie del blanco, que son calculadas mediante la aplicación de PO. En la figura 3.7 se muestran las contribuciones en el orden k debidas a la radiación de las corrientes J_{k-1} del orden $k-1$, para una de las facetas contenida en el conjunto de facetas S_{k-1}^{lit} , representada en amarillo. Cada faceta de este conjunto induce unas corrientes J_k sobre las facetas visibles correspondientes y realiza su contribución sobre los puntos de observación.

De esta forma, la ecuación 3.3 de la formulación PO puede ser reescrita en términos de la formulación

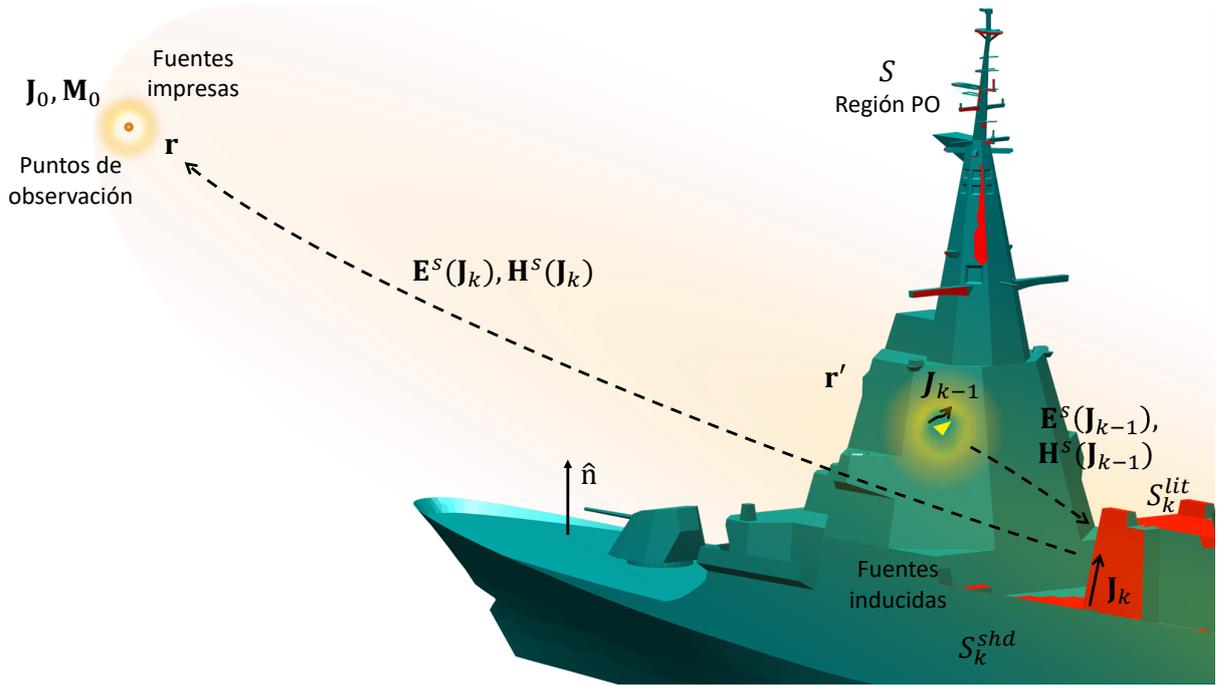


Figura 3.7: Representación de un problema equivalente, para una geometría arbitraria, como ejemplo de desarrollo de la formulación IPO, para el caso de una contribución de orden k .

IPO para el 1º orden como:

$$\mathbf{J}_1(\mathbf{r}') = 2\hat{n} \times \mathbf{H}^i(\mathbf{r}'). \quad (3.27)$$

Este proceso puede ser generalizado para una iteración k , donde las facetas *hijas*, *nietas*, ... iluminadas por la iteración anterior (representada en amarillo en la figura 3.7 para el caso de una de estas facetas, contenida en la superficie S_{k-1}^{lit}), sustituyen a las fuentes impresas como nuevos focos de radiación, tal que:

$$\mathbf{J}_k(\mathbf{r}') = 2\hat{n} \times \mathbf{H}^s(\mathbf{J}_{k-1}), \quad (3.28)$$

donde \mathbf{H}^s , descrito en la ecuación (3.4) para PO, es ahora evaluada a partir de las corrientes de la iteración anterior de la forma:

$$\mathbf{H}^s(\mathbf{J}_{k-1}) = \oint_S \mathbf{J}_{k-1}(\mathbf{r}') \times \nabla G(\mathbf{r} - \mathbf{r}') dS'. \quad (3.29)$$

Al desarrollar la ecuación (3.29) en base a la función de Green (2.9), como hicimos para el campo incidente en (3.2), obtenemos:

$$\mathbf{H}^s(\mathbf{J}_{k-1}) = \oint_S \left(j\beta + \frac{1}{R} \right) \hat{R} \times \mathbf{J}_{k-1}(\mathbf{r}') \frac{e^{-j\beta R}}{4\pi R} dS'. \quad (3.30)$$

Así pues, la densidad de corriente total viene dada por la suma de todas las contribuciones parciales como:

$$\mathbf{J}(\mathbf{r}') = \sum_{k=1}^K \mathbf{J}_k(\mathbf{r}'). \quad (3.31)$$

De igual modo, el campo \mathbf{E} y \mathbf{H} radiado en los puntos de observación puede ser expresado respectiva-

mente, tal y como recoge:

$$\mathbf{E}^s(\mathbf{r}) = \sum_{k=1}^K \mathbf{E}^s(\mathbf{J}_k), \quad (3.32)$$

$$\mathbf{H}^s(\mathbf{r}) = \sum_{k=1}^K \mathbf{H}^s(\mathbf{J}_k). \quad (3.33)$$

En la formulación original de IPO, el requisito de muestreo de las incógnitas viene impuesto por el teorema de Nyquist, para modelar correctamente el comportamiento oscilatorio de la corriente inducida. Si uno asume que la interacción entre los dos subdominios se puede aproximar mediante una onda plana, es decir, si cumple el criterio de campo lejano, entonces la corriente inducida en el dominio receptor se puede expresar como:

$$\mathbf{J}_k^{mn}(\mathbf{r}', \hat{\mathbf{v}}_{mn}) = \mathbf{J}_k(\mathbf{r}'_m) \cdot e^{-j\beta \hat{\mathbf{v}}_{mn} \cdot \mathbf{r}'}, \quad \mathbf{r}' \in \Delta_m, \quad (3.34)$$

siendo Δ_m el m -ésimo subdominio triangular y \mathbf{v}_{mn} el vector de incidencia entre la faceta m y n . La integral de radiación de un subdominio triangular plano, incluyendo el término de fase correspondiente, puede ser evaluada usando la formulación LPIT [68], desarrollada en el apartado 3.1.1.1, o de manera alternativa, usando la formulación de Gordon [69]. El único requisito es que el punto de observación debe obedecer la condición de campo lejano, aunque este criterio puede ser algo más relajado sin suponer una pérdida significativa de precisión.

Con la inclusión de los términos de fase, se puede reducir en consecuencia la densidad de malla. De esta manera es posible mantener una variación de amplitud lenta, a la vez que el término de fase de cada triángulo permite aproximar el comportamiento oscilatorio de las corrientes inducidas para una excitación determinada. La densidad de corriente inducida total en la m -ésima faceta debido a la radiación de la iteración anterior se puede calcular entonces como la siguiente superposición:

$$\mathbf{J}_k^m(\mathbf{r}') = \sum_{n=1}^N \mathbf{J}_k^{mn}(\mathbf{r}', \hat{\mathbf{v}}_{mn}). \quad (3.35)$$

Es importante recordar que esta suma sobre todo el conjunto de facetas es tan solo formal. En la práctica, se limita a aquellas facetas iluminadas en la iteración $k-1$ cuya línea de visión con el dominio m -ésimo no esté obstruida.

Como resultado, el coste computacional del algoritmo es de $O(N \cdot (K-1))$ en el caso de usar una discretización de la corriente mediante funciones de fase constante, y de $O(N^K)$ en el caso de usar discretización con términos de fase. En este último caso, tal como acabamos de mencionar, el número de facetas necesarias para modelar la corriente es tan sólo una fracción muy reducida de las necesarias en la resolución mediante técnicas *full-wave*, llegando a ser del orden de unas pocas facetas por longitud de onda cuadrada, lo que contrasta con las alrededor de 300 incógnitas por longitud de onda cuadrada necesarias en los métodos *full-wave*. En la práctica, al trabajar con blancos reales, el número de facetas en cada iteración de IPO, ya de por sí reducido gracias a la inclusión de los términos de fase, se limita aún más al ser necesario considerar tan sólo aquellas facetas iluminadas por cada faceta madre de la iteración anterior, reduciendo el coste computacional asociado al IPO.

Queda manifiesto por tanto la necesidad de disponer de las relaciones de visibilidades entre facetas, además de la visibilidad desde la fuente. Esto requiere de una nueva matriz de visibilidades que denominaremos *iVis*. Esta matriz supone el elemento con mayor coste de almacenamiento en memoria, de $O(N^2)$, con N el número de facetas LPIT. Sobre blancos reales se trata de uno de los procesos de cómputo más costoso, pese a definirse como una matriz de relaciones simétricas (las facetas visibles por una faceta ven a su vez a dicha faceta). No obstante, se trata de un cálculo dependiente únicamente de la geometría, por lo que solo requiere de ser realizado una sola vez para el conjunto de simulaciones que no requieran de modificaciones geométricas. Este es el caso del cálculo de la RCS monoestática para diversas configuraciones de elevación, acimut, frecuencia, etc. Estas y otras cuestiones, junto a los procedimientos y consideraciones necesarios para la obtención de esta matriz y la matriz *iLit*, serán desarrollados en la sección 3.2.5, tras introducir brevemente los conocimientos requeridos sobre el trazado de rayos.

3.1.2.1. Mallado adaptativo

El mallado de partida para la solución propuesta en esta Tesis no necesita ser conformado, como ya ha sido mencionado con anterioridad. Esto se debe a que la implementación permite la existencia de *T-vértices* (vértices cuya ubicación ha de encontrarse exactamente en el borde de una faceta, pero no tienen por qué ser el punto final del mismo). O lo que es lo mismo, no hay funciones de corriente definidas entre los triángulos adyacentes como, sucede en la formulación del MoM al utilizar bases RWG. Esta peculiaridad aporta un grado de libertad adicional y ayuda a reducir el tamaño inicial de la malla, que como hemos podido apreciar hasta ahora, conviene que resulte ser lo más reducido posible en cuanto a número de facetas a fin de minimizar el coste computacional asociado. Se puede partir por tanto de un mallado grueso, en el que la resolución obedece tan solo a restricciones de fidelidad geométrica y perfilado de zonas visibles/no visibles (perfilado de sombras). Posteriormente, se llevará a cabo un procedimiento de refinamiento adaptativo del mallado, siguiendo los criterios que se indican a continuación.

En primer lugar es preciso atender a los criterios de tamaño en base a la correcta aplicación de las funciones base LPIT con modelado lineal para la fase. Por regla general, se escoge el criterio estándar que divide un triángulo cuando es visto por otro desde una distancia inferior a la dictada por la región de Fraunhofer (campo lejano). No obstante, es posible escoger una condición más relajada, pero igual de representativa que el criterio estándar. Esto es posible ya que la necesidad de estipular una separación mínima entre facetas visibles surge como consecuencia de radiar desde una faceta a una región mayor que la contenida por el ángulo sólido de la faceta receptora.

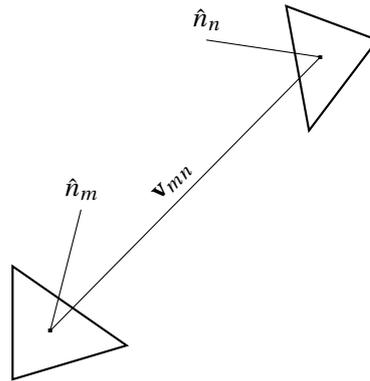


Figura 3.8: Esquema geométrico del criterio de subdivisión.

Para obtener dicho criterio, definiremos una serie de parámetros donde $R_{mn} = \|\mathbf{v}_{mn}\|$ es la distancia entre dos facetas m y n , $A_{m,n}$ son sus respectivas áreas, y los términos trigonométricos están dados por la proyección del correspondiente vector normal a cada triángulo en la dirección de iluminación, es decir, $\cos(\theta_m) = \hat{\mathbf{v}}_{mn} \cdot \hat{\mathbf{n}}_m$ y $\cos(\theta_n) = -\hat{\mathbf{v}}_{mn} \cdot \hat{\mathbf{n}}_n$. Para ello, partiremos considerando que el haz de la faceta m apunta a una n -ésima faceta, tal y como muestra la figura 3.8. La *ganancia* G_m de dicha faceta (aplicando aquí un símil de antenas de apertura) podría aproximarse por:

$$G_m = \frac{4\pi A_m}{\lambda^2} \cos\theta_m. \quad (3.36)$$

E inmediatamente el ángulo sólido del haz como:

$$\Omega_m = \frac{\lambda^2}{A_m \cos\theta_m}. \quad (3.37)$$

Por otro lado, de manera análoga, el ángulo subtendido de la faceta n , visto desde el centro de la faceta m , vendría dado por:

$$\Omega_n = \frac{A_n \cos\theta_n}{R_{mn}^2}. \quad (3.38)$$

Recordando que la condición de muestreo requiere que $\Omega_m > \Omega_n$, podemos combinar la ecuación (3.37) y (3.38), hasta deducir la condición:

$$R_{mn} > \zeta \frac{\sqrt{A_m A_n \cos(\theta_m) \cos(\theta_n)}}{\lambda}. \quad (3.39)$$

Finalmente, se introduce un factor multiplicativo ζ para controlar el crecimiento de la malla. Este factor determinará el tamaño de las matrices de visibilidades que posteriormente desarrollaremos en el subapartado 3.1.2.2. Cabe destacar que, en el caso de enfrentarse dos facetas idénticas, la ecuación (3.39) se reduciría a la fórmula habitual de campo lejano, hasta un determinado valor o factor de escala. La presencia de ambas facetas en la ecuación (3.39) es una forma de *reciprocidad*. Un concepto que se aplica a lo largo de todo el proceso de subdivisión, para cada una de las tareas restantes de la implementación propuesta, de la siguiente forma:

- La distancia mínima para todas las facetas que componen la malla se comprueba salvo para aquellas que cumplan la condición de control, es decir, aquellas que sean más pequeñas que un determinado valor. Este mecanismo de parada permite regular el tamaño final de la malla.
- Si una faceta no cumple la regla (3.39) (con respecto a todas las facetas visibles desde su posición), se descompone en 4 nuevas facetas que reemplazarán a la anterior faceta. La subdivisión se realiza generando 3 nuevos vértices en el punto medio de cada lado, que combinados con los 3 anteriores, conducen a la formación de 4 facetas con la misma relación de aspecto que la faceta original. Este proceso da lugar de manera natural a mallados no conformados, pero tal como se explicó anteriormente, esto no representa ningún inconveniente, más bien al contrario, permite optimizar el coste computacional sin afectar a la precisión.
- El proceso se repite hasta que todas las facetas cumplen el criterio de campo lejano, siendo necesario a continuación la actualización de las correspondientes relaciones de visibilidades entre facetas almacenadas en la matriz de visibilidades *iVis*. Esto puede realizarse mediante el cálculo de una nueva matriz, o mediante herencia de la matriz calculada originalmente, dependiendo del nivel de precisión de sombras deseado.

3.1.2.2. Matrices de visibilidades

Como podemos deducir del planteamiento expuesto a lo largo de esta formulación, se requieren estructuras que nos permitan relacionar y almacenar las visibilidades/ocultaciones entre facetas (matriz denominada *iVis*), además de las visibilidades/ocultaciones desde la fuente o la dirección de incidencia (vector *iLit*). Estas ocultaciones pueden llegar a definirse como binarias en el caso de considerarse completamente visibles o, por el contrario, completamente ocultas. Sin duda, esto supone ciertas deformaciones en los contornos de sombra en función de la resolución o discretización de la malla en las proximidades al contorno de la sombra. No obstante, este efecto resulta ciertamente mitigado, al menos parcialmente, en las áreas conflictivas debido a los procedimientos adaptativos de remallado comentados anteriormente.

La figura 3.9 muestra este efecto para las facetas visibles (representadas en rojo) interferidas desde la faceta de observación (representada en amarillo). Este caso correspondería a las relaciones de visibilidades almacenadas en la matriz *iVis* para una única faceta de observación. Contiene por tanto la información de todas las posibles facetas interferidas (visibles) desde su posición. A la vez, en la figura 3.10 se muestra un detalle de las ocultaciones (representadas en gris) sobre áreas conflictivas, cuyo mallado adaptativo permite precisar en mayor medida el contorno de las sombras. En ambos casos, estas matrices permiten obviar las contribuciones menos significativas de las regiones no interferidas, evitando además la formación de cavidades ficticias. Dichas cavidades ficticias se hubiesen formado por facetas de gran tamaño determinando triedros en las vértices que, como podemos observar, son descartados por el contorno de sombra.

El criterio para determinar la visibilidad entre facetas se realiza sobre el baricentro de cada triángulo, definiendo un grado de complejidad del problema de $O(N^2)$. Aunque el número de facetas no llega a ser tan elevado como el número de triángulos equivalente que requeriría un problema *full-wave*, éste sigue siendo considerable. Por ello, el coste computacional resulta excesivo, especialmente en el caso de la matriz $iVis$. En esta Tesis se propone una alternativa rápida en comparación con algoritmos de determinación de visibilidades/ocultaciones convencionales, basados en estructuras jerárquicas como los *kd-trees*. La implementación propuesta está basada en la utilización de algoritmos de trazado de rayos sobre GPU, que proporcionan unos resultados precisos en tiempos de ejecución realmente reducidos. Estas rutinas permiten calcular y almacenar las relaciones de visibilidades entre facetas en una matriz lógica. En particular, se ha hecho uso de la interfaz de programación de aplicaciones (*Application Programming Interface, API*) de NVIDIA OptiX™ [70], tal como se desarrollará detenidamente en la siguiente sección 3.2, basada en un motor de trazado de rayos sobre hardware específico. La implementación se basa especialmente en la serie de tarjetas gráficas NVIDIA RTX™, que poseen hardware dedicado para los procesos del trazado de rayo que requerimos en esta Tesis.

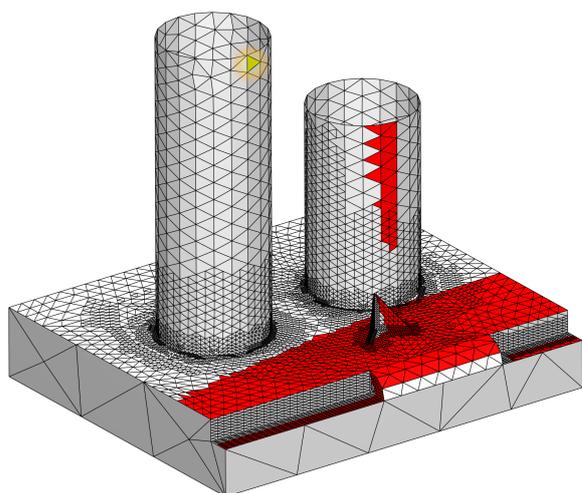


Figura 3.9: Relación de visibilidades $iVis$ representadas en rojo, para una faceta de observación representada en amarillo.

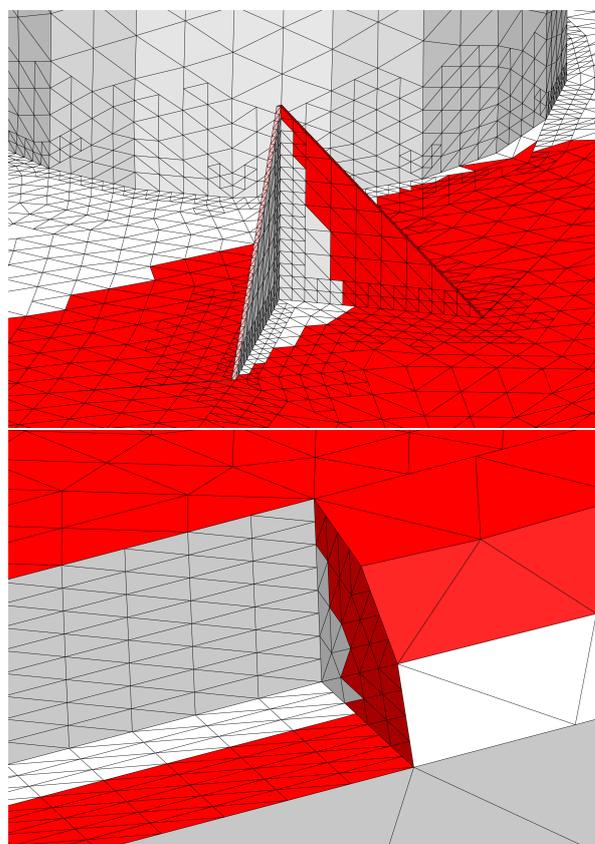


Figura 3.10: Contorno de sombra sobre áreas conflictivas debido al mallado.

3.1.2.3. Detalles de implementación

En este apartado desarrollamos el algoritmo de implementación de la metodología IPO de manera eficiente. Para ello partimos del algoritmo base (algoritmo 1, denominado IPO-Naive), describiendo un esquema básico de implementación, para posteriormente presentar una aceleración del mismo (algoritmo 2, denominado IPO-GO). Esta mejora presenta una aceleración importante en los tiempos de cómputo, la cual incorpora una aproximación basada en la óptica geométrica. Esta solución permite sustituir las iteraciones de mayor coste computacional (3º orden), por la radiación producida por una única faceta equivalente. Esta equivalencia mejora drásticamente la eficiencia del algoritmo y simplifica los procesos de radiación del algoritmo base.

Algoritmo 1 Algoritmo base (IPO-Naive).

```

1:  $\mathbf{E}^r = \mathbf{0}$ 
2:  $m \leftarrow \text{lit}(\theta, \phi)$ 
3: for  $m = \dots$  do ▷ 1º orden
4:    $\mathbf{J}_1^m \leftarrow 2\hat{n} \times \mathbf{H}^{i,m}$ 
5:    $\mathbf{E}^r \leftarrow \mathbf{E}^r + \mathbf{E}^s(\mathbf{J}_1^m)$ 
6:    $n \leftarrow \text{visible}(m)$ 
7:   for  $n = \dots$  do ▷ 2º orden
8:      $\mathbf{J}_2^n \leftarrow 2\hat{n} \times \mathbf{H}^s(\mathbf{J}_1^m)$ 
9:     if  $\text{lit}(\theta, \phi)$  then
10:        $\mathbf{E}^r \leftarrow \mathbf{E}^r + \mathbf{E}^s(\mathbf{J}_2^n)$ 
11:     end if
12:      $p \leftarrow \text{visible}(n)$ 
13:     for  $p = \dots$  do ▷ 3º orden
14:        $\mathbf{J}_3^p \leftarrow 2\hat{n} \times \mathbf{H}^s(\mathbf{J}_2^n)$ 
15:       if  $\text{lit}(\theta, \phi)$  then
16:          $\mathbf{E}^r \leftarrow \mathbf{E}^r + \mathbf{E}^s(\mathbf{J}_3^p)$ 
17:       end if
18:     end for
19:   end for
20: end for

```

Algoritmo 2 Algoritmo acelerado (IPO-GO).

```

1:  $\mathbf{E}^r = \mathbf{0}$ 
2:  $m \leftarrow \text{lit}(\theta, \phi)$ 
3: for  $m = \dots$  do ▷ 1º orden
4:    $\mathbf{J}_1^m \leftarrow 2\hat{n} \times \mathbf{H}^{i,m}$ 
5:    $\mathbf{E}^r \leftarrow \mathbf{E}^r + \mathbf{E}^s(\mathbf{J}_1^m)$ 
6:    $n \leftarrow \text{visible}(m)$ 
7:   for  $n = \dots$  do ▷ 2º orden
8:      $\mathbf{J}_2^n \leftarrow 2\hat{n} \times \mathbf{H}^s(\mathbf{J}_1^m)$ 
9:     if  $\text{lit}(\theta, \phi)$  then
10:        $\mathbf{E}^r \leftarrow \mathbf{E}^r + \mathbf{E}^s(\mathbf{J}_2^n)$ 
11:     end if
12:      $\mathbf{J}_2^{\text{GO}} \leftarrow \text{raytrace}(m)$ 
13:      $p \leftarrow \text{visible}(\mathbf{r}_{\text{GO}})$  ▷ 3º orden
14:      $\mathbf{J}_3^p \leftarrow 2\hat{n} \times \mathbf{H}^s(\mathbf{J}_2^{\text{GO}})$ 
15:     if  $\text{lit}(\theta, \phi)$  then
16:        $\mathbf{E}^r \leftarrow \mathbf{E}^r + \mathbf{E}^s(\mathbf{J}_3^p)$ 
17:     end if
18:   end for
19: end for
20:

```

En el primer caso, el proceso comienza identificando todas las facetas m que son visibles desde una ubicación radar determinada. Cada una de ellas proporciona una contribución $\mathbf{E}^s(\mathbf{J}_1^m)$ al campo total reflejado \mathbf{E}^r , a partir de las corrientes inducidas \mathbf{J}_1^m , constituyendo la contribución de 1º orden. No obstante, dichas facetas, que podemos considerar *madres*, también irradian sobre otras facetas n en la superficie del blanco, que pueden ser consideradas facetas *hija*, cuya contribución $\mathbf{E}^s(\mathbf{J}_2^n)$ estará condicionada a pertenecer al mismo tiempo al conjunto de facetas iluminadas por el radar, en cuyo caso constituirán la contribución de 2º orden, acumulándose a la anterior contribución de 1º orden. De igual modo que una faceta *madre*, una faceta *hija* puede radiar a su vez sobre otras facetas p , o facetas *nietas*, dando lugar por tanto a las contribuciones de 3º orden, siempre bajo la misma condición de que estas facetas nietas sean visibles también desde el radar. En resumen, el proceso completo puede verse como un árbol de expansión recursiva en tres niveles, fácilmente extensible iterativamente al número de niveles (orden) necesario.

Este algoritmo resulta el esquema más básico de implementar a partir de la formulación del método anteriormente descrita, cuya complejidad es de $O(N^3)$, con N el número de llamadas a las rutinas de radiación. Esta complejidad crece exponencialmente con cada orden de contribución, aunque sigue siendo dependiente de las visibilidades geométricas para desencadenar los sucesivos ordenes de reflexión. Como ya hemos mencionado anteriormente, en esta Tesis consideraremos tres órdenes como condición suficientes para caracterizar correctamente la dispersión incluyendo detalles y cavidades.

En el caso de objetivos realistas a frecuencias de interés, el algoritmo IPO-Naive resulta sencillo, pero costoso por cada nivel recursivo implementado (especialmente el de mayor orden). No obstante, es posible reducir la complejidad computacional del algoritmo de $O(N^3)$ a $O(N^2)$, manteniendo los efectos de 3º orden que son cruciales en objetivos reales. Por medio del trazado de rayos basado en el método GO, es posible sustituir el proceso iterativo de 3º orden sin repercutir sobre las reflexiones de 1º y 2º orden, que se mantienen intactas.

Esta aproximación utiliza las técnicas de trazado de rayos de la aproximación de GO para definir una faceta ficticia p , equivalente a la región iluminada de 3º orden. Cuya corriente superficial equivalente en dicha región es denotada por \mathbf{J}_2^{GO} , procedente del campo incidente de la fuente $\mathbf{H}^{i,n}$. Una vez obtenido \mathbf{J}_2^{GO} y p , se induce la corriente superficial equivalente \mathbf{J}_3^p sobre la faceta y finalmente se radia una única vez hacia el punto de observación, conformando así la contribución de 3º orden completa.

3.1.2.4. Ejemplo de aplicación

A continuación, expondremos un par de ejemplos canónicos, consistente en el triedro y el Slicy analizados en la sección 3.1.1.2 mediante PO. Como solución de contraste se emplearán los resultados de RCS monoestática calculados mediante MLFMA usando discretización uniforme con resolución de $\lambda/10$ ($h = 0,1$) y un objetivo de error residual en las corrientes equivalentes de 10^{-6} .

Durante el análisis de los resultados obtenidos para estos dos ejemplos, nos centraremos principalmente en 3 aspectos de mejora. El primero consistirá en la precisión del cálculo de RCS con respecto a la simulación MLFMA, para el cual analizaremos el grado de mejora de los resultados finales con respecto al método anterior. En segundo lugar, valoraremos el estudio paramétrico concerniente al mallado adaptativo, junto con su estimación de tiempos y el coste computacional asociado al crecimiento de la malla. En este estudio obtendremos los procesos necesarios para la obtención de las matrices de visibilidades, considerándolos despreciables para la estimación de tiempo total de cómputo (asumiendo su obtención en tiempo real, como detallaremos en apartados posteriores). Por último, destacaremos la aceleración del algoritmo GO frente al algoritmo base, comparándolos en términos de precisión y tiempo total para un mismo mallado.

Consideremos el problema canónico introducido en el apartado 3.1.1.2, consistente en un triedro PEC cuyas aristas están alineadas con los ejes. Las dimensiones del triedro son de $50 \times 50 \times 50 \lambda$ de arista, y se analiza a la frecuencia de 300 MHz, tal y como muestra la figura 3.11. La RCS monoestática se calcula en un barrido comprendido entre $0^\circ \leq \phi \leq 90^\circ$, con resolución angular de $\Delta\phi = 0,25^\circ$ (361 direcciones de incidencia), manteniendo una elevación de $\theta = 54,7^\circ$. La polarización considerada es la HH.

Para esta geometría se ha realizado un estudio paramétrico concerniente al mallado adaptativo, atendiendo a los dos parámetros fundamentales que definen el tamaño de la malla: el tamaño de la faceta en longitudes de onda, h , y el factor multiplicativo que controla el crecimiento de malla, ζ , tal como se recoge en la ecuación (3.39). La figura 3.12 se muestra como un ejemplo representativo de la influencia de ambos parámetros sobre el mallado adaptativo.

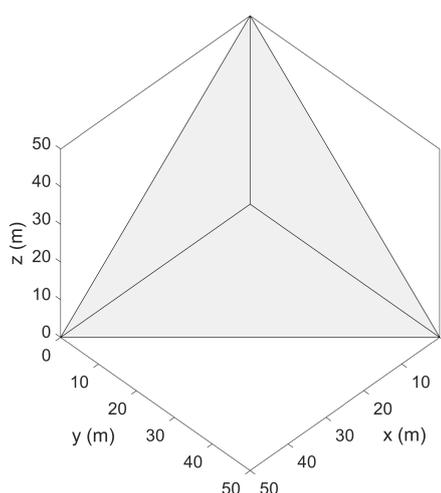


Figura 3.11: Triedro de 50λ de arista a 300 MHz.

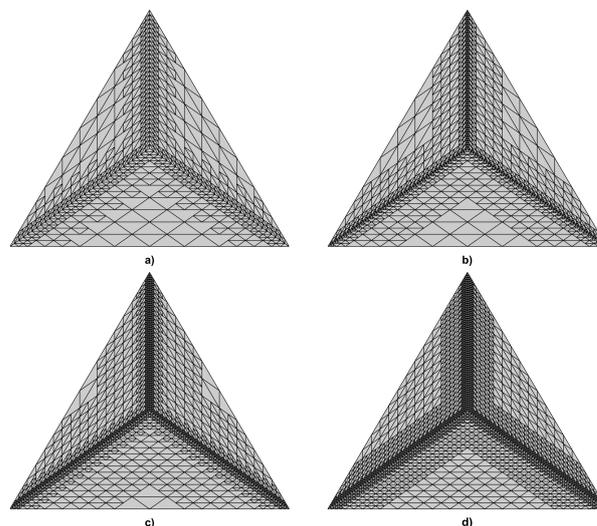


Figura 3.12: Mallado adaptativo para $h = 1$ y a) $\zeta = 1/2$, b) $\zeta = 2/3$, c) $\zeta = 1$ y d) $\zeta = 2$.

Los resultados obtenidos mediante IPO, dispuestos en las figuras 3.13 y 3.14, muestran en todos los casos mejoras evidentes con respecto a los resultados obtenidos mediante PO, recogidos en la figura 3.4. Esta mejora sustancial se debe a la inclusión de las contribuciones de 2º y 3º orden que incorpora

Tabla 3.1: Resultados de simulación del estudio paramétrico de un triedro de 50λ a 300 MHz mediante IPO

Parámetros	$h = 5$	$h = 4$	$h = 3$	$h = 2$	$h = 1$	$\zeta = 1/2$	$\zeta = 2/3$	$\zeta = 1$	$\zeta = 2$	$\zeta = 4$	MLFMA
Incógnitas	348	554	978	2115	8643	1083	1731	2631	3819	6060	1.3e6
Tiempo IPO-Naive (min)	0.1	0.29	2.07	19.43	830.61	1.52	5.69	21.05	78.56	274.57	2.45e3
Tiempo IPO-GO (min)	0.058	0.065	0.071	0.14	0.62	0.052	0.11	0.24	0.37	0.8	

la metodología IPO. En la tabla 3.1 se pueden observar los valores de tamaño de discretización (h) y factor de campo lejano (ζ), así como el número de facetas resultante para cada una de las simulaciones consideradas en el estudio paramétrico.

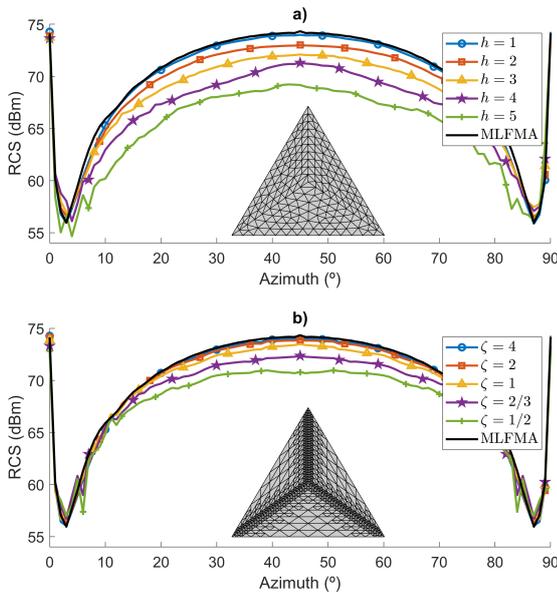


Figura 3.13: RCS monoestática de un triedro de 50λ a 300 MHz resuelto por el algoritmo IPO naive (a) malla uniforme (b) malla adaptativa.

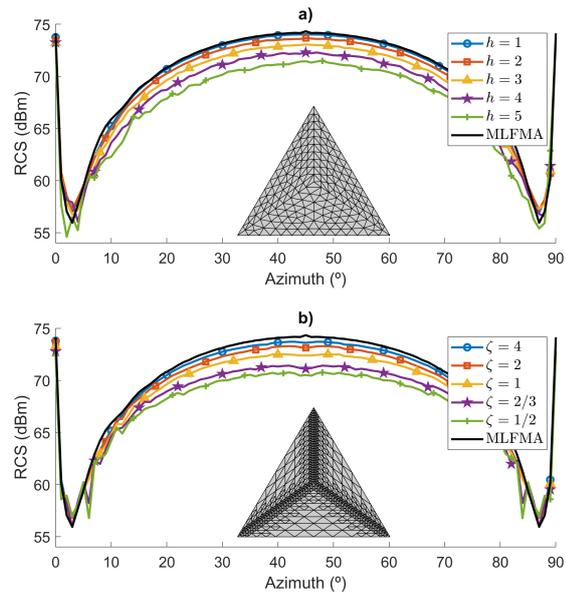


Figura 3.14: RCS monoestática de un triedro 50λ a 300 MHz resuelto por el algoritmo IPO-GO (a) malla uniforme (b) malla adaptativa.

En la figura 3.13 (a) se muestran los resultados que corresponden al estudio del parámetro h aplicando en todos los casos una malla uniforme, mientras que la figura 3.13 (b) muestra el estudio paramétrico para ζ considerando un tamaño de la malla uniforme de partida de $h = 50$ (es decir, se modela cada cara del triedro con una sola faceta) y una resolución de campo cercano de $h = 1$. Ambos resultados han sido obtenidos mediante la implementación naive de IPO (algoritmo base). Observando la figura 3.13 (a), se puede apreciar una mejora significativa como consecuencia de aumentar uniformemente la densidad del mallado pasando de $h = 5$ a $h = 1$, mostrando una elevada concordancia con el resultado de referencia para el caso de utilizar facetas de 1λ ($h = 1$). El nivel de precisión alcanzado con respecto a MLFMA resulta increíble. Esto permite deducir que el incremento del número de facetas mejora la estimación de contribuciones significativas, a diferencia de PO, donde la mejora con la discretización resulta irrelevante. La reducción en el número de facetas y tiempo de cómputo del algoritmo IPO con respecto a MLFMA, mostrados en la tabla 3.1, es indiscutible. Pasando ahora a analizar los resultados aplicando un mallado adaptativo en función del parámetro de campo lejano ζ , se observa que es posible reducir de manera considerable el número de facetas y, por ende, el tiempo de cómputo asociado. Se han aplicado factores de campo lejano entre $\zeta = 1/2$ y $\zeta = 4$. A mayor valor de ζ , mayor será la distancia considerada en campo cercano, dando lugar a una mayor extensión de refinamiento del mallado en torno a las discontinuidades geométricas. Así pues, para un resultado equivalente a ambos casos, el mallado uniforme $h = 1$ requiere de 13.84 horas, lo que contrasta con el resultado más preciso obtenido en este caso para el mallado adaptativo con $\zeta = 4$ (implicando 6060 facetas), que ha sido de 4.57 horas. Para valores de ζ menores,

los tiempos pueden llegar a ser tan reducidos como 1.52 minutos para $\zeta = 1/2$. Cabe destacar, por otra parte, que este ejemplo resulta bastante conservador, puesto que todas las facetas son vistas para todas las direcciones de incidencia del estudio y visibles entre sí, por lo que los tiempo de cómputo son los más elevados posibles para una geometría dada (todas las facetas poseen contribuciones de segundo y tercer orden).

Por último, queda analizar el grado de mejora del algoritmo acelerado, frente al algoritmo base. En este caso dispuesto en la figura 3.14, de manera análoga a la figura 3.13. Esta mejora resulta poco representativa al comparar ambas figuras, en la que podemos intuir una leve mejora por parte del algoritmo acelerado con respecto al base. No obstante, al acudir a los valores de tiempo recogidos en la tabla 3.1, observamos la mejora sustancial. La aceleración resulta extraordinaria, concluyendo que los cálculos de las contribuciones de 3º orden suponen prácticamente la totalidad del tiempo de cómputo.

El siguiente ejemplo trata de representar un problema arbitrario para una geometría de referencia compuesta por diferentes formas canónicas, denominada Slicy [67]. La geometría PEC descrita en la figura 3.15 posee una dimensiones de 24.06 x 29.2 x 26.6 cm, la cual es analizada a 8, 14 y 60 GHz para obtener el cálculo de una RCS monoestática. Dicha RCS se calcula para un barrido angular comprendido entre $-90^\circ \leq \phi \leq 90^\circ$, con una resolución angular $\Delta\phi = 0,5^\circ$ (361 direcciones de incidencia), manteniendo una elevación de $\theta = 90^\circ$. La polarización considerada es la VV. La polarización considerada es la VV.

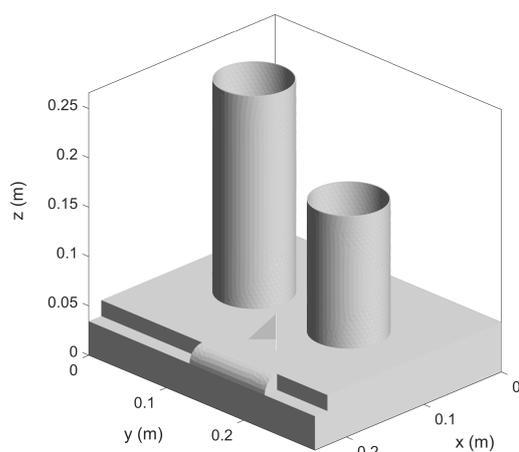


Figura 3.15: Geometría de referencia (Slicy).

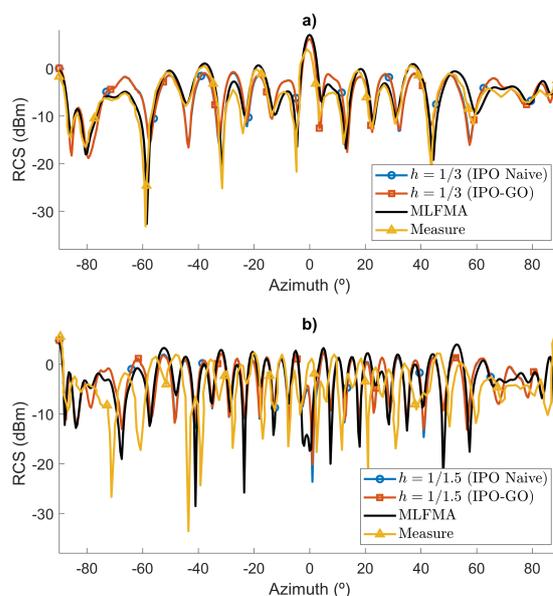


Figura 3.16: Comparativa de medidas y solución MLFMA e IPO de RCS monoestática de un slicy a) a 8GHz b) a 14 GHz.

Los resultados obtenidos mediante IPO, dispuestos en la figura 3.16, muestran en todos los casos la buena correlación entre los resultados IPO y la solución de contraste obtenida por MLFMA y las medidas recogidas en la cita anterior. La figura 3.16 (a) muestra la comparativa de resultados obtenidos entre el algoritmo naive, el algoritmo GO (ambos para $h = 1/3$) y las soluciones de contraste para un análisis a 8 GHz, mientras que la figura 3.16 (b) muestra la misma comparativa para un análisis a 14 GHz ($h = 1/1,5$). Ambos resultados resultan en la línea de los obtenidos mediante PO, recogido en la figura 3.5. Al comparar ambas figuras la mejora resulta poco representativa, en la que podemos intuir una leve mejora por parte de cualquiera de los algoritmos IPO con respecto a PO. No obstante, al acudir a los valores de tiempo recogidos en la tabla 3.2, observamos, al igual que en el ejemplo del triedro, una aceleración sustancial.

Tabla 3.2: Resultados de simulación del estudio paramétrico de un Slicy mediante IPO.

Frecuencia (GHz)	8		14				60								
Parámetros	$h=1/3$	MLFMA	$h=1/1,5$	MLFMA	$h=5$	$h=4$	$h=3$	$h=2$	$h=1$	$\zeta=1/2$	$\zeta=2/3$	$\zeta=1$	$\zeta=2$	$\zeta=4$	MLFMA
Incógnitas	5000	137340	5000	279366	1934	2226	3822	8186	33102	2570	2885	4127	6962	11444	4.97e6
Tiempo IPO-Naive (min)	0.051	152.8	0.051	1.91e3	0.024	0.029	0.051	0.1	0.45	0.05	0.052	0.064	0.1	0.12	1.14e4
Tiempo IPO-GO (min)	0.049		0.049		0.024	0.025	0.042	0.075	0.37	0.036	0.039	0.051	0.07	0.1	

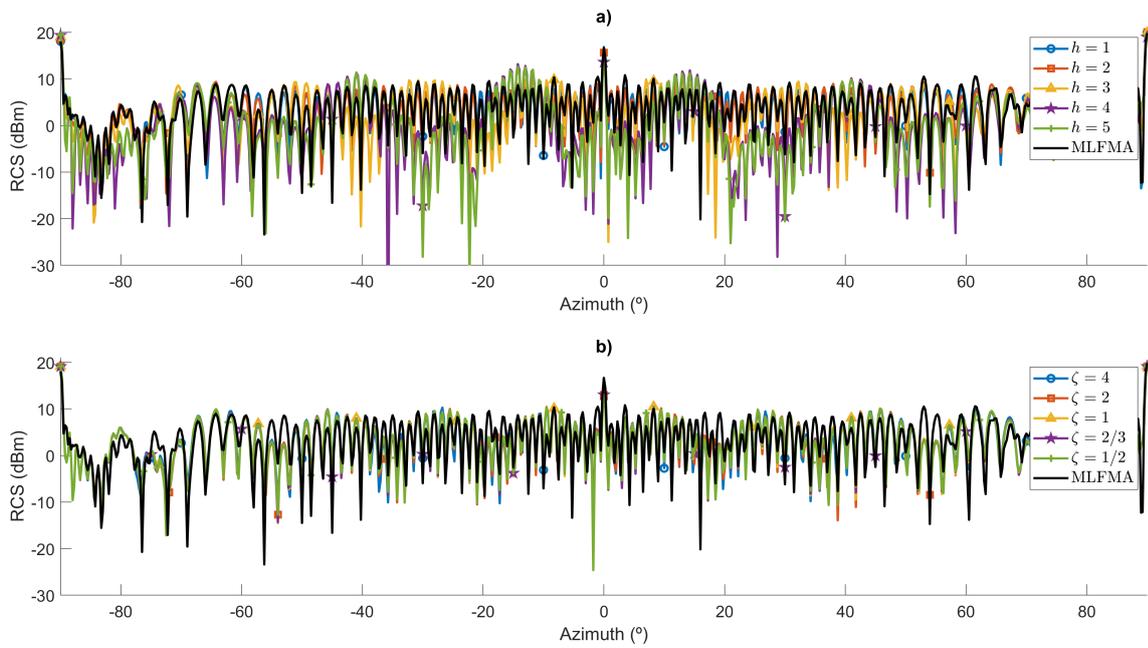


Figura 3.17: RCS monoestática de un Slicy a 60GHz resuelto por el algoritmo IPO naive (a) malla uniforme (b) malla adaptativa.

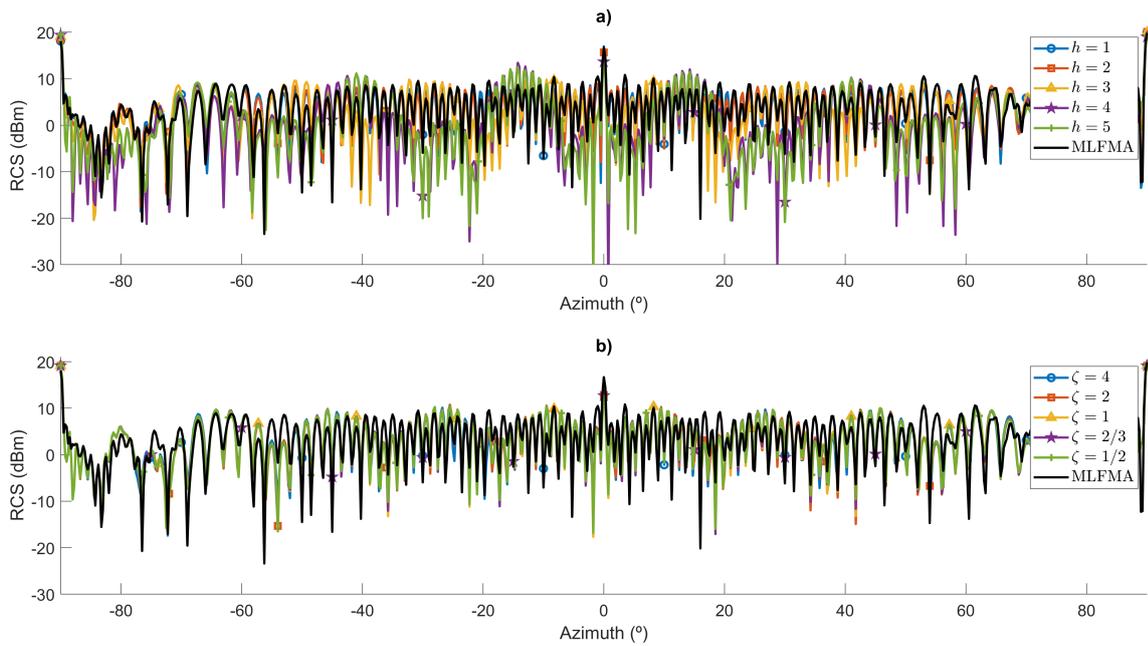


Figura 3.18: RCS monoestática de un Slicy a 60GHz resuelto por el algoritmo IPO GO (a) malla uniforme (b) malla adaptativa.

Estas conclusiones pueden deducirse, de igual forma, del estudio paramétrico concerniente al mallado adaptativo, atendiendo a los dos parámetros estudiados en el ejemplo anterior (h y ζ), para un análisis a 60 GHz. En las figuras 3.17 (a) y 3.18 (a) se muestran los resultados que corresponden al estudio del parámetro h aplicando en todos los casos una malla uniforme, mientras que las figuras 3.17 (b) y 3.18 (b) muestran el estudio paramétrico para ζ considerando un mallado de partida lo más grueso posible, manteniendo la curvatura de las superficies, y una resolución de campo cercano de $h = 0,5$. Ambos pares de representaciones obtenidos mediante la implementación naive y GO de IPO respectivamente. Observando el primer par de imágenes se puede apreciar una mejora significativa como consecuencia de aumentar uniformemente la densidad del mallado pasando de $h = 5$ a $h = 1$, al igual que en triedro, y observando el segundo par de imágenes una considerable reducción del número de facetas y, por ende, el tiempo de cómputo asociado, aplicando factores de campo lejano entre $\zeta = 1/2$ y $\zeta = 4$. Ante este considerable incremento en la complejidad del problema, tal y como recoge la tabla 3.2 en base al número de facetas necesario para cada caso, y la precisión de la solución obtenida, podemos asegurar la escalabilidad del método IPO con respecto a la frecuencia de análisis. En esta tabla se puede apreciar fácilmente la magnitud de la aceleración del método IPO, al reducir drásticamente el tiempo de cómputo necesario mediante MLFMA de 7.91 días a 6 segundos para la obtención del resultado más preciso ($\zeta = 4$) y de 1.44 segundos para el resultado más rápido ($h = 5$) mediante IPO.

3.2. Trazado de rayos: NVIDIA OptiX API

Hasta hace poco, las CPUs (*Central Processing Unit*) habían constituido el principal centro de cómputo para las labores de simulación, renderizado y cálculo computacional. No obstante, con el desarrollo de la tecnología y la incesante demanda de contenido susceptible de procesamiento gráfico, las GPUs han experimentado una considerable evolución, con el fin de explotar los elevados grados de paralelización que poseen. Este auge ha permitido extender su ámbito de aplicación desde el procesamiento gráfico a la ejecución masivas de operaciones simples para el minado de criptomonedas y otras muchas aplicaciones, en auge hoy en día. Por esta razón, la tecnología ha mejorado y se han abaratado los costes de producción, permitiendo disponer de un hardware específico de altas prestaciones, de manera asequible con respecto a la potencia equivalente provista por las CPUs.

Originalmente, las GPUs fueron diseñadas para la paralelización masiva de instrucciones simples mediante cientos de núcleos (*Single Instruction Multiple Threads*, SIMT). Esto significa que la misma instrucción se ejecutaba simultáneamente en múltiples hilos. La idea detrás de SIMT era reducir la sobrecarga al obtener instrucciones, por ello, las GPUs fueron optimizadas para un alto rendimiento en el intercambio de memoria, a diferencia de las CPUs centradas en obtener un acceso de baja latencia, tal y como muestra la figura 3.19. Esta latencia con respecto a las CPUs, quedó enmascarada debido al diseño de una arquitectura hardware eficiente que aumentó el número de transistores (representados en verde) dedicados al procesamiento de datos [71]. Este aspecto supuso un cambio con respecto a la arquitectura CPU tradicional, que junto al gran ancho de banda y la velocidad de los transistores, redujo la necesidad de una memoria caché. En definitiva, el cambio de la arquitectura y la ejecución SIMT redujeron la necesidad de un control de flujo y una aceleración considerable en la paralelización masiva de procesos.

Estas excelentes consideraciones hicieron de la arquitectura GPU la mejor opción para el desarrollo de aplicaciones en el trazado de rayos, las cuales requerían la ejecución masiva de tareas sencillas. La paralelización de un número elevado de ejecuciones simples permitió crear algoritmos capaces de predecir y simular el comportamiento de la luz. No obstante, y hasta el momento, los algoritmos de trazado de rayos han resultado ser irregulares e ineficientes en su balanceo de carga, limitando así su completo potencial en el desarrollo de aplicaciones complejas.

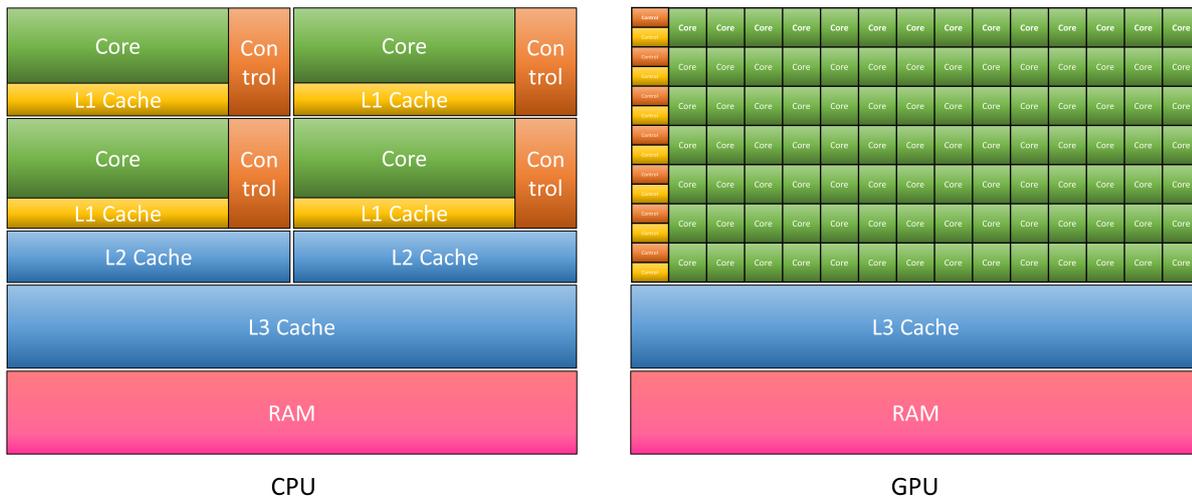


Figura 3.19: Comparación entre la arquitectura CPU vs GPU.

3.2.1. Aplicaciones del trazado de rayos

El origen del trazado de rayos tal y como lo conocemos, puede atribuírsele a Turner Whitted, el cual lo introdujo por primera vez en [72]. Aunque este a su vez, se basó en la técnica de emisión de rayos (*ray casting*) de Arthur Appel, presentada en [73], extendiendo el *ray casting* de manera recursiva para obtener reflexiones y sombras de manera computacional. En cualquier caso, desde aquel entonces hasta nuestros días han surgido un sin fin de avances tecnológicos y aportaciones, que abarcan un amplio rango de aplicaciones muy diversas. Por citar alguna de las más recientes, la detección de colisionamientos en métodos como GO y UTD [74], la propagación de ondas electromagnéticas [75, 76] o el renderizado de imágenes tridimensionales de gran tamaño en el contexto de la biomedicina y las nanopartículas [77, 78]. Aunque principalmente se encuentran centradas en aplicaciones de videojuegos y renderizado de imágenes y vídeos [79]. Para todo este abanico de aplicaciones, parece haber un elemento común a todos ellos y se trata de la tecnología del trazado de rayos de NVIDIA.

Recientemente, NVIDIA aportó una solución novedosa mediante algoritmos de trazado de rayos para el desarrollo de aplicaciones sobre hardware específico, NVIDIA OptiX API [70]. Para ello, desarrolló un conjunto de herramientas de código libre y hardware privativo, constituidas por dos elementos interrelacionados: una API propia, que define estructuras de datos apropiadas para *ray tracing*, y un sistema basado en los lenguajes de programación C++ y CUDA (*Compute Unified Device Architecture*). Estos dos elementos permiten operar a tan bajo nivel que lo convierten en una herramienta flexible, con la que poder aprovechar todo el potencial del hardware en determinadas tareas, como el cálculo de visibilidades IPO que desarrollaremos en esta Tesis. No obstante, existen otras alternativas como la API estandarizada Vulkan Ray Tracing o DirectX Ray Tracing, diseñadas por Vulkan y Microsoft respectivamente en colaboración con NVIDIA. Estas alternativas se basan en sombreadores de cálculo (*shaders*) en vez de hardware específico, los cuales resultan muy similares, aunque algo menos potente. Por tanto, OptiX supone una aplicación mucho más optimizada y la opción escogida para esta Tesis.

NVIDIA provee de una documentación bastante completa para cada versión de la API de OptiX (en nuestro caso la 5.1 [80]), junto a una colección de ejemplos implementados por OptiX SDK (*Software Development Kit*). Este conjunto de ejemplos abarca diversas aplicaciones representativas de las capacidades de la herramienta, los cuales se recomienda su estudio en profundidad para una mayor comprensión. A su vez, NVIDIA publica artículos científicos y de divulgación, recogidos en [81, 82], de las aplicaciones más novedosas en materia de trazado de rayos en base a su tecnología.

3.2.2. Estructura de la arquitectura OptiX

OptiX es un sistema de programación de trazado de rayos, diseñado por NVIDIA GPUs, que permite mediante un arquitectura altamente paralelizable sintetizar las principales claves de los algoritmos de trazado de rayos en un reducido conjunto de operaciones programables. El núcleo fundamental de OptiX reside en la compilación de sentencias específicas en tiempo real para la generación de rayos a disposición de los programas definidos por el usuario. La ejecución masiva de estas tareas sencillas resulta extremadamente rápida debido a la escalabilidad y el elevado número de núcleos disponibles en las arquitecturas de GPUs. Esto permite la implementación de una amplia variedad de algoritmos de trazado de rayos y aplicaciones de renderizado, inteligencia artificial o, como la que nos atañen, detección de colisiones y simulación. Además, OptiX provee de múltiples aceleradores y módulos compactos que permiten la optimización de recursos en función de las aplicaciones a desarrollar.

Como ya hemos mencionado anteriormente, se trata de una tecnología basada en hardware específico, provista de núcleos destinados a la aceleración del cálculo de trazado de rayos. En concreto, la aceleración de la jerarquía del contorno del volumen (*Bounding Volume Hierarchy, BVH*), de la que posteriormente se hablará en la memoria, destinada a las funciones de detección de colisiones de rayos y triángulos. El cálculo de estas interacciones resulta realmente costoso en términos de cómputo mediante procedimientos convencionales, para los que el hardware se encuentra además limitado por tres aspectos fundamentalmente: el acceso a memoria, el control del flujo de datos, y el elevado computo de operaciones en coma flotante.

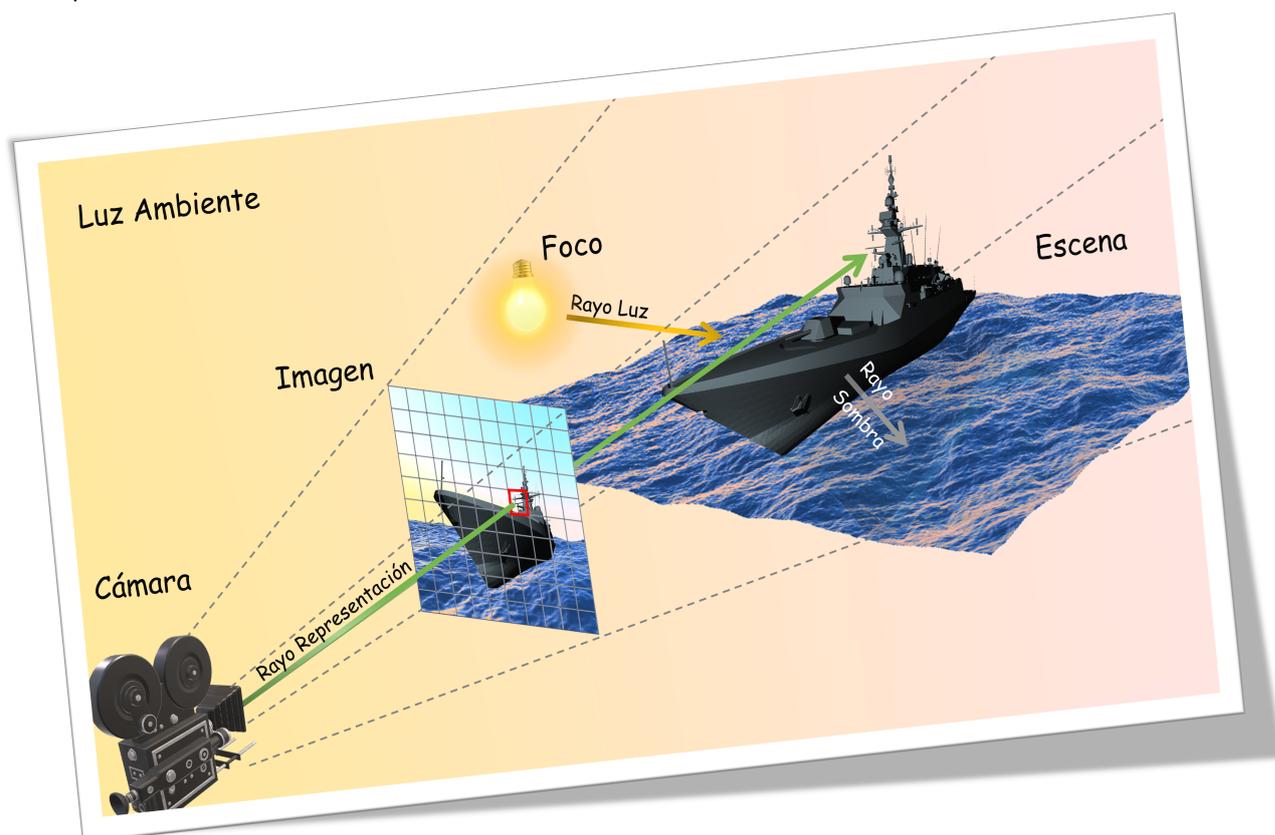


Figura 3.20: Esquema gráfico del trazado de rayos a través de la arquitectura OptiX.

Por otra parte, la arquitectura CUDA, diseñada por NVIDIA para codificar algoritmos de GPU, ha sido tradicionalmente compleja. La programación de rutinas de cierta dificultad resultan excesivamente tediosas con este lenguaje, y la interconexión con otros sistemas es también costosa en términos de implementación y operación, al no tener acceso directo a la memoria principal de la CPU. La ventaja del conjunto de herramientas de desarrollo proporcionadas por las librerías de OptiX reside en el

primer elemento anteriormente mencionado, la implementación de una API. Ésta es la encargada de gestionar la comunicación entre sistemas, entendiéndose por sistemas las CPUs asociadas a los códigos en C++ y las GPUs, asociadas a los códigos en CUDA. Mediante un conjunto reducido de sentencias, resulta relativamente sencillo realizar la adecuada gestión de las variables comunes a ambos sistemas. La paralelización, que por otra parte es la pieza fundamental de la aceleración, queda relegada a operaciones de alto nivel, basadas en perfiles escogidos por el usuario y gestionada de manera interna. Esto facilita enormemente la implementación, sin menoscabo de la adecuación de las características de los procesos a las necesidades del usuario.

Las herramientas proporcionadas por la API permiten simplificar y abstraer en gran medida la complejidad del problema, desligando la escena de la observación. Esta escena atañe a la realidad del problema, mientras que la observación atañe a la representación de los resultados obtenidos, como muestra la figura 3.20. Gracias a ello, la compatibilidad con otras herramientas se vuelve inmediata, como es el caso de los entornos gráficos que proveen OpenGL y Direct3D. Se trata de dos de las herramientas más utilizadas para proveer de un entorno de representación inmediato, por medio de una serie de librerías de funciones implementadas en C++. Estos entornos permiten representar gráficamente los resultados obtenidos en OptiX, asociando las consecuencias de cada uno de los rayos generados a un pixel de la imagen resultante, como se mostrará en apartados posteriores.

3.2.3. Programas definidos por el usuario

El proceso de trazado de rayos es un mecanismo recursivo, cuyo origen parte de una fuente y los procesos de reflexión, refracción, etc. determinan su fin o continuidad. Se trata de un esquema de procesos que involucra diversos factores que pueden ser independizados y secuenciados. A estos conjuntos de procesos se les denomina *programas*.

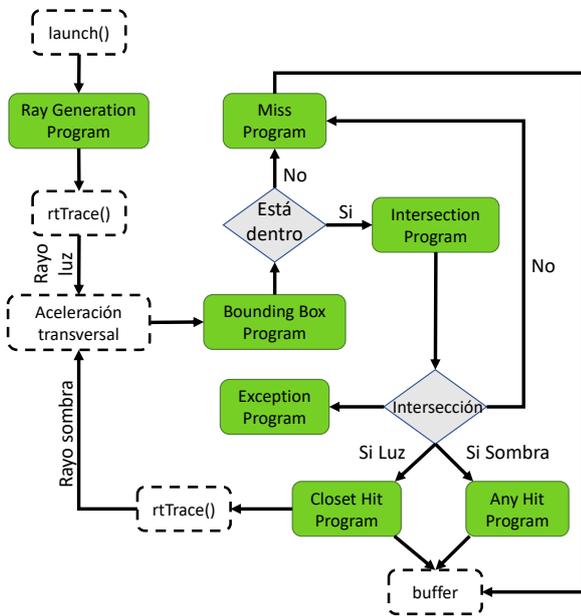


Figura 3.21: Esquema de procesos de OptiX.

Algoritmo 3 Pseudocódigo del esquema de procesos de OptiX.

```

1: context-> createAcceleration("Trbvh");
2: context-> launch( width, height );
3: for launch_index = width*height do
4:   Ray_Generation_Program();
5:   rtTrace(object, launch_index);
6:   if Bounding_Box_Program( launch_index ) then
7:     if Intersection_Program( launch_index ) then
8:       if launch_index = PerRayData_radiance then
9:         Closest_Hit_Program();
10:        output_buffer = color;
11:      else if launch_index = PerRayData_shadow then
12:        Any_Hit_Program();
13:        output_buffer[launch_index] = color*atten;
14:      else
15:        Exception_program();
16:        code = rtGetExceptionCode();
17:        output_buffer[launch_index] = bad_color;
18:      end if
19:    else
20:      Miss_Program();
21:      output_buffer = color_background;
22:    end if
23:  else
24:    Miss_Program();
25:    output_buffer = color_background;
26:  end if
27: end for
28: Buffer_out = getOutputBuffer( output_buffer );
29: displayBufferGL( Buffer_out );
30: write_iLit1_iLit2_iVis();

```

Los *programas* son conjuntos de procesos con un objetivo común, correspondiente a una de las etapas del esquema de procesos de OptiX, representados en verde en la figura 3.21 y dispuestos en el algoritmo 3. Son aplicables a todos los rayos y programados en CUDA, por tanto ejecutados

concurrentemente. Estos *programas* son implementados de manera independiente e interconectados con la API a través de los ficheros de ejecución paralela multihilo (*Parallel Thread Execution*, PTX), que a su vez, son generados por el compilador que provee CUDA SDK, *nvcc*. Este compilador actúa como una máquina virtual de ejecución de instrucciones paralelas a bajo nivel, permitiendo traducir dichos procesos a la arquitectura de conjunto de instrucciones (*Industry Standard Architecture*, ISA) reconocida por la GPU del equipo sobre el que se esté ejecutando. Hasta el momento ha sido testada la versión CUDA 10.0 sobre el sistema operativo Linux en la distribución Ubuntu (versión 18.04.2 LTS (Bionic Beaver)) y openSUSE (versión openSUSE Leap 42.3).

En los siguientes apartados se abordan los programas aplicados en esta Tesis, describiéndolos brevemente y detallando las consideraciones tenidas en cuenta a la hora de su implementación.

3.2.3.1. Ray Generation

Constituye el inicio del esquema de procesos y uno de los programas indispensables en el proceso de desarrollo basado en OptiX. Este programa es precedido por la sentencia 2 del algoritmo 3, que instancia la llamada al *Ray Generation Program* y da comienzo a la ejecución concurrente desde el código en C++. Esta sentencia fija el número de rayos totales que intervendrán en la escena (*width* x *height*), distribuidos uniformemente sobre la cuadrícula del sistema de coordenadas imagen. Este sistema de coordenadas resulta independiente del sistema de coordenadas píxel, como veremos en apartados posteriores. El programa de generación de rayos es ejecutado simultáneamente para cada rayo de representación, que finalmente constituirá uno o varios píxeles de la imagen resultante, como muestra la figura 3.22.

Para cada hilo el programa calcula la dirección de incidencia de un rayo como un vector que parte del origen (cámara) hacia el destino (escena). En su recorrido se realizan transformaciones a cuatro sistemas de coordenadas distintos. Dichas transformaciones se realizan tanto en el código C++ como en CUDA. En el caso del *Ray Generation Program*, al tratarse de procesos que transcurren en el ámbito de la escena, se trabaja en el sistema de coordenadas de la escena (variables *U*, *V*, *W*). No obstante, intervienen factores que requieren del sistema de coordenadas imagen (definido por la variable *d* en la línea 7 y 13 del código código 3.1) y el sistema de coordenadas de la cámara (a través de la variable *ray_origin* definida en la línea 3 del código código 3.1).

En el código 3.1 se muestra un fragmento con los elementos indispensables que han de estar presentes en un programa de generación de rayos como es el caso de la estructura *Ray*, *PerRayData_radiance*, la sentencia *rtTrace* y el *buffer* de salida *output_buffer*, considerando la variables no definidas como variables globales externas al programa. Ambas estructuras *Ray* y *PerRayData_radiance* se encuentran asociadas e identificados por el índice *launch_index*. La primera contiene los parámetros de origen y destino y la segunda, parámetros de interés que el usuario puede asociar a dicho rayo como el brillo en caso de ser atenuado o absorbido por materiales, o el resultado o color. La sentencia *rtTrace* es la puerta de entrada y salida hacia los sucesivos programas, en el orden dictado por el esquema de procesos. De esta manera, queda ligado el concepto de rayo de representación a lo que posteriormente será reinterpretado como color de un píxel. Dichos valores de color se irán almacenando en posiciones ordenadas del *buffer* de salida para ser transferidos al código C++.

En la línea 10 se aplica la transformación del sistema de coordenadas de la cámara al de la escena, basada en la perturbación del vector de incidencias del rayo de representación para obtener una representación en perspectiva. A su vez, se incluye la transformación del sistema de coordenadas de la escena de cartesianas a esféricas, mediante la bandera *camera_fish*. Esta transformación será utilizada para las posteriores representaciones de la imagen con el denominado efecto *ojo de pez*, que permite representar los 360° en acimut y 135° en elevación (de -45° a 90°) de la escena 3D, en una única imagen 2D.

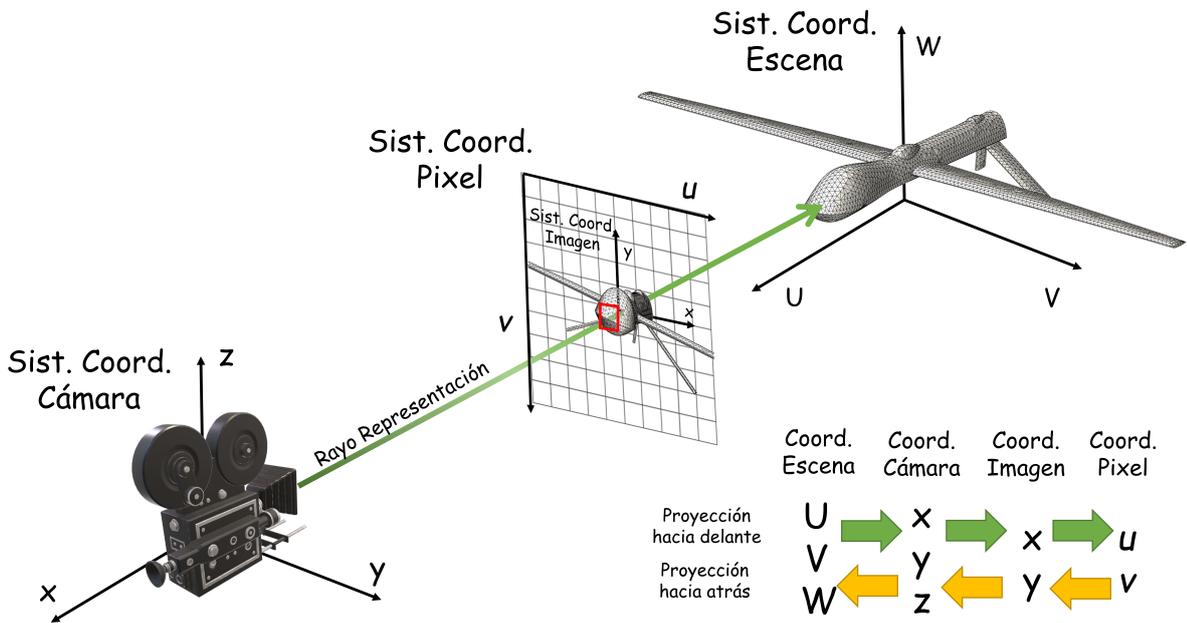


Figura 3.22: Esquema gráfico del trazado de rayos a través de los distintos sistemas de coordenadas de la arquitectura OptiX.

Código 3.1: Código del Ray Generation Program

```

1 void Ray_Generation_Program() {
2     float2 screen = make_float2(output_buffer.size())-1.0;
3     float3 ray_origin = eye;
4     float3 ray_direction;
5
6     if( camera_fish == 0 ){
7         float2 d = make_float2(launch_index)/screen*2.f-1.f; // Coord. Imagen
8
9         // Vista en perspectiva
10        ray_direction = d.x*U+d.y*V+W; // Coord. Escena
11    }
12    else {
13        float2 d = make_float2(launch_index) / screen * make_float2(2.0f*M_PIf,
14        3.0f*M_PIf/4.0f) + make_float2(M_PIf+M_PIf/2, M_PIf/4.0f);
15        float3 angle = make_float3(-cos(d.x)*sin(d.y), -cos(d.y), sin(d.x)*sin(d.y));
16        ray_direction = normalize(angle.x*normalize(U) + angle.y*normalize(V) +
17        angle.z*normalize(W));
18    }
19
20    optix::Ray ray(ray_origin, ray_direction, 0, scene_epsilon );
21
22    PerRayData_radiance prd; // Rayo luz
23    prd.importance = 1.f;
24    prd.depth = 0;
25
26    rtTrace(object, ray, prd);
27
28    output_buffer[launch_index] = make_color( prd.result );
29 }

```

3.2.3.2. Bounding Box

Tras la ejecución de la sentencia `rtTrace()`, entran en juego las estructuras de aceleración definidas en C++. Como ya anticipamos en el apartado 3.2.2, OptiX provee de algoritmos de aceleración transversales basadas en BVH. Dichas estructuras utilizan los *Bounding Box Programs* para delimitar la extensión espacial de las escenas a ejecutar, acelerando el rendimiento del recorrido de rayos, ignorando las regiones fuera del objetivo de la cámara.

Como podemos observar en la figura 3.23, OptiX establece una jerarquía en cuanto a estructuras geométricas. Esto permite acelerar el total de la escena a distintos niveles y vincular diferentes programas o procesos a distintas partes de la escena. El esquema gráfico de nodos de OptiX parte de los elementos más fundamentales, las *primitivas*, hasta el conjunto más elevado que engloba el total de la escena, el *grupo*. Las *primitivas* pueden estar formadas por una o varias facetas, que serán los niveles más bajos para los que se ejecutarán individualmente los programas. Estas *primitivas* serán contenidas a su vez, por las estructuras *geometría* y *material*, que podrán asociar distintos programas a los diferentes conjuntos de *primitivas*. La *geometría* asocia programas de *Intersection* y *Bounding Box* y el *material* programas de *Closest Hit* y *Any Hit*. Ambas estructuras forman parte de otra a nivel superior, denominada *geometría instanciada*, que permite recoger y reutilizar estructuras geométricas de nivel inferior ya definidas y asociarlas a elementos de la escena. En ocasiones, puede resultar útil definir uno de estos elementos mediante varias *geometrías instanciadas*, en cuyo caso pueden reagruparse en una estructura de nivel superior denominada *grupo de geometrías*. Esta configuración permite ser optimizada mediante estructuras de aceleración definidas por OptiX a nivel de *geometría instanciada* y *grupo de geometrías* y adicionalmente, establecer otra estructura de aceleración independiente a nivel superior, *grupo*.

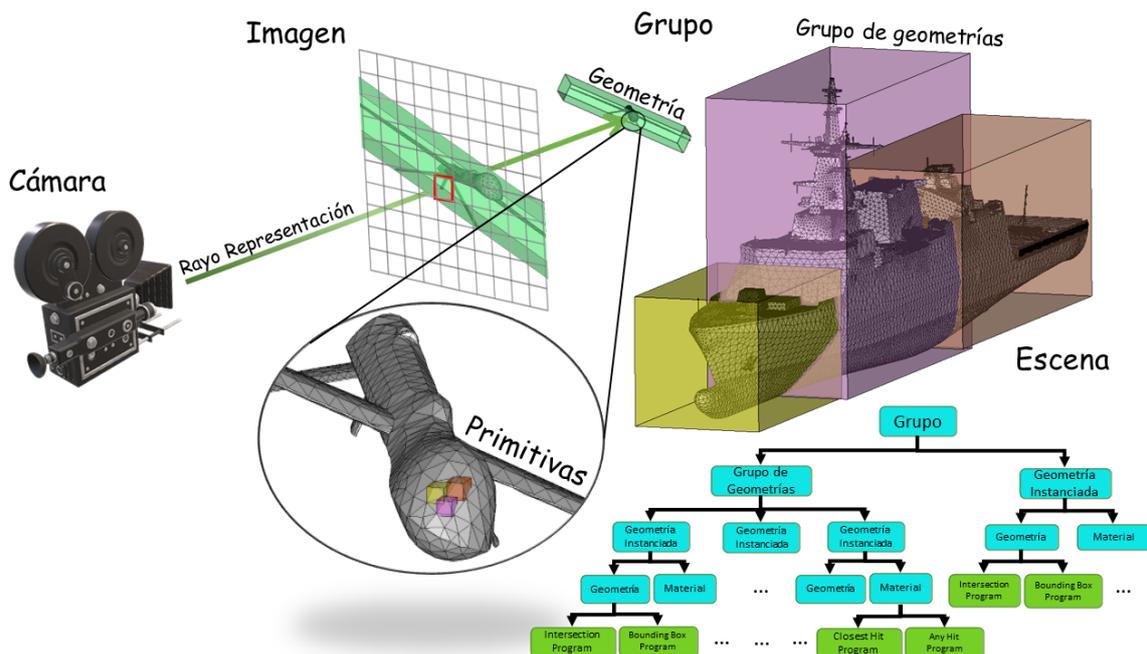


Figura 3.23: Esquema gráfico del trazado de rayos en la elección de los contornos de aceleración de la arquitectura OptiX.

Cada estructura podría ser ampliamente desarrollada y la discusión sobre la estrategia a la hora subdividir la escena extensamente razonada. No obstante, para el desarrollo de esta Tesis partiremos del ejemplo más simple: una única estructura de tipo *geometría* independientemente del tamaño, con una única estructura *material* asociada. En este caso, se tratará de un material opaco, para el que posteriormente en los resultados se ajustarán las características de iluminación y reflexión para obtener diferentes efectos gráficos asociados (en el caso de requerir renderizaciones, para la visualización de sectores ciegos, por ejemplo). De igual modo, existen diferentes estructuras de aceleración ligadas a los respectivos niveles de la jerarquía de nodos, incluso programas como el *Visit Programs* para

aceleraciones de alto nivel. No obstante, para esta Tesis se optó por el constructor TRBVH (*Treelet Restructuring Bounding Volume Hierarchy*) basado en la compilación BVH de la GPU. Este constructor ha resultado ser uno de los constructores más rápidos de entre los cuatro definidos hasta el momento (*Trbvh*, *Sbvh*, *Bvh* y *NoAccel*), caracterizado por disponer de memoria adicional comúnmente necesaria para los procesos finales del BVH.

Con el fin de poder aplicar las estructuras de aceleración sobre una geometría arbitraria es necesario que OptiX pueda consultar los límites del contorno de las primitivas de manera individual. Por tanto, la finalidad del *Bounding Box Program* consiste en definir el mínimo contorno cúbico que contenga por completo la *primitiva* especificada en su primer argumento de entrada, para almacenar el resultado en el segundo, tal y como se implementa en el código 3.2. Estos contornos delimitadores son siempre especificados en el espacio de los objetos, por lo que el usuario no debe aplicarles ninguna transformación. No obstante, el movimiento de la cámara requiere de operaciones de rotación, traslación, ..., que afectan a la determinación de estos volúmenes. Para ello, se hace uso de la librería de funciones *ArcBall* que realiza las transformaciones pertinentes.

Código 3.2: Código del Bounding Box Program

```

1  void Bounding_Box_Program (int primIdx, float result[6]) {
2      const int3 v_idx = index_buffer[primIdx];
3
4      const float3 v0 = vertex_buffer[ v_idx.x ];
5      const float3 v1 = vertex_buffer[ v_idx.y ];
6      const float3 v2 = vertex_buffer[ v_idx.z ];
7      const float  area = length(cross(v1-v0, v2-v0));
8
9      optix::Aabb* aabb = (optix::Aabb*)result;
10
11     if(area > 0.0f && !isinf(area)) {
12         aabb->m_min = fminf( fminf( v0, v1), v2 );
13         aabb->m_max = fmaxf( fmaxf( v0, v1), v2 );
14     } else {
15         aabb->invalidate();
16     }
17 }

```

3.2.3.3. Intersection

Para definir las primitivas, ha de asignarse un *Intersection Program* a la geometría de manera que permita proporcionar los mecanismos necesarios para definir la intersección entre cualquier tipo de primitiva con un rayo. Para ello, los parámetros de entrada al programa han de ser el índice de la *primitiva* y el potencial rayo a ser interferido (siendo éste una variable global concurrente definida por el programa de generación de rayos). Esta *primitiva* contendrá los vértices de un triángulo o faceta necesarios para el cálculo de la intersección, que estarán almacenados en *buffers* de intercambio entre C++ y CUDA.

En el código 3.3 podemos observar un fragmento de las funciones necesarias para generar un programa de intersección asociado a una geometría, donde *primIdx* es la faceta indexada contenida en el *buffer* de índices de facetas *index_buffer*. A partir de las coordenadas de los vértices almacenados en *vertex_buffer* la función *intersect_triangle*, provista por los ejemplos de OptiX SDK, determina la intersección entre rayo y faceta. Esto permite transferir parámetros como la normal, el vector de incidencia, las coordenadas de la textura de la faceta interferida, entre otros, al resto de programas mediante variables globales.

Código 3.3: Código del Intersection Program

```

1 void Intersection_Program( int primIdx ) {
2     const int3 v_idx = index_buffer[primIdx];
3     const float3 p0 = vertex_buffer[ v_idx.x ];
4     const float3 p1 = vertex_buffer[ v_idx.y ];
5     const float3 p2 = vertex_buffer[ v_idx.z ];
6
7     float3 n;
8     float t, beta, gamma; // Deteccion de colisiones con triangulos
9     if( intersect_triangle( ray, p0, p1, p2, n, t, beta, gamma ) ) {
10        if( rtPotentialIntersection( t ) ) {
11            index_triangle = primIdx;
12            geometric_normal = normalize( n );
13            shading_normal = geometric_normal;
14            if( texcoord_buffer.size() == 1 ) {
15                float2 t0 = texcoord_buffer[ v_idx.x ];
16                float2 t1 = texcoord_buffer[ v_idx.y ];
17                float2 t2 = texcoord_buffer[ v_idx.z ];
18                texcoord = make_float3( t1*beta + t2*gamma + t0*(1.0f-beta-gamma) );
19            }
20            rtReportIntersection(material_buffer[primIdx]);
21        }
22    }
23 }

```

3.2.3.4. Closest Hit

En función de los criterios definidos por el usuario para la detección de colisiones en el programa de intersección, el *Closest Hit Program* define las consecuencias de la colisión entre rayo y *primitiva*. El *Closest Hit Program* supone uno de los elementos clave en la obtención de los resultados por parte del usuario. En él se definen los procesos de actuación en caso de colisión como el registro del índice de facetas interferidas, la determinación de las características del rayo reflejado en función del material o la incidencia (el color), generar la casuística que determine el trazado de rayos de sombra, por nombrar alguno de los procesos más relevantes.

El código 3.4 muestra la consecución de los procesos desarrollados en esta Tesis para la obtención de los posteriores resultados. Para ello, partimos de las variables globales de salida del *Intersection Program*, en caso de haberse reportado una colisión, definiendo la normal de la geometría *world_shade_normal*, el vector de incidencia del rayo en la faceta *world_geo_normal* y la normal de la faceta en la dirección de iluminación *ffnormal*. Es necesario realizar esta distinción, dado que el trazado de rayos a priori no distingue entre caras, pudiendo atravesar un objeto cerrado e iluminarlo desde dentro. Esta distinción dependerá de la variable *world_geo_normal* en función de las características del material como en materiales transparentes, donde el trazado de rayos no es truncado sino particularizado en función de la leyes de Snell o cualquier otro comportamiento que determine el usuario.

A continuación, se definen una serie de casuísticas propias de la implementación con el fin de definir la variable *color* que irá acumulando las consecuencias de los procesos que sucedan (de 0 a 255 en la escala RGB de 8 bits). Estos procesos concluirán el color que se deduzca de los factores involucrados en la escena con cada intersección, durante el recorrido del rayo desde la fuente a la cámara. Algunos de estos factores son las propiedades del material (*color_geo*, *Ka*, *Kd*, *Ks*) y la iluminación, ya sea ambiental (*ambient_light_color*) o particular (a través de la estructura *BasicLight*). Puesto que consideraremos un único material y la posibilidad de diversas fuentes, tan solo será necesario especificar los procesos para cada foco particular mediante un bucle (línea 22). En función de la condición del producto escalar entre la normal de la faceta en la dirección de iluminación y el vector de incidencia (línea 30), se sucederán los procesos de sombra y penumbra.

Código 3.4: Código del Closest Hit Program

```

1 void Closest_Hit_Program() {
2     float3 hit_point = ray.origin + t_hit * ray.direction;
3
4     // Normal interpretada por la interseccion de rayo
5     float3 world_geo_normal = normalize( rtTransformNormal( RT_OBJECT_TO_WORLD,
6     geometric_normal ));
7
8     // Normal de la geometria
9     float3 world_shade_normal = normalize( rtTransformNormal( RT_OBJECT_TO_WORLD,
10    shading_normal ) );
11    // Normal en la direccion de iluminacion, siempre frente a la iluminacion
12    float3 ffnormal = faceforward( world_shade_normal, -ray.direction,
13    world_geo_normal );
14
15    float3 color;
16    if( color_flag == 1 || color_flag == 3 ) color = color_geo;
17    else color = Ka * ambient_light_color;
18
19    int num_visib = visib_light.size();
20    if( num_visib == 0 ) num_visib = 1;
21    if( color_flag == 3 ) num_visib = 0;
22
23    for(int i = 0; i < num_visib; ++i) {
24        BasicLight light;
25        if( visib_light.size() == 0 ) light = lights[pos];
26        else light = lights[visib_light[i]];
27
28        float3 L = normalize(light.pos - hit_point);
29        float nDl = dot( ffnormal, L);
30
31        if( nDl > 0.0f ){
32            // Generacion de rayo sombra
33            PerRayData_shadow shadow_prd;
34            shadow_prd.attenuation = make_float3(1.0f);
35            float Ldist = length(light.pos - hit_point);
36            optix::Ray shadow_ray( hit_point, L, 1, scene_epsilon, Ldist );
37            rtTrace(top_shadower, shadow_ray, shadow_prd);
38            float3 light_attenuation = shadow_prd.attenuation;
39
40            if( fmaxf(light_attenuation) > 0.0f ){
41                float3 Lc = light.color * light_attenuation;
42                color += Kd * nDl * Lc;
43
44                // Simula la iluminacion ponderada phong al estar frente al foco
45                float3 H = normalize(L);
46                float nDh = dot( ffnormal, H );
47                if(nDh > 0) color += Ks * Lc * pow(nDh, phong_exp);
48            }
49        }
50
51    // Reflexion de rayos en función del criterio: material, nº reflexiones, ...
52    float importance = prd_radiance.importance * optix::luminance( reflectivity );
53    if( reflex==1 && importance>importance_cutoff && prd_radiance.depth<max_depth ) {
54        PerRayData_radiance refl_prd;
55        refl_prd.importance = importance;
56        refl_prd.depth = prd_radiance.depth+1;
57

```

```

58     float3 R = reflect( ray.direction, fnormal );
59     optix::Ray refl_ray( hit_point, R, 0, scene_epsilon );
60
61     rtTrace(top_object, refl_ray, refl_prd);
62     color += reflectivity * refl_prd.result;
63 }
64 prd_radiance.result = color;
65 }

```

La definición del trazado de rayos de sombra implica un nuevo trazado de rayos y por tanto, de una función recursiva (líneas 31-37). En este caso, la estructura *Ray* es particularizada con el origen en el punto de colisión y el destino en la dirección del vector de incidencia de la luz. La estructura *PerRayData_shadow* es definida por el usuario en base a una escala de 0 a 1, donde 0 implica una atenuación total y 1 ninguna atenuación, ponderando así el grado de atenuación. Una vez definidas las nuevas estructuras, la sentencia *rtTrace()* instancia de nuevo la ejecución del esquema de procesos OptiX. Finalmente, una vez realizado el trazado de sombra, los resultados son almacenados en *light_attenuation*. El primer resultado posible es 1 como consecuencia de no haber modificado la variable *shadow_prd* y haber sido descartado por el programa de intersección, dando paso al *Miss Program*. Este resultado permite proseguir la ejecución caracterizando la penumbra como cualquier modificación sobre la variable *color* debido a los diversos parámetros del material o la iluminación (líneas 40-46). El segundo resultado posible es 0 como consecuencia de haber detectado colisión y ejecutarse el programa *Any Hit Program*, concluyendo el *Closest Hit Program*.

Adicionalmente, a partir de la línea 51 se modela el proceso de reflexión en caso de cumplir las condiciones de iluminación y *flag* requeridos. Este proceso supone una particularización del trazado de sombra a partir de rayos de luz. Este procedimiento podría ser ejecutado recursivamente en caso de colisión hasta un número de veces estipulado en la variable de control *max_depth*. Dichas consecuencias serían recogidas en la variable *color* y finalmente devueltas al *Ray Generation Program* mediante *prd_radiance*, para ser almacenadas en la posición correspondiente del *buffer* de salida *output_buffer*.

3.2.3.5. Any Hit

Al igual que el programa *Closest Hit Program*, el programa *Any Hit Program* corresponde al caso de detección de una colisión entre rayo de sombra y una *primitiva*. En él pueden definirse todos los procesos que el usuario requiera para caracterizar la casuística de los rayos de sombra. En el desarrollo de esta Tesis, este programa ha actuado como una mera máscara con la que poder descartar, gráfica y cuantitativamente, las facetas iluminadas de las no iluminadas.

Como anticipamos en el apartado anterior, el código 3.5 actualiza el argumento de la estructura rayo de sombra *prd_shadow.attenuation* para modificar su valor por defecto 1 a 0. Dicho valor permite ponderar la variable *color* en el *Closest Hit Program*, refiriéndola a 0 (negro).

Código 3.5: Código del Any Hit Program

```

1 void Any_Hit_Program() {
2     prd_shadow.attenuation = make_float3(0);
3     rtTerminateRay();
4 }

```

3.2.3.6. Miss

Al contrario que los programas *Closest Hit Program* y *Any Hit Program*, el programa *Miss Program* es el encargado de recoger los rayos descartados que no interfieren con las *primitivas*. El resultado por tanto, es la obtención de la reflexión del color de fondo definido por la variable *bg_color*. No obstante, es posible realizar procesos para mapear una imagen 2D al igual que los procesos que empleamos para obtener el efecto de ojo de pez. Sin embargo, resultan poco relevantes para nuestro cometido y su implementación se relegó a futuras mejoras. Para el desarrollo de las aplicaciones en esta Tesis, el color de fondo se ha definido como blanco, como recoge el código 3.6.

Código 3.6: Código del Miss Program

```
1 void Miss_Program() {
2     prd_radiance.result = bg_color;
3 }
```

3.2.3.7. Exception

Por último, *Exception Program* es definido como un programa auxiliar. Su finalidad se centra en labores de depuración de código y registro de errores. La ejecución de código concurrente, al margen de las herramientas en C++, supone una complejidad añadida en la detección de errores de los códigos ejecutados en CUDA, para lo que una de las pocas herramientas a nuestra disposición son los programas de excepciones. OptiX declara un conjunto de banderas por defecto, asociadas a unas pocas sentencias con las que identificar posibles fallos asociados al código. De esta manera, el código 3.7 muestra por pantalla el resultado de dichas sentencias y, en el caso de no suponer un error fatal, permite identificar los rayos perturbados mediante la variable *bad_color*.

Se trata de una técnica poco intuitiva y más precaria que las herramientas de depurado disponibles en C++. No obstante, resulta recomendable su uso en el desarrollo de programas en OptiX.

Código 3.7: Código del Exception Program

```
1 void Exception_Program() {
2     const unsigned int code = rtGetExceptionCode();
3     rtPrintExceptionDetails();
4
5     output_buffer[launch_index] = make_color( bad_color );
6 }
```

3.2.4. OpenGL

Hasta el momento nos hemos centrado meramente en las aportaciones tecnológicas (hardware y software) por parte de NVIDIA en materia del trazado de rayos. No obstante, el mérito de la API de NVIDIA OptiX no solo reside en sus aportaciones tecnológicas, sino en su accesibilidad y compatibilidad con las herramientas y estándares de representación gráfica popularmente extendidos como OpenGL, Direct3D (API que forma parte del conjunto DirectX) o Vulkan.

Al igual que OptiX, OpenGL es considerado una API que proporciona un gran conjunto de funciones destinadas al manejo de gráficos e imágenes. En esta herramienta, la sucesión de procesos llevados a cabo por las variables definidas se equiparan a una máquina de estados que define cómo debe operar

OpenGL. Dicha API está fundamentalmente implementada en C++, por lo que permite una inmediata compatibilidad con el desarrollo de nuestra aplicación, aunque contempla adaptaciones a diferentes lenguajes de programación de alto nivel. Por parte de OptiX, este soporte es adicionalmente extendido en OptiX SDK, incorporando OpenGL a sus implementaciones.

OpenGL supone una herramienta igualmente extensa a la anteriormente descrita, cuyo desarrollo de estructuras y funciones supondría un capítulo aparte. El cometido de este apartado se limita a una breve introducción de esta herramienta en el ámbito de esta Tesis. No obstante, existen otras alternativas como ya hemos comentado, para las que el planteamiento de este apartado es equivalente.

La implementación desarrollada en esta Tesis está basada principalmente en *OpenGL Utility Toolkit* (GLUT). Una de las librerías más comunes que permite gestionar fácilmente la representación de escenas generadas por OpenGL, en función del sistema operativo (*Operating System*, OS). Esto es así debido a que la visualización de ventanas requiere de operaciones específicas en base al OS. Por tanto, esta librería permite simplificar y abstraer al usuario a un nivel superior, donde precisa manipular únicamente parámetros de entrada y salida, sin necesidad de conocimientos específicos de OS.

Existen diversas librerías equivalentes a GLUT para la gestión de los aspectos más condicionantes de OpenGL. Entre las más conocidas se encuentran *Simple DirectMedia Layer* (SDL), *Simple and Fast Multimedia Library* (SFML), *OpenGL Extension Wrangler Library* (GLEW) o *Graphics Library Framework* (GLFW). Se trata de librerías muy similares entre sí, cuya diferencia radica básicamente en las funcionalidades implementadas. La elección de librería en esta Tesis se basó exclusivamente en la herencia de los ejemplos de partida de OptiX SDK, que han sido comentados en líneas anteriores. No obstante, posteriormente se extendió el código a implementaciones más interactivas basadas en ImGUI (*Immediate-mode Graphical User Interface*), como se mostrará más adelante.

A continuación se muestra el código simplificado 3.8 con la estructura general de la implementación basada en OpenGL para la representación de los resultados obtenidos mediante OptiX. Recordamos que esta implementación concierne a las estructuras de representación en C++ y, por tanto, ejecutadas secuencialmente en la CPU. El primer bloque de comandos (de la línea 4 a la línea 9) son los encargados de inicializar las estructuras necesarias para la representación mediante la librería GLUT. Con ellos podemos especificar parámetros fundamentales de la ventana a mostrar como el ancho y el largo (*width* y *height*), la disposición en la pantalla representada, el nombre de la cabecera que tendrá la ventana (*PROGRAM_NAME*) o el formato de entrada de los valores que representará (*GLUT_RGB* | *GLUT_ALPHA* | *GLUT_DEPTH* | *GLUT_DOUBLE*). Una vez inicializadas, resulta útil obtener información de la configuración final y mostrarlo por pantalla (línea 12).

La transformación de un elemento 3D a una representación 2D requiere la secuencia de los procesos de la máquina de estados de OpenGL. En nuestro caso, OpenGL no determinará la transformación mediante *shaders*, como realizaban las API de Vulkan y Microsoft, sino que partirá de los valores obtenidos por OptiX almacenados en los *buffers*. No obstante, la presencia de estos *shaders* seguirán siendo necesarios para las operaciones de representación de cuadrículas y postprocesados que mencionaremos en el apartado de resultados numéricos. Por tanto, será necesario definir estos programas ejecutados concurrentemente en los núcleos de la GPU, mediante el *OpenGL Shading Language* (GLSL). Estos estados están destinados a determinar el color del píxel correspondiente a la representación 2D a partir de las coordenadas 3D de la geometría. Para ello, estas coordenadas son almacenadas en el *Vertex Buffer Objects* (VBO), que transfiere las posiciones a la memoria de la tarjeta gráfica para reducir los tiempos de acceso a la memoria. En nuestro caso, reservamos memoria para los datos obtenidos por OptiX y el VBO (líneas 15 a 19). Ya que el uso de OptiX, independiza la resolución de la ventana de representación de OpenGL (*width* y *height*) de la resolución obtenida por los rayos de OptiX (*width_resolut* * *height_resolut*), se requiere de VBO para renderizar mediante el denominado *fragment shader* aquellos casos en que ambas resoluciones difieran, adaptando de esta manera los resultados obtenidos a la ventana de representación.

Código 3.8: Representación gráfica mediante OpenGL, basado en la librería GLUT.

```

1  int main(int argc, char** argv )
2  {
3      // Inicializa estructuras GLUT
4      glutInit( argc, argv );
5      glutInitDisplayMode( GLUT_RGB | GLUT_ALPHA | GLUT_DEPTH | GLUT_DOUBLE );
6      glutInitWindowSize( width, height );
7      glutInitWindowPosition( 0, 0 );
8      glutCreateWindow( PROGRAM_NAME );
9      glutHideWindow();
10
11     // Muestra informacion de OpenGL
12     glInfo glInfo;      glInfo.getInfo();      glInfo.printSelf();
13
14     // Configuracion del buffer de salida
15     GLuint vbo = 0;
16     glGenBuffers( 1, &vbo );
17     glBindBuffer( GL_ARRAY_BUFFER, vbo );
18     glBufferData( GL_ARRAY_BUFFER, 4 * width_resolut * height_resolut, 0, GL_STREAM_DRAW);
19     glBindBuffer( GL_ARRAY_BUFFER, 0 );
20
21     // Inicializa la maquina de estados
22     glMatrixMode(GL_PROJECTION);
23     glLoadIdentity();
24     glOrtho( 0, 1, 0, 1, -1, 1 );
25
26     glMatrixMode(GL_MODELVIEW);
27     glLoadIdentity();
28
29     glViewport(0, 0, width, height);
30
31     glutShowWindow();
32     glutReshapeWindow( width, height);
33
34     // Registro de llamadas
35     glutDisplayFunc( glutDisplay );
36     glutIdleFunc( glutDisplay );
37     glutReshapeFunc( glutResize );
38     glutKeyboardFunc( glutKeyPress );
39     glutMouseFunc( glutMousePress );
40     glutMotionFunc( glutMouseMotion );
41
42     registerExitHandler();
43
44     glutMainLoop();
45 }

```

La implementación de los *shaders* supone uno de los retos más complejos en la configuración de OpenGL, para lo que las versiones más actuales de OpenGL requieren de la implementación propia de al menos el *vertex* y *fragment shader*. Estas dos herramientas posibilitan el tratamiento de la geometría y el acabado de la representación en ausencia de hardware específico. No obstante, resultan difíciles de programar y depurar, requiriendo de conocimientos previos sobre renderizado y GLSL. Esto resulta coherente, puesto que OptiX en parte surgió para dar solución a este complejo sistema de representación, el cual trata de determinar el valor de un píxel sin trazado de rayos. OptiX SDK implementa unos *shaders* bastante genéricos, para aquellas necesidades que requieran de un postprocesado en OpenGL, sin embargo, siguen siendo difíciles de adaptar. En nuestro caso, fue necesario para la representación en coordenadas polares, que se introducirán posteriormente.

A estas alturas del código, la estructura del esquema de procesos de OptiX en C++ se había generado a la par, cargando la geometría, asociando los programas necesarios en CUDA, generando la escena y configurando los puntos de observación (cámara) e iluminación. Como anticipamos en el subapartado 3.2.3.1 en especial en la figura 3.22, existen diversos sistemas de referencia en función del ámbito de actuación requerido. Esto nos permite realizar una implementación ciertamente versátil y flexible, a cambio eso sí de incrementar el nivel de complejidad. En OpenGL existen principalmente cinco sistemas de referencia, estrechamente asociados a los cuatro anteriormente descritos en OptiX:

- El *local space* y el *world space* encargados de describir y posicionar la geometría o geometrías en el espacio. Ambos considerados el mismo en nuestro caso, ya que no contemplamos la posibilidad de escenarios móviles, cuyos objetos describiesen trayectorias o desplazamientos con respecto al origen (0,0,0). No obstante, estos sistemas concuerdan con el sistema de coordenadas de la escena (U,V,W), que ya describimos.
- El *view space* asociado al sistema de coordenadas de la cámara (x,y,z), se encarga de identificar las coordenadas del espacio 3D visibles por la cámara o punto de observación que serán representadas en el espacio 2D.
- El *clip space* o sistema de coordenadas imagen (x,y) proyecta las anteriores coordenadas sobre el espacio 2D, renormalizando el sistema en rangos de (-1,1) que determinan los vértices del sistema de coordenadas de la ventana OpenGL (línea 22-27). Esta transformación conlleva aspectos de proyección y perspectivas que requerirán de la librería matemática de OpenGL (*OpenGL Mathematics*, GML).
- El *screen space* correspondiente al último sistema de coordenadas del píxel (u,v), el cual finalmente transforma las coordenadas del espacio renormalizado, mediante la sentencia *glViewport* (línea 29), en un rasterizado con el que posteriormente pasará por el *fragment shader*.

Todas estas transformaciones están disponibles una vez inicializada la máquina de estados de OpenGL y accesibles convenientemente en función del sistema de coordenadas en el que nos encontremos. Para ello, a partir de la línea 35, la librería GLUT define diversas funciones para actuar sobre el bucle de ejecución (máquina de estado) en tiempo real. Esta se define como un bucle infinito controlado por un registro de llamadas que permiten intervenir la ejecución mediante dichas sentencias. Estas sentencias, a su vez, definen interrupciones en base a actuadores como las teclas del teclado o el ratón, que desencadenan la serie de procesos definida por el usuario. Entre estos actuadores se requiere de la definición del proceso de finalización que intervenga la ejecución y finalice el proceso iterativo, liberando la memoria de las estructuras de almacenamiento y *buffers* y cerrando la ventana y los procesos de OpenGL.

Adicionalmente, en el desarrollo de esta Tesis se extendió de manera equivalente esta implementación a una librería complementaria. La librería ImGUI es una extensión de la librería *Graphical User Interface* (GUI), basada en la representación de un entorno gráfico mediante OpenGL, al igual que GLUT. No obstante, esta librería implementa un conjunto de funcionalidades gráficas mucho más interactivas, permitiendo al usuario configurar parámetros de entrada sin necesidad de precompilar. Se trata de una librería especialmente diseñada para la configuración de parámetros de entrada, mediante una gran selección de actuadores gráficos editables. En la figura 3.24 se muestra un ejemplo de alguna de sus funcionalidades más representativas, como la representación de subventanas, desplegadas, introducción de caracteres y números, potenciómetros, etc., configurables en tamaño, posición, color y transparencia.

Estas funcionalidades hacen de la librería ImGUI un elemento adecuado para el desarrollo de aplicaciones interactivas que requieran de actuadores gráficos para la obtención de resultados de manera intuitiva, mejorando la usabilidad y el manejo del entorno y el posicionamiento de los elementos implicados en la escena. Todo ello, proporciona una experiencia más inmersiva y un análisis más cualitativo de los resultados obtenidos.

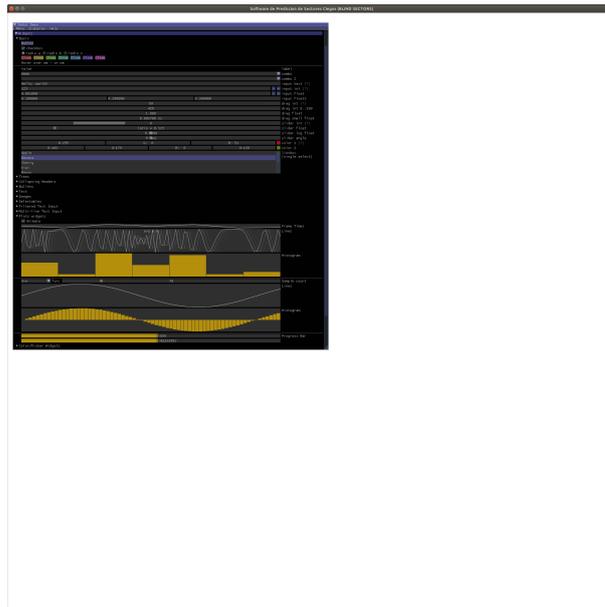


Figura 3.24: Representación del conjunto de funcionalidades gráficas de la librería ImGui de OpenGL.

3.2.5. Módulo de visibilidades de IPO

Una vez introducida y expuesta de manera generalizada la arquitectura OptiX, dedicamos un apartado a concretar el módulo de visibilidades de IPO. Para ello, realizamos dos distinciones a la hora de abordar la obtención de dichas matrices, una para $iLit$ y otra para $iVis$. Ambos desarrollos son contextualizados para la obtención de las matrices de visibilidades necesarias en el cálculo de la RCS monoestática, no obstante, estas consideraciones pueden ser extrapoladas a cualquier otro tipo de aplicaciones o configuraciones.

En primer lugar, partiremos del desarrollo expuesto anteriormente en los diferentes subapartados, considerando la configuración de escena más simple, tal y como representábamos en la figura 3.20. Para ello, recordamos que la matriz $iLit$ correspondía con la necesidad de una estructura capaz de almacenar las relaciones de visibilidades de las facetas iluminadas, para las distintas direcciones de incidencia desde los puntos de observación. Estas direcciones de incidencia son generalmente descritas en rangos de acimut y elevación para determinadas resoluciones angulares. Por tanto, requerimos de un proceso sistemático que permita parametrizar las distintas posiciones de observación e iluminación. Estos serán los únicos elementos susceptibles de cambio en el estudio de las diferentes direcciones de incidencia para una misma escena o geometría, como muestra la figura 3.25.

Para el desarrollo de este apartado, consideraremos indistintamente la posición del observador (cámara) y de la fuente (iluminación) como una misma, ya que resultan completamente independientes y unificarlas facilita el entendimiento a la hora de ser representadas. No obstante, tan solo el punto de observación resulta un factor determinante a la hora de obtener las matrices de visibilidades. Esto se debe a que la representación gráfica es considerada un añadido, por la naturaleza a la que va destinada esta tecnología. El módulo de visibilidades IPO no requiere de una interfaz gráfica para determinar las visibilidades, aunque resulta muy útil como método o herramienta de depuración de errores. Esta consideración trata de ser evidente en la figura 3.26, que muestra la diferencia entre unificar observador y fuente para distintas direcciones de incidencia a partir de una fuente omnidireccional (parte superior) y el caso de fijar el punto de observación y variar el punto de iluminación (parte inferior). En ambos casos se iluminarían las mismas facetas, aunque como veremos a continuación, el segundo caso no permite acceder al estado de dichas facetas, al no ser representadas.

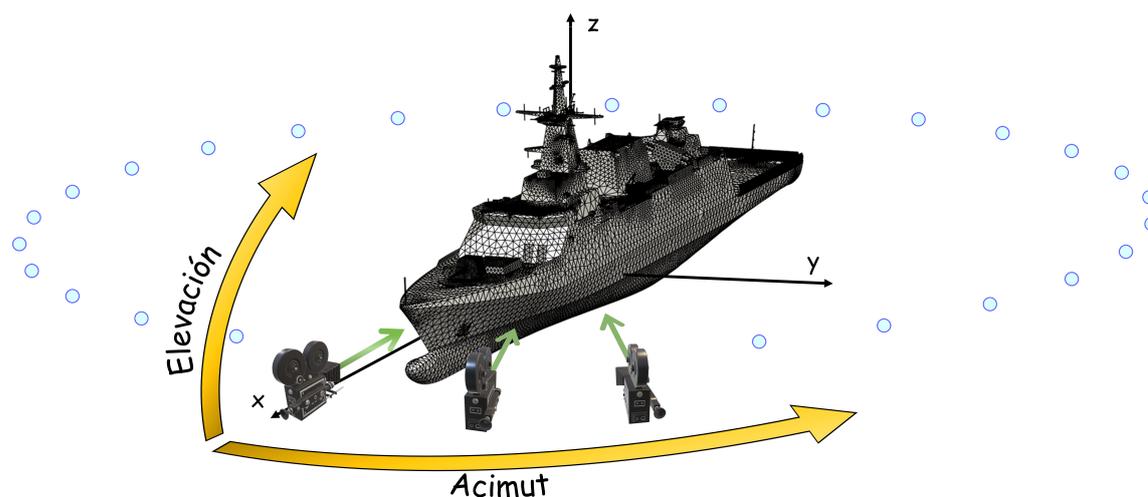


Figura 3.25: Parametrización de las direcciones de incidencia para la determinación de las matrices de visibilidades IPO (*iLit*) en aplicaciones de RCS monoestática.

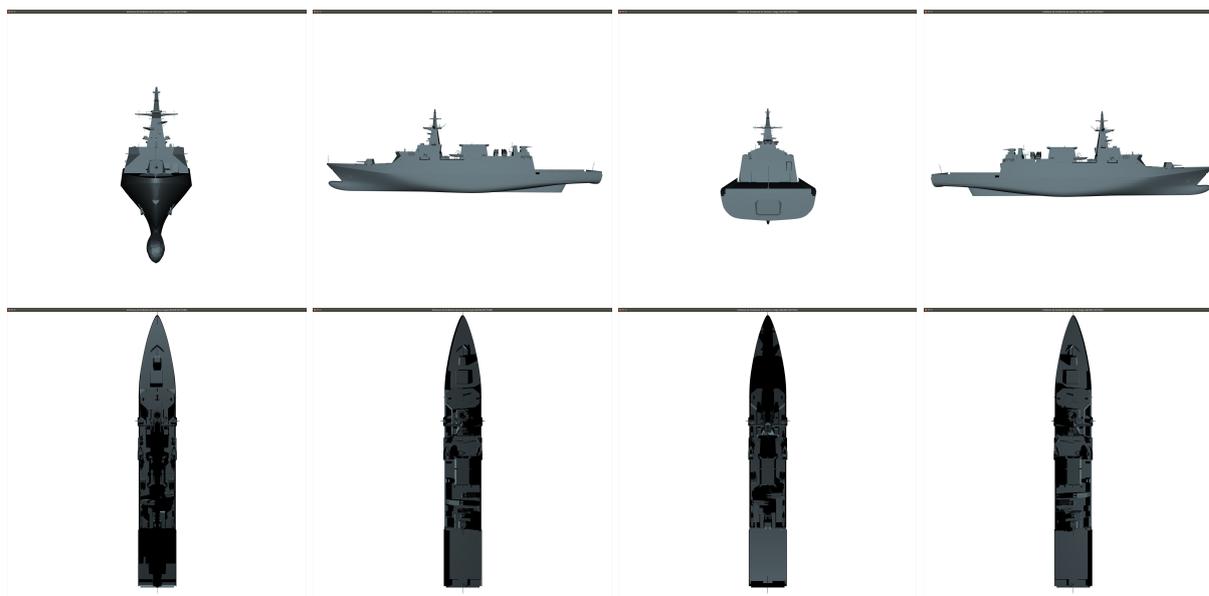


Figura 3.26: Comparativa de la obtención de la matriz *iLit* para 4 direcciones de incidencia, considerando mismo posicionamiento fuente y observador, en la parte superior, y fijando el punto observador y variando el punto fuente, en la parte inferior.

En OptiX el punto de iluminación únicamente determina el color o brillo de las facetas iluminadas que se encuentre representada, pudiendo reportar el identificador de la faceta visible a partir de un cierto umbral. Sin embargo, esto no resulta representativo de las facetas verdaderamente visibles por la fuente, dado que puede darse el caso de facetas iluminadas no representadas por la disposición del punto de observación, como ocurría en el segundo caso de la figura 3.26. Esto se debe a que la determinación del color o brillo de la faceta, en base a la fuente, es realizado en el *Closest Hit Program* y no en el *Intersection Program*. Las facetas visibles al *Closest Hit Program* ya han sido escogidas (para ser representadas) y son visibles a la observación. Por tanto, esto demuestra que el poder de selección viene determinado por el punto de observación, que controla la trayectoria del rayo mediante el *Ray Generation Program* (y determina la colisión o visibilidad de la faceta mediante el *Intersection Program*).

La aclaración ante este paradigma resulta de gran importancia debido a la tendencia natural de querer

asociar facetas visibles a facetas iluminadas. Y que como hemos visto, no es un requisito suficiente para determinar la visibilidad, sino que requiere de ser visible por el observador, es decir, ser representada. Este problema resultaría inexistente de considerar siempre posiciones de fuente y observador idénticas, no obstante, llevaría a errores en configuraciones independientes entre observador y fuente. Por tanto, la implementación del módulo de visibilidades de IPO requiere de la disposición superior de la figura 3.26 y no la inferior, que únicamente resulta útil como método de comprobación.

La elección del punto de observación como herramienta de selección o detección de colisiones implica una serie de consideraciones a tener en cuenta a la hora de posicionarse como el ángulo de visión, la perspectiva, el apuntamiento, etc. Estos parámetros resultan posibles de ajustar mediante el control de la trayectoria del rayo en el *Ray Generation Program*. En el caso de la matriz *iLit*, la parametrización de las direcciones de incidencia conlleva una corrección en el ángulo de observación a la vez que se posiciona el punto de observación, considerando la geometría centrada en el origen como describía la figura 3.25. Esta solución resulta efectiva, pero sensiblemente inestable ya que el número de rayos determina la precisión con que detectar colisiones. Inicialmente, podemos asumir una distribución uniforme del número de rayos asociados a cada píxel de una imagen, identificando cada rayo con la porción de superficie contenida en el interior de la cuadrícula definida por un píxel. No obstante, resulta extremadamente ineficiente en configuraciones con mallados gruesos y alta densidad de rayos, donde varios rayos colisionen una misma facetas o no lleguen a colisionar ninguna, o por el contrario, en mallados muy finos donde la resolución sea escasa e insuficiente para determinar cada faceta visible de manera individual.

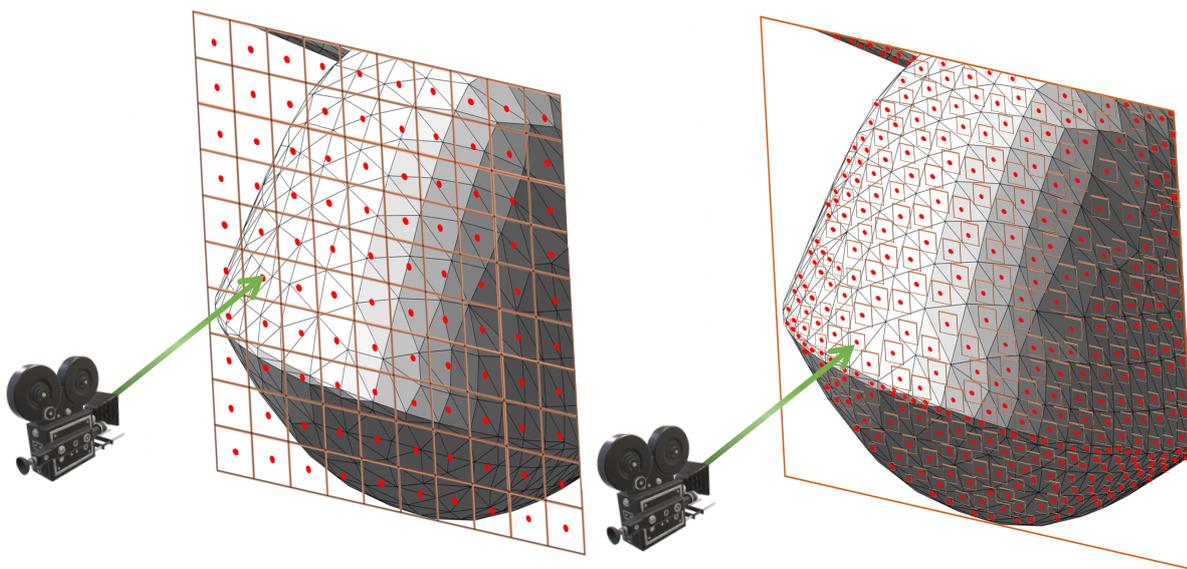


Figura 3.27: Comparativa del trazado de rayo basado en la representación coherente de la escena y la solución eficiente para únicamente determinar la visibilidad de las facetas.

Para ello, el módulo de visibilidades de IPO implementa una solución muy eficiente, requiriendo de un número de rayos igual al número de facetas de la geometría. Mediante el control de la trayectoria del rayo, es posible únicamente trazar rayos dirigidos a los centroides de cada faceta, estableciendo el centro como criterio suficiente para determinar la visibilidad de la faceta. Esta solución ahorra un sobrecoste en el tiempo de cómputo de trazados de rayos y una precisión excelente, independientemente del tamaño de las facetas, tal y como trata de representar la figura 3.27. En consecuencia, la representación gráfica de comprobación es sacrificada, ya que el color de cada píxel adquiriría el valor de visibilidad del centroide de cada faceta de manera secuenciada, sin una interpretación visual coherente a priori.

Una vez comprendido las aportaciones incorporadas al módulo de visibilidades *iLit*, la implementación del módulo *iVis* resulta inmediata. La matriz de visibilidades *iVis* contenía las relaciones de visibilidades

de cada faceta con el resto de facetas, convirtiéndola en una matriz pesada de $O(N^2)$, con N el número de facetas. A diferencia de la matriz *iLit* de dimensiones $N \times M$, que requería M direcciones de incidencias (M simulaciones de N rayos), la matriz *iVis* tan solo requiere de ser obtenida una sola vez para cada geometría (N simulaciones de N rayos).

Para ello secuenciamos N proceso de N rayos, posicionando el punto de observación sobre el centroide de cada faceta. Esta disposición simula un nuevo foco de dispersión o reflexión, dirigido a todas las direcciones susceptibles de interés (el resto de centroides). Con este proceso determinamos convenientemente la visibilidad de una faceta con respecto a las demás, contemplando las ocultaciones al igual que en el caso *iLit*. Este procedimiento describe la implementación eficiente mencionada en el apartado anterior, la cual no permite la visualización coherente de los resultados. No obstante, es posible corroborar los resultados obtenidos mediante la implementación ineficiente por defecto o el postprocesado geométrico en matlab del código M^3 . De esta forma, la figura 3.28 muestra las visibilidades (representadas en rojo) correspondientes a una faceta (representada en amarillo) obtenidas para la matriz *iVis* y las facetas iluminadas para una dirección de incidencia correspondiente a $\theta = 90^\circ$ y $\phi = -5^\circ$ para la matriz *iLit*.

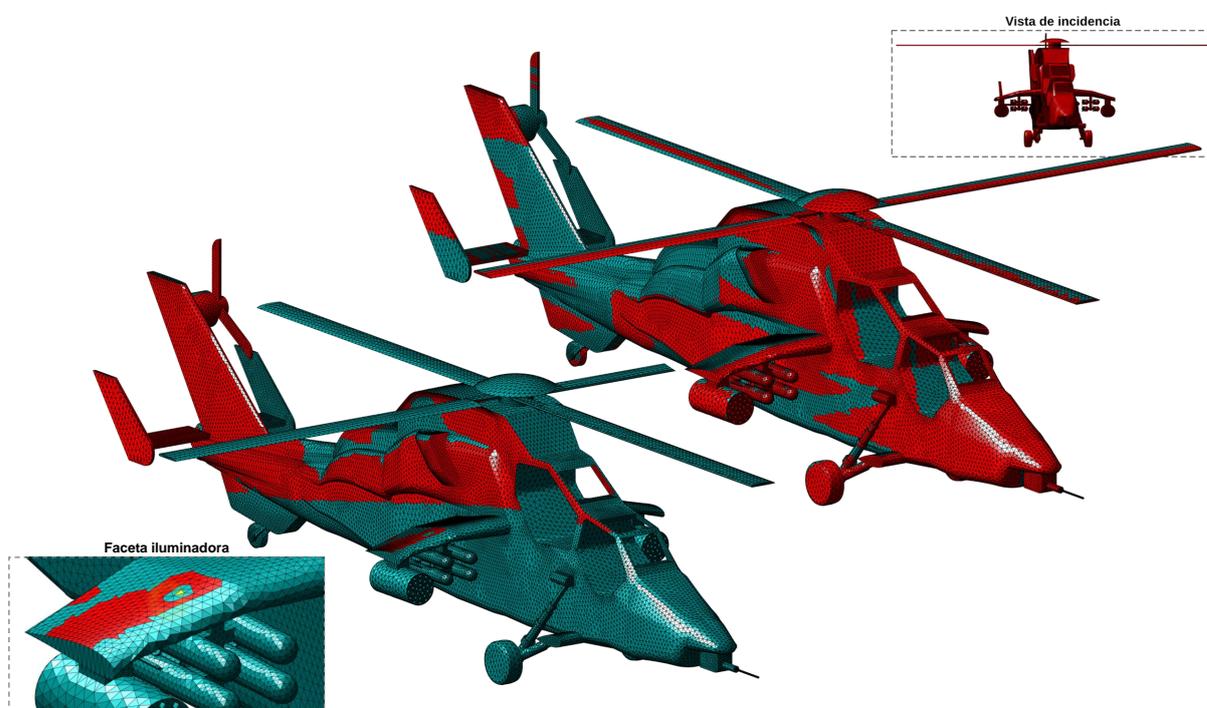


Figura 3.28: Representación de los resultados obtenidos para la implementación eficiente del módulo de visibilidades IPO de la matriz *iVis* a la izquierda e *iLit* a la derecha, para el caso de un Eurocopter EC665 Tigre.

Adicionalmente, el proceso de la obtención de la matriz *iVis* se puede acelerar aprovechando la simetría del problema, y por tanto, de la matriz. Como ya mencionamos en otra ocasión, la visibilidad entre facetas resulta recíproca, es decir, la faceta que determina visible a otra faceta es a su vez contenida en el conjunto de facetas visibles de esta. Este hecho define una matriz simétrica con respecto a la diagonal, cuyos elementos representan la autodeterminación como facetas visibles y que puede ser obviada. Por tanto, tan solo es necesario realizar la mitad del proceso, reduciendo el número de simulaciones, el tiempo de cómputo y los recursos en memoria a la mitad. En consecuencia se incrementará levemente la complejidad a la hora de acceder a los datos, en función del almacenamiento de la información contenida en la matriz triangular superior o inferior.

4 Resultados numéricos

Contenido

4.1. Sección equivalente radar (RCS)	66
4.2. Detección de sectores ciegos (Blind Sectors)	72
4.2.1. Estudio de diagramas de cobertura óptica	73
4.2.1.1. Configuración y posicionamiento	74
4.2.1.2. Representación de los diagramas de cobertura óptica y sectores ciegos	76
4.2.2. Versiones beta	81
4.3. Método híbrido DDM-IPO	83

En este capítulo, analizaremos tres ejemplos de aplicación para poner de manifiesto la utilidad de las principales aportaciones de esta Tesis. En primer lugar, comenzaremos calculando la RCS monoestática de un buque real mediante el método IPO. Para ello, aplicaremos los desarrollos realizados en materia de preprocesado geométrico y cálculo de las matrices de visibilidades mediante OptiX. A continuación, modelaremos la presencia del plano de mar y su influencia en la estimación y contribución a la RCS total. Se comparará la precisión del método IPO desarrollado con respecto a medidas reales disponibles para este buque.

En el segundo ejemplo se ilustran los procedimientos necesarios para la obtención de los diagramas de cobertura óptica y matrices de visibilidades mediante el software de detección de sectores ciegos. Para ello, comenzaremos especificando los requerimientos hardware y las características técnicas de los casos testados. Proseguiremos describiendo los procedimientos de configuración y posicionamiento de la escena, hasta obtener la representación de los diagramas de cobertura óptica y sectores ciegos buscados. Concluiremos con una breve descripción del estado de versiones en fase de desarrollo, cuyos resultados no pudieron ser recogidos en esta Tesis.

Por último, se muestra un ejemplo de aplicación de la metodología IPO como un solver rápido en el contexto del método DDM. Partiendo del problema inicial de cálculo de la RCS monoestática a frecuencia radar mediante IPO, se incorporará la resolución de dominios específicos con gran cantidad de detalle geométrico mediante metodologías *full-wave*, manteniendo el IPO para la caracterización del resto de la superestructura.

4.1. Sección equivalente radar (RCS)

El blanco realista propuesto en la figura 4.1, corresponde al patrullero Tabarca (P-28) de la Armada Española. Se trata de un buque militar utilizado para el adiestramiento y el desempeño de operaciones reales, por lo que se encuentra dotado de muchos elementos presentes en plataformas portante de

mayor dimensión. Este blanco posee unas dimensiones totales de 40 m de eslora, 6 m de manga, 3 m desde la línea de flotación hasta la cubierta principal, aproximadamente, y 13 m desde la línea de flotación hasta la parte superior del mástil. El modelo es considerado completamente PEC, por lo que tan solo son contemplados dispersores y cavidades metálicas en ausencia de materiales dieléctricos. El cálculo de la RCS monoestática es realizado a 8 GHz, mediante un barrido comprendido entre $0^\circ \leq \phi \leq 360^\circ$, con una resolución angular de $\Delta\phi = 0,5^\circ$ (721 direcciones), manteniendo una elevación de $0,05^\circ$ ($\theta = 89,95^\circ$). La polarización considerada es la HH.

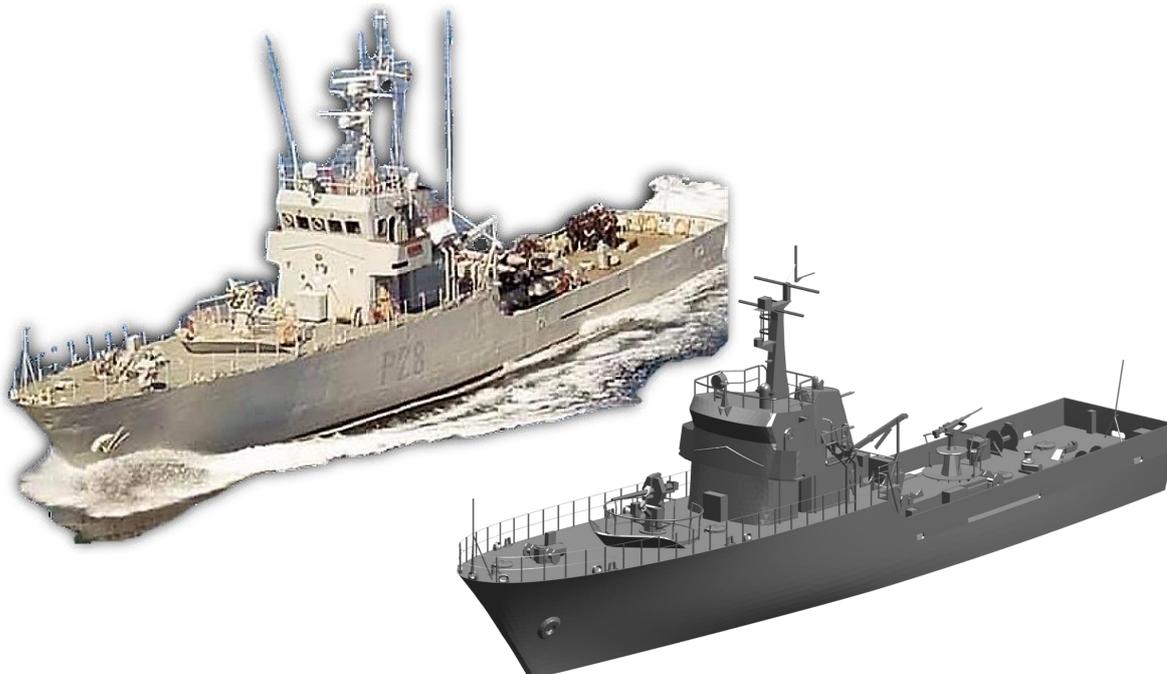


Figura 4.1: Ejemplo de objetivo de ámbito militar real, correspondiente al patrullero Tabarca (P-28) de la Armada Española.

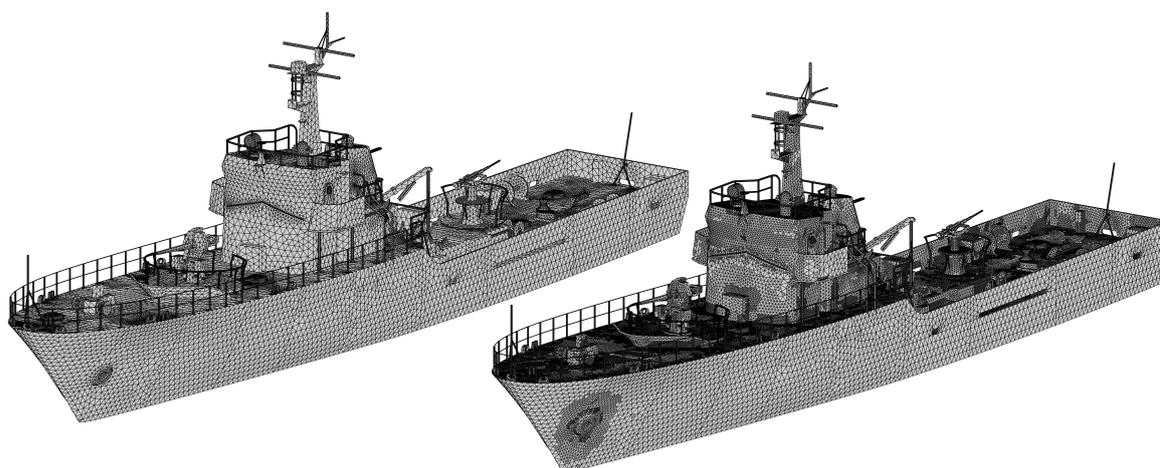


Figura 4.2: Comparativa entre el mallado uniforme y el mallado adaptativo del buque Tabarca para $h = 0,6$ y $\zeta = 2$.

Para esta geometría se ha alcanzado un equilibrio entre el número de facetas y la estimación de una precisión elevada, en base al estudio de los dos parámetros fundamentales que definen el tamaño de la malla (h y ζ), realizado en el apartado 3.1.2.4. Se ha optado en este caso por un tamaño de mallado con $h = 0,6$ en las zonas de mayor resolución, y un parámetro de campo lejano $\zeta = 2$, generando una malla de

213 400 facetas. El mallado original partió de una discretización más o menos uniforme de $h = 8$ que se adecuase correctamente a los contornos de las superficies, de un total de 118 414 facetas. Ambos mallados son representados en la figura 4.2 y mostrados en detalle en la figura 4.3.

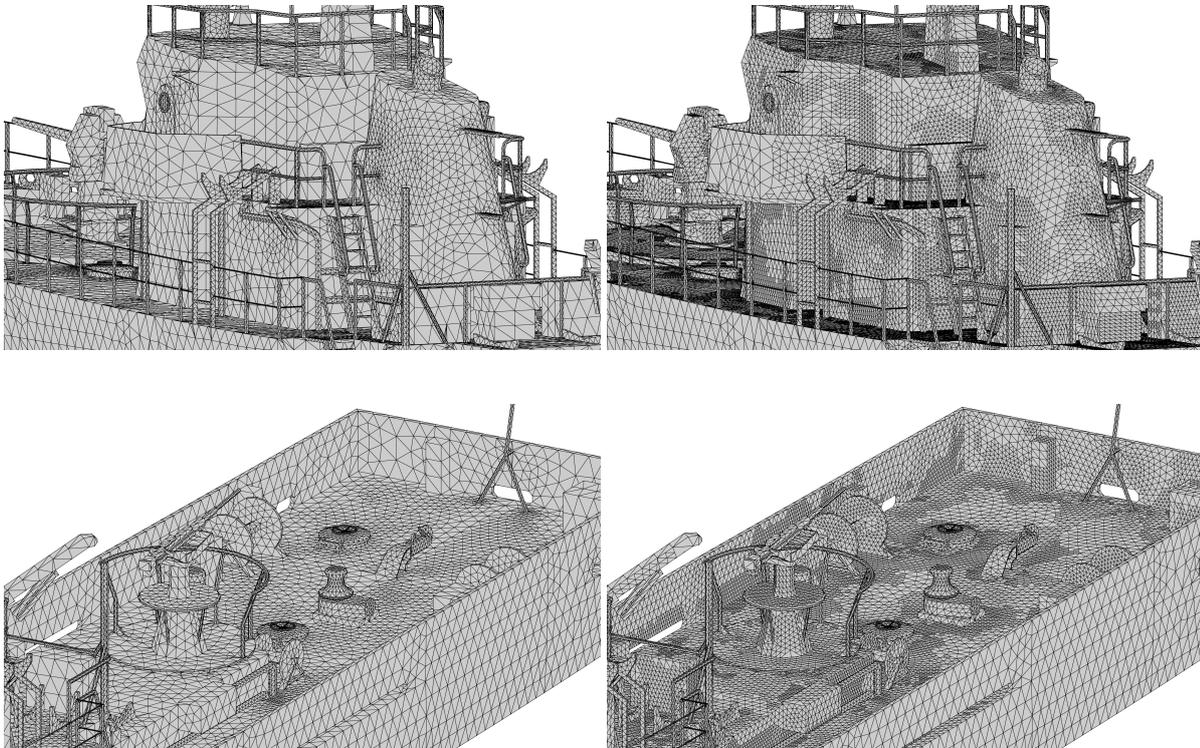


Figura 4.3: Representación en detalle de las diferencias del mallado uniforme con respecto al adaptativo para el buque Tabarca.

La definición de los márgenes angulares de la RCS monoestática del problema (θ y ϕ) determinan las dimensiones de las matrices de visibilidades. En el caso de la matriz *iLit* de 213400 x 721 elementos y en el caso de la matriz *iVis* de 213400 x 213400 elementos. Estas dimensiones suponen un coste de almacenamiento de 150 MB y 22.2 GB respectivamente, mientras que la simulación sistematizada mediante OptiX consta de 721 y 213400 configuraciones respectivamente. Para el cálculo de estos elementos se ha usado la distribución Ubuntu 18.04.2 LTS (Bionic Beaver) y la tarjeta gráfica Quadro P2000. El desarrollo de los procedimientos necesarios para la obtención de las matrices será abordado con mayor detalle en el ejemplo siguiente.

Como muestra de los resultados obtenidos mediante los desarrollos llevados a cabo en OptiX, la figura 4.4 representa las visibilidades/ocultaciones correspondientes a la primera dirección de incidencia almacenada en la matriz *iLit* (en particular, con la dirección $\theta = 89,95^\circ$ y $\phi = 0^\circ$).

La predicción realista de RCS monoestática llevada a cabo en esta Tesis contempla el cálculo de las contribuciones debidas a la presencia del plano de mar. Se aplica para ello un esquema basado en cuatro contribuciones, tal como se representa en la figura 4.5. Este esquema modela las trayectorias directa-directa, directa-reflejada, reflejada-directa y reflejada-reflejada, de la radiación en base a la teoría de imágenes y las leyes de Snell. Por tanto, este esquema asume el comportamiento del mar a frecuencias de microondas como un plano infinito cuya contribución total se compone de la radiación en los puntos de observación y observación imagen por parte de geometría.

Para llevar a cabo la simulación mediante la metodología anterior, no se requiere de ninguna adaptación geométrica de la estructura; no obstante, se incrementa a algo más del doble el coste computacional asociado, surgiendo además la necesidad de calcular una segunda matriz de visibilidades/ocultaciones

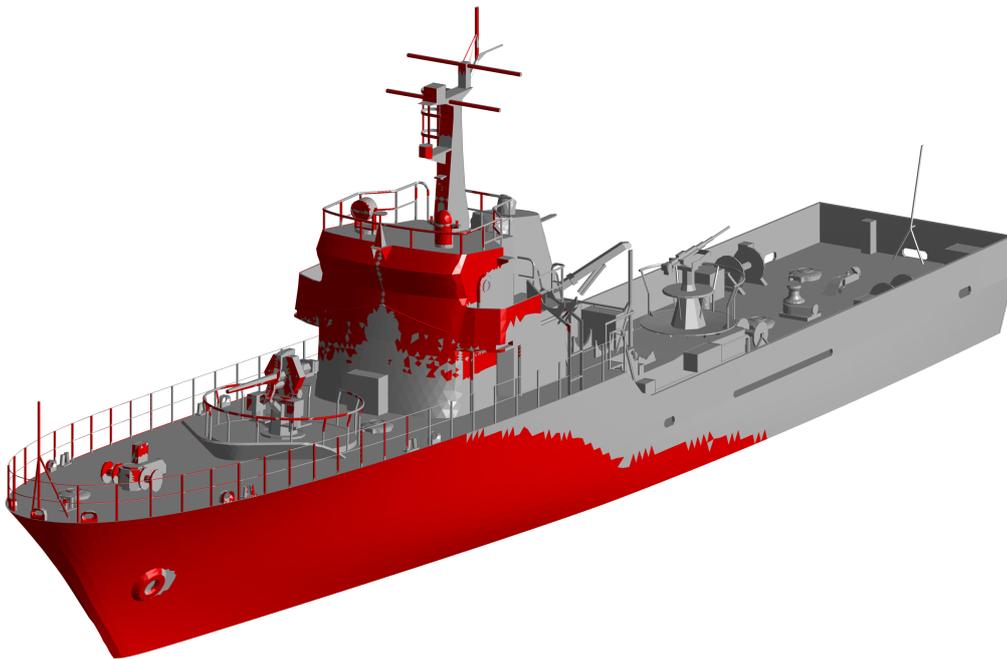


Figura 4.4: Iluminación para una dirección de incidencia sobre el buque Tabarca.

(*iLit*). Esta nueva matriz surge como respuesta a la necesidad de conocer las facetas visibles a la dirección de incidencia reflejada ($-\psi$), requiriendo por tanto un nuevo proceso sistematizado en OptiX (equiparable al procedimiento para *iLit*), particularizado para la nueva dirección de incidencia. Por el contrario, no es preciso calcular una nueva matriz de visibilidades entre facetas (*iVis*), puesto que dicha matriz es independiente de la dirección de incidencia.

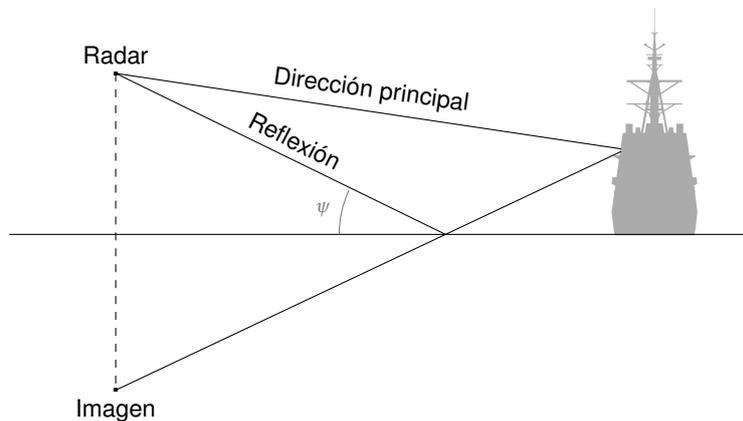


Figura 4.5: Esquema del modelo de 4 rayos para contemplar la presencia del plano de mar en la predicción del cálculo de RCS.

El método IPO tal como se ha desarrollado en la sección 3.1.2 contempla solamente la contribución directa-directa. No obstante, es posible obtener las contribuciones directa-directa y directa-reflejada en un solo proceso iterativo, tratando el vector de incidencia en cada momento a partir de los parámetros originales del estudio. Esta implementación aprovecha el esquema original para desdoblar estas dos contribuciones implicando un leve sobrecoste computacional. Sin embargo, las contribuciones reflejada-directa y reflejada-reflejada requieren de un segundo proceso iterativo que particularice el problema análogo a partir de los puntos de observación y fuente imágenes, duplicando el coste computacional.

Se muestran a continuación los resultados obtenidos por el método IPO para el Tabarca. En la figura 4.6 se observa la RCS monoestática han obtenido un nivel de precisión elevado con respecto a las medidas realizadas. No obstante, estos resultados requieren de ciertas consideraciones a la hora de ser analizados y conviene desarrollarlos.

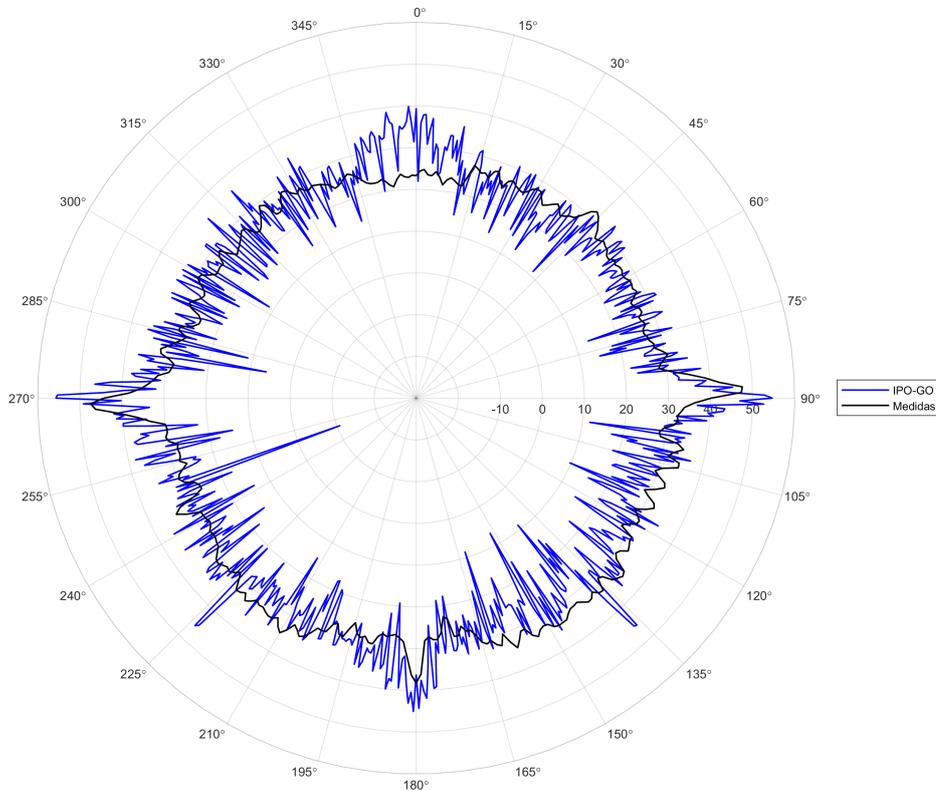


Figura 4.6: Resultados numéricos de la RCS monoestática del buque Tabarca.

Comenzaremos analizando los factores involucrados en la realización de las medidas cuya influencia ha de tenerse en cuenta a la hora de comparar estos valores con la RCS calculada. Las estructuras en presencia de plano de mar resultan significativamente dependientes del estado de ésta. El movimiento constante de la estructura y las leves deformaciones, torsiones y vibraciones que soporta pueden significar desplazamientos relativos de varias longitudes de onda a frecuencias radar, dando lugar a fluctuaciones importantes en la medida debido a las múltiples y cambiantes sumas en fase o contrafase entre diversos dispersores presentes en la plataforma. La RCS resulta por tanto un proceso centelleante, por lo que debe considerarse como un proceso estocástico.

Con respecto a los resultados IPO de este ejemplo, han sido obtenidos mediante el algoritmo IPO-GO, que tal como se ha mostrado en apartados anteriores, ofrece unos tiempos de cómputo sensiblemente reducidos para un nivel de precisión requerido. El tiempo de cómputo total necesario para la estimación de la RCS monoestática de este blanco ha sido de tan solo 2.16 horas en una única máquina AMD EPYC 7763 64-Core Processor. Sirva como contraste que el cálculo necesario para resolver este problema mediante MLFMA o DDM habría requerido alrededor de 4 horas por dirección, aproximadamente 120 días de simulación intensiva para resolver la RCS monoestática en todo el margen angular acimutal.

Teniendo en cuenta la naturaleza estocástica previamente mencionada de la RCS, la presentación de resultados medidos o simulados en presencia de mar típicamente va acompañada de un postprocesado estadístico. Han de obtenerse indicadores que permitan discriminar leves perturbaciones sobre los resultados y faciliten la interpretación de los mismos. Por otra parte, estos indicadores estadísticos

permiten extraer conclusiones sobre la firma radar de un blanco, así como contrastarla con la obtenida para diferentes plataformas.

A continuación se introducen tres estadísticos comúnmente aplicados para evaluar la firma radar de blancos reales sobre el plano de mar. Se trata de la media, la mediana, y el percentil 90. En el caso de la RCS de buques, se suele excluir del cálculo de la media y la mediana unos márgenes de guarda en torno a las direcciones principales (o direcciones de sacrificio, en las que resulta muy difícil contener el valor de la RCS). Estas direcciones son las de proa, popa, babor y estribor, tal como se muestra en la figura 4.7. En esta Tesis hemos considerado un margen de ± 5 grados en torno a dichas direcciones de sacrificio. En el cálculo del percentil 90, sin embargo, se ha considerado el margen angular completo.

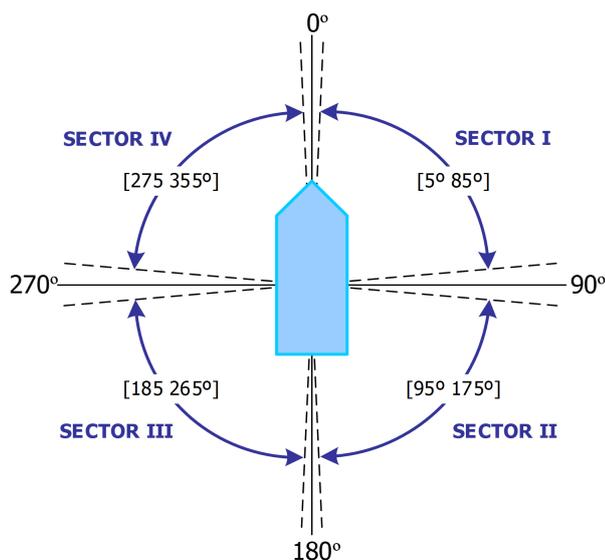


Figura 4.7: Definición de sectores angulares.

Observando los valores estadísticos recogidos en la tabla 4.1, podemos concluir que la solución IPO-GO contrasta muy bien con las medidas. En concreto, se observan diferencias de 1.27 dB para el caso de la media, 0.61 dB para la mediana, y de 2.45 dB para el percentil 90. Este último contemplando las direcciones de sacrificio, caracterizadas por su carácter desfavorable en el valor de RCS.

En ocasiones se realiza también una comparativa por sectores principales (correspondientes a la amura y la aleta de estribor y a la aleta y la amura de babor, respectivamente tal y como se observa en la figura 4.7). En este caso, podemos concluir que el sector II muestra una mejor correlación con las medias, mientras que el sector IV presenta una correlación algo menor.

Tabla 4.1: Estadísticos para los resultados de RCS monoestática obtenidos por IPO-GO y las medidas.

Estadísticos		Media (dBsm)	Mediana (dBsm)	Percentil 90 (dBsm)
IPO-GO	Total	30.80	27.38	36.18
	Sector I	28.22	26.10	31.40
	Sector II	31.29	27.36	33.66
	Sector III	32.61	28.78	35.87
	Sector IV	29.91	27.19	33.81
Medidas	Total	29.53	27.99	33.73
	Sector I	27.28	27.06	29.09
	Sector II	31.80	31.51	34.09
	Sector III	30.26	29.66	32.83
	Sector IV	26.80	26.19	28.87

4.2. Detección de sectores ciegos (Blind Sectors)

El software de detección de *blind sectors* (véase figura 4.8), permite calcular de manera muy precisa las direcciones de visión o campo de actuación de los sensores y elementos radiantes a bordo de estructuras complejas. Partiendo del software basado en OptiX para la detección de visibilidades/ocultaciones presentado en secciones anteriores, se ha implementado una interfaz gráfica interactiva que permite extraer dichos diagramas de manera sencilla e interactiva. Se trata de un software que aporta valor añadido a los estudios de compatibilidad electromagnética de sensores, transceptores y otros elementos sobre estructuras portantes, proporcionando la capacidad de detectar bloqueos o interferencias electromagnéticas entre diferentes elementos dentro de entornos *cosite*.

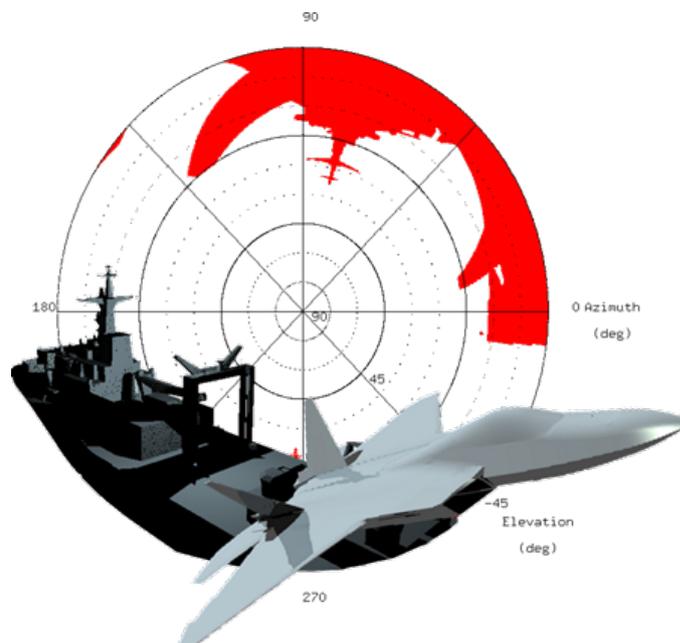


Figura 4.8: Portada del software de predicción de sectores ciegos.

El software provee de un conjunto de funciones capaces de representar en tiempo real diagramas de cobertura óptica en cualquier punto del espacio mediante modelado 3D. Si bien el diagrama de cobertura óptica y la detección de sectores ciegos no determinan por completo el comportamiento de las antenas, a menudo resultan de gran utilidad para inferir sus prestaciones, especialmente en el caso de antenas altamente directivas como la de los radares o las antenas de comunicación satelital.

El software desarrollado ha requerido de las herramientas que proporciona la arquitectura CUDA, mencionadas en la sección 3.2. Se hace uso de la NVIDIA OptiX API, que permite la correcta comunicación entre lenguajes de alto nivel y las funciones implementadas en esta arquitectura.

Para el desarrollo de las aplicaciones llevadas a cabo en esta Tesis, se escogió la instalación por defecto del compilador propio de NVIDIA (*nvcc*) y el propio de Linux (*gcc* versión 4.8.5 para openSUSE y 7.5.0 para Ubuntu). La instalación de ambos compiladores es precisa, puesto que el proceso de compilación requiere de *nvcc* para generar los PTX y *gcc* para la composición de ejecutables de alto nivel sobre los que se implementa la aplicación (C++). En el caso del kernel de Linux se dispuso de la versión 4.4.76-1-default para la distribución openSUSE y 4.15.0-143-generic para Ubuntu, ambas compatibles con OptiX.

A efectos de desarrollo e implementación, se dispuso además de la herramienta CMake 3.10.2 como plataforma de empaquetado, lo que ha permitido unificar los procesos de compilación entre diversos OS.

Tabla 4.2: Salida de terminal del software de sectores ciegos para dos casos de estructuras complejas.

Buque	Avión
<pre> OptiX 5.1.1 Number of Devices = 1 Device 0: Quadro P2000 Compute Support: 6.1 Total Memory: 4236312576 Clock Rate: 1607000 kHz Max. Threads per Block: 1024 Streaming Multiprocessor Count: 6 Execution Timeout Enabled: 1 Max. Hardware Texture Count: 1048576 TCC Driver enabled: 0 CUDA Device Ordinal: 0 Mesh: /home/OptiX/data/Ship.stl num vertices : 3800559 num triangles: 1266853 num materials: 1 has normals : 1 has texcoords: 0 bbox min : (-1024.1,-126.97,-390.053) bbox max : (1198.9,127.007,356.261) Tiempo en inicializar: 16.3237 seconds createContext(): 15.7796 seconds createGeometry(): 8.1e-05 seconds Setup(): 8.1e-05 seconds validate(): 1.5e-05 seconds </pre> 	<pre> OptiX 5.1.1 Number of Devices = 1 Device 0: Quadro P2000 Compute Support: 6.1 Total Memory: 4236312576 Clock Rate: 1607000 kHz Max. Threads per Block: 1024 Streaming Multiprocessor Count: 6 Execution Timeout Enabled: 1 Max. Hardware Texture Count: 1048576 TCC Driver enabled: 0 CUDA Device Ordinal: 0 Mesh: /home/OptiX/data/Avion.msh num vertices : 427800 num triangles: 1711192 num materials: 1 has normals : 0 has texcoords: 0 bbox min : (0,0,0) bbox max : (15.1413,19.9074,4.20194) Tiempo en inicializar: 0.842256 seconds createContext(): 0.596442 seconds createGeometry(): 2.1e-05 seconds Setup(): 2.1e-05 seconds validate(): 8e-06 seconds </pre> 

Esto ha permitido utilizar un único fichero de compilación (.cmake) para generar ejecutables (o proyectos, reconocibles por los sistemas más comunes de compilación, como *gcc* o Microsoft Visual Studio) en los que se especifican todos los parámetros y ficheros pertinentes para la correcta compilación, teniendo en cuenta versiones de compilador, ejecutables, sistemas operativos y ficheros y directorios generados tras su ejecución. Con respecto a los recursos hardware, OptiX requiere de núcleos RT especialmente diseñados para las operaciones del trazado de rayo como se ha mencionado en capítulos anteriores, por tanto, requiere de hardware específico. Para el desarrollo de estos ejemplos dispusimos de una tarjeta gráfica Quadro P2000.

En la tabla 4.2 se muestra la salida de terminal proporcionada por el software implementado, incluyendo parámetros de descripción del hardware y tiempos requeridos para la inicialización de la aplicación. Se muestran estos valores para dos casos de estructuras complejas, consistentes en una fragata y una aeronave respectivamente. Una vez realizada la inicialización, la ejecución posterior del software es completamente interactiva y se realiza en tiempo real.

4.2.1. Estudio de diagramas de cobertura óptica

El software de predicción de sectores ciegos cuenta con una interfaz gráfica controlada mediante el teclado y el ratón del ordenador, capaz de interactuar y obtener los resultados deseados en tiempo real. En este apartado nos centraremos en el estudio de los diagramas de cobertura de un sistema radar, representado en la figura 4.9, del buque mostrado en la tabla 4.2.

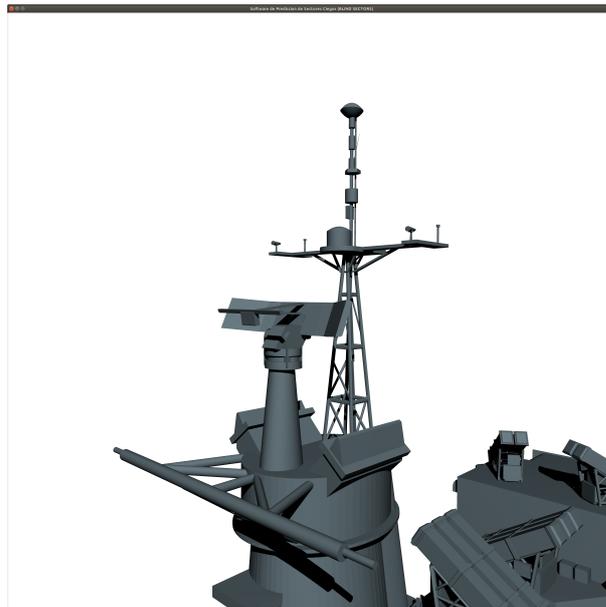


Figura 4.9: Sistema radar móvil escogido para el estudio de los diagramas de cobertura óptica y sectores ciegos.

4.2.1.1. Configuración y posicionamiento

Partiremos de una configuración por defecto en la que la cámara estará localizada sobre una posición arbitraria en el espacio. El programa desarrollado proporciona entonces al usuario los grados de libertad necesarios para permitir posicionar la cámara en la ubicación del elemento bajo estudio y obtener los diagramas de cobertura óptica deseados. Este posicionamiento requiere de una serie de funcionalidades que han sido implementadas para la cámara y que resultan accesibles a través de los actuadores (ratón y teclado), configurando de esta manera un entorno virtual sobre el que desplazarse a discreción y apuntar en las direcciones deseadas. Así, se cuenta con procesos para variar el ángulo de elevación y acimut a lo largo de arcos de 360° en direcciones ortogonales de la imagen, pudiendo dirigir el apuntamiento de la cámara hacia cualquier dirección del espacio a la vez que se mantiene anclado su posicionamiento (que coincide con el punto de observación). Es posible también rotar el posicionamiento de la cámara a lo largo del eje perimetral de una hipotética circunferencia centrada en el centro de la imagen, tal y como muestra la figura 4.10. Se implementan asimismo funcionalidades que permiten el desplazamiento a lo largo, ancho y alto de la escena con respecto a la dirección de observación, de modo que es posible alcanzar cualquier posición del espacio 3D. Todas estas funcionalidades permiten ubicar el posicionamiento de la cámara sobre el elemento bajo estudio y los apuntamientos y ángulos de visión en las direcciones deseadas. Todas estas operaciones son ejecutadas en tiempo real y representadas con tasas de refresco suficientemente elevadas para proporcionar una ejecución fluida.

Eventualmente, puede resultar conveniente ajustar las dimensiones de la imagen, a fin de modificar la relación de aspecto de la ventana en las representaciones. Con respecto a esta cuestión, conviene destacar consideraciones en el comportamiento de la cámara, en base a la implementación que OpenGL comúnmente provee. Este ajuste se realiza, al igual que en otras aplicaciones convencionales, mediante el cursor sobre cualquiera de las esquinas de la ventana de representación. No obstante, a priori puede resultar confusa y llevar a errores de interpretación. Este comportamiento define un escalado de la representación para desplazamientos diagonales, un aumento en el ángulo de visión acimutal en desplazamientos horizontales y un escalado o zoom para desplazamientos verticales (no representa un aumento en el ángulo de elevación), tal y como muestra la figura 4.11. Esta configuración asimétrica resulta propia de aplicaciones de representación gráfica, pero alteran la proyección en el caso de las aplicaciones de esta Tesis. Por tanto, no se recomienda realizar modificaciones sobre las dimensiones de la ventana de representación de cara a la obtención de diagramas de cobertura óptica.

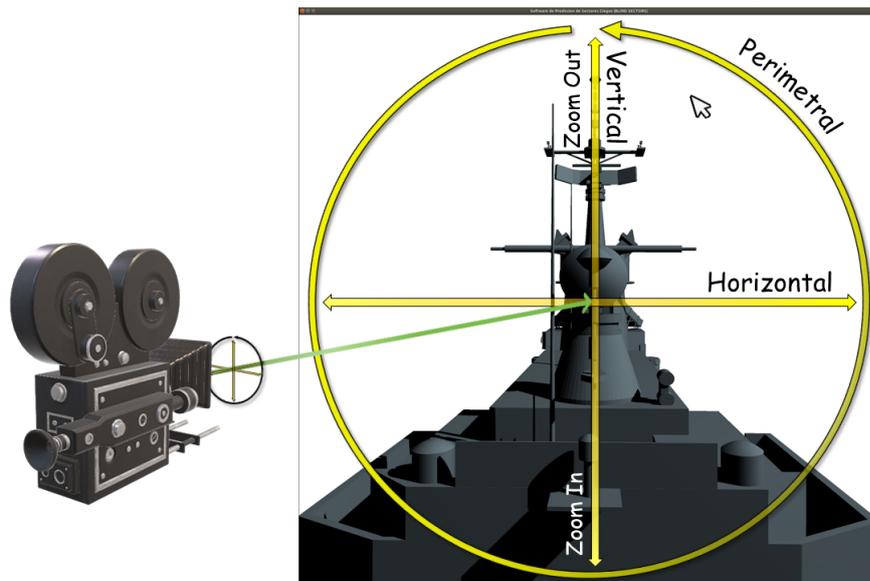


Figura 4.10: Desplazamientos y rotaciones del punto de observación (cámara) disponibles, mediante las funcionalidades del ratón.

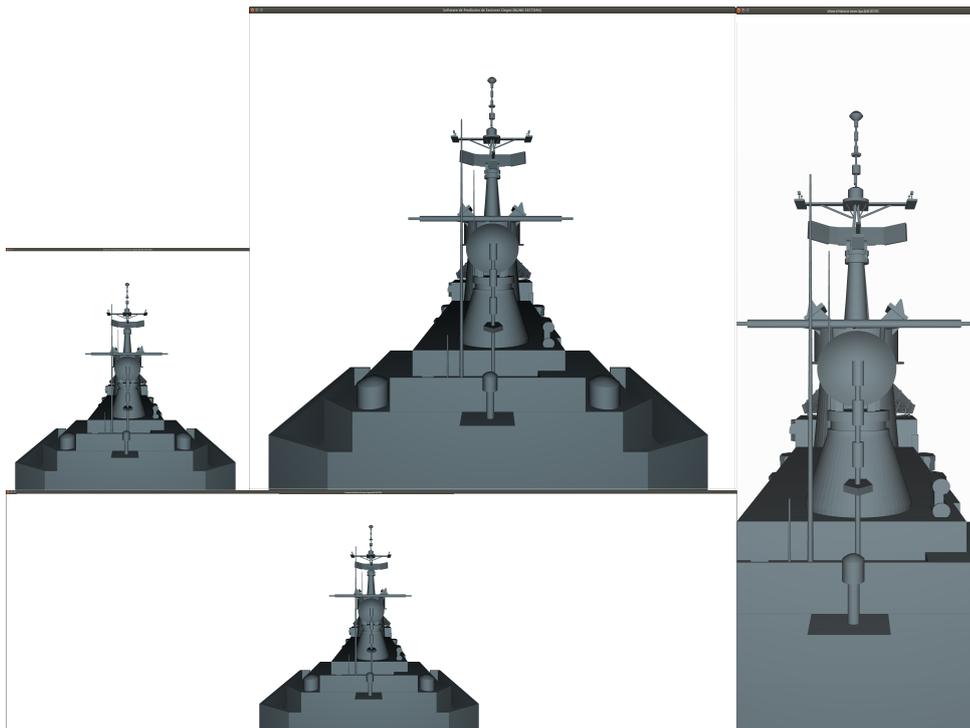


Figura 4.11: Comparativa de los resultados obtenidos al redimensionar la ventana de representación de referencia (imagen superior izquierda), realizando un escalado (imagen superior central), un aumento del rango acimutal (imagen inferior izquierda) y un zoom (derecha), mediante un desplazamiento diagonal, horizontal y vertical respectivamente.

Como solución a este problema, el software de predicción de sectores ciegos también permite configurar el posicionamiento y tamaño de la ventana mediante código. Esta solución permite especificar de manera inequívoca los parámetros de la configuración deseada, aunque requiere de recompilar nuevamente el código para hacer efectiva las especificaciones. La configuración mediante código define listas de posicionamiento e iluminación independientes para distintos sistemas, anticipando las diversas configuraciones de interés, evitando reconfigurar el código o desplazarse hasta la ubicación deseada. La elección de estas configuraciones se especifica mediante los actuadores del teclado correspondientes.

4.2.1.2. Representación de los diagramas de cobertura óptica y sectores ciegos

Una vez posicionado el punto de observación en la ubicación del elemento radiante (sistema radar en nuestro ejemplo), es necesario modificar el rango de representación de los diagramas de cobertura óptica mediante la transformación denominada *ojo de pez*. Este efecto se define como una representación de 360° en acimut y en torno a 135° en elevación (especificados de -45° a 90° en nuestro caso). El resultado de esta transformación puede observarse en la figura 4.12. Esta representación permite visualizar e identificar los elementos que interfieren en la línea de visión del sistema escogido. La herramienta también permite la superposición de los lóbulos principales de los diagramas de radiación (para el ancho de haz deseado especificado por el usuario) sobre la figura de cobertura, lo cual permite analizar con mayor precisión el efecto de los diferentes dispersores sobre las prestaciones de los sensores.

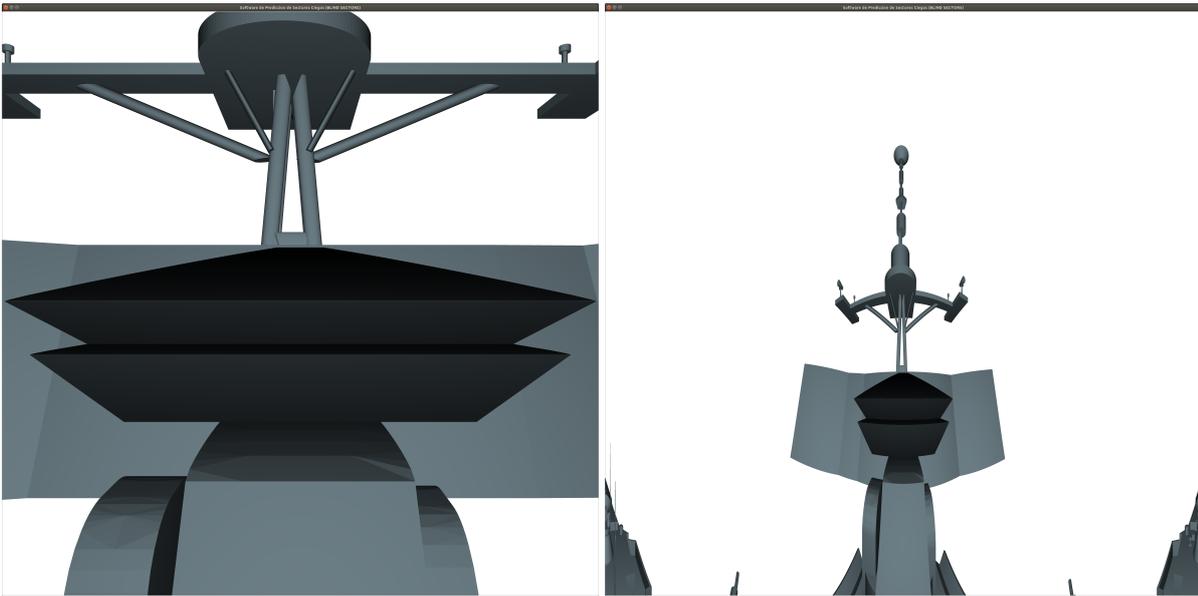


Figura 4.12: Representación del efecto *ojo de pez* a la derecha, de la antena dispuesta en la representación de la izquierda. En un ejemplo de línea de visión de una antena directiva en aplicaciones radar.

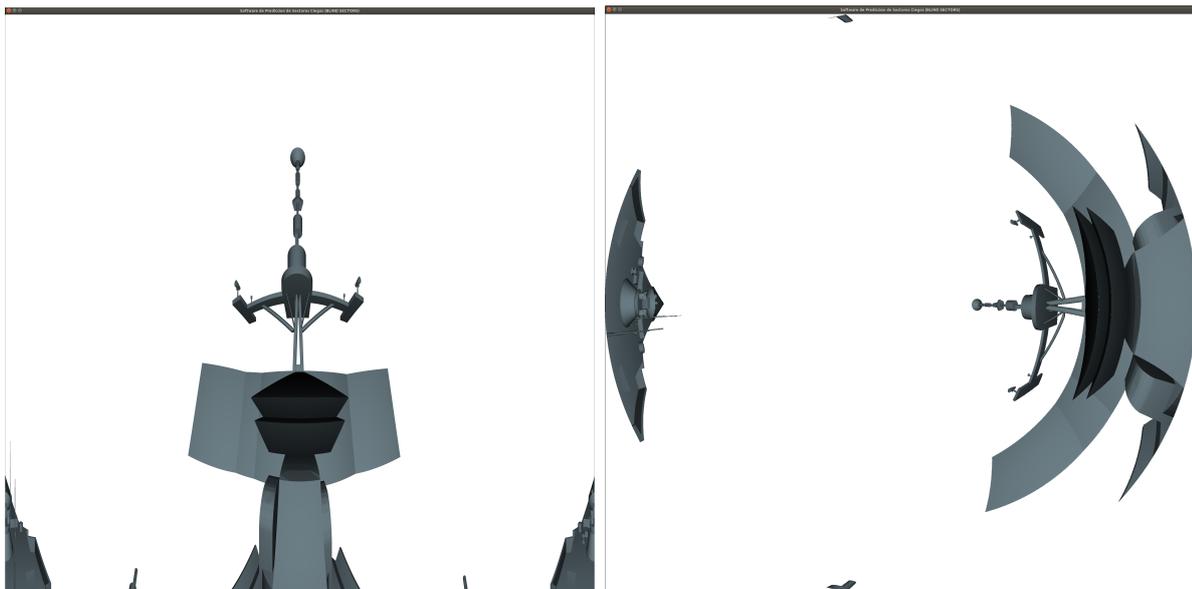


Figura 4.13: Representación de los 360° en acimut (efecto ojo de pez) y de -45° a 90° en elevación, a la izquierda en coordenadas cartesianas y a la derecha en coordenadas polares.

La representación de diagramas de cobertura óptica por defecto se realiza en coordenadas cartesianas; no obstante, resulta de utilidad poder representar los diagramas de cobertura óptica con el efecto *ojo de pez* en coordenadas polares, tal como se muestra en la figura 4.13. Esta representación requiere de un postprocesado gráfico en OpenGL mediante los denominados *shaders*, tal como se ha descrito en el apartado 3.2.4.

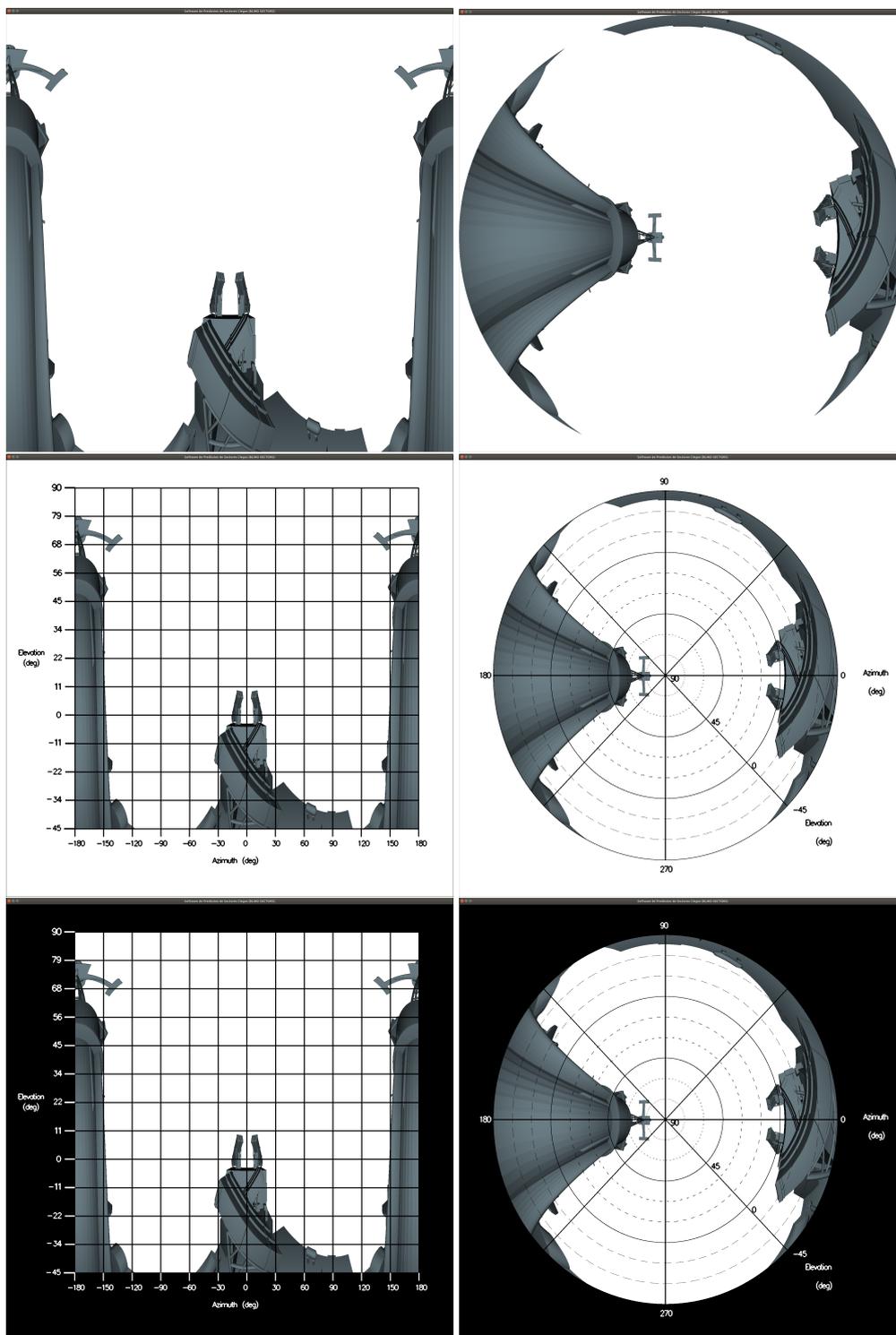


Figura 4.14: Representación de la cuadrícula y numeración de los ejes tanto en la representación cartesiana a izquierdas como en la representación polares a derechas. Se muestran las diferentes configuraciones conmutables entre sí de arriba a abajo.

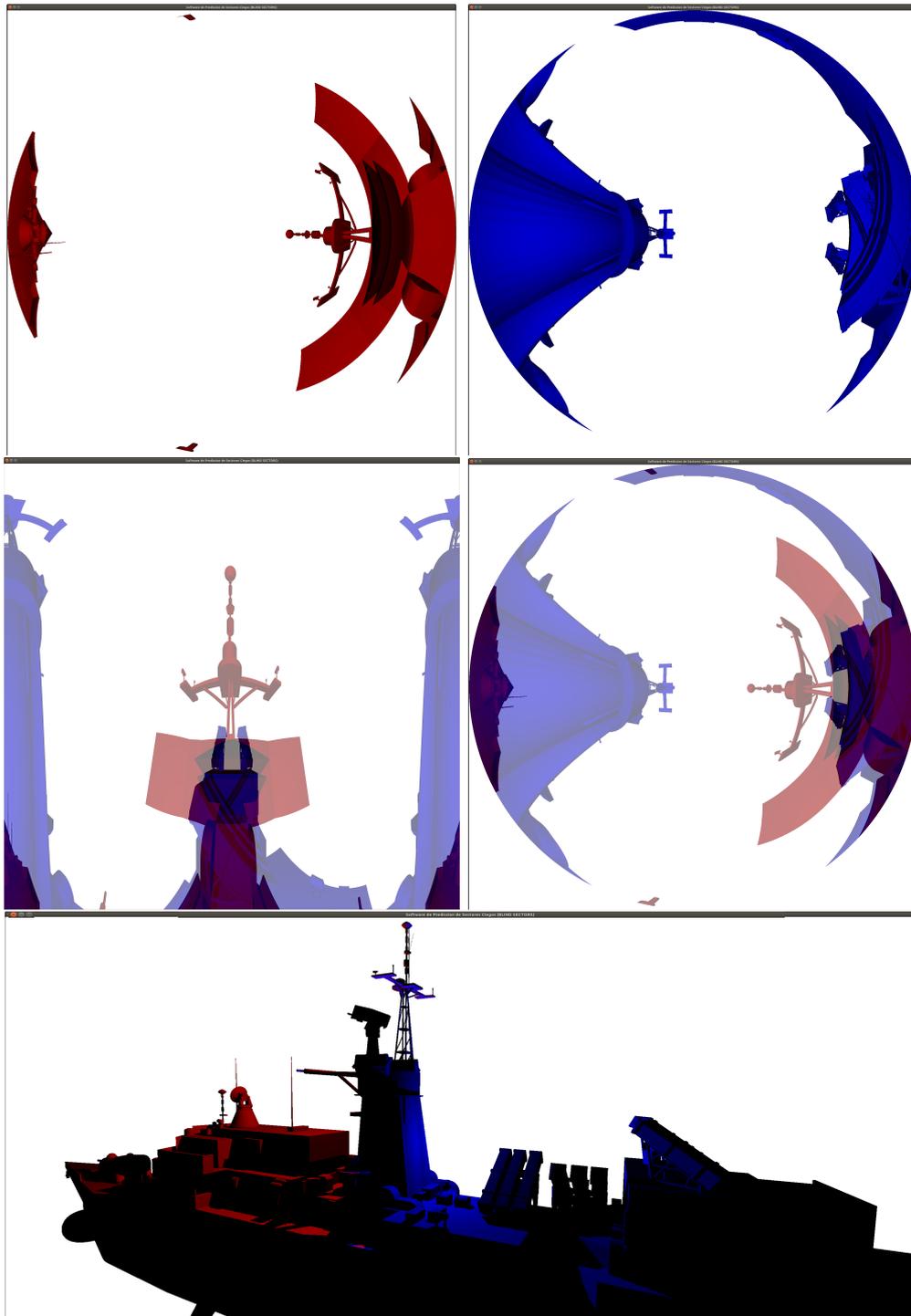


Figura 4.15: Representación de superposiciones en distintas configuraciones. Iluminación individualizada en los 4 primeros casos y simultánea en el último.

El postprocesado de OpenGL también permite representar ejes graduados para ambas configuraciones, lo que resulta de gran utilidad a la hora de generar resultados más fácilmente interpretables. Se ha configurado una cuadrícula en el caso de aplicar coordenadas cartesianas, con ángulos de acimut dispuestos sobre el eje x y de elevación en el eje y . En el caso de diagramas en coordenadas polares, se han incorporado radiales y circunferencias concéntricas para especificar los barridos en acimut y elevación respectivamente. Esta disposición puede verse en la figura 4.14 para el caso de un sistema ubicado en popa. La representación por defecto abarca los márgenes de -180° a 180° en acimut y de

-45° a 90° en elevación, aunque el direccionamiento de la cámara permite actualizar en tiempo real dicha representación, lo que se reflejará de manera automática con la modificación necesaria de los ejes graduados correspondientes. Esto permite obtener los diagramas de sectores en márgenes angulares arbitrarios.

Además de las funcionalidades anteriormente descritas, que permiten la obtención de diagramas de cobertura y sectores ciegos de manera convencional, el software desarrollado en esta Tesis provee algunas funcionalidades adicionales que mejoran su usabilidad y versatilidad. Además de representar los diagramas de cobertura de manera individualizada, es posible superponer en la misma gráfica los diagramas correspondientes a diferentes sistemas. Esta funcionalidad puede apreciarse en la figura 4.15 para dos sistemas, cuyos diagramas son ejecutados secuencialmente y representados como un promediado RGB de los píxeles finales. La configuración de este procedimiento se realiza de igual modo que las listas de posicionamiento e iluminación mencionadas anteriormente, pudiendo ser particularizados con diferentes parámetros de representación como el acabado o el comportamiento lumínico del material que compone la geometría, tal y como muestra la figura 4.16. La posibilidad de configurar estos parámetros permite identificar los diferentes sistemas a bordo que puedan operar simultáneamente, delimitando las áreas de actuación mediante la asignación de diferentes colores. De esta forma, la figura 4.17 muestra la interferencia producida por múltiples sistemas operando conjuntamente a los que se han asignado colores diferentes, como una composición lineal de los colores involucrados.

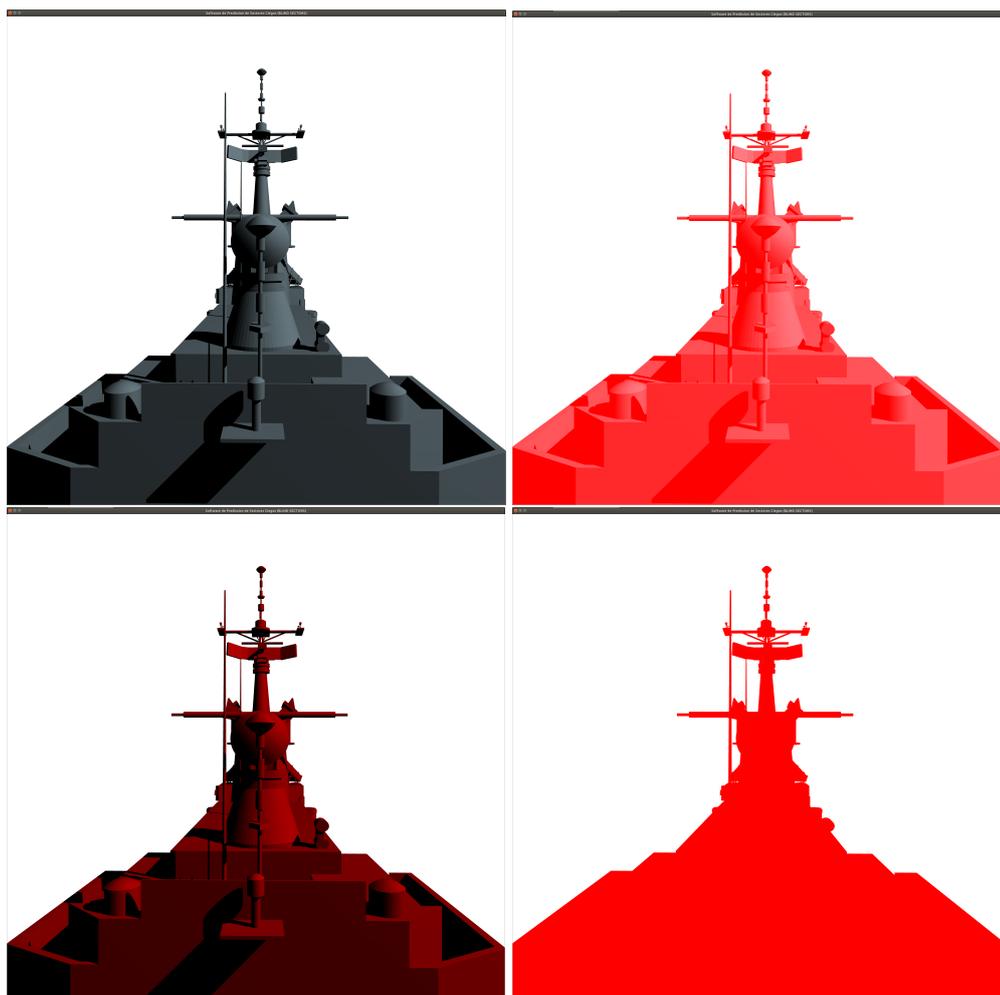


Figura 4.16: Representación en color, por defecto (rojo), de los diferentes acabados disponibles en función de los aspectos de iluminación y reflexión u absorción del material.

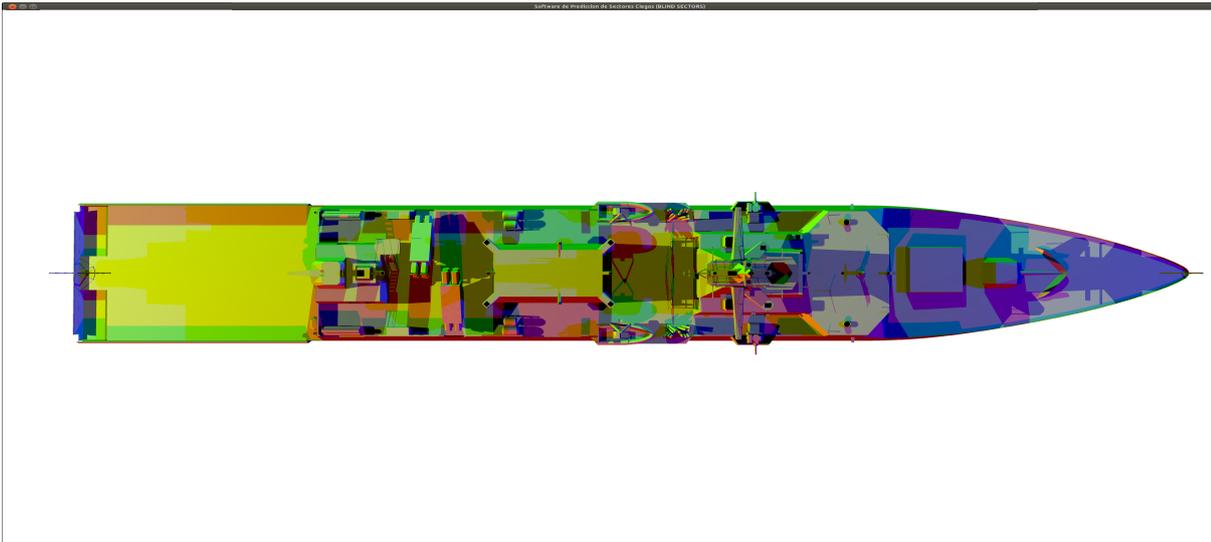


Figura 4.17: Representación simultánea de diversos puntos de iluminación, con los que identificar los ocultamientos e interferencia entre sistemas.

Una vez representados los diagramas de cobertura óptica, el software de predicción de sectores ciegos permite almacenar los resultados obtenidos en diferentes formatos de representación (png, ppm o tga) sin pérdida de resolución. Esto se debe a que el valor de los píxeles representados corresponde a los resultados obtenidos por la resolución del trazado de rayos de OptiX y por tanto, accesible para ser almacenado convenientemente, con independencia de la resolución de la pantalla.

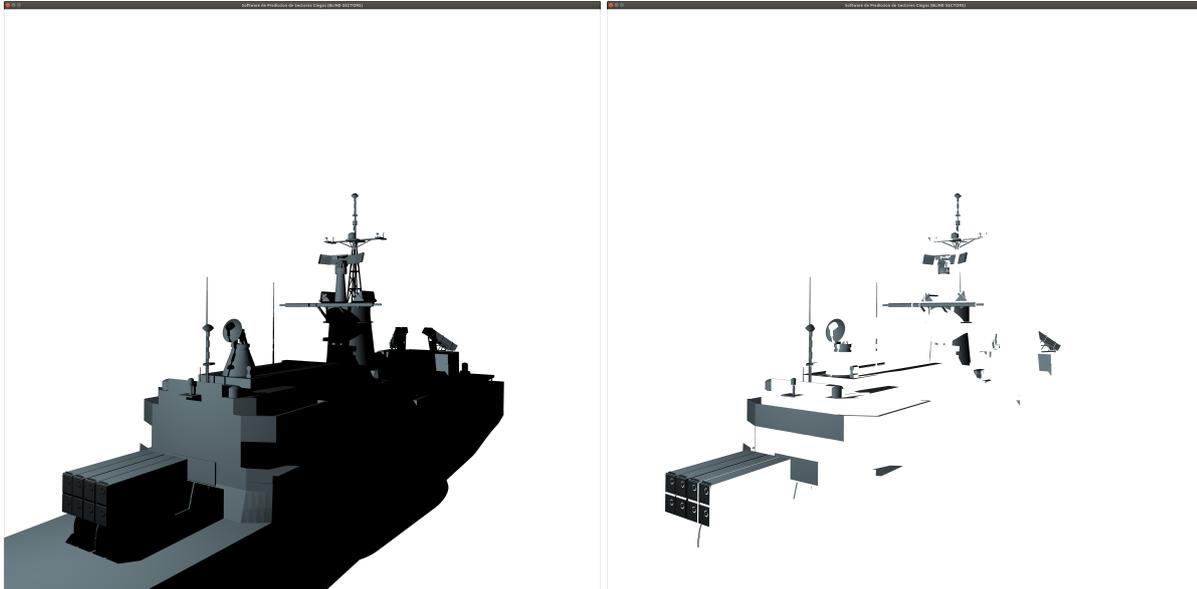


Figura 4.18: Representación de las facetas visibles para una dirección de incidencia (método de depuración de la matriz *iLit*).

Las funcionalidades del software de predicción de sectores ciegos no se limitan únicamente a la obtención de los diagramas de cobertura óptica, sino que además actúan como una herramienta de depuración y obtención de las matrices de visibilidades de IPO (*iLit* e *iVis*). Los procedimientos descritos en el apartado 4.19 son automatizados y asignados a actuadores con los que secuenciar el procedimiento de obtención de dichas matrices. El acceso a los datos durante las etapas del trazado de rayos, permite definir representaciones de interés. Este es el caso de la figura 4.18, para un posicionamiento e iluminación arbitrarios, que permite identificar las facetas iluminadas para una dirección de incidencia.

Esta representación permite depurar las columnas de la matriz *iLit* o *iVis*, previsualizando los resultados obtenidos antes de su almacenamiento. También resulta de interés la representación gráfica de procedimientos recursivos como la reflexión, mostrada en la figura 4.19 (se incluye una imagen de fondo para apreciar mejor la reflexión sobre la superficie). Esta representación permite deducir el origen de la contribución por reflexión de manera visual, observando el elemento reflejado desde la posición deseada.



Figura 4.19: Representación de la reflexión recursiva de un solo nivel.

4.2.2. Versiones beta

Los resultados expuestos anteriormente han sido obtenidos mediante la implementación del software de predicción de sectores ciegos basada en la librería GLUT, que describimos en el apartado 3.2.4. No obstante, actualmente existe una versión en fase de desarrollo, que denominamos *beta*, basada en la librería ImGui. Esta versión incorpora algunas diferencias con respecto a las funcionalidades vistas en la versión anterior, cuya principal mejora reside en la usabilidad de software.

La incorporación más evidente de esta mejora se encuentra en la interfaz gráfica, la cual muestra ventanas desplegadas que permiten organizar e identificar las funcionalidades disponibles por la herramienta. En la figura 4.20 se puede apreciar el conjunto de actuadores gráficos que implementa la librería ImGui para la configuración de la escena. La incorporación de casillas editables, desplegadas, barras de scroll, etc. son las principales aportaciones, junto con los actuadores ya implementados (ratón y teclado), que permiten a esta versión combinar la ejecución con la configuración en tiempo real. Por tanto, esta versión permite independizar al usuario por completo de los procesos de compilación que requería la versión anterior, centrándose en el manejo del ejecutable que provee de una interfaz similar a las aplicaciones o programas de uso común.

Junto a esta versión existen implementaciones paralelas cuyos resultados no han podido ser incorporados a esta Tesis, pero merece la pena citar. Este es el caso de la implementación basada en uno de los ejemplos provistos por OptiX SDK (*OptiX Ocean*), basada en las librerías GLFW e ImGui (optimizadas para aplicaciones de renderizado y animación). Este ejemplo resulta de especial interés debido a su potencial aplicación en el desarrollo de proyectos, el cual provee de un modelo de mar realista capaz de contemplar distintas iluminaciones y estados de mar. El objetivo tras esta implementación es combinar el software de predicción de sectores con este modelo, a fin de obtener una estimación aun más realista de

las contribuciones de RCS en presencia del plano de mar, además de una representación animada de un escenario real.



Figura 4.20: Representación gráfica de la versión beta del software de detección de sectores ciegos.

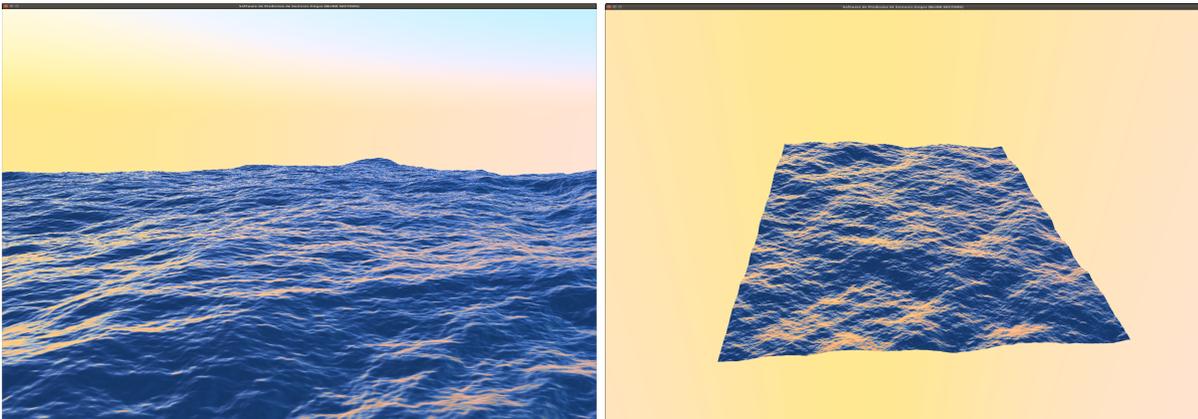


Figura 4.21: Animación del plano de mar basada en el ejemplo *OptiX Ocean* provisto por NVIDIA.

4.3. Método híbrido DDM-IPO

En esta sección se propone la resolución del problema de RCS anterior para el buque Tabarca mediante un esquema DDM-IPO. Esta solución persigue mejorar la estimación de la RCS, resolviendo de manera independiente dominios de menor tamaño y mayor complejidad, mediante métodos *full-wave*, mientras que las regiones de gran tamaño, líneas más simples y superficies más suaves se incluyen mediante IPO.

Los dispersores escogidos en este caso para ser analizados mediante MLFMA se muestran en la figura 4.22, representados en amarillo.

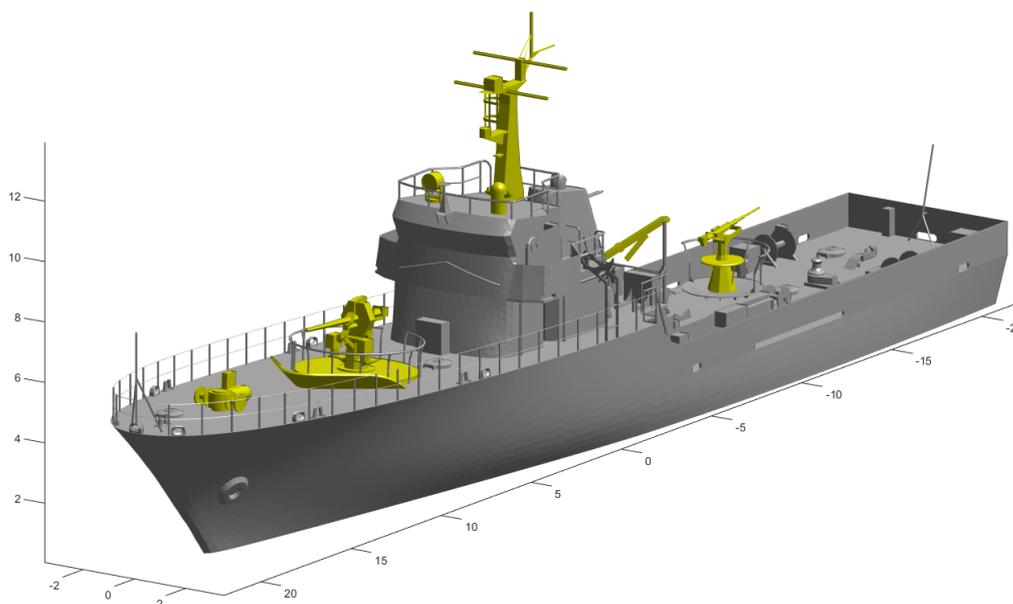


Figura 4.22: Configuración de los dispersores tenidos en cuenta para el esquema DDM-IPO.

Previo al análisis electromagnético, se determina las matrices de visibilidades de las regiones IPO con respecto a los dominios *full-wave*. Este procedimiento requiere del software de predicción de sectores ciegos para definir las facetas de la región IPO que resultan visibles desde cada dispersor. Se ha considerado como punto de referencia el centro de masas geométrico de cada dispersor. En la figura 4.23 y 4.24 pueden observarse los diagramas de cobertura óptica representados en coordenadas cartesianas y polares respectivamente para cada uno de los dispersores contemplados en este ejemplo. Para ello, es preciso configurar un escenario en ausencia del elemento bajo estudio y en presencia del resto. La observación se sitúa entonces en el centro de masa de cada dispersor en cuestión y se calculan así las matrices de visibilidades. Estas matrices permiten determinar los acoplos mutuos entre dominios *full-wave* y el dominio IPO. El acoplo mutuo entre los dominios *full-wave* se obtiene de manera directa, y también de manera indirecta a través de la región IPO. A estas contribuciones se suma la estimación directa de la RCS del dominio IPO en ausencia de los dispersores.

En la tabla 4.3, puede consultarse el número de incógnitas y tiempos de cómputo necesarios para la resolución de cada dominio de manera independiente. Estos tiempos fueron obtenidos sobre una máquina AMD EPYC 7763 64-Core Processor, disponiendo de un total de 4 máquinas para la ejecución del método DDM-IPO. El resultado de la RCS monoestática calculada para cada dominio puede verse en la figura 4.25.

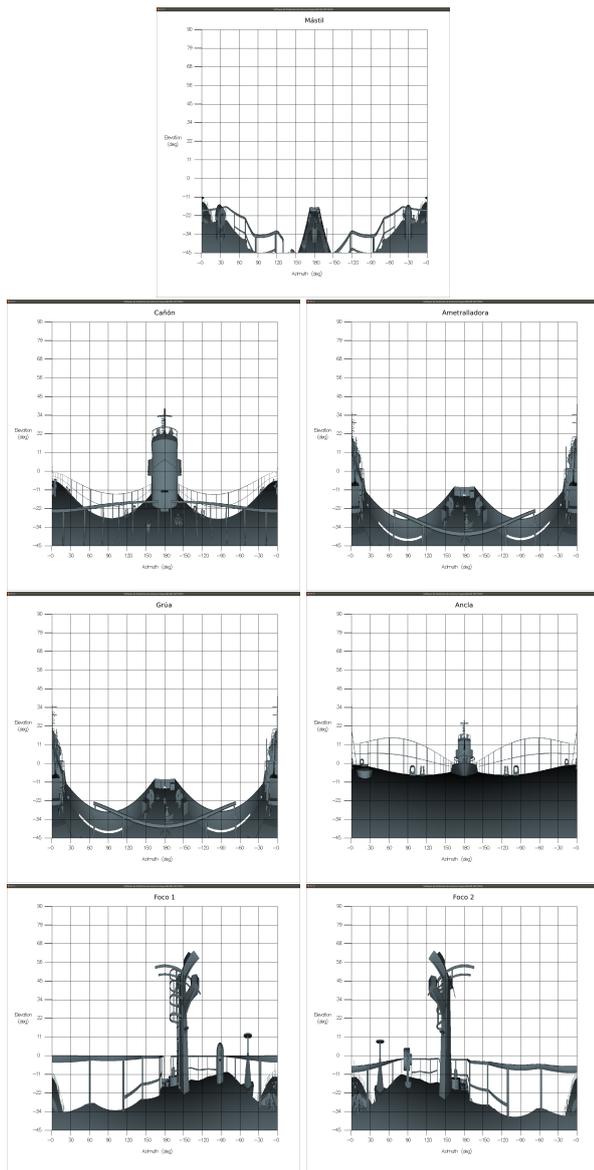


Figura 4.23: Contribuciones de RCS de los diferentes elementos del problema enmascaradas.

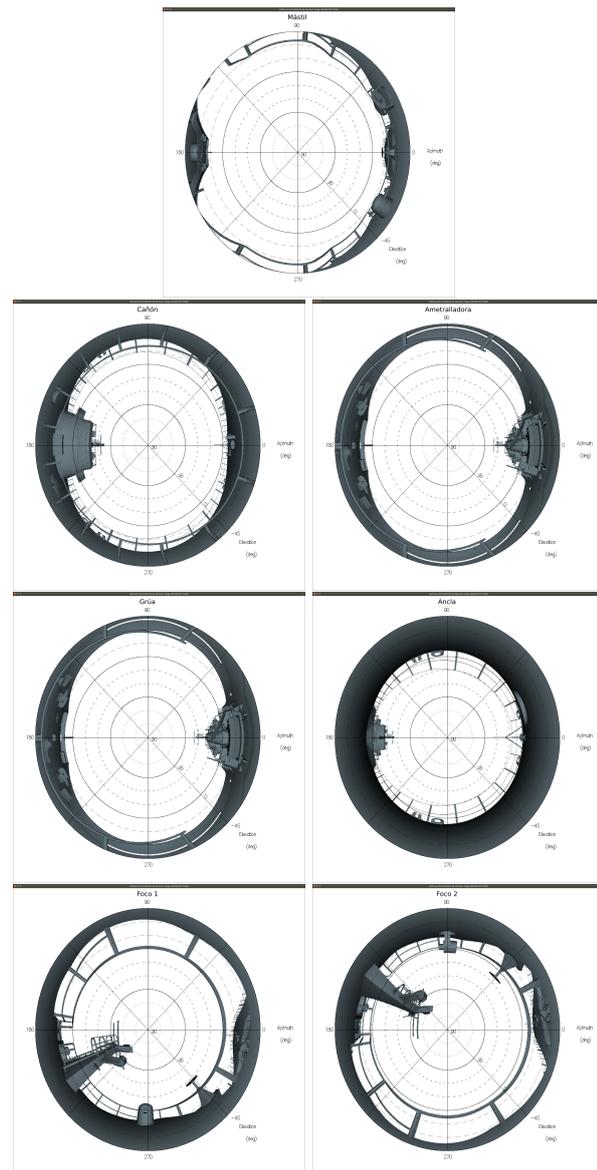


Figura 4.24: Contribuciones de RCS de los diferentes elementos del problema.

Tabla 4.3: Número de incógnitas y tiempo de cómputo de los diferentes dominios del esquema DDM-IPO.

Dispersores	Estructura	Mástil	Cañón	Ametralladora	Grúa	Ancla	Foco 1	Foco 2
Incógnitas	213233	5057444	12105171	2486064	1554897	2008644	610413	459696
Tiempo (h)	0.84	199.99	493.08	53.61	128.27	93.71	18.29	45.28

A partir de las matrices de visibilidades se obtienen las máscaras de visibilidades/ocultaciones para cada dispersor. Esta máscara permite ponderar la contribución de la RCS, obtenida de la resolución individualizada de cada dominio *full-wave* en presencia de la región IPO. La figura 4.26 muestra la contribución de los resultados representados en la figura 4.25. Esta representación permite inferir la contribución de cada dispersor para cada dirección de incidencia.

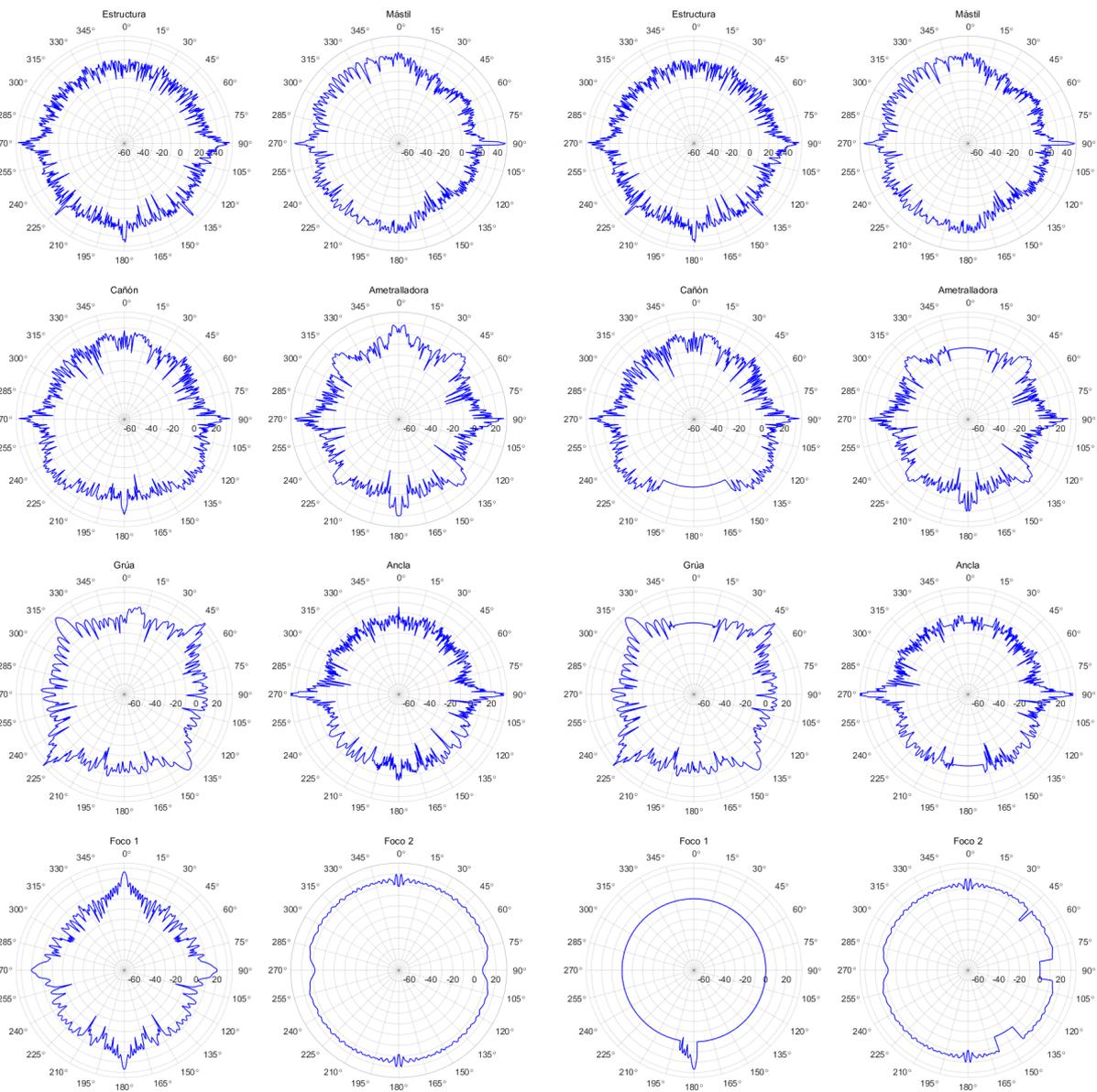


Figura 4.25: Contribuciones de RCS de los diferentes elementos del problema.

Figura 4.26: Contribuciones de RCS de los diferentes elementos del problema enmascaradas.

Finalmente, el resultado obtenido mediante el método DDM-IPO, junto con los resultados de IPO-GO obtenidos previamente y las medidas, se muestra en la figura 4.27. Esta solución ha sido alcanzada tras la tercera iteración del GMRES externo, proporcionando un residuo de 10^{-3} .

La principal mejora de la solución propuesta puede apreciarse en las direcciones de proa, para las que las figuras 4.28 y 4.29 muestran los resultados con un promediado angular de 2° y en sectores angulares de 10° respectivamente. En la tabla 4.4 se recogen los estadísticos de la solución propuesta con respecto a las medidas y la solución obtenida en la sección 4.1. Estos valores pueden ser consultados para la totalidad de los valores de la RCS o en base a los sectores descritos en la figura 4.7. De igual modo, en la tabla 4.5 se recogen los estadísticos obtenidos sobre la RCS promediada mediante la aplicación de una ventana deslizante de 2° en acimut, que se representaba la figura 4.28.

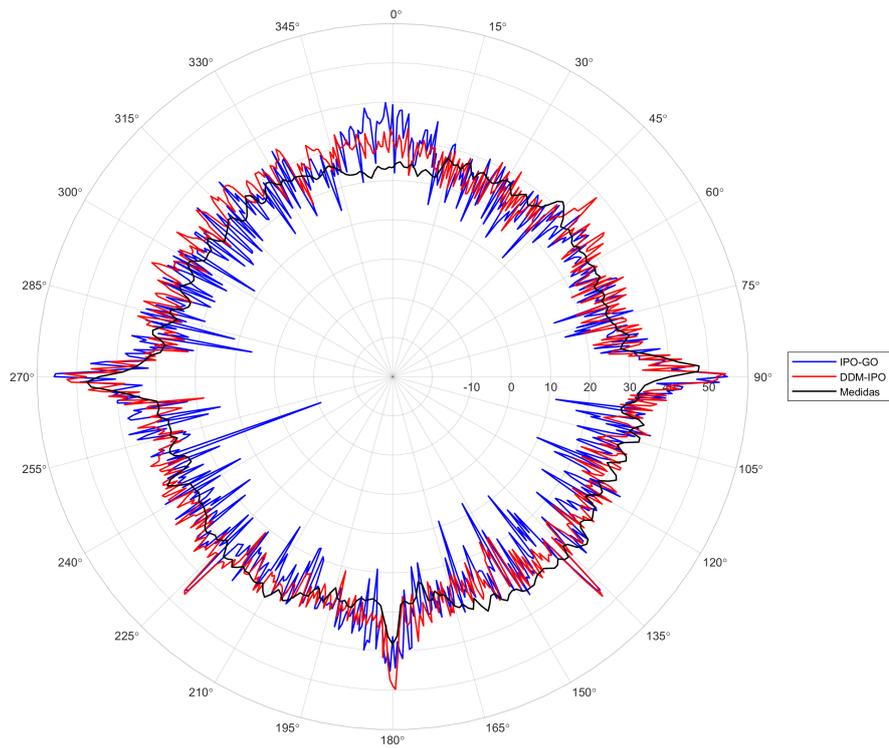


Figura 4.27: RCS monoestática del buque Tabarca obtenidas por IPO-GO, DDM-IPO y las medidas.

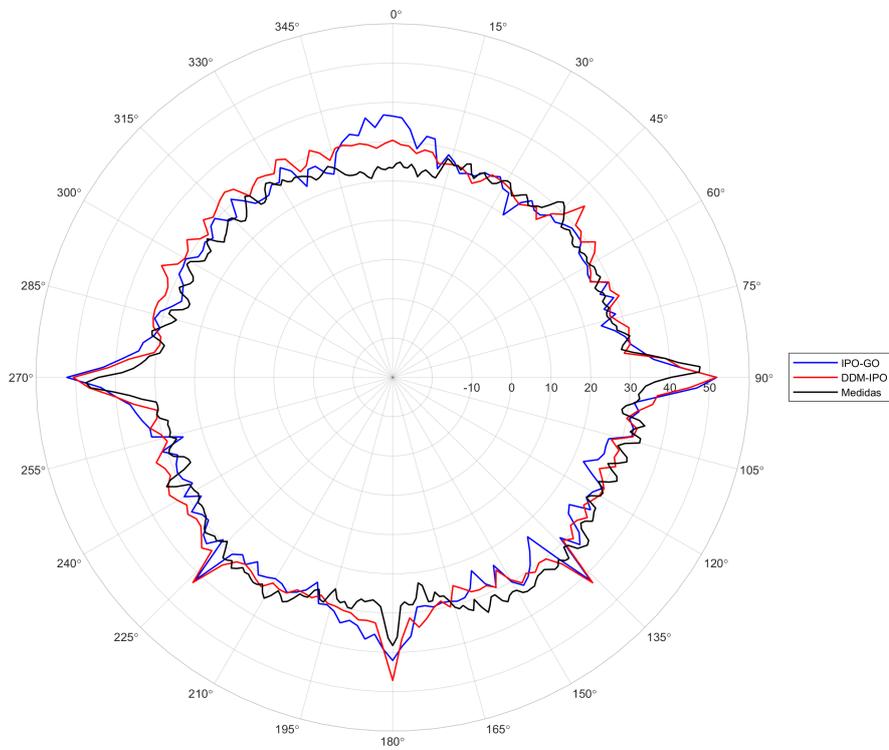


Figura 4.28: RCS monoestática del buque Tabarca obtenidas por IPO-GO, DDM-IPO y las medidas con un promediado angular de 2° .

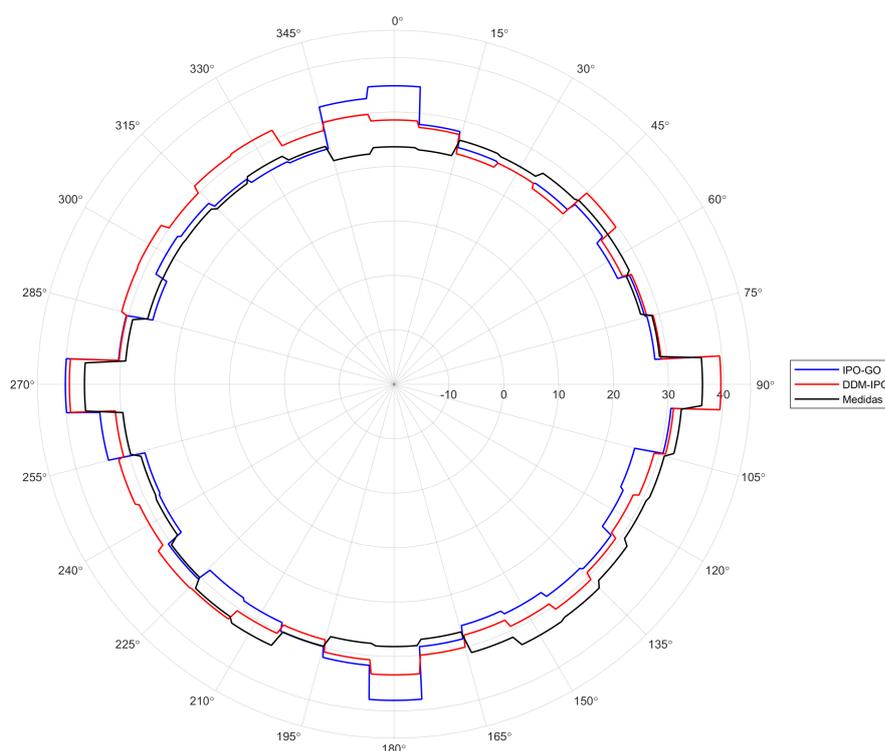


Figura 4.29: Mediana de la RCS monoestática del buque Tabarca obtenidas por IPO-GO, DDM-IPO y las medidas representada en sectores angulares de 10°.

De los valores contenidos de las tablas 4.5 y 4.4, se extrae que la solución obtenida por el esquema DDM-IPO en el sector II sigue siendo la que ofrece mejor correlación con las medidas. No obstante, para el caso del percentil 90, que contempla todas las direcciones, sin descartar direcciones de sacrificio, la solución obtenida mediante IPO-GO difiere 2.74 dB con respecto a las medidas, mientras que la solución del DDM-IPO difiere en 1.24 dB, por lo que se puede concluir que el análisis mediante técnicas full-wave de dispersores complejos, combinado con el uso de IPO para las superficies de mayor tamaño en el marco de una solución basada en DDM, ofrece un compromiso muy razonable entre eficiencia de cómputo y precisión en la predicción de la RCS monoestática de grandes plataformas portantes a las frecuencias de interés en radar.

Tabla 4.4: Estadísticos para los resultados de RCS monoestática obtenidos por IPO-GO, DDM-IPO y las medidas.

Estadísticos		Media (dBsm)	Mediana (dBsm)	Percentil 90 (dBsm)
IPO-GO	Total	30.80	27.38	36.18
	Sector I	28.22	26.10	31.40
	Sector II	31.29	27.36	33.66
	Sector III	32.61	28.78	35.87
	Sector IV	29.91	27.19	33.81
DDM-IPO	Total	31.45	29.49	34.82
	Sector I	28.61	27.22	32.66
	Sector II	31.96	29.24	33.73
	Sector III	33.01	31.22	34.77
	Sector IV	31.07	30.39	33.71
Medidas	Total	29.53	27.99	33.73
	Sector I	27.28	27.06	29.09
	Sector II	31.80	31.51	34.09
	Sector III	30.26	29.66	32.83
	Sector IV	26.80	26.19	28.87

Tabla 4.5: Estadísticos para los resultados de RCS monoestática obtenidos por IPO-GO, DDM-IPO y las medidas, con promedio angular de 2°.

Estadísticos		Media (dBsm)	Mediana (dBsm)	Percentil 90 (dBsm)
IPO-GO	Total	30.69	28.89	36.47
	Sector I	28.10	26.94	31.04
	Sector II	31.05	28.75	32.53
	Sector III	32.49	30.50	35.41
	Sector IV	30.01	28.52	33.09
DDM-IPO	Total	31.35	30.21	34.97
	Sector I	28.51	27.32	30.60
	Sector II	31.88	29.94	33.39
	Sector III	32.85	31.26	33.98
	Sector IV	31.07	30.85	32.63
Medidas	Total	29.53	27.99	33.73
	Sector I	27.28	27.06	29.09
	Sector II	31.80	31.51	34.09
	Sector III	30.26	29.66	32.83
	Sector IV	26.80	26.19	28.87

CAPÍTULO 5 **Conclusiones**

Contenido

5.1. Conclusiones	88
5.2. Líneas futuras	89

5.1. Conclusiones

En esta tesis se han propuesto una serie de procedimientos orientados a la aceleración y mejora del método IPO en aplicaciones de cálculo de la RCS monoestática a frecuencias radar para objetos de gran tamaño eléctrico.

Se ha implementado en primer lugar una resolución eficiente y rigurosa de la MFIE mediante el algoritmo IPO combinado con una discretización de la corriente inducida sobre facetas triangulares con modelado constante en amplitud y lineal para la fase. Esto ha permitido reducir de manera muy significativa el coste computacional asociado, facilitando el cómputo de múltiples interacciones sobre la estructura (alcanzando reflexiones de tercer orden) en blancos de interés real como puede ser un buque a frecuencias radar. Se ha desarrollado posteriormente una mejora del algoritmo, mediante la inclusión de GO para modelar las contribuciones de tercer orden, con el que se han logrado reducir los tiempos de cómputo, mejorando la eficiencia, sin reducir por ello el nivel de precisión del algoritmo original.

La segunda vía de trabajo llevada a cabo ha consistido en acelerar el cálculo de las matrices de visibilidades en IPO mediante el trazado de rayos sobre GPU. Se han aplicado los avances más recientes mediante el empleo de la arquitectura OptiX, derivando el cómputo masivo de los procesos que determinan las visibilidades/ocultaciones al hardware específico, con excelentes resultados. El desempeño de procesos SIMT mediante GPU ha resultado ser una solución adecuada con la que obtener estos resultados prácticamente en tiempo real. Adicionalmente, se ha desarrollado un software de detección de sectores ciegos que permite determinar de manera muy eficaz diagramas de cobertura óptica y ocultaciones para todo tipo de objetos a bordo, incluyendo sensores, antenas, elementos dispersores, etc.

Finalmente, los desarrollos anteriores se han integrado en una metodología DDM-IPO, que ha demostrado ofrecer un buen compromiso entre precisión y coste computacional en el cálculo de la RCS monoestática de grandes plataformas a frecuencias radar.

En definitiva, la colección de herramientas propuestas y desarrolladas en esta Tesis proporciona una alternativa viable con la que abordar proyectos que requieran de simulaciones intensivas, ofreciendo además gran flexibilidad para la realización y evaluación de cambios sobre las estructuras y garantizando unas precisiones suficientes con unos tiempos de cálculo reducidos.

5.2. Líneas futuras

Las soluciones implementadas en esta Tesis ofrecen un amplio repertorio de posibles mejoras que podrían ser de interés para la industria. El previsible incremento de la complejidad y tamaño eléctrico de los problemas a resolver, debido al desarrollo de sistemas en bandas de frecuencias cada vez más elevadas, demandarán soluciones que permitan combinar eficiencia de cálculo y agilidad a la hora de realizar simulaciones con una precisión suficiente sobre la que sostener la toma de decisiones desde las etapas iniciales de los proyectos.

En este contexto, resultaría interesante la implementación de algoritmos de refinado y criterios de subdivisión geométricas aprovechando las capacidades de paralelización masiva sobre GPUs, procesos que actualmente se realizan sobre CPU. Incluso el cálculo electromagnético, o al menos parte, podría derivarse también hacia las GPUs, además de los cálculos puramente geométricos, lo que reduciría más aún los tiempos de computación. Teniendo en cuenta el carácter estocástico de la RCS, esto permitiría simular de forma más realista los procedimientos de medidas, considerando diferentes entornos, estados de la mar, etc.

Continuando con la obtención de resultados de especial interés para el desarrollo de proyectos, el cálculo de imágenes radar 3D tendría un papel relevante como futura línea de investigación. Para ello sería necesario extender el cálculo de la RCS mediante IPO a los procedimientos de *k-space* [83]. Este estudio tendría por objeto determinar los elementos de mayor contribución a la RCS de blancos complejos en función de la dirección.

Por otro lado, una vía natural de continuidad del software de detección de sectores ciegos sería extender las capacidades de representación a un entorno animado, provisto de múltiples escenarios con elementos móviles que permitiesen evaluar diferentes configuraciones. En consecuencia, sería conveniente también incrementar la potencia de cálculo mediante la interconexión de equipos en paralelo, para mantener la ejecución en tiempo real y adaptarla al nuevo nivel de exigencia. También sería deseable la inclusión de diversos estados de la mar mediante técnicas de trazado de rayos basadas en GPUs, lo que facilitaría la caracterización realmente estocástica de la RCS.

Otras líneas de actuación implicarían la incorporación de mejoras en el método DDM-IPO, incluyendo desde aceleraciones de los procesos de acoplamiento entre dominios mediante radiación a la incorporación de diferentes preconditionadores para mejorar la convergencia. Otro aspecto a mejorar en el contexto del cálculo de la RCS, consistiría en convenir criterios que determinen los dispersores que requieren de un análisis *full-wave*, frente a aquellos que pueden caracterizarse con suficiente precisión mediante IPO. Aunque se han llevado a cabo estudios preliminares de esta naturaleza a lo largo de esta Tesis, en el marco de su aplicación a diferentes proyectos, no se ha realizado un estudio pormenorizado y exhaustivo en este aspecto. Esto permitiría optimizar los recursos permitiendo alcanzar la precisión necesaria en el menor tiempo de cómputo posible.

Bibliografía

- [1] N.N. Youssef. «Radar cross section of complex targets». En: *Proceedings of the IEEE* 77.5 (1989), págs. 722-734. DOI: 10.1109/5.32062.
- [2] J.M. Rius, M. Ferrando y L. Jofre. «High-frequency RCS of complex radar targets in real-time». En: *IEEE Transactions on Antennas and Propagation* 41.9 (1993), págs. 1308-1319. DOI: 10.1109/8.247759.
- [3] R.A. Shore y A.D. Yaghjian. «Incremental diffraction coefficients for planar surfaces». En: *IEEE Transactions on Antennas and Propagation* 36.1 (1988), págs. 55-70. DOI: 10.1109/8.1075.
- [4] Vall-Ilossera Rius J.M. y M. & Cardama. «Greco : graphical processing methods for high-frequency RCS prediction». En: *Annales Des Télécommunications* 0.50 (1995), págs. 551-556. DOI: 10.1007/BF02995755.
- [5] D. Andersh et al. «Xpatch 4: the next generation in high frequency electromagnetic modeling and simulation software». En: *Record of the IEEE 2000 International Radar Conference [Cat. No. 00CH37037]*. 2000, págs. 844-849. DOI: 10.1109/RADAR.2000.851945.
- [6] Robert J. Burkholder et al. «Iterative Physical Optics for Radar Scattering Predictions». En: 2009.
- [7] J. L. Rodriguez F. Obelleiro y R. J. Burkholder. «An iterative physical optics approach for analyzing the electromagnetic scattering by large open-ended cavities». En: *IEEE Transactions on Antennas and Propagation* 43.4 (1995), págs. 356-361. DOI: 10.1109/8.376032.
- [8] R.J. Burkholder et al. «Iterative physical optics: its not just for cavities anymore [EM wave propagation]». En: *2005 IEEE Antennas and Propagation Society International Symposium*. Vol. 1A. 2005, 18-21 Vol. 1A. DOI: 10.1109/APS.2005.1551230.
- [9] Cai-Cheng Liu y Weng Cho Chew. «Fast far field approximation for calculating the RCS of large objects». En: *IEEE Antennas and Propagation Society International Symposium. 1995 Digest*. Vol. 1. 1995, 22-25 vol.1. DOI: 10.1109/APS.1995.529955.
- [10] H. E. Pocklington. «Electrical oscillations in wires». En: *Camb. Phil. Soc. Proc.* 9 (1897), págs. 324-332.
- [11] R. F. Harrington. *Field Computation by Moment Method*. IEEE Press, 1993.
- [12] R. Coifman, V. Rokhlin y S. Wandzura. «The fast multipole method for the wave equation: a pedestrian prescription». En: *IEEE Antennas and Propagation Magazine* 35.3 (1993), págs. 7-12. DOI: 10.1109/74.250128.
- [13] J. M. Taboada et al. «High Scalability FMM-FFT Electromagnetic Solver for Supercomputer Systems». En: *IEEE Antennas and Propagation Magazine* 51.6 (2009), págs. 20-28. DOI: 10.1109/MAP.2009.5433091.
- [14] X. Q. Sheng et al. «Solution of combined-field integral equation using multilevel fast multipole algorithm for scattering by homogeneous bodies». En: *IEEE Transactions on Antennas and Propagation* 46.11 (1998), págs. 1718-1726. DOI: 10.1109/8.736628.
- [15] M. G. Araújo et al. «Solution of large-scale plasmonic problems with the multilevel fast multipole algorithm». En: *Opt. Lett.* 37.3 (feb. de 2012), págs. 416-418. DOI: 10.1364/OL.37.000416.
- [16] J. M. Song y W. C. Chew. «Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering». En: *Microwave and Optical Technology Letters* 10.1 (1995), págs. 14-19. DOI: <https://doi.org/10.1002/mop.4650100107>.

- [17] J. Song, Cai-Cheng Lu y Weng Cho Chew. «Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects». En: *IEEE Transactions on Antennas and Propagation* 45.10 (1997), págs. 1488-1493. DOI: 10.1109/8.633855.
- [18] Xiao-Min Pan y X Sh. «A Sophisticated Parallel MLFMA for Scattering by Extremely Large Targets [EM Programmer's Notebook]». En: *IEEE Antennas and Propagation Magazine* 50 (2008), págs. 129-138.
- [19] J.M. Song et al. «Fast Illinois solver code (FISC)». En: *IEEE Antennas and Propagation Magazine* 40.3 (1998), págs. 27-34. DOI: 10.1109/74.706067.
- [20] W.C. Chew et al. *Fast and Efficient Algorithms in Computational Electromagnetics*. USA: Artech House, Inc., 2001. ISBN: 1580531520.
- [21] J. M. Taboada et al. «MLFMA-FFT Parallel Algorithm for the Solution of Extremely Large Problems in Electromagnetics». En: *Proceedings of the IEEE* 101.2 (2013), págs. 350-363. DOI: 10.1109/JPROC.2012.2194269.
- [22] J. M. Bertolo L. Landesa F. Obelleiro J. M. Taboada M. G. Araújo y J. L. Rodriguez. «MLFMA-FFT Parallel Algorithm for the Solution of Large-Scale Problems in Electromagnetics (Invited Paper)». En: *Progress In Electromagnetics Research* 105.3 (2010), págs. 15-30. DOI: 10.2528/PIER10041603.
- [23] S. Velampambil y Weng Cho Chew. «Analysis and performance of a distributed memory multilevel fast multipole algorithm». En: *IEEE Transactions on Antennas and Propagation* 53.8 (2005), págs. 2719-2727. DOI: 10.1109/TAP.2005.851859.
- [24] R.L. Wagner, Jiming Song y W.C. Chew. «Monte Carlo simulation of electromagnetic scattering from two-dimensional random rough surfaces». En: *IEEE Transactions on Antennas and Propagation* 45.2 (1997), págs. 235-245. DOI: 10.1109/8.560342.
- [25] J. Jin. *The Finite Element Method in Electromagnetics*. Wiley, New York, 2002.
- [26] F. Hao et al. «Plasmon Resonances of a Gold Nanostar». En: *Nano Letters* 7.3 (2007), págs. 729-732. DOI: 10.1021/nl062969c.
- [27] D. B. Davidson. *Computational electromagnetics for RF and microwave engineering*. Google Books, 2010.
- [28] <http://www.comsol.com>.
- [29] <https://www.cst.com>.
- [30] P.H. Pathak. *Techniques for High-Frequency Problems*. Antenna Handbook Theory, Applications y Design, 1988. ISBN: Y.T., Lee S.W., Eds. New York:Van Nostrand Reinhold.
- [31] H. Ling, R.-C. Chou. y S.-W. Lee. «Shooting and bouncing rays: calculating the RCS of an arbitrarily shaped cavity». En: *IEEE Transactions on Antennas and Propagation* 37.2 (1989), págs. 194-205. DOI: 10.1109/8.18706.
- [32] R.J. Burkholder y P.H. Pathak. «Analysis of EM penetration into and scattering by electrically large open waveguide cavities using Gaussian beam shooting». En: *Proceedings of the IEEE* 79.10 (1991), págs. 1401-1412. DOI: 10.1109/5.104215.
- [33] A. Garcia-Pino, F. Obelleiro y J. L. Rodriguez. «Scattering from conducting open cavities by generalized ray expansion (GRE)». En: *IEEE Transactions on Antennas and Propagation* 41.7 (1993), págs. 989-992. DOI: 10.1109/8.237634.
- [34] J.B. Keller. *Geometrical theory of diffraction*. J. Opt. Soc. Am., 1962. ISBN: 52, 116-130.
- [35] R.G. Kouyoumjian y P.H. Pathak. «A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface». En: *Proceedings of the IEEE* 62.11 (1974), págs. 1448-1461. DOI: 10.1109/PROC.1974.9651.
- [36] Y. Zhao. «Fast physical optics calculation for SAR imaging of complex scatterers». En: *M.S. thesis, Elect. Comput. Eng., The Ohio State Univ., Columbus, OH, US* (2012).
- [37] Felipe Cátedra et al. «Efficient Techniques for Accelerating the Ray-Tracing for Computing the Multiple Bounce Scattering of Complex Bodies Modeled by Flat Facets». En: 2010.

- [38] R. M. James. «A contribution to scattering calculation for small wavelengths-the high frequency panel method». En: *IEEE Transactions on Antennas and Propagation* 38.10 (1990), págs. 1625-1630. DOI: 10.1109/8.59776.
- [39] K.R. Aberegg y A.F. Peterson. «Application of the integral equation-asymptotic phase method to two-dimensional scattering». En: *IEEE Transactions on Antennas and Propagation* 43.5 (1995), págs. 534-537. DOI: 10.1109/8.384199.
- [40] C.S. Kim e Y. Rahmat-Samii. «Low profile antenna study using the physical optics hybrid method (POHM)». En: *Antennas and Propagation Society Symposium 1991 Digest* (1991), 1350-1353 vol.3. DOI: 10.1109/APS.1991.175100.
- [41] R.E. Hodges e Y. Rahmat-Samii. «An iterative current-based hybrid method for complex structures». En: *IEEE Transactions on Antennas and Propagation* 45.2 (1997), págs. 265-276. DOI: 10.1109/8.560345.
- [42] J. M. Taboada Varela. *Estudio de Compatibilidad Electromagnética e Interferencia en Sistemas Radiantes a Bordo de Estructuras Complejas*. Universidad de Vigo, 2001.
- [43] H. A. Schwarz. *Gesammelte mathematische Abhandlungen*. Berlin: J. Springer, 1890.
- [44] A. Toselli y O. Widlund. *Domain Decomposition Methods: Algorithms and Theory*. Berlin: J. Springer, 2005.
- [45] C.T. Wolfe, U. Navsariwala y S.D. Gedney. «A parallel finite-element tearing and interconnecting algorithm for solution of the vector wave equation with PML absorbing medium». En: *IEEE Transactions on Antennas and Propagation* 48.2 (2000), págs. 278-284. DOI: 10.1109/8.833077.
- [46] B. Stupfel y M. Mognot. «A domain decomposition method for the vector wave equation». En: *IEEE Transactions on Antennas and Propagation* 48.5 (2000), págs. 653-660. DOI: 10.1109/8.855483.
- [47] Kezhong Zhao et al. «A Domain Decomposition Method With Nonconformal Meshes for Finite Periodic and Semi-Periodic Structures». En: *IEEE Transactions on Antennas and Propagation* 55.9 (2007), págs. 2559-2570. DOI: 10.1109/TAP.2007.904107.
- [48] Yu-Jia Li y Jian-Ming Jin. «A New Dual-Primal Domain Decomposition Approach for Finite Element Simulation of 3-D Large-Scale Electromagnetic Problems». En: *IEEE Transactions on Antennas and Propagation* 55.10 (2007), págs. 2803-2810. DOI: 10.1109/TAP.2007.905954.
- [49] Mao-Kun Li y Weng Cho Chew. «Multiscale Simulation of Complex Structures Using Equivalence Principle Algorithm With High-Order Field Point Sampling Scheme». En: *IEEE Transactions on Antennas and Propagation* 56.8 (2008), págs. 2389-2397. DOI: 10.1109/TAP.2008.926785.
- [50] A. Greenwood B. MacKie-Mason y Z. Peng. «Adaptive and Parallel Surface Integral Equation Solvers for Very Large-Scale Electromagnetic Modeling and Simulation (Invited Paper)». En: *Progress In Electromagnetics Research* 154 (2015), págs. 143-162. DOI: doi:10.2528/PIER15113001.
- [51] J. L. Rodríguez J. M. Taboada F. Obelleiro y J. O. Rubiños. «Modeling the Physical Optics Currents in a Hybrid Moment-Method-Physical-Optics Code». En: *IEICE Transactions on Electronics* E84-C.9 (2001), págs. 1207-1214. DOI: 0916-8516.
- [52] F. Obelleiro J. M. Taboada y J. L. Rodríguez. «Improvement of the hybrid moment method-physical optics method through a novel evaluation of the physical optics operator». En: *Microwave and Optical Technology Letters* 30.5 (2001), págs. 357-363. DOI: <https://doi.org/10.1002/mop.1314>.
- [53] F. Obelleiro et al. «HEMCUVI: a software package for the electromagnetic analysis and design of radiating systems on board real platforms». En: *IEEE Antennas and Propagation Magazine* 44.5 (2002), págs. 44-61. DOI: 10.1109/MAP.2002.1077776.
- [54] D. M. Solís, F. Obelleiro y J. M. Taboada. «Surface Integral Equation-Domain Decomposition Scheme for Solving Multiscale Nanoparticle Assemblies With Repetitions». En: *IEEE Photonics Journal* 8.5 (2016), págs. 1-14. DOI: 10.1109/JPHOT.2016.2614895.
- [55] J. M. Bértolo Seara. *Desarrollo de métodos avanzados de supercomputación en electromagnetismo*. Universidad de Vigo, 2011.
- [56] V. F. Martín Martínez. *Parallel multisolver methods for very long-scale electromagnetic modelling*. Universidad de Extremadura, 2022.

- [57] P. Ylä-Oijala, M. Taskinen y S. Järvenpää. «Surface integral equation formulations for solving electromagnetic scattering problems with iterative methods». En: *Radio Science* 40.06 (2005), págs. 1-19. DOI: 10.1029/2004RS003169.
- [58] S. Rao, D. Wilton y A. Glisson. «Electromagnetic scattering by surfaces of arbitrary shape». En: *IEEE Transactions on Antennas and Propagation* 30.3 (1982), págs. 409-418. DOI: 10.1109/TAP.1982.1142818.
- [59] V. Rokhlin. «Rapid solution of integral equations of classical potential theory». En: *Journal of Computational Physics* 60.2 (1985), págs. 187-207. DOI: [https://doi.org/10.1016/0021-9991\(85\)90002-6](https://doi.org/10.1016/0021-9991(85)90002-6).
- [60] J. M. Song y W. C. Chew. «Fast multipole method solution using parametric geometry». En: *Microwave and Optical Technology Letters* 7.16 (1994), págs. 760-765. DOI: <https://doi.org/10.1002/mop.4650071612>.
- [61] Nail Gumerov, Ramani Duraiswami y Eugene Borovikov. «Data Structures, Optimal Choice of Parameters, and Complexity Results for Generalized Multilevel Fast Multipole Methods in d Dimensions». En: (sep. de 2013).
- [62] Y. Saad y M. H. Schultz. «GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems.» En: *SIAM J. Sci. Stat. Comput.* 7.3 (1986), págs. 856-869.
- [63] T. F. Chan et al. «A Quasi-Minimal Residual Variant of the Bi-CGSTAB Algorithm for Nonsymmetric Systems». En: *SIAM Journal on Scientific Computing* 15.2 (1994), págs. 338-347.
- [64] Mario A Echeverri Bautista et al. «A Nonconformal Domain Decomposition Scheme for the Analysis of Multiscale Structures». En: *IEEE Transactions on Antennas and Propagation* 63.8 (2015), págs. 3548-3560. DOI: 10.1109/TAP.2015.2430873.
- [65] Diego M. Solís et al. «Accurate EMC Engineering on Realistic Platforms Using an Integral Equation Domain Decomposition Approach». En: *IEEE Transactions on Antennas and Propagation* 68.4 (2020), págs. 3002-3015. DOI: 10.1109/TAP.2019.2950862.
- [66] F.J.S. Moreira y A. Prata. «A self-checking predictor-corrector algorithm for efficient evaluation of reflector antenna radiation integrals». En: *IEEE Transactions on Antennas and Propagation* 42.2 (1994), págs. 246-254. DOI: 10.1109/8.277219.
- [67] R. L. Eigel. *Bistatic Radar Cross Section (RCS) Characterization of Complex Objects*. Air Force Institute of Technology, 1999.
- [68] W.A. Imbriale y R.E. Hodges. *Linear-Phase Approximation in the Triangular Facet Near-Field Physical Optics Computer Program*. Telecommunications y Data Acquisition Progress Report, 1990.
- [69] W. Gordon. «Far-field approximations to the Kirchoff-Helmholtz representations of scattered fields». En: *IEEE Transactions on Antennas and Propagation* 23.4 (1975), págs. 590-592. DOI: 10.1109/TAP.1975.1141105.
- [70] S. G. Parker et al. «OptiX: A General Purpose Ray Tracing Engine». En: *ACM Transactions on Graphics* (ago. de 2010).
- [71] https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf.
- [72] Turner Whitted. «An Improved Illumination Model for Shaded Display». En: *Commun. ACM* 23.6 (jun. de 1980), págs. 343-349. ISSN: 0001-0782. DOI: 10.1145/358876.358882. URL: <https://doi.org/10.1145/358876.358882>.
- [73] Arthur Appel. «Some Techniques for Shading Machine Renderings of Solids». En: AFIPS '68 (Spring). New York, NY, USA: Association for Computing Machinery, 1968, págs. 37-45. ISBN: 9781450378970. DOI: 10.1145/1468075.1468082. URL: <https://doi.org/10.1145/1468075.1468082>.
- [74] Andreas Kern Manuel Schiller y Alois Knoll. «Real-time Electromagnetic Wave Propagation using OptiX for Simulation of Car-to-Car-Communication». En: *NVIDIA GTC* (2014).

- [75] Robert Felbecker et al. «Electromagnetic wave propagation in the millimeter wave band using the NVIDIA OptiX GPU ray tracing engine». En: mar. de 2012, págs. 488-492. ISBN: 978-1-4577-0918-0. DOI: 10.1109/EuCAP.2012.6206198.
- [76] Zhengqing Yun y Magdy F. Iskander. «Ray Tracing for Radio Propagation Modeling: Principles and Applications». En: *IEEE Access* 3 (2015), págs. 1089-1100. DOI: 10.1109/ACCESS.2015.2453991.
- [77] John E. Stone, Kirby L. Vandivort y Klaus Schulten. «GPU-Accelerated Molecular Visualization on Petascale Supercomputing Platforms». En: *Proceedings of the 8th International Workshop on Ultrascale Visualization*. UltraVis '13. New York, NY, USA: Association for Computing Machinery, 2013. ISBN: 9781450325004. DOI: 10.1145/2535571.2535595. URL: <https://doi.org/10.1145/2535571.2535595>.
- [78] Blyth Simon C. «Opticks : GPU Optical Photon Simulation for Particle Physics using NVIDIA® OptiX™». En: *Journal of Physics: Conference Series* 898 (oct. de 2017), pág. 042001. DOI: 10.1088/1742-6596/898/4/042001. URL: <https://doi.org/10.1088/1742-6596/898/4/042001>.
- [79] Andrés Navarro Cadavid, Dinael Guevara Ibarra y Sebastian Londoño Salcedo. «Using 3-D Video Game Technology in Channel Modeling». En: *IEEE Access* 2 (2014), págs. 1652-1659. DOI: 10.1109/ACCESS.2014.2370758.
- [80] <https://developer.nvidia.com/optix>.
- [81] Eric Haines y Tomas Akenine-Moller. *Ray Tracing Gems: High-Quality and Real-Time Rendering with DXR and Other APIs*. USA: Apress, 2019. ISBN: 1484244265.
- [82] A. Marrs, P. Shirley e I. Wald. *Ray Tracing Gems II: Next Generation Real-Time Rendering with DXR, Vulkan, and OptiX*. Apress, 2021. ISBN: 9781484271872. URL: <https://books.google.es/books?id=pLJ3zgEACAAJ>.
- [83] John F. Shaeffer, Brett Cooper y Kam W. Hom. «Analytical Bistatic k Space Images Compared to Experimental Swept Frequency EAR Images,2004.