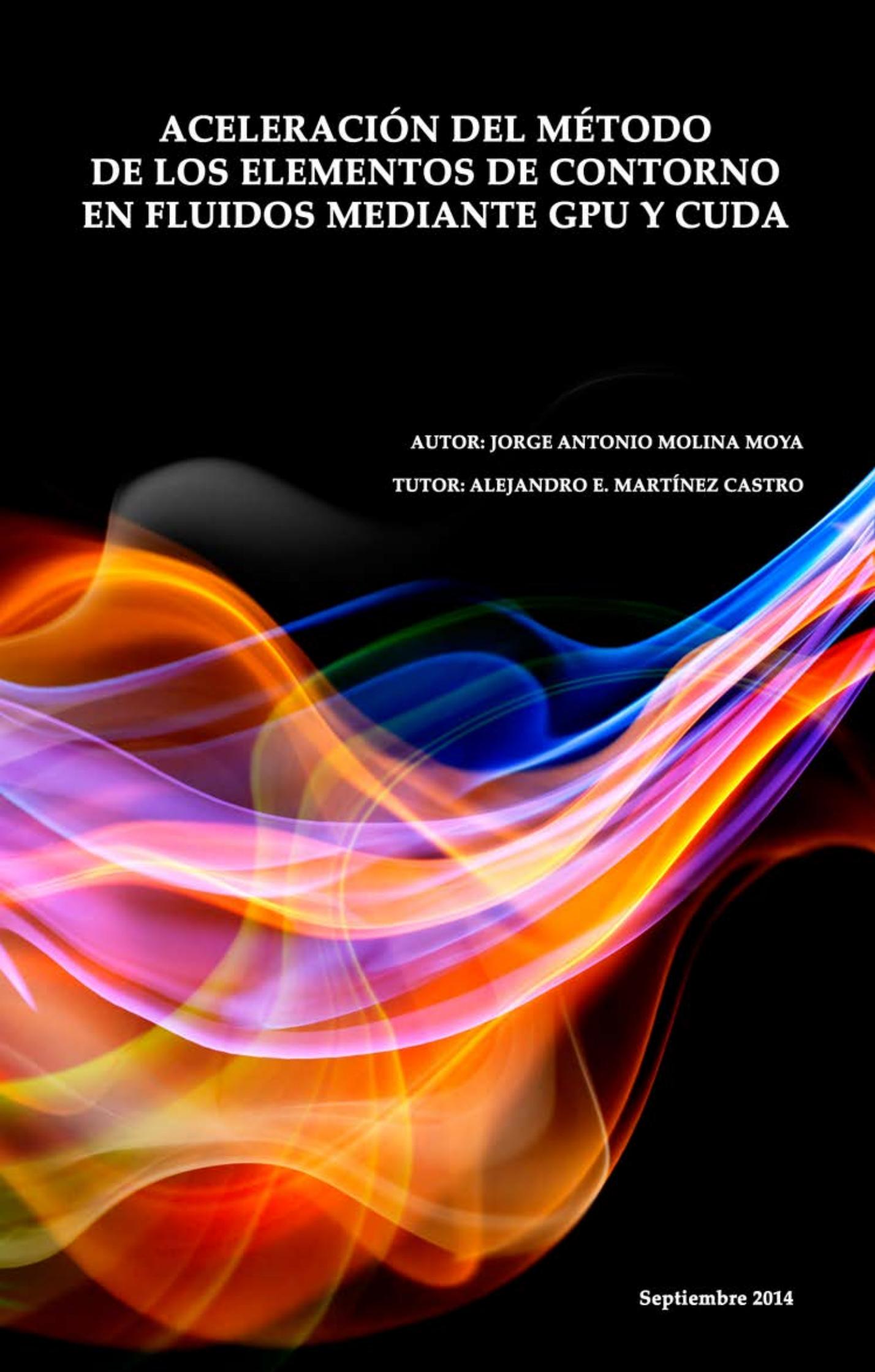


ACELERACIÓN DEL MÉTODO DE LOS ELEMENTOS DE CONTORNO EN FLUIDOS MEDIANTE GPU Y CUDA

AUTOR: JORGE ANTONIO MOLINA MOYA

TUTOR: ALEJANDRO E. MARTÍNEZ CASTRO

Septiembre 2014

The background of the cover features a dynamic, abstract visualization of fluid flow. It consists of multiple overlapping, semi-transparent layers of color, including shades of orange, yellow, purple, blue, and red. These layers are curved and layered, creating a sense of depth and movement, similar to a 3D rendering of a complex flow field or a simulation of fluid dynamics. The colors are vibrant and contrast sharply against the black background.

ACELERACIÓN DEL MÉTODO DE LOS ELEMENTOS DE CONTORNO EN FLUIDOS MEDIANTE GPU Y CUDA

PROYECTO FIN DE CARRERA

Por

Jorge Antonio Molina Moya
Ingeniería de Caminos, Canales y Puertos

Tutor

Alejandro E. Martínez Castro

Departamento de Mecánica de Estructuras ,
Universidad de Granada
Edificio Politécnico Fuentenueva, C/ Severo Ochoa s/n, CP 18002
Granada (España)

Septiembre de 2014

© Copyright 2014
por
Jorge Antonio Molina Moya

Resumen

El Método de los Elementos de Contorno (Boundary Element Method), también conocido como 'Panel Method', supone hoy en día una alternativa al conocido Método de los Elementos Finitos. Uno de los campos donde este método cobra bastante importancia es el campo de la Mecánica de Fluidos, más concretamente, el estudio del oleaje y su interacción con estructuras offshore. Sin embargo, el cuello de botella de dicha metodología radica en la gran cantidad de grados de libertad que tiene presentes; lo cual da como resultado tiempos de computación y resolución demasiado elevados ya que los sistemas de ecuaciones resultantes de dicho método tienen una matriz de coeficientes llena. Muchos han sido los investigadores que han tratado de resolver este problema, logrando un gran éxito con el llamado 'Fast Multipole Method'. En el presente documento, sin embargo, se propone una alternativa a éste, que consiste en implementar la programación en tarjetas gráficas o 'GPU' gracias a CUDA-C.

Dicho esto, se requiere la búsqueda de un método de resolución de sistemas de ecuaciones lineales más rápido que el método tradicional de Gauss que es inviable para grandes sistemas. El método elegido para este proyecto es el 'Generalized Minimum Residual Method' o GMRES, cuyo 'punto débil' radica en la ingente cantidad de multiplicaciones matriz-vector que se deben implementar; por ello, será en esta parte del código donde se implementará la programación paralela mediante GPU y CUDA-C.

Una vez logrado este primer objetivo, se han concebido diversas herramientas informáticas para ayudar a conocer mejor los esfuerzos dinámicos a los que están sometidas estructuras offshore como consecuencia de su interacción con el oleaje. Para ello se ha estudiado el cálculo y obtención de la matriz de Masa Añadida y la Matriz de Amortiguamiento Hidrodinámico, las cuales se pueden introducir en un programa de cálculo de estructuras y obtener así resultados más realistas y menos conservadores.

Otra herramienta informática desarrollada resuelve la Ecuación de Mild-Slope para Fondo Constante y Variable. Gracias a esta herramienta se puede realizar un estudio de agitación de un recinto situado en la costa así como un estudio de propagación de ondas oceánicas.

En el presente proyecto se podrá encontrar un Capítulo 0 en el cual se expone los objetivos buscados y la metodología empleada. El resto del documento está dividido en dos partes. Una primera parte en la que se resume toda la base teórica y científica en la que se ha apoyado el autor; aquí se explica el Método de los Elementos de Contorno, Teoría Lineal de Ondas, Teoría de Flotabilidad, GMRES y CUDA. En la segunda parte constan todas las pruebas numéricas de validación realizadas con los algoritmos desarrollados. Estos algoritmos han sido 4: dos para resolver regiones gobernadas por la ecuación $\nabla^2\Phi = 0$ en 2D y en 3D respectivamente; y otros dos para resolver regiones gobernadas por la ecuación de Mild-Slope en fondo constante y en fondo variable respectivamente.

Agradecimientos

Han sido seis años de una etapa que ahora se cierra, en la que me he sentido rodeado de personas que han estado a mi lado siempre que lo he necesitado.

En primer lugar, agradecer a mis padres y a mi hermana todo su apoyo, su confianza, que se alegren y disfruten de mis éxitos y mis logros, y el hecho de saber que siempre estarán ahí para seguir acompañándome en el resto de etapas. También agradecer a mis padres, que no tuvieron ocasión de estudiar, haberme brindado a mí esa oportunidad y haber podido hacer esta ingeniería que era mi sueño. Ahora vienen momentos difíciles y de incertidumbre pero sé que podré contar con ellos.

Igualmente comparto esta alegría con mi amiga, mi compañera y mi amor, María. Empezamos esta carrera y hemos crecido juntos, nos hemos ayudado mutuamente y hemos conseguido superarlo. En muchas ocasiones has sido mi sustento, te debo muchas cosas y te quiero mucho.

Por otro lado, agradecer a Alejandro E. Martínez Castro, tutor del presente proyecto, su orientación, su compromiso y entusiasmo aportados durante este tiempo. Gracias a él, a sus nociones, el material que me ha facilitado y la constante disponibilidad en cualquier momento que le he necesitado, ha sido posible la existencia de este documento. No quiero dejar de agradecerle los buenos consejos que me ha dado en otros ámbitos distintos a este proyecto.

Por último, y no menos importante, agradezco el apoyo al resto de mi familia y amigos, a mis compañeros y profesores de la Universidad que siempre los llevaré conmigo, en especial a David Trujillo por su incondicional e inestimable compañerismo durante todo este tiempo; y a todas las personas que en estos años de duro esfuerzo han formado parte de mi vida.

Se cumple una etapa llena de momentos de agobio y desesperación que me han hecho aprender, madurar y crecer como persona; y llena de grandes momentos de felicidad y orgullo que han superado con creces a los anteriores.

Índice General

Resumen	III
Agradecimientos	V
Capítulo 0. Introducción, Objetivos, Metodología	1
0.1. Introducción	1
0.2. Objetivos	1
0.3. Metodología	2
I Base Teórica y Matemática	3
Capítulo 1. Revisión del Estado del Arte	5
Capítulo 2. Fundamentos del Movimiento Oscilatorio de la Superficie Marítima	7
2.1. Introducción a la Ingeniería Marítima	7
2.1.1. Introducción	7
2.1.2. Tipos de Ondas Marinas	7
2.1.3. El Oleaje y sus Características Principales	9
2.2. Teoría Lineal de Ondas	9
2.2.1. Definición del Problema	9
2.2.2. Fluido Incompresible y No Viscoso	11
2.2.3. Condiciones de Contorno	12
2.2.4. Resolución del Problema	15
2.2.5. Solución General	18
Capítulo 3. Ecuaciones Linealizadas del Movimiento de un Cuerpo Flotante	21
3.1. Condición Cinemática	21
3.2. Conservación del Momento Lineal	23
3.3. Conservación del Momento Angular	25
3.4. Ecuaciones Dinámicas para un Cuerpo Flotante (Forma Matricial)	27
3.5. Descomposición en un Problema de Difracción y Problemas de Radiación	29
3.6. Masa añadida y Amortiguamiento por Radiación	30
Capítulo 4. Método de los Elementos de Contorno	33
4.1. Introducción	33
4.2. Diferencias entre FEM y BEM	33
4.3. Conceptos Básicos	34
4.4. Ecuación Integral de Contorno para la Ecuación de Laplace	37
4.5. Método de los Elementos de Contorno	41

4.6. Evaluación de Singularidades	45
4.7. Elementos Lineales	47
4.8. Conclusiones	51
Capítulo 5. Ecuación de Mild-Slope y el Método de los Elementos de Contorno de Doble Reciprocidad	53
5.1. Ecuación de Mild-Slope	53
5.2. Método de los Elementos de Contorno de Doble Reciprocidad	55
5.2.1. Ecuación de Helmholtz	55
5.2.2. Ecuación Completa	55
5.2.3. Elección de la Función de Interpolación	56
5.2.4. Evaluación de Singularidades	57
5.3. Aplicación del Método de los Elementos de Contorno de Doble Reciprocidad a la Ecuación de Mild-Slope	58
5.3.1. Resolución de la Ecuación de MildSlope con Fondo Variable	58
5.3.2. Discretización Numérica	61
5.4. Ecuación de Mild-Slope para Fondo Constante	63
Capítulo 6. Generalized Minimum Residual Method	67
6.1. Introducción	67
6.2. Subespacios de Krylov	67
6.3. Técnicas de Proyección	68
6.4. Método de Arnoldi	69
6.5. Método de Mínimo Residuo Generalizado (GMRES)	71
6.6. GMRES Implementado y sus particularidades	73
Capítulo 7. CUDA y la Programación Paralela	79
7.1. Introducción	79
7.2. ¿Qué es una GPU?	80
7.3. CUDA y sus Principales Elementos	81
7.4. Implementación de CUDA en el Software	85
7.5. Conclusiones	86
II Programas Desarrollados y Pruebas Numéricas de Validación	91
Capítulo 8. Preproceso y Postproceso	93
8.1. Introducción	93
8.2. Módulo Geometry	93
8.3. Módulo MESH	93
8.4. Exportación de la Malla	96
8.5. Módulo ParaView	97
8.6. Exportación de Resultados para el PostProceso	99
Capítulo 9. Test de Verificación para BEM2D	101
9.1. Descripción	101
9.2. Prueba de Validación	102
9.3. Conclusiones	104

Capítulo 10. Test de Verificación para BEM3D	105
10.1. Descripción	105
10.2. Prueba de Validación	107
10.3. Conclusiones	109
Capítulo 11. Test de Verificación para Mild-Slope con Fondo Constante	117
11.1. Descripción	117
11.2. Prueba de Validación	118
11.3. Conclusiones	119
Capítulo 12. Test de Verificación para Mild-Slope con Fondo Variable	125
12.1. Descripción	125
12.2. Prueba de Validación 1 - Isla Homma	125
12.3. Prueba de Validación 2 - Experimento de Kyung Doug Suh	131
12.4. Conclusiones	135
Capítulo 13. Conclusiones, Aportaciones y Desarrollo Futuro	137
13.1. Conclusiones	137
13.2. Aportaciones	137
13.3. Desarrollo Futuro	137
Bibliografía	139

Lista de Tablas

2.1. Tipos de Ondas Marítimas	8
9.1. Coordenadas de los Puntos de Integración y Funciones Peso para el caso bidimensional	102
9.2. Precisión de los resultados	103
10.1. Coordenadas de los Puntos de Integración y Funciones Peso para el caso tridimensional	107

Lista de Figuras

2.1. Principales parámetros de una onda marítima	8
2.2. Definición del Problema de Estudio	10
2.3. Definición del Volumen de Control	10
2.4. Esquema de una Onda General	19
3.1. Esquema del Cuerpo Flotante	21
3.2. Matrices $[M]$ y $[K]$. Fuente [1]	28
3.3. Descomposición en un Problema de Difracción y Problemas de Radiación	29
4.1. Diferencias entre el Método de Elementos Finitos y el Método de Elementos de Contorno. Fuente[2]	34
4.2. Definición del sentido del vector normal y de la frontera	35
4.3. Deformación de la frontera en el caso bidimensional	39
4.4. Deformación de la frontera en el caso tridimensional	39
4.5. Cambio de coordenadas en el caso bidimensional	40
4.6. División del Contorno en Elementos y Tipos de Elementos	42
4.7. Esquema para la evaluación de singularidades en 2D	46
4.8. Evaluación de Singularidades con elementos triangulares constantes.	47
4.9. Esquema para la resolución de la integral obtenida en el estudio de la singularidad	48
4.10. Esquema de los Elementos Lineales	48
4.11. Ángulo entre Elementos Lineales	49
4.12. Funciones de Interpolación	49
5.1. Esquema para la evaluación de singularidades	58
6.1. Comparativa entre el Método de Eliminación de Gauss y el GMRES	77
7.1. Procesamiento en Serie vs Procesamiento en Paralelo	79
7.2. Treads, Blocks, Grid y Kernel (Fuente: www.nvidia.com)	82
7.3. Comparativa entre un programa con CUDA implementado y un Programa Secuencial al ejecutar una Multiplicación Matriz-Vector	88
7.4. Comparativa entre el método GMRES normal y el método GMRES optimizado con CUDA	89
8.1. Captura de pantalla del inicio del Módulo Geometry	94
8.2. Captura de pantalla de figuras hechas con el Módulo Geometry	94
8.3. Captura de pantalla del inicio del Módulo MESH	95
8.4. Captura de pantalla de la malla creada a partir de un toroide	95
8.5. Captura de pantalla del inicio de ParaView	98
8.6. Captura de pantalla de resultados obtenidos con ParaView	98
9.1. Captura de pantalla de los resultados obtenidos, se muestra el valor del potencial en todo el dominio.	103
9.2. Resultados del potencial en la línea $y = 100$	104
10.1. Cambio de Coordenadas para la integración.	106

10.2. Geometría del problema del validación	107
10.3. Malla para la prueba de validación de BEM3D.	110
10.4. Resultados del Movimiento Surge	111
10.5. Resultados del Movimiento Heave	112
10.6. Resultados del Movimiento Pitch	113
10.7. Superficie del Mar debido al desplazamiento modal Surge, $k=0.1$ y Amplitud = 1 metro (Unidades en metros)	114
10.8. Superficie del Mar debido al desplazamiento modal Heave, $k=0.1$ y Amplitud = 1 me- tro (Unidades en metros)	114
10.9. Superficie del Mar debido al desplazamiento modal Pitch, $k=0.1$ y Amplitud = 1 radian (Unidades en metros)	115
11.1. Geometría del problema del validación	119
11.2. Resultados para el caso con todos los bordes absorbentes	120
11.3. Solución para bordes totalmente absorbentes obtenida en [3]	121
11.4. Resultados para el caso con todos los bordes reflexivos	122
11.5. Solución para bordes totalmente reflexivos obtenida en [3]	123
11.6. Resultados para Alta Frecuencia ($T=2\text{seg}$)	124
12.1. Geometría de la Isla Homma($a = 10000m$, $b = 30000m$, $h_0 = 4000m$ y $h_i = 444m$)	126
12.2. Definición de los nodos utilizados para la Isla Homma	127
12.3. Resultados de la prueba de la Isla Homma obtenidos en [4].	128
12.4. Resultados de la prueba de la Isla Homma con elementos constantes.	129
12.5. Comparación de los resultados obtenidos por el autor y los obtenidos por Huan-Wen Liu para $T=410$ segundos.	130
12.6. Resultados del experimento de Kyung Doug Suh para $k \cdot h = 3$. Puntos = Resultados Experimentales; Línea Continua = Resultados con el Modelo MMSE; Línea Disconti- nua = Resultados con el Modelo MSE	132
12.7. Resultados obtenidos de la prueba con el Experimento de Kyung Doug Suh.	133
12.8. Comparación de los resultados obtenidos por el autor y los resultados experimentales obtenidos por Kyung Doug Suh	134

Lista de Algoritmos

1.	Algoritmo de Arnoldi	70
2.	Algoritmo de Arnoldi Modificado Gram-Schmidt	71
3.	Algoritmo GMRES	72
4.	Algoritmo Restarted GMRES	72
5.	Restarted GMRES implementado	73

CAPÍTULO 0

Introducción, Objetivos, Metodología

0.1. Introducción

Uno de los métodos más utilizados para resolver problemas en los que intervengan ondas oceánicas y la interacción de éstas con estructuras es el famoso Método de los Elementos de Contorno. El gran reto de las últimas décadas ha sido la reducción de tiempo de computación que conlleva dicho método. Hasta ahora, la metodología que mejores resultados ha obtenido ha sido el conocido Método Multipolo o 'Fast Multipole Method' introducido por Rokhlin y Greengard. Dicho método, a grandes rasgos, consiste en la expansión multipolar de la función de Green utilizada; de este modo las fuentes que están cercanas unas de otras se tratan como si fuese una sola.

En este proyecto se ofrece una vía alternativa en la que se exploran las posibilidades que ofrece la programación paralela en tarjetas gráficas o GPU mediante CUDA. Para el uso de dicha herramienta es preferible la programación en lenguaje C ya que Fortran-Cuda es propietario y no funciona de la manera adecuada en ciertas ocasiones.

En pos del uso de software de libre licencia, para las fases de preproceso (generación de geometría y malla) y postproceso (visualización de resultados) se va a utilizar el software SALOME. Las características de dicho software serán explicadas en el Capítulo 8.

En cuanto a los problemas que se pueden resolver con el Método de los Elementos de Contorno cabe destacar la necesidad de resolver problemas de flotabilidad. De dichos problemas se obtienen importantes parámetros que se deben usar para un correcto análisis dinámico de estructuras offshore. Por otro lado, es importante también la resolución de problemas de difracción de ondas marítimas, ya sea debido a obstáculos (diques, muelles o cualquier tipo de estructuras) o a un fondo marino variable. Estos problemas serán resueltos con las herramientas desarrolladas previamente.

0.2. Objetivos

Los objetivos que se han logrado alcanzar mediante la realización de este Proyecto se resumen en los siguientes puntos:

- Generación de malla a partir de software libre, generando la entrada y salida aprovechando herramientas relativamente recientes y de licencia libre como son Salomé y ParaView.
- Implementación de códigos en CUDA-C basados en C++, de forma tal que supongan una base para, posteriormente en un futuro, crecer mediante recursos propios de C++.
- Implementación del método BEM basado en elementos constantes para el problema de potencial de ondas 3D.
- Implementación del método BEM basado en elementos constantes para la ecuación de la Mild-Slope.

0.3. Metodología

La metodología seguida a lo largo de este proyecto, para completar satisfactoriamente el presente documento, se resume en los siguientes pasos:

- Revisión del estado del arte.
- Estudio y Desarrollo de códigos propios en C++.
- Estudio y desarrollo de códigos propios en CUDA basados en CUDA-6.
- Estudio y desarrollo de formulaciones BEM para problemas de dinámica de fluidos 3D (tanto formulación 3D como formulación relativa a la ecuación de Mild-Slope).
- Desarrollo de código para la lectura de archivos 'UNV' que contienen los datos de mallas.
- Desarrollo de ficheros 'VTK' para exportar y visualizar resultados.
- Desarrollo e implementación del algoritmo GMRES para aceleración de solver.
- Implementación de test numéricos y discusión de resultados.

Parte I

Base Teórica y Matemática

CAPÍTULO 1

Revisión del Estado del Arte

Prácticamente todos los fenómenos físicos que ocurren en la naturaleza pueden ser descritos por ecuaciones diferenciales y condiciones de contorno. Sin embargo, solamente se conocen soluciones para geometrías sencillas y para problemas en los que se asumen importantes simplificaciones. Por ello, para resolver problemas de la vida real se debe recurrir a soluciones aproximadas. Para ello hay dos técnicas posibles que se pueden aplicar. Éstas consisten en, o bien encontrar una solución que cumpla perfectamente las condiciones de contorno y minimice el error cometido en satisfacer la ecuación diferencial; o bien buscar una solución que verifique completamente la ecuación diferencial y que minimice el error cometido al satisfacer las condiciones de contorno.

La primera técnica fue propuesta por Ritz[5] y supuso el primer paso para que Richard Courant desarrollase el conocido Método de los Elementos Finitos o FEM (acrónimo de Finite Elements Method en inglés) en 1943.

La segunda técnica, propuesta por Trefftz, es la utilizada por el Método de los Elementos de Contorno o BEM (acrónimo Boundary Elements Method en inglés), siendo la que se ha utilizado en el presente documento.

Muchos son los autores que han tratado los elementos de contorno, entre ellos cabe destacar C.A. Brebbia, cuyo libro [6], sirve como perfecto manual introductorio a este método. También se debe mencionar a Stephen Kirkup y su trabajo en acústica [7] y [8], en los que se apoya en los elementos de contorno para resolver la Ecuación de Helmholtz. Otros autores como Gernot Beer et al.[9], dan una gran cantidad de herramientas con el objetivo de ayudar al lector a crear algoritmos para resolver problemas que incluyen campos potenciales con el apoyo de los ya mencionados elementos de contorno. Finalmente, se debe mencionar a Alok Sutradhar et al.[10]; éstos exponen un método especial con el que, aprovechando las propiedades simétricas de la función de Green, transforman el problema inicial en otro más rápidamente resoluble computacionalmente hablando.

Como se ha mencionado anteriormente, desde los últimos años se han buscado vías para acelerar el Método de los Elementos de Contorno. La más importante es el llamado 'Fast Multipole Method', por ejemplo, Yijun Liu en [11] y [12] expone dicho método y lo utiliza para problemas potenciales. Otro método de resolución rápida de sistemas de ecuaciones es el basado en matrices jerárquicas, por ejemplo I. Benedetti en [13] aplica este procedimiento para el Método de los Elementos de Contorno. Sin embargo, como se ha mencionado anteriormente, el método que finalmente se ha utilizado ha sido el 'GMRES' mejorado con CUDA-C. Sobre el método GMRES, se puede encontrar la descripción y obtención del mismo en [14] y [15]. Finalmente, en cuanto a CUDA, se puede consultar el manual oficial [16], por otro lado, ya ha habido autores que han utilizado CUDA en el Método de los Elementos de Contorno como es el ejemplo de Shoji Hamada [17] el cual usa CUDA y el 'Fast Multipole Method' para análisis de modelos vóxel¹. Otro ejemplo se puede encontrar con Toru Takahashi y

¹El vóxel (del inglés volumetric pixel) es la unidad cúbica que compone un objeto tridimensional. Constituye la unidad mínima procesable de una matriz tridimensional y es, por tanto, el equivalente del píxel en un objeto 3D.

Tsuyoshi Hamada [18], que han estudiado la ecuación de Helmholtz en 3D, acelerado con GPU y CUDA, para el estudio de la propagación de ondas acústicas.

En cuanto al estudio de ondas marítimas, flotabilidad e ingeniería oceánica en general, cabe destacar a Chiang C. Mei et al. y su libro [1]. Otros documentos relevantes son, por ejemplo, [19] de C.M. Linton y P. McIver que estudian la interacción oleaje-estructura y [20] de Turgut 'Sarp' Sarpkaya que profundiza en el estudio de las fuerzas dinámicas del oleaje en estructuras offshore. Más específicamente en este tema, son muchos los autores que han estudiado soluciones analíticas a problemas de flotabilidad como son el caso de [21], [22], [23], [24] y [25]. Todos ellos han estudiado las acciones hidrodinámicas así como la masa añadida y amortiguamiento hidrodinámico de un cilindro. La ventaja de este tipo de geometría es que se puede hacer un cambio a coordenadas polares, y, por ello, la ecuación $\nabla^2\phi = 0$ se transforma en una ecuación de Helmholtz. La solución general a este problema es una suma infinita de términos que incluyen la función de Hankel.

Un enfoque diferente se puede encontrar en el artículo de Simon Bogner y Ulrich Rüde [26]. En dicho artículo se aborda el problema utilizando el método de Lattice Boltzmann. Dicho método consiste, a grandes rasgos, en la modelización de un número limitado de partículas que, en este caso, forman el fluido mediante la resolución de la ecuación de Boltzmann. El estudio de los procesos de colisión e interacciones entre las partículas y el sólido permite obtener los movimientos y esfuerzos a los que está sometido de dicho sólido.

En el presente documento, se han resuelto las ecuaciones en el dominio de la frecuencia. Otro camino que se podría tomar es la resolución en el dominio del tiempo, como es el caso de los artículos de C.F. Liu et al. [27] y Michael H. Meylan [28] donde también se estudian las fuerzas inducidas por el oleaje en estructuras.

Adicionalmente, en este proyecto se ha estudiado la resolución de la ecuación de Mild-Slope con fondo variable mediante el Método de Elementos de Contorno de Doble Reciprocidad. P.W. Partridge et al. en [29] explicaron este método y estudiaron la precisión de los resultados obtenidos por varias funciones de interpolación². Otros autores que han estudiado la resolución de la ecuación de Mild-Slope de fondo variable son, por ejemplo, Song-Ping Zhu et al. [30], Huan-Wen Liu en [4], Sung-Shan Hsiao et al. en [31], P.G. Chamberlain y D. Porter en [32] y Huanwen LIU en [33]. Todos ellos han usado el método de los elementos de contorno de doble reciprocidad, aunque también se podría resolver con el método de los volúmenes finitos, como es el caso de Asu Inan y Lale Balas en [34].

²Ver el Capítulo 5

CAPÍTULO 2

Fundamentos del Movimiento Oscilatorio de la Superficie Marítima

2.1. Introducción a la Ingeniería Marítima

2.1.1. Introducción

Las ondas que se observan en cualquier cuerpo de agua en contacto con la atmósfera se deben a fuerzas que actúan sobre el fluido tratando de deformarlo y que se compensan con las fuerzas de la gravedad y las tensiones superficiales, que tratan de restaurar el equilibrio. La forma de estas ondas y su contenido energético dependen de la magnitud de la fuerza que las originó. El estudio de este tipo de oscilaciones es de gran importancia, ya que cualquier elemento dentro de una masa de agua o cerca de ella, puede verse sometido a su acción. La Ingeniería Marítima estudia, entre otros aspectos, el efecto que los distintos tipos de ondas existentes en el mar tienen sobre playas, obras de protección del litoral, puertos y estructuras off-shore[35].

2.1.2. Tipos de Ondas Marinas

En la tabla 2.1 se puede apreciar un resumen de las principales ondas marinas clasificadas según su periodo, magnitud y fuerzas generadoras y restauradoras.

Es conocido que, en un análisis espectral, las ondas de gravedad son las que contienen mayor cantidad de energía, por lo tanto son las ondas que interesan estudiar con mayor profundidad.

Por otro lado se puede hacer otra clasificación de las ondas según su movimiento:

- Ondas Progresivas: son ondas que se propagan en una dirección específica conservando la forma y se pueden describir matemáticamente según $\eta = A \cdot \cos(ax + bt)$ donde a y b son parámetros que se definirán a posteriori.
- Ondas Estacionarias: son ondas que se pueden formar cuando se encuentran dos ondas progresivas, de características idénticas y de sentido de avance contrario. Se describen matemáticamente según $\eta = 2A \cos(ax) \cos(bt)$. Esta onda no avanza con el tiempo, sino que, en cada posición x , la superficie varía entre $-2A \cos(ax)$ y $2A \cos(ax)$. Aquellos puntos donde $\cos(ax) = 0$ se llaman 'nodos' y son puntos en los que la elevación del mar es constantemente 0. Por otro lado, los puntos donde $\cos(ax) = 1$ se llaman 'antinodos' y son puntos en los que la elevación del mar es máxima.

Tipo de Onda	Periodo	Longitud	Altura	Fuerzas Generadoras	Fuerzas Restauradoras
Ondas Capilares	0-0.1 s.	2-7 cm	1-2 mm	Viento	Tensión Superficial
Ondas de Ultragravedad	0.1-1s	O(cm)	O(cm)	Viento	Tensión Superficial
Oleaje (Ondas de Gravedad)	3-30 s.	Metros a Cientos de metros	Desde cm hasta 15 m.	Viento	Gravedad
Ondas de Infragravedad	30s.-30min.	100-200 m.	Pequeña	Viento	Gravedad, Coriolis
Tsunamis (maremotos)	5-20 min.	O(km)	O(Decenas de m.)	Movimientos Tectónicos	Gravedad
Ondas de largo Periodo	5min.-24h.	Incluso escala planetaria	1-5m.	Atracción de cuerpos celestes	Gravedad, Coriolis
Transmarea	Más de 24h.	-	0-12m	Oscilaciones climáticas	Gravedad, Coriolis

Cuadro 2.1: Tipos de Ondas Marítimas

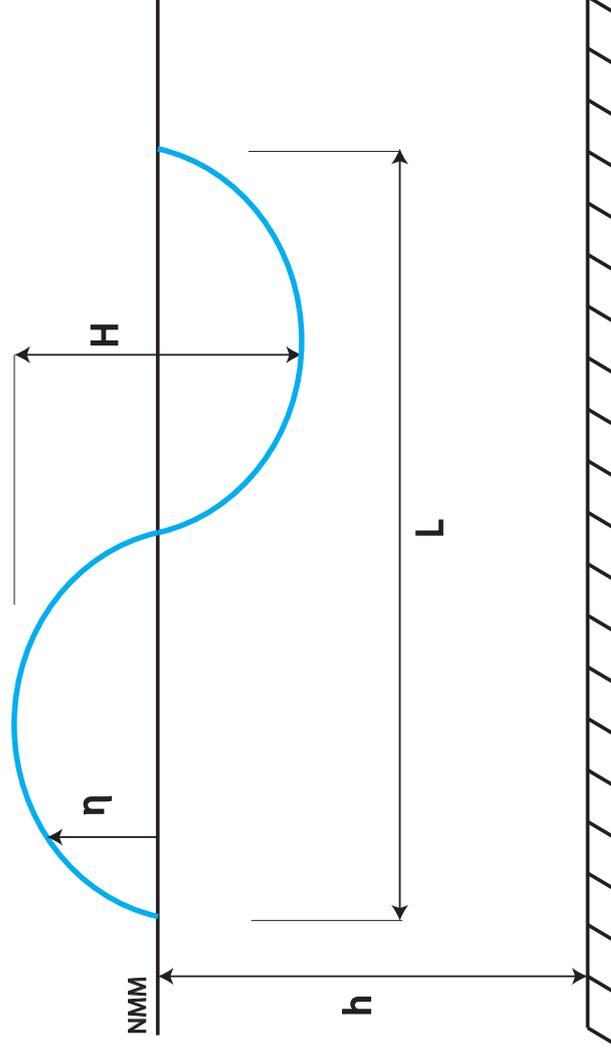


Figura 2.1: Principales parámetros de una onda marítima

2.1.3. El Oleaje y sus Características Principales

En el estudio de obras marítimas, es necesario un análisis profundo del oleaje. Las principales características de las ondas marítimas son (ver figura 2.1):

- Longitud de Ola(L): es la distancia entre cresta y cresta de la onda
- Altura de Ola (H): es la distancia entre la cresta y el seno de la onda
- Periodo de Ola (T): es el tiempo que transcurre desde que, por un mismo punto, pasan dos crestas consecutivas de una onda.
- Profundidad (h): es la distancia desde el nivel medio del mar (NMM) hasta el fondo oceánico.
- Elevación (η): es la elevación (positiva o negativa) de cada punto de la superficie marítima debido al paso de una ola.
- Peralte de ola (H/L): es la relación entre la altura de ola y su longitud y mide la pendiente del oleaje.

El oleaje nace en mar abierto como consecuencia de la acción del viento, y se propaga desde alta mar hasta la costa. En esta propagación, dichas ondas sufren transformaciones debidas a la constante acción del viento (las ondas crecen en altura) y a la influencia del fondo.

Cuando las ondas llegan a profundidades reducidas y empiezan a 'sentir el fondo', las olas se peraltan aumentando la altura de las crestas y disminuyendo la de los senos y comienzan a perder simetría de eje horizontal. Este fenómeno, denominado **asomeramiento**, se hace más acusado a medida que la onda se propaga hacia aguas de menor profundidad. Otro conocido efecto del fondo sobre el oleaje es la llamada **refracción**, que consiste en el cambio de dirección del oleaje en planta. Debido a este efecto, los frentes de ondas tienden a situarse paralelos a las líneas batimétricas.

Por último, los efectos más importantes que se tendrán en cuenta en este proyecto son la **reflexión** y la **transmisión**. Estos efectos son la devolución y absorción, respectivamente, de energía de la oscilación oceánica cuando ésta se encuentra con un obstáculo.

2.2. Teoría Lineal de Ondas

2.2.1. Definición del Problema

El problema a estudiar consiste en una porción de mar con un fondo del que se conoce su profundidad en cada punto y con un tren de ondas de oleaje entrante como se observa en la figura 2.2.

Para el estudio del problema se debe definir un volumen de control donde hacer cumplir la ecuación de gobierno y con una frontera donde imponer las condiciones de contorno.

El método del volumen de control consiste en trabajar con un volumen de control fijo que podrá ser diferencial o no. Con este método las leyes físicas pueden ser formuladas en términos de flujo de masa, energía o cantidad de movimiento a través de las paredes del volumen de control y de la tasa temporal de variación de masa, energía o cantidad de movimiento dentro del volumen de control. Así, las ecuaciones del movimiento pueden obtenerse a partir de la segunda ley de Newton que establece que la suma de las fuerzas externas que actúan sobre el volumen de control igualan la tasa temporal de variación de la cantidad de movimiento. Este método se usa generalmente en análisis en los que el interés primario se centra en conocer las variaciones de las características del flujo en la dirección del mismo [35].

En la figura 2.3 se puede ver un esquema con el volumen de control elegido.

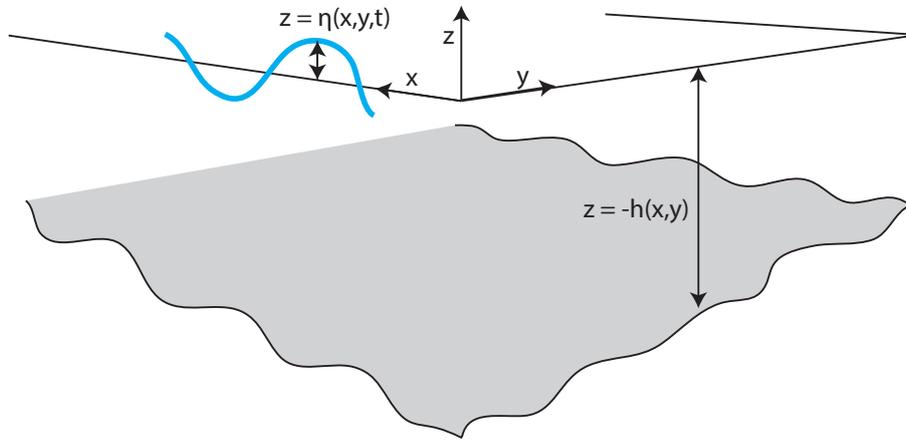


Figura 2.2: Definición del Problema de Estudio

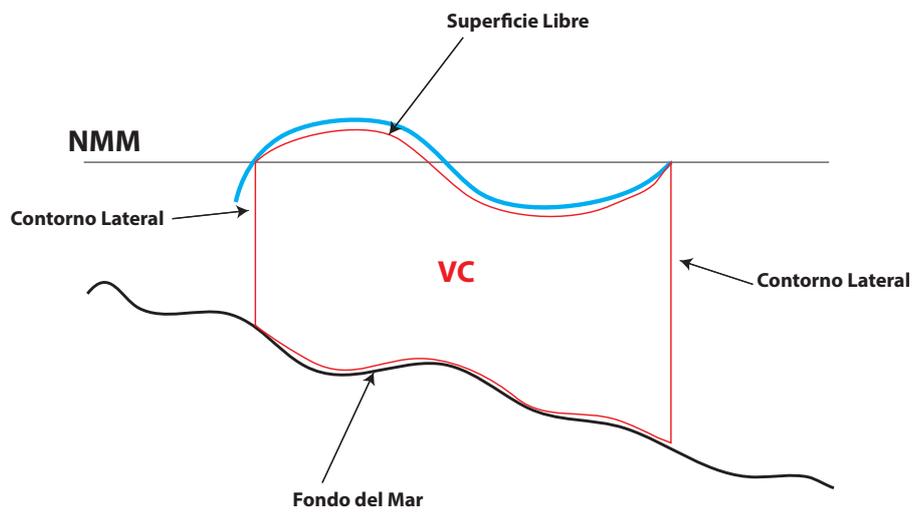


Figura 2.3: Definición del Volumen de Control

2.2.2. Fluido Incompresible y No Viscoso

En una gran cantidad de problemas que incluyen ondas de gravedad, la variación de la densidad del agua es insignificante en comparación con las escalas temporales y espaciales que son de interés en la ingeniería.

Las ecuaciones de conservación fundamentales se describen a continuación:

$$\text{Conservación de Masa: } \nabla \cdot \mathbf{u} = 0 \quad (2.1)$$

$$\text{Conservación del Momento: } \left(\frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla \right) \mathbf{u} = -\nabla \left(\frac{P}{\rho} + gz \right) + \nu \nabla^2 \mathbf{u} \quad (2.2)$$

donde $\mathbf{u}(x, t)$ es el vector de velocidad (u, v, w) , $P(x, t)$ la presión, ρ es la densidad, g es la aceleración de la gravedad y ν es la viscosidad cinemática del fluido.

En este sistema, el eje z está definido en el sentido vertical con sentido hacia arriba. Combinando las ecuaciones 2.1 y 2.2 se obtiene

$$\left(\frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla \right) \Omega = \Omega \cdot \nabla \mathbf{u} + \nu \nabla^2 \Omega \quad (2.3)$$

Siendo $\Omega = \nabla \times \mathbf{u}$, el vector vorticidad.

Para el agua $\nu \cong 10^{-6} m^2/s$, siendo despreciable para la tipología de problemas que se van a estudiar. La ecuación anterior resulta:

$$\left(\frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla \right) \Omega = \Omega \cdot \nabla \mathbf{u} \quad (2.4)$$

Se puede demostrar (ver a modo de ejemplo el Capítulo 1 de [1]) que $\Omega = 0$, es decir, que el fluido es irrotacional, por tanto, se puede expresar el campo de velocidades a través de una función escalar potencial de la forma:

$$\mathbf{u} = \nabla \Phi \quad (2.5)$$

Para cumplir la ecuación de conservación de la masa debe darse la condición de que:

$$\nabla^2 \Phi = 0 \quad (2.6)$$

que es la Ecuación de Gobierno del problema a estudiar.

Transformando la ecuación 2.2 según las consideraciones que se han mencionado, se obtiene:

$$\mathbf{u} \cdot \nabla \mathbf{u} = \nabla \frac{\mathbf{u}^2}{2} - \mathbf{u} \times (\nabla \times \mathbf{u}) \quad (2.7)$$

e introduciendo la irrotacionalidad del sistema:

$$\nabla \left[\frac{\partial \Phi}{\partial t} + \frac{1}{2} |\nabla \Phi|^2 \right] = -\nabla \left(\frac{P}{\rho} + gz \right) \quad (2.8)$$

que, integrando respecto las variables espaciales:

$$\frac{\partial \Phi}{\partial t} + \frac{1}{2} |\nabla \Phi|^2 + \frac{P}{\rho} + gz = 0 \quad (2.9)$$

Siendo ésta la conocida Ecuación de Bernoulli, que será usada en las siguientes apartados para obtener las condiciones de contorno de un problema de ondas.

2.2.3. Condiciones de Contorno

En este tipo de problemas se pueden plantear dos tipos de condiciones de contorno:

- Condiciones de Contorno Cinemáticas: son aquellas que especifican como es el movimiento en los contornos.
- Condiciones de Contorno Dinámicas: son las que especifican como es la distribución de presiones en los contornos.

Los contornos donde interesa aplicar dichas condiciones en un problema de ondas debidas a oleaje son:

- Fondo del mar
- Superficie Libre
- Contornos Laterales
- Cualquier superficie que interfiera en el progreso de dichas ondas

Un contorno es una superficie que se puede expresar de la siguiente forma:

$$F(x, t) = z - f(x, y, t) = 0 \quad (2.10)$$

Donde z es la altura y se ha definido $z = 0$ en la superficie del mar.

Las condiciones de contorno cinemáticas en una superficie se imponen con la derivada material de la superficie evaluada en la superficie, esto es:

$$\frac{DF}{Dt} = 0 \text{ en } F = 0 \quad (2.11)$$

Que es equivalente a:

$$\frac{\partial F}{\partial t} + \mathbf{u} \cdot \mathbf{n} \cdot |\nabla F| = 0 \text{ en } F = 0 \quad (2.12)$$

Siendo \mathbf{n} la normal a la superficie, es decir

$$\mathbf{n} = \frac{\nabla F}{|\nabla F|} \quad (2.13)$$

Por otro lado, las condiciones de contorno dinámicas se aplican imponiendo la Ecuación de Bernoulli 2.9 en dicha superficie.

Una vez hechas estas definiciones, se procede a imponer las condiciones de contorno en la superficie libre y en el fondo.

▪ Condición Cinemática en el Fondo del Mar

Para un fondo impermeable, la condición es que el flujo sea nulo a través de la misma. Ésto se expresa matemáticamente como:

$$F = z + h(x, y) \quad (2.14)$$

Aplicando la ecuación 2.12:

$$u \cdot \frac{\partial h}{\partial x} + v \cdot \frac{\partial h}{\partial y} + w = 0 \quad \text{en } z = -h(x, y) \quad (2.15)$$

En este caso se ha supuesto que el fondo permanece constante con el paso del tiempo. Esta hipótesis es válida para las escalas de tiempo que se manejan en este tipo de problemas.

Simplificando:

$$\frac{\partial \Phi}{\partial n} = 0 \quad \text{en } z = -h(x, y) \quad (2.16)$$

■ **Condición Cinemática en la Superficie Libre**

La condición de contorno cinemática, en este caso, exige que no haya flujo a través de la superficie aire-agua, es decir, que las partículas que constituyen la superficie libre, permanezcan en ella y no la abandonen. Todo esto se expresa matemáticamente de la siguiente manera:

$$F = z - \eta(x, y, t) \quad (2.17)$$

Aplicando la ecuación 2.12:

$$\frac{\partial \eta}{\partial t} + u \cdot \frac{\partial \eta}{\partial x} + v \cdot \frac{\partial \eta}{\partial y} - w = 0 \quad \text{en } z = \eta(x, y, t) \quad (2.18)$$

■ **Condición Dinámica en la Superficie Libre**

Al considerarse el fluido no viscoso, en cualquier contorno, las únicas fuerzas actuantes son las fuerzas de presión. La superficie libre del fluido responde a las variaciones en el campo de presiones en orden a mantener la presión uniforme en toda la superficie. Para las hipótesis adoptadas, esto es, fluido Newtoniano, incompresible, no viscoso e irrotacional, la ecuación que se debe aplicar, como se ha mencionado anteriormente, es la Ecuación de Bernoulli 2.9.

$$\frac{\partial \Phi}{\partial t} + \frac{1}{2} \left[\left(\frac{\partial \Phi}{\partial x} \right)^2 + \left(\frac{\partial \Phi}{\partial y} \right)^2 + \left(\frac{\partial \Phi}{\partial z} \right)^2 \right] + \frac{P}{\rho} + g \cdot \eta = 0 \quad \text{en } z = \eta(x, y, t) \quad (2.19)$$

■ **Condiciones de Contorno en los Laterales**

En los laterales no hay cambio de fase, por lo tanto no se puede aplicar una condición de contorno dinámica. Sin embargo, se deben aplicar condiciones de periodicidad espacial y temporal dado que se está tratando con un problema de ondas, esto es:

$$\Phi(x, y, z, t) = \Phi(x + L, y, z, t) \quad \text{Periodicidad Espacial} \quad (2.20)$$

$$\Phi(x, y, z, t) = \Phi(x, y, z, t + T) \quad \text{Periodicidad Temporal} \quad (2.21)$$

A estas condiciones se les puede aplicar la hipótesis de fondo horizontal. Dicha hipótesis no es real, pero, dado que las pendientes que se encuentran en mar abierto son pequeñas, el error cometido es despreciable. Al aplicar esta simplificación, el problema ya no depende de la variable y , y por ello, los términos $\frac{\partial}{\partial y} = 0$. Al final del capítulo se planteará la formulación para el caso que depende de las variables x e y .

El problema planteado hasta ahora tiene el inconveniente de que las condiciones de contorno en la superficie libre vienen dadas en términos de η , que es una incógnita. Por ello, para la resolución del problema, se debe introducir la simplificación de que η es pequeña, es decir, las ondas son de pequeña amplitud. De esta forma, haciendo un desarrollo en Series de Taylor centradas en $z = 0$, se puede aproximar las condiciones de contorno en $z = \eta$ por condiciones de contorno en $z = 0$.

El desarrollo en Series de Taylor tiene la siguiente formulación:

$$f(x_0 + \Delta x) = f(x_0) + \Delta x \cdot \left[\frac{\partial f(x)}{\partial x} \right]_{x=x_0} + \frac{1}{2}(\Delta x)^2 \cdot \left[\frac{\partial^2 f(x)}{\partial x^2} \right]_{x=x_0} + \dots + \frac{1}{n!}(\Delta x)^n \cdot \left[\frac{\partial^n f(x)}{\partial x^n} \right]_{x=x_0} \quad (2.22)$$

Aplicamos dicha formulación a las condiciones de contorno en la superficie libre:

■ **Condición de Contorno Cinemática en la Superficie Libre**

$$\left[\frac{\partial \eta}{\partial t} + u \cdot \frac{\partial \eta}{\partial x} - w \right]_{z=\eta} = \left[\frac{\partial \eta}{\partial t} + u \cdot \frac{\partial \eta}{\partial x} - w \right]_{z=0} + \eta \cdot \left[\frac{\partial}{\partial z} \left(\frac{\partial \eta}{\partial t} + u \cdot \frac{\partial \eta}{\partial x} - w \right) \right]_{z=0} + \dots \quad (2.23)$$

Despreciando los términos cuadrático y de orden superior (su valor es despreciable frente a los términos de primer orden) se obtiene:

$$\frac{\partial \eta}{\partial t} - w = 0 \quad \text{en } z = 0 \quad (2.24)$$

■ **Condición de Contorno Dinámica en la Superficie Libre**

$$\begin{aligned} \left[\frac{\partial \Phi}{\partial t} + \frac{1}{2} \left[\left(\frac{\partial \Phi}{\partial x} \right)^2 + \left(\frac{\partial \Phi}{\partial z} \right)^2 \right] + \frac{P}{\rho} + g \cdot z \right]_{z=\eta} &= \\ &= \left[\frac{\partial \Phi}{\partial t} + \frac{1}{2} \left[\left(\frac{\partial \Phi}{\partial x} \right)^2 + \left(\frac{\partial \Phi}{\partial z} \right)^2 \right] + \frac{P}{\rho} + g \cdot z \right]_{z=0} + \\ &+ \eta \cdot \left[\frac{\partial}{\partial z} \left(\frac{\partial \Phi}{\partial t} + \frac{1}{2} \left[\left(\frac{\partial \Phi}{\partial x} \right)^2 + \left(\frac{\partial \Phi}{\partial z} \right)^2 \right] + \frac{P}{\rho} + g \cdot z \right) \right]_{z=0} + \dots \end{aligned} \quad (2.25)$$

De nuevo, despreciando términos de orden superior y asumiendo que la presión en la superficie del mar es la presión atmosférica, es decir, $P = 0$, resulta:

$$\frac{\partial \Phi}{\partial t} + g\eta = 0 \quad \text{en } z = 0 \quad (2.26)$$

Ya se está en posición de formular el problema completo que se resolverá en el siguiente apartado:

■ **Ecuación de Gobierno**

$$\nabla^2 \Phi = \frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial z^2} = 0 \quad \text{en } 0 \leq x \leq L; \quad -h \leq z \leq 0 \quad (2.27)$$

■ **Condición de Contorno Cinemática en el Fondo**

$$w = 0 \quad \text{en } z = -h \quad (2.28)$$

■ **Condición de Contorno Cinemática en la Superficie Libre**

$$\frac{\partial \eta}{\partial t} - w = 0 \quad \text{en } z = 0 \quad (2.29)$$

- **Condición de Contorno Dinámica en la Superficie Libre**

$$\frac{\partial \Phi}{\partial t} + g\eta = 0 \quad \text{en } z = 0 \quad (2.30)$$

- **Condiciones de Contorno en los Laterales**

$$\Phi(x, y, z, t) = \Phi(x + L, y, z, t) \quad (2.31)$$

$$\Phi(x, y, z, t) = \Phi(x, y, z, t + T) \quad (2.32)$$

2.2.4. Resolución del Problema

Se procederá a resolver el problema por el conocido Método de Separación de Variables.

Se considera que la función potencial $\Phi(x, z, t)$ se puede descomponer en un producto de tres funciones, esto es:

$$\Phi(x, z, t) = X(x)Z(z)\Psi(t) \quad (2.33)$$

Para la función $\Psi(t)$, se debe cumplir que sea periódica por la ecuación 2.32. Por ello, debe tener la siguiente forma:

$$\Psi(t) = E\cos(\sigma t) + F\sin(\sigma t) \quad (2.34)$$

Aplicando 2.32,

$$E\cos(\sigma t) + F\sin(\sigma t) = E\cos(\sigma(t + T)) + F\sin(\sigma(t + T)) \quad (2.35)$$

Para que la igualdad se cumpla y T sea el periodo de la función:

$$\sigma = \frac{2\pi}{T} \quad (2.36)$$

Siendo la denominada frecuencia angular del oleaje.

Ya obtenido este resultado, se sustituye 2.33 en la Ecuación de Gobierno 2.27.

$$\frac{\partial^2 X(x)}{\partial x^2} Z(z)\Psi(t) + \frac{\partial^2 Z(z)}{\partial z^2} X(x)\Psi(t) = 0 \quad (2.37)$$

Simplificando y dividiendo la ecuación anterior por Ψ :

$$\frac{1}{X(x)} \frac{\partial^2 X(x)}{\partial x^2} + \frac{1}{Z(z)} \frac{\partial^2 Z(z)}{\partial z^2} = 0 \quad (2.38)$$

En la ecuación anterior se observa que un miembro solo depende de la variable x y el otro solo depende de la variable z . Para que la ecuación tenga solución, ambos miembros deben ser constantes y opuestos, esto es:

$$\frac{1}{Z(z)} \frac{\partial^2 Z(z)}{\partial z^2} = -\frac{1}{X(x)} \frac{\partial^2 X(x)}{\partial x^2} = k^2 \quad (2.39)$$

Según el valor de k , se pueden dar los siguientes casos:

- Si k es real, $k^2 > 0$, las soluciones son:

$$X(x) = A\cos(kx) + B\sin(kx) \quad (2.40)$$

$$Z(z) = Ce^{kz} + De^{-kz} \quad (2.41)$$

- Si $k = 0$, las soluciones son:

$$X(x) = Ax + B \quad (2.42)$$

$$Z(z) = Cz + D \quad (2.43)$$

- Si $k^2 < 0$, es decir, si k es un número imaginario, esto es, $k = ik'$, siendo k' un número real:

$$X(x) = Ae^{k'x} + Be^{-k'x} \quad (2.44)$$

$$Z(z) = C\cos(k'x) + D\sin(k'x) \quad (2.45)$$

Debido a la condición de contorno de periodicidad espacial dada por la ecuación 2.31, la solución buscada debe ser periódica en x , por tanto, k debe ser real y distinto de cero. De esta forma, Φ tiene el siguiente aspecto:

$$\Phi(x, z, t) = (A\cos(kx) + B\sin(kx))(Ce^{kz} + De^{-kz})(E\cos(\sigma t) + F\sin(\sigma t)) \quad (2.46)$$

Para determinar las constantes A, B, C, D, E, F se deben aplicar las condiciones de contorno.

■ Condición de Contorno de Periodicidad Espacial

Para que el potencial sea periódico en x , debe darse:

$$\begin{aligned} A\cos(kx) + B\sin(kx) &= A\cos(k(x+L)) + B\sin(k(x+L)) = \\ &= A(\cos(kx)\cos(kL) - \sin(kx)\sin(kL)) + B(\cos(kx)\cos(kL) + \sin(kx)\sin(kL)) \end{aligned} \quad (2.47)$$

Para que esta igualdad se cumpla:

$$k = \frac{2\pi}{L} \quad (2.48)$$

Donde k es el conocido número de onda y L es la longitud de dicha onda. Dado que $X(x)$ y $\Psi(t)$ son una combinación lineal de senos y cosenos, ambas expresiones se pueden combinar, dejando al potencial Φ de la siguiente forma:

$$\Phi(x, z, t) = (Ce^{kz} + De^{-kz})(B_1\sin(kx - \sigma t + \zeta_1) + B_2\cos(kx + \sigma t - \zeta_2)) \quad (2.49)$$

■ Condición de Contorno Cinemática en el Fondo

$$w = \frac{\partial \Phi}{\partial z} = X(x)\Psi(t)k \cdot [(Ce^{kz} + De^{-kz})]_{z=-h} = X(x)\Psi(t)k \cdot (Ce^{-kh} + De^{kh}) = 0 \quad (2.50)$$

Para que esta igualdad se dé, independientemente del punto x del volumen de control, de debe cumplir que:

$$(Ce^{-kh} - De^{kh}) = 0 \iff C = De^{2kh} \quad (2.51)$$

Sustituyendo en el potencial Φ

$$\begin{aligned} \Phi(x, z, t) &= X(x)\Psi(t)D(e^{2kh}e^{kz} + e^{-kz}) = \\ &= X(x)\Psi(t)De^{kh}(e^{k(h+z)} + e^{-k(h+z)}) = \\ &= \cosh(k(h+z)) \cdot [(B'_1\sin(kx - \sigma t + \zeta_1) + B'_2\cos(kx + \sigma t - \zeta_2))] \end{aligned} \quad (2.52)$$

Donde $B'_1 = 2B_1De^{kh}$ y $B'_2 = 2B_2De^{kh}$

■ **Condición de Contorno Dinámica en la Superficie Libre**

$$\eta = -\frac{1}{g} \left[\frac{\partial \Phi}{\partial t} \right]_{z=0} \quad (2.53)$$

$$\begin{aligned} \eta &= \frac{\sigma}{g} \cosh(k(h+z)) [(B'_1 \cos(kx - \sigma t + \zeta_1) - B'_2 \sin(kx + \sigma t - \zeta_2))]_{z=0} = \\ &= \frac{\sigma \cosh(kh)}{g} [(B'_1 \cos(kx - \sigma t + \zeta_1) - B'_2 \sin(kx + \sigma t - \zeta_2))] \end{aligned} \quad (2.54)$$

Esta expresión es equivalente a dos oscilaciones de amplitudes:

$$A_1 = \frac{\sigma B'_1 \cosh(kh)}{g} ; A_2 = -\frac{\sigma B'_2 \cosh(kh)}{g} \quad (2.55)$$

Si se despejan B'_1 y B'_2 y se vuelven a sustituir en la expresión del potencial Φ , se obtiene:

$$\begin{aligned} \Phi(x, z, t) &= \frac{g}{\sigma} \frac{\cosh(k(h+z))}{\cosh(kh)} [A_1 \sin(kx - \sigma t + \zeta_1) - A_2 \sin(kx + \sigma t - \zeta_2)] = \\ &= \Re \left\{ -\frac{ig}{\sigma} \frac{\cosh(k(h+z))}{\cosh(kh)} [A'_1 e^{ikx} + A'_2 e^{-ikx}] e^{-i\sigma t} \right\} \end{aligned} \quad (2.56)$$

Donde $A'_1 = A_1 e^{i\zeta_1}$ y $A'_2 = A_2 e^{i\zeta_2}$.

La ecuación anterior representa la superposición de una onda progresiva de amplitud A_1 que se propaga en el sentido positivo del eje x y otra onda progresiva de amplitud A_2 que se propaga en el sentido negativo del eje x . La celeridad de ambas ondas son iguales y de valor $c = \frac{\sigma}{k} = \frac{L}{T}$. Si A_1 o A_2 es cero, se obtiene la expresión general de una onda progresiva, mientras que si A_1 y A_2 son distintos de cero se obtiene la ecuación de una onda estacionaria ($A_1 = A_2$) o parcialmente estacionaria ($A_1 \neq A_2$).

■ **Condición de Contorno Cinemática en la Superficie Libre**

Esta condición tenía la siguiente forma

$$\frac{\partial \eta}{\partial t} - w = 0 \quad \text{en } z = 0 \quad (2.57)$$

Donde, la expresión para η se puede obtener mediante la formula

$$\eta = A e^{i(kx - \sigma t)} \quad (2.58)$$

En consecuencia

$$\frac{\partial \eta}{\partial t} = -i\sigma A e^{i(kx - \sigma t)} \quad (2.59)$$

De la misma manera, w es:

$$\begin{aligned} w &= \left[\frac{\partial \Phi}{\partial z} \right]_{z=0} = \\ &= -\frac{igAk}{\sigma} e^{i(kx - \sigma t)} \left[\frac{\sinh(k(h+z))}{\cosh(kh)} \right]_{z=0} = \\ &= -\frac{igAk}{\sigma} \tanh(kh) e^{i(kx - \sigma t)} \end{aligned} \quad (2.60)$$

Volviendo de nuevo a la condición de contorno:

$$\frac{\partial \eta}{\partial t} - w = 0 \iff -iA\sigma e^{i(kx-\sigma t)} = -\frac{igAk}{\sigma} \tanh(kh) e^{i(kx-\sigma t)} \quad (2.61)$$

Despejando el valor de σ^2 , se obtiene:

$$\sigma^2 = gk \tanh(kh) \quad (2.62)$$

siendo la conocida Ecuación de Dispersión, que relaciona las variables h , T y k . De aquí en adelante, se le llamará 'Factor de Profundidad' $Kp(z)$ a la expresión:

$$Kp(z) = \frac{\cosh(k(h+z))}{\cosh(kh)} \quad (2.63)$$

Teniendo esta expresión en cuenta, la solución del problema propuesto, dado por la expresión del potencial, queda de la siguiente manera:

$$\Phi(x, z, t) = \Re \left\{ -\frac{ig}{\sigma} Kp(z) \left[A'_1 e^{ikx} + A'_2 e^{-ikx} \right] e^{-i\sigma t} \right\} \quad (2.64)$$

2.2.5. Solución General

Hasta ahora, se ha supuesto que la onda de estudio se propagaba en la dirección x , sin embargo, en la vida real, el oleaje toma diferentes direcciones. En este caso, es necesario proporcionar una solución que dependa de la variable x y de la variable y .

Se supone que la onda se propaga según un eje x' que forma un cierto ángulo θ con el eje x (ver figura 2.4).

A partir de la ecuación 2.64, se obtiene:

$$\Phi(x, z, t) = \Re \left\{ -\frac{ig}{\sigma} Kp(z) \left[A'_1 e^{ikx'} + A'_2 e^{-ikx'} \right] e^{-i\sigma t} \right\} \quad (2.65)$$

Mediante el siguiente cambio de base:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.66)$$

Se obtiene:

$$\Phi(x, z, t) = \Re \left\{ -\frac{ig}{\sigma} Kp(z) \left[A'_1 e^{i(k_x x + k_y y)} + A'_2 e^{-i(k_x x + k_y y)} \right] e^{-i\sigma t} \right\} \quad (2.67)$$

Donde:

$$k_x = k \cdot \cos(\theta) ; k_y = k \cdot \sin(\theta) \quad (2.68)$$

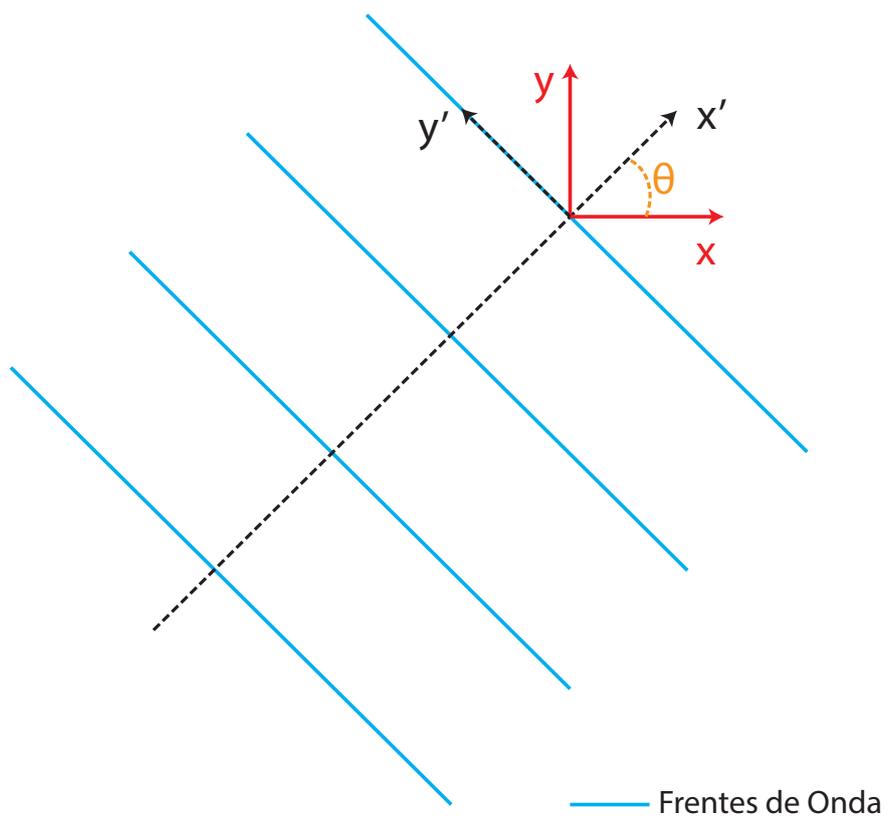


Figura 2.4: Esquema de una Onda General

CAPÍTULO 3

Ecuaciones Linealizadas del Movimiento de un Cuerpo Flotante

Este apartado es un resumen del Capítulo 8 de [1].

Se considera un cuerpo parcial o totalmente sumergido (ver figura 3.1), al que se llamará S_B . La posición de su contorno viene dada por la función $z = f(x, y, t)$. Para obtener las ecuaciones de su movimiento, es necesario aplicar la Condición Cinemática y las Ecuaciones de Conservación del Momento Lineal y del Momento Angular.

3.1. Condición Cinemática

Si aplicamos la condición de contorno cinemática a dicho cuerpo, se debe cumplir:

$$\Phi_x f_x + \Phi_y f_y + f_t = \Phi_z \quad \text{en } z = f(x, y, t) \quad (3.1)$$

Donde los subíndices indican la derivada parcial respecto del subíndice indicado.

Para movimientos de pequeña amplitud, se puede expandir la función f en potencias de la pendiente del oleaje ($\epsilon = kA$). De esta forma se caracteriza el movimiento del cuerpo de la siguiente forma:

$$z = f^{(0)}(x, y) + \epsilon f^{(1)}(x, y, t) + \epsilon^2 f^{(2)}(x, y, t) + \dots \quad (3.2)$$

Donde $f^{(0)}(x, y)$ no depende del tiempo ya que es la posición del cuerpo cuando está en reposo, esto es, $S_B^{(0)}$. De igual forma, se puede hacer la misma descomposición con el potencial de la velocidad del fluido:

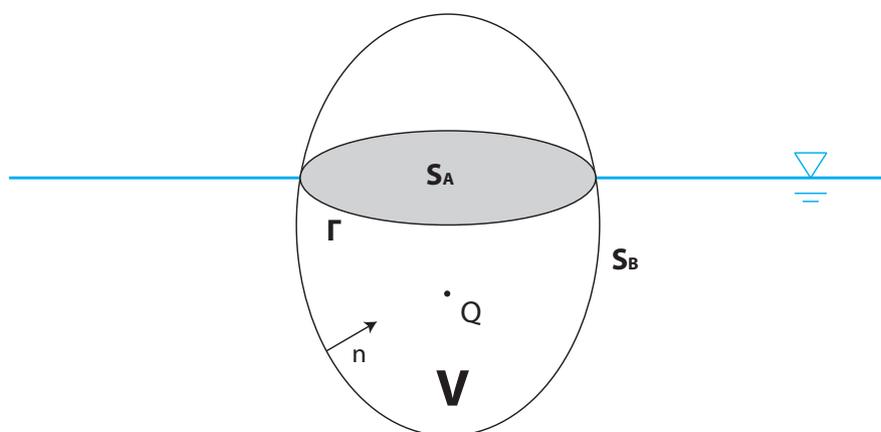


Figura 3.1: Esquema del Cuerpo Flotante

$$\Phi = \epsilon \Phi^{(1)} + \epsilon^2 \Phi^{(2)} + \dots \quad (3.3)$$

Evaluando de nuevo la ecuación 3.1 con estas modificaciones y despreciando los términos que van multiplicados por ϵ^2 y de mayor orden¹, se obtiene:

$$\Phi_x^{(1)} f_x^{(0)} + \Phi_y^{(1)} f_y^{(0)} + f_t^{(1)} = \Phi_z^{(1)} \quad \text{en } z = f^{(0)}(x, y) \quad (3.4)$$

De esta ecuación, es necesario encontrar el valor de $f^{(1)}$.

Al centro de rotación del cuerpo se le nombrará con la letra Q y sus coordenadas serán

$$\mathbf{X}(t) = \mathbf{X}^{(0)} + \epsilon \mathbf{X}^1(t) + \epsilon^2 \mathbf{X}^2(t) + \dots \quad \mathbf{X} = (x, y, z) \quad (3.5)$$

donde $\mathbf{X}^{(0)}$ es la posición de reposo de Q y es independiente de la variable t .

Se define ahora, la coordenada $\bar{\mathbf{x}}$, que es una coordenada fijada en el cuerpo, de tal manera que, la igualdad $\mathbf{x} = \bar{\mathbf{x}}$ solo se cumplirá en la posición de reposo. Si ahora se considera un desplazamiento angular del cuerpo $\epsilon \theta^{(1)}(t)$ cuyas componentes son $\epsilon \alpha$, $\epsilon \beta$ y $\epsilon \gamma$ alrededor de los ejes x , y y z respectivamente, los sistemas de coordenadas \mathbf{x} y $\bar{\mathbf{x}}$ se relacionan como se muestra en la siguiente fórmula:

$$\bar{\mathbf{x}} = \mathbf{x} - \epsilon \left[\mathbf{X}^{(1)} + \theta^{(1)} \times (\mathbf{x} - \mathbf{X}^{(0)}) \right] + O(\epsilon^2) \quad (3.6)$$

cuyas componentes, escritas por separado, son:

$$\bar{x} = x - \epsilon \left[X^{(1)} + \beta(z - Z^{(0)}) - \gamma(y - Y^{(0)}) \right] \quad (3.7)$$

$$\bar{y} = y - \epsilon \left[Y^{(1)} + \gamma(x - X^{(0)}) - \alpha(z - Z^{(0)}) \right] \quad (3.8)$$

$$\bar{z} = z - \epsilon \left[Z^{(1)} + \alpha(y - Y^{(0)}) - \beta(x - X^{(0)}) \right] \quad (3.9)$$

Dado que, por definición $\mathbf{x} = \bar{\mathbf{x}}$, se cumple que $\bar{z} = f^{(0)}(\bar{x}, \bar{y})$. Sustituyendo en la ecuación anterior las ecuaciones 3.7, 3.8 y 3.9 y comparando con la ecuación 3.2 queda:

$$\begin{aligned} f^{(1)} = & Z^{(1)} + \alpha(y - Y^{(0)}) - \beta(x - X^{(0)}) \\ & - f_x^{(0)} \left[X^{(1)} + \beta(z - Z^{(0)}) - \gamma(y - Y^{(0)}) \right] \\ & - f_y^{(0)} \left[Y^{(1)} + \gamma(x - X^{(0)}) - \alpha(z - Z^{(0)}) \right] \end{aligned} \quad (3.10)$$

Finalmente, combinando las ecuaciones 3.10 y 3.4 se obtiene la condición cinemática de primer orden:

¹Bajo la hipótesis de movimientos de pequeña amplitud, $\epsilon = kA \ll 1$

$$\begin{aligned}
-\Phi_x^{(1)} f_x^{(0)} - \Phi_y^{(1)} f_y^{(0)} - \Phi_z^{(1)} = & \\
& -f_x^{(0)} \left[X_t^{(1)} + \beta_t(z - Z^{(0)}) - \gamma_t(y - Y^{(0)}) \right] \\
& -f_y^{(0)} \left[Y_t^{(1)} + \gamma_t(x - X^{(0)}) - \alpha_t(z - Z^{(0)}) \right] \\
& + Z_t^{(1)} + \alpha_t(y - Y^{(0)}) - \beta_t(x - X^{(0)}) \quad (3.11)
\end{aligned}$$

Otra forma de expresar la ecuación anterior es mediante el vector normal a la superficie con sentido hacia dentro del cuerpo. Dicho vector tiene la siguiente forma:

$$\mathbf{n} = (-f_x^{(0)}, -f_y^{(0)}, 1) \left[1 + (f_x^{(0)})^2 + (f_y^{(0)})^2 \right]^{-1/2} \quad (3.12)$$

Teniendo en cuenta la expresión del vector \mathbf{n} , la ecuación 3.11 cambia de la siguiente forma:

$$\frac{\partial \phi^{(1)}}{\partial \mathbf{n}} = \mathbf{X}_t^{(1)} \cdot \mathbf{n} + \boldsymbol{\theta}_t^{(1)} \cdot \left[(\mathbf{x} - \mathbf{X}^{(0)}) \times \mathbf{n} \right] \quad (3.13)$$

Alternativamente, se puede introducir los vectores de seis elementos $\{\mathbf{X}_\alpha\}$ y $\{\mathbf{n}_\alpha\}$ definidos como sigue:

$$\{\mathbf{X}_\alpha\} = \{X^{(1)}, Y^{(1)}, Z^{(1)}, \alpha, \beta, \gamma\} = \{\mathbf{X}^{(1)}, \boldsymbol{\theta}^{(1)}\} \quad (3.14)$$

$$\{\mathbf{n}_\alpha\} = \{\mathbf{n}, (\mathbf{x} - \mathbf{X}^{(0)}) \times \mathbf{n}\} \quad (3.15)$$

De este modo se obtiene la forma final de la condición cinemática:

$$\frac{\partial \phi^{(1)}}{\partial \mathbf{n}} = \sum_{\alpha=1}^6 (X_\alpha)_t n_\alpha = \{\mathbf{X}_t\}^T \{n\} \quad (3.16)$$

3.2. Conservación del Momento Lineal

Se le llamará M a la masa total del cuerpo flotante, y $x^c(t)$ será la posición del centro de masas. La ecuación de conservación del momento lineal se formula de la siguiente manera:

$$Mx_{tt}^c = \int_{S_B} P \mathbf{n} dS - M g \mathbf{e}_3 + F \quad (3.17)$$

Donde S_B es la superficie sumergida del cuerpo, \mathbf{e}_3 es el vector $(0, 0, 1)$ y F es una fuerza aplicada al cuerpo por un agente externo. De igual forma que en la ecuación 3.6, se expresa la posición del centro de masas como sigue:

$$\mathbf{x}^c = \bar{\mathbf{x}}^c + \epsilon \left[\mathbf{X}^{(1)} + \boldsymbol{\theta}^{(1)} \times (\bar{\mathbf{x}}^c - \mathbf{X}^{(0)}) \right] \quad (3.18)$$

Donde se han despreciado los términos de orden ϵ^2 y superiores.

Por otra parte, a partir de la ecuación linealizada de Bernoulli se puede obtener:

$$P = -\rho g f - \epsilon \rho \Phi_t^{(1)} \quad (3.19)$$

Teniendo en cuenta las ecuaciones 3.18 y 3.19, la ecuación 3.17 resulta:

$$\epsilon M \left[\mathbf{x}_{tt}^{(1)} + \boldsymbol{\theta}_{tt}^{(1)} \times (\bar{\mathbf{x}}^c - \mathbf{X}^{(0)}) \right] = \int_{S_B^{(0)}} (-\rho g f - \epsilon \rho \Phi_t^{(1)}) \mathbf{n} dS - Mg \mathbf{e}_3 + F \quad (3.20)$$

Teniendo en cuenta que el término $\mathbf{n} dS$ es:

$$\mathbf{n} dS = (-f_x, -f_y, 1) dx dy \quad (3.21)$$

las integrales de la ecuación anterior se evalúan como sigue:

$$-\rho g \int_{S_B^{(0)}} f n_3 dS = -\rho g \int_{S_A^{(0)}} (f^{(0)} + \epsilon f^{(1)}) dx dy \quad (3.22)$$

Donde S_A es la intersección que se define por un corte entre la superficie del fluido y el cuerpo flotante (mirar figura 3.1). Esta última integral, sustituyendo la expresión de $f^{(1)}$ queda:

$$-\rho g \int_{S_B^{(0)}} f n_3 dS = -\rho g \int_{S_A^{(0)}} f^{(0)} dx dy + \epsilon \rho g \int_{S_A^{(0)}} (Z^{(1)} + \alpha(y - Y^{(0)}) - \beta(x - X^{(0)})) dx dy \quad (3.23)$$

Finalmente, se separan los términos de orden 0 y de orden 1 de la ecuación 3.20. Para los términos de orden 0 se obtiene la siguiente ecuación:

$$Mg = \rho g V^{(0)} + F_3^{(0)} \quad (3.24)$$

Donde $V^{(0)}$ es el volumen sumergido, resultado de la integral $\int_{S_A^{(0)}} f^{(0)} dx dy$; y $F_3^{(0)}$ es la fuerza externa en sentido vertical de orden 0. Como se puede apreciar, ésta es la ecuación correspondiente a la Ley de Arquímedes. De la misma forma, agrupando los términos de orden 1 en el eje z:

$$M \left[Z_{tt}^{(1)} + \alpha_{tt}(\bar{y}^c - Y^{(0)}) - \beta_{tt}(\bar{x}^c - X^{(0)}) \right] = -\rho \int_{S_B^{(0)}} \Phi_t^{(1)} n_3 dS + F_3^{(1)} - \rho g (I_2^A \alpha - I_1^A \beta + Z^{(1)} A^{(0)}) \quad (3.25)$$

$$I_1^A = \int_{S_A^{(0)}} (x - X^{(0)}) dx dy \quad (3.26)$$

$$I_2^A = \int_{S_A^{(0)}} (y - Y^{(0)}) dx dy \quad (3.27)$$

Nótese la siguiente igualdad para unos de los términos en x e y :

$$\rho g \int_{S_B} f n_1 dS = -\rho g \int_{S_B} f f_x dx dy = \oint_{\Gamma} \left[\frac{1}{2} f^2 \right] dy = O(\epsilon^2) \quad (3.28)$$

$$\rho g \int_{S_B} f n_2 dS = -\rho g \int_{S_B} f f_y dx dy = \oint_{\Gamma} \left[\frac{1}{2} f^2 \right] dx = O(\epsilon^2) \quad (3.29)$$

Se comprueba entonces que dichos términos son despreciables. Si se aíslan ahora los demás términos de orden 1 en los ejes x e y , se obtiene:

$$M \left[X_{tt}^{(1)} + \beta_{tt}(\bar{z}^c - Z^{(0)}) - \gamma_{tt}(\bar{y}^c - Y^{(0)}) \right] = -\rho \int_{S_B^{(0)}} \Phi_t^{(1)} n_1 dS + F_1^{(1)} \quad (3.30)$$

$$M \left[Y_{tt}^{(1)} + \gamma_{tt}(\bar{x}^c - X^{(0)}) - \alpha_{tt}(\bar{z}^c - Z^{(0)}) \right] = -\rho \int_{S_B^{(0)}} \Phi_t^{(1)} n_2 dS + F_2^{(1)} \quad (3.31)$$

3.3. Conservación del Momento Angular

Se llamará V_b al volumen total del cuerpo (incluyendo tanto la parte sumergida como la parte emergida). La conservación del momento angular se aplica con la siguiente fórmula:

$$\frac{\partial \mathbf{L}}{\partial t} = \frac{\partial}{\partial t} \int_{V_b} (\mathbf{x} \times \mathbf{x}_{tt}) dm = \int_{V_b} (\mathbf{x} \times \mathbf{x}_{tt}) dm \quad (3.32)$$

Donde dm es la masa del cuerpo por unidad de volumen y $\mathbf{x}(t)$ es la posición instantánea de un punto del cuerpo rígido.

La ecuación 3.32 se expresa como sigue:

$$\frac{\partial \mathbf{L}}{\partial t} = \int_{V_b} (\mathbf{x} \times \mathbf{x}_{tt}) dm = \int_{S_B} (\mathbf{x} \times P\mathbf{n}) dS + \mathbf{x}^c \times (-Mg\mathbf{e}_3) + \mathbf{T} \quad (3.33)$$

Donde \mathbf{T} es el momento ejercido por un agente externo.

Nótese también que las siguientes igualdades son ciertas:

$$\int_{V_b} \mathbf{x} dm = M\mathbf{x}^c \quad (3.34)$$

$$\int_{V_b} \mathbf{x}_{tt} dm = \int_{S_B} P\mathbf{n} dS - Mg\mathbf{e}_3 + \mathbf{F} \quad (3.35)$$

Así, haciendo el producto vectorial de la ecuación 3.35 con el vector \mathbf{X} y restando el resultado a la ecuación 3.33 se obtiene la ecuación de conservación del momento angular con respecto al centro de rotación Q (ver Figura 3.1).

$$\begin{aligned} \frac{\partial \mathbf{L}^Q}{\partial t} &= \int_{V_b} ((\mathbf{x} - \mathbf{X}) \times \mathbf{x}_{tt}) dm = \\ &= \int_{S_B} ((\mathbf{x} - \mathbf{X}) \times P\mathbf{n}) dS + (\mathbf{x}^c - \mathbf{X}) \times (-Mg\mathbf{e}_3) + (\mathbf{T} + \mathbf{F} \times \mathbf{X}) \end{aligned} \quad (3.36)$$

A partir de aquí, se supondrá que no hay fuerzas externas actuando en el cuerpo rígido.

En primer lugar, se estudiarán los términos de orden 0, así, la ecuación anterior resulta:

$$- \rho g \int_{S_B^{(0)}} ((\bar{\mathbf{x}} - \mathbf{X}^{(0)}) \times \mathbf{n} f^{(0)}) dS + (\bar{\mathbf{x}}^c - \mathbf{X}^{(0)}) \times (-Mg\mathbf{e}_3) = 0 \quad (3.37)$$

Siguiendo el mismo procedimiento que en la ecuación 3.23, la integral de la ecuación anterior se separa en sus distintas componentes. La componente en el eje x sería:

$$- \rho g \left[\int_{S_A^{(0)}} (y - Y^{(0)}) f^{(0)} dx dy + \int_{S_A^{(0)}} f^{(0)} (f^{(0)} - Z^{(0)}) f_y^{(0)} dx dy \right] \quad (3.38)$$

La segunda integral desaparece debido al término $f_y^{(0)}$. Por otro lado, se cumple la igualdad $f^{(0)} = - \int_{f^{(0)}}^0 dz$, por ello, la ecuación anterior se reescribe como:

$$\rho g \int_{V^{(0)}} (y - Y^{(0)}) dx dy dz = \rho g I_2^V = \rho g V^{(0)} (y^V - Y^{(0)}) \quad (3.39)$$

Donde $V^{(0)}$ es el volumen sumergido, I_2^V es el primer momento sobre el plano $y = Y^{(0)}$ y y^V es la coordenada y del centro de flotabilidad. Con esta nomenclatura, la ecuación 3.37 se reescribe como:

$$\rho g I_2^V - Mg(y^c - Y^{(0)}) = 0 \quad (3.40)$$

De forma similar, la componente y es:

$$-\rho g I_1^V + Mg(\bar{x}^c - X^{(0)}) = 0 \quad (3.41)$$

De las dos últimas ecuaciones se deduce que los centros de masas y de flotabilidad están en la misma vertical, esto es $x^c = x^V$ y $y^c = y^V$. Se recuerda la hipótesis que se ha realizado anteriormente en la que las fuerzas externas son 0; y de no cumplirse esta hipótesis, esta igualdad no sería cierta.

Ahora considerando los términos de orden ϵ , la ecuación 3.36 queda:

$$\frac{1}{\epsilon} \frac{\partial \mathbf{L}^Q}{\partial t} = \left(\int_{V_b} (\bar{\mathbf{x}} - \mathbf{X}^{(0)}) dm \right) \times \mathbf{X}_{tt}^{(1)} + \boldsymbol{\theta}_{tt}^{(1)} \int_{V_b} (\bar{\mathbf{x}} - \mathbf{X}^{(0)})^2 dm - \left(\boldsymbol{\theta}_{tt}^{(1)} \cdot \int_{V_b} (\bar{\mathbf{x}} - \mathbf{X}^{(0)}) \right) (\bar{\mathbf{x}} - \mathbf{X}^{(0)}) dm \quad (3.42)$$

Esta ecuación se puede separar en sus distintas componentes:

$$\frac{1}{\epsilon} \frac{\partial \mathbf{L}_1^Q}{\partial t} = I_2^b Z_{tt}^{(1)} - I_3^b Y_{tt}^{(1)} + (I_{22}^b + I_{33}^b) \alpha_{tt} - I_{21}^b \beta_{tt} - I_{31}^b \gamma_{tt} \quad (3.43)$$

$$\frac{1}{\epsilon} \frac{\partial \mathbf{L}_2^Q}{\partial t} = I_3^b X_{tt}^{(1)} - I_1^b Z_{tt}^{(1)} + (I_{33}^b + I_{11}^b) \beta_{tt} - I_{32}^b \gamma_{tt} - I_{12}^b \alpha_{tt} \quad (3.44)$$

$$\frac{1}{\epsilon} \frac{\partial \mathbf{L}_3^Q}{\partial t} = I_1^b Y_{tt}^{(1)} - I_2^b X_{tt}^{(1)} + (I_{11}^b + I_{22}^b) \gamma_{tt} - I_{13}^b \alpha_{tt} - I_{23}^b \beta_{tt} \quad (3.45)$$

$$I_1^b = \int_{V_b} (\bar{x} - X^{(0)}) dm = M(\bar{x}^c - X^{(0)}) \quad (3.46)$$

$$I_{11}^b = \int_{V_b} (\bar{x} - X^{(0)})^2 dm \quad (3.47)$$

$$I_{12}^b = \int_{V_b} (\bar{x} - X^{(0)}) (\bar{y} - Y^{(0)}) dm \quad (3.48)$$

El resto de términos se calculan de forma similar.

Por otra parte, el miembro derecho de la ecuación 3.36 queda:

$$-\epsilon \int_{S_B^{(0)}} \rho \Phi_t^{(1)} (\mathbf{x} - \mathbf{X}^{(0)}) \times \mathbf{n} dS - \rho g \left(\int_{S_B} f(\mathbf{x} - \mathbf{X}) \times \mathbf{n} dS - \int_{S_B^{(0)}} f^{(0)}(\mathbf{x} - \mathbf{X}^{(0)}) \times \mathbf{n} dS \right) + \epsilon (\boldsymbol{\theta}^{(1)} \times (\bar{\mathbf{x}}^c - \mathbf{X}^{(0)})) \times (-Mg\mathbf{e}_3) \quad (3.49)$$

Estos términos se tratarán de forma independiente:

■ Torsor Hidrodinámico

$$-\epsilon \int_{S_B^{(0)}} \rho \Phi_t^{(1)} (\mathbf{x} - \mathbf{X}^{(0)}) \times \mathbf{n} dS = -\epsilon \int_{S_B^{(0)}} \rho \Phi_t^{(1)} n_\alpha dS \quad \text{para } \alpha = 4, 5, 6 \quad (3.50)$$

■ **Torsor por Flotabilidad**

Despreciando términos de segundo orden y haciendo algunas operaciones (ver Capítulo 8 de [1]) el torsor de flotabilidad queda:

$$\begin{aligned} -\rho g \left(\int_{S_B} f(\mathbf{x} - \mathbf{X}) \times \mathbf{n} dS - \int_{S_B^{(0)}} f^{(0)}(\mathbf{x} - \mathbf{X}^{(0)}) \times \mathbf{n} dS \right) = \\ = -\epsilon \rho g \left[Z^{(1)} I_2^A + \alpha I_{22}^A - \beta I_{22}^A - \gamma I_1^V + \alpha I_3^V \right] \cdot \mathbf{e}_1 - \\ - \epsilon \rho g \left[Z^{(1)} I_1^A + \alpha I_{21}^A - \beta I_{11}^A - \beta I_3^V + \gamma I_2^V \right] \cdot \mathbf{e}_2 \quad (3.51) \end{aligned}$$

$$I_1^A = \int_{S_A} (\bar{x} - X^{(0)}) dm \quad (3.52)$$

$$I_{11}^A = \int_{S_A} (\bar{x} - X^{(0)})^2 dm \quad (3.53)$$

$$I_{12}^A = \int_{S_A} (\bar{x} - X^{(0)})(\bar{y} - Y^{(0)}) dm \quad (3.54)$$

El resto de términos se calcula de forma similar

■ **Torsor de Inercia**

$$\begin{aligned} \epsilon(\boldsymbol{\theta}^{(1)} \times (\bar{\mathbf{x}}^c - \mathbf{X}^{(0)})) \times (-Mg\mathbf{e}_3) = \\ = (-Mg)(\gamma(\bar{x}^c - X^{(0)}) - \alpha(\bar{z}^c - Z^{(0)})) \cdot \mathbf{e}_1 + \\ + (Mg)(\beta(\bar{z}^c - Z^{(0)}) - \gamma(\bar{y}^c - Y^{(0)})) \cdot \mathbf{e}_2 \quad (3.55) \end{aligned}$$

3.4. Ecuaciones Dinámicas para un Cuerpo Flotante (Forma Matricial)

A partir de todas las ecuaciones que se han formulado en este capítulo, es posible formar una sistema con la siguiente apariencia:

$$[M] \{\ddot{X}\} + [K] \{X\} = -\rho \int_{S_B^{(0)}} dS \Phi_t^{(1)} \{n\} + \{F\} \quad (3.56)$$

Donde $[M]$ y $[K]$ son matrices simétricas cuadradas de dimensión 3 para el caso bidimensional y de dimensión 6 para el caso tridimensional (ver figura 3.2) y F es el vector de fuerzas externas².

²Para más información se puede consultar el Capítulo 8 de [1].

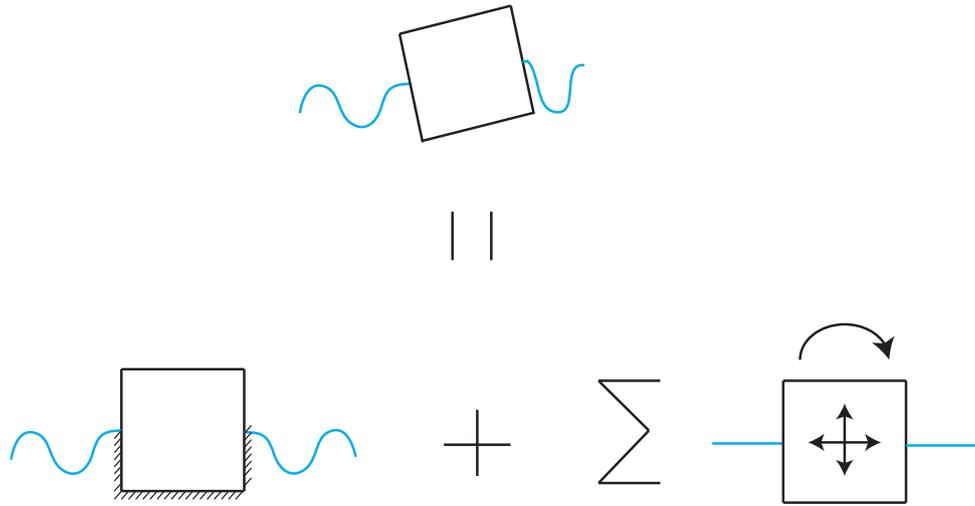


Figura 3.3: Descomposición en un Problema de Difracción y Problemas de Radiación

Los términos realmente interesantes para los problemas de ingeniería son los de amortiguamiento hidrodinámico y de masa añadida. Estos términos se obtendrán en los próximos apartados.

3.5. Descomposición en un Problema de Difracción y Problemas de Radiación

Cualquier problema de flotabilidad se puede descomponer en un problema de difracción y otros problemas de radiación (ver Figura 3.3). En el problema de difracción se supone que el cuerpo permanece inmóvil ante la acción de un oleaje entrante. De este problema se obtiene las fuerzas que excitan el movimiento. Por otro lado, en los problemas de radiación, se supone que el medio líquido están en calma (no hay potencial entrante) y el cuerpo sólido es excitado con un movimiento armónico en sus diferentes grados de libertad. Las fuerzas resultantes de estos problemas son las fuerzas restauradoras del movimiento.

Si se introduce la notación compleja y se desacopla el término temporal del resto de términos se tiene que:

$$\{\Phi, \dot{X}_\alpha\}^T = \text{Re} \{\phi, V_\alpha\}^T e^{i\sigma t} \quad (3.57)$$

Donde V_α es la amplitud de la velocidad del cuerpo.

Como se ha mencionado anteriormente, la descomposición en dos problemas se formula como:

$$\phi = \phi^D + \sum_{\alpha} V_\alpha \phi_\alpha \quad \alpha = 1, 2, \dots, 6 \quad (3.58)$$

El potencial ϕ^D está gobernado por las siguientes condiciones de contorno:

$$\nabla^2 \phi^D = 0 \quad \text{En el fluido} \quad (3.59)$$

$$\frac{\partial \phi^D}{\partial z} - \frac{\sigma^2}{g} \phi^D = 0 \quad \text{En } z = 0 \quad (3.60)$$

$$\frac{\partial \phi^D}{\partial z} = 0 \quad \text{En } z = -h \quad (3.61)$$

$$\frac{\partial \phi^D}{\partial n} = 0 \quad \text{En } S_B \quad (3.62)$$

$$\phi^D - \phi^I \quad \text{En el infinito} \quad (3.63)$$

Donde ϕ^D representa el potencial incidente.

Para los problemas de radiación, se define ϕ_α para que satisfaga las tres primeras condiciones impuestas anteriormente. Adicionalmente, para cada problema de radiación se debe imponer la siguiente condición:

$$\frac{\partial \phi_\alpha}{\partial n} = n_\alpha \quad \text{En } S_B \quad (3.64)$$

También se cumple la siguiente igualdad:

$$\frac{\partial \phi}{\partial n} = \sum_{\alpha} V_{\alpha} n_{\alpha} \quad \text{En } S_B \quad (3.65)$$

En las anteriores ecuaciones, el potencial incidente tiene esta apariencia:

$$\phi^I = \frac{-igA \cosh(k(z+h))}{\sigma \cosh(kh)} e^{ikr \cos(\theta - \theta_I)} \quad (3.66)$$

donde θ_I es la dirección del potencial incidente con respecto al eje X

3.6. Masa añadida y Amortiguamiento por Radiación

La α -ésima componente de la fuerza hidrodinámica en el cuerpo es:

$$F_{\alpha} = \int_{S_B} p n_{\alpha} dS = \text{Re}(F_{\alpha} e^{-i\sigma t}) \quad (3.67)$$

$$F_{\alpha} = i\rho\sigma \int_{S_B} \phi n_{\alpha} dS \quad (3.68)$$

Sustituyendo la ecuación 3.58 en la ecuación anterior, se obtiene:

$$F_{\alpha} = i\rho\sigma \int_{S_B} \phi^D n_{\alpha} dS + \sum_{\beta} i\rho\sigma V_{\beta} \int_{S_B} \phi_{\beta} n_{\alpha} dS \quad (3.69)$$

Con la siguiente notación:

$$F_{\alpha}^D = i\rho\sigma \int_{S_B} \phi^D n_{\alpha} dS \quad (3.70)$$

$$f_{\beta\alpha} = i\rho\sigma \int_{S_B} \phi_{\beta} n_{\alpha} dS \quad (3.71)$$

La ecuación 3.69 se reescribe como:

$$F_{\alpha} = F_{\alpha}^D + \sum_{\beta} V_{\beta} f_{\beta\alpha} \quad (3.72)$$

Considerando ahora solamente la parte real de la fuerza restauradora sin sumar sobre β resulta:

$$\begin{aligned}
Re(V_\beta f_{\beta\alpha} e^{-i\sigma t}) &= Re \left[(Re f_{\beta\alpha} + i Im f_{\beta\alpha}) V_\beta e^{-i\sigma t} \right] \\
&= Re \left[\left(i\rho\sigma \int_{S_B} Re\phi_\beta n_\alpha dS - \rho\sigma \int_{S_B} Im\phi_\beta n_\alpha dS \right) \cdot V_\beta e^{-i\sigma t} \right] = \\
&= - \left(\rho \int_{S_B} Re\phi_\beta n_\alpha dS \right) Re \frac{\partial}{\partial t} (V_\beta e^{i\sigma t}) - \\
&\quad - \left(\rho\sigma \int_{S_B} Im\phi_\beta n_\alpha dS \right) Re(V_\beta e^{i\sigma t}) = \\
&= - \left(\rho \int_{S_B} Re\phi_\beta n_\alpha dS \right) \ddot{X}_\beta - \left(\rho\sigma \int_{S_B} Im\phi_\beta n_\alpha dS \right) \dot{X}_\beta
\end{aligned} \tag{3.73}$$

La primera integral es proporcional a la aceleración del cuerpo y es la inercia hidrodinámica. Los términos de la matriz que forman se calculan como:

$$\mu_{\beta\alpha} = \rho \int_{S_B} Re\phi_\beta n_\alpha dS = \frac{1}{\sigma} Im f_{\beta\alpha} \tag{3.74}$$

Dichos términos forman la matriz de masa añadida (o added mass matrix).

La segunda integral de la ecuación anterior es proporcional a la velocidad del cuerpo y es el amortiguamiento hidrodinámico. Los términos de la matriz que forman se calculan como:

$$\lambda_{\beta\alpha} = \rho\sigma \int_{S_B} Im\phi_\beta n_\alpha dS = -Re f_{\beta\alpha} \tag{3.75}$$

Dichos términos forman la matriz de amortiguamiento por radiación (o radiation damping matrix).

Teniendo en cuenta estos nuevos términos obtenidos, la ecuación 3.56 se transforma en:

$$\left[-\sigma^2 ([M] + [\mu]) + [K] - i\sigma[\lambda] \right] \{\xi\} = F^D + F \tag{3.76}$$

Donde ξ es la amplitud de X :

$$\{X\} = Re \{\xi\} e^{-i\sigma t} \tag{3.77}$$

Estas dos matrices son de vital importancia para el estudio de un problema de ingeniería que incluya cuerpos total o parcialmente sumergidos. Dichas matrices se pueden introducir en programas de cálculo de estructuras con el objetivo de obtener resultados más realistas (y menos conservadores) en los estudios dinámicos de dichas estructuras.

CAPÍTULO 4

Método de los Elementos de Contorno

4.1. Introducción

Prácticamente todos los fenómenos físicos que ocurren en la naturaleza pueden ser descritos por ecuaciones diferenciales y condiciones de contorno. Con la solución de estos problemas se pretende determinar la respuesta del medio a dichas condiciones de contorno. Por ejemplo, problemas típicos son determinar la respuesta de una masa de rocas debido a la excavación de un túnel o determinar la respuesta de una estructura debido a excitaciones dinámicas (terremotos). Las soluciones analíticas que satisfacen las ecuaciones diferenciales y que cumplen las condiciones de contorno son difíciles de determinar, tanto es así que solo se conocen soluciones analíticas de unos pocos problemas con condiciones de contorno sencillas y que, en la mayoría de los casos, se corresponden poco con la realidad. Para el caso propuesto anteriormente, de conocer el estado de una masa de rocas ante la excavación de un túnel, existen soluciones analíticas para el caso en el que la roca es homogénea, pero esta restricción limita mucho su uso en la realidad[9]. Para resolver problemas de la vida real se debe recurrir a soluciones aproximadas, para ello hay dos técnicas posibles que se pueden aplicar. Éstas consisten en, o bien encontrar una solución que cumpla perfectamente las condiciones de contorno y minimice el error cometido en satisfacer la ecuación diferencial, o bien buscar una solución que verifique completamente la ecuación diferencial y que minimice el error cometido al satisfacer las condiciones de contorno. La primera técnica fue propuesta por Ritz[5] y supuso el primer paso para que Richard Courant desarrollase el conocido Método de los Elementos Finitos o FEM (acrónimo de Finite Elements Method en inglés) en 1943. La segunda técnica fue propuesta por Trefftz; esta técnica es utilizada por el Método de los Elementos de Contorno o BEM (acrónimo Boundary Elements Method en inglés), y es la que se va a tratar en el presente capítulo.

4.2. Diferencias entre FEM y BEM

El Método de los Elementos Finitos es ampliamente conocido. Consiste en dividir el dominio en pequeños elementos y aplicar una forma discretizada de la ecuación diferencial que rige el sistema (nótese que cuanto más pequeñas son las divisiones, menos error se comete al aplicar la forma discretizada de la ecuación diferencial). En este método existe una incógnita por cada elemento y las ecuaciones necesarias para su resolución se obtienen de imponer las relaciones entre cada elemento, los elementos que hay a su alrededor y los valores en el contorno del sistema (que son conocidos). En cambio, el Método de los Elementos de Contorno no requiere la división en elementos del dominio ya que las funciones usadas para aproximar la solución dentro del dominio son elegidas para que satisfagan exactamente la ecuación diferencial (como se ha mencionado en el apartado anterior). De forma parecida a como pasaba con el FEM, se debe minimizar el error cometido al satisfacer las condiciones de contorno. Esta técnica requiere una integral en la frontera del dominio. Numéricamente, para evaluar esta integral, el contorno debe ser dividido en elementos en los cuales los valores

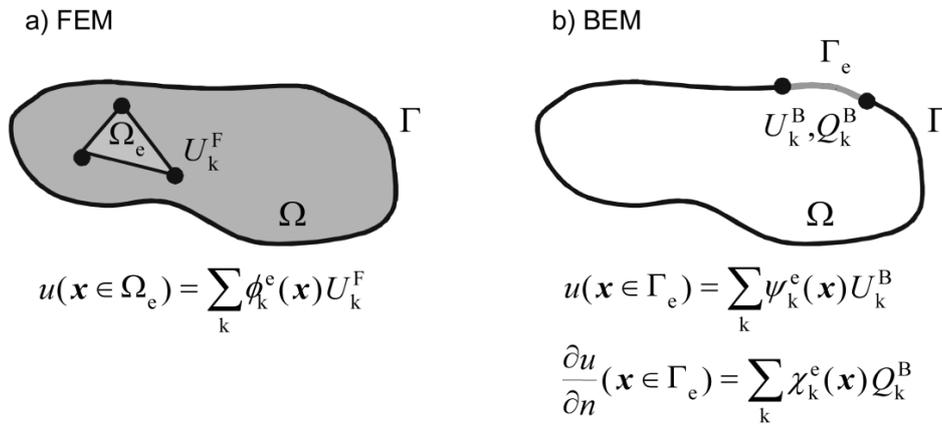


Figura 4.1: Diferencias entre el Método de Elementos Finitos y el Método de Elementos de Contorno. Fuente[2]

son interpolados. Según el grado de interpolación, se conciben varios tipos de Métodos de Elementos de Contorno como por ejemplo el método con elementos constantes (se supone que el valor de la variable es constante en todo el elemento) o el método con elementos lineales (se supone una variación lineal de la variable en cada elemento). La ventaja de este método es obvia, la dimensionalidad del problema se reduce en un orden, ya que en los problemas 3D se tratan con elementos 2D y en problemas 2D se trata con elementos lineales. Por otro lado, el número de elementos con los que se trabaja se reduce drásticamente, en especial para problemas 3D. Sin embargo, este método presenta un inconveniente, y es que, como se verá en las próximas secciones, cuando se tiene planteado el sistema de ecuaciones de la forma:

$$A \cdot \mathbf{x} = \mathbf{b} \quad (4.1)$$

La matriz A de coeficientes es una matriz llena. Este sistema de ecuaciones es más difícil de resolver que los sistemas resultantes de aplicar el FEM, en los que la matriz A es simétrica. Este simple hecho, dependiendo del problema que se tenga entre manos, puede hacer mucho más viable aplicar FEM que BEM aun teniendo muchos más elementos en el primer método. En la siguiente figura 4.1 se puede apreciar a modo de resumen las diferencias descritas en esta sección entre FEM y BEM.

4.3. Conceptos Básicos

Una ecuación importante en ingeniería es la conocida Ecuación de Poisson. Esta ecuación rige diversos problemas de electrostática y potencial gravitatorio. La forma bidimensional de esta ecuación se escribe como:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = b \text{ en un dominio } \Omega \quad (4.2)$$

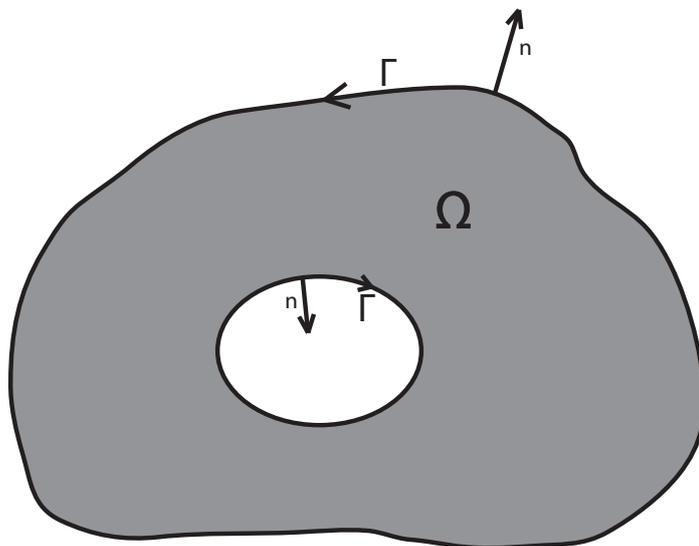


Figura 4.2: Definición del sentido del vector normal y de la frontera

O lo que es lo mismo:

$$\nabla^2 u = b \quad x \in \Omega \quad (4.3)$$

Siendo x un punto de coordenadas (x, y) y $\nabla^2(\) = \frac{\partial^2(\)}{\partial x^2} + \frac{\partial^2(\)}{\partial y^2}$ el llamado operador laplaciano. b es una función conocida en el dominio Ω que depende de (x, y) . Se llamará Γ al contorno que rodea a Ω y el vector normal a dicho contorno será aquel cuya dirección sea saliente del contorno, como se puede apreciar en la siguiente figura: La ecuación de Poisson en su forma homogénea, esto es $b = 0$, es la ecuación de Laplace. Dicha ecuación gobierna numerosos tipos de problemas ingenieriles como los análisis de acuíferos, conducción de calor, procesos de difusión, movimientos de fluidos etc. . .

A continuación se procede a explicar varios principios matemáticos, como el Teorema de Green, gracias a los cuales se obtiene el Método de los Elementos de Contorno. Para profundizar más en este paso es recomendable la lectura del capítulo 1 de [6].

En primer lugar, se introduce la idea de multiplicar la ecuación original de Poisson por una función arbitraria w , de esta forma se cumple:

$$\int_{\Omega} (\nabla^2 u - b) \cdot w \, d\Omega = 0 \quad (4.4)$$

Integrando por partes en términos de x e y se obtiene:

$$\int_{\Omega} \left(-\frac{\partial u}{\partial x} \frac{\partial w}{\partial x} - \frac{\partial u}{\partial y} \frac{\partial w}{\partial y} - bw \right) d\Omega + \int_{\Gamma} \frac{\partial u}{\partial n} w d\Gamma = 0 \quad (4.5)$$

Integrando por partes de nuevo:

$$\int_{\Omega} \left(\frac{\partial^2 w}{\partial x^2} u + \frac{\partial^2 w}{\partial y^2} u - bw \right) d\Omega + \int_{\Gamma} \frac{\partial u}{\partial n} w d\Gamma - \int_{\Gamma} u \frac{\partial w}{\partial n} d\Gamma = 0 \quad (4.6)$$

o lo que es lo mismo

$$\int_{\Omega} \left(\nabla^2 w \cdot u - bw \right) d\Omega + \int_{\Gamma} \frac{\partial u}{\partial n} w d\Gamma - \int_{\Gamma} u \frac{\partial w}{\partial n} d\Gamma = 0 \quad (4.7)$$

Combinando las ecuaciones 4.4 y 4.7 se obtiene:

$$\int_{\Omega} \left(\nabla^2 u \right) \cdot w d\Omega = \int_{\Omega} \left(\nabla^2 w \cdot u - bw \right) d\Omega + \int_{\Gamma} \frac{\partial u}{\partial n} w d\Gamma - \int_{\Gamma} u \frac{\partial w}{\partial n} d\Gamma \quad (4.8)$$

donde el término b se anula al estar presente en ambas ecuaciones. Por último, se expresa el conocido Teorema de Green que se obtiene como resultado de reorganizar los miembros de 4.8

$$\int_{\Omega} \left(\nabla^2 u \right) \cdot w - \left(\nabla^2 w \right) \cdot u d\Omega = \int_{\Gamma} \left(\frac{\partial u}{\partial n} w - u \frac{\partial w}{\partial n} \right) d\Gamma \quad (4.9)$$

Se considera, a partir de este punto, que la frontera está dividida en dos partes Γ_1 y Γ_2 . En la primera tenemos datos de u y en la segunda tenemos datos de $\frac{\partial u}{\partial n}$, esto es:

$$u = \bar{u} \text{ en } \Gamma_1$$

$$q = \frac{\partial u}{\partial n} = \bar{q} \text{ en } \Gamma_2$$

Basándose en las condiciones, la ecuación 4.7 se reescribe como:

$$\int_{\Omega} \left(\nabla^2 w \cdot u - bw \right) d\Omega + \int_{\Gamma_1} q w d\Gamma + \int_{\Gamma_2} \bar{q} w d\Gamma - \int_{\Gamma_1} \bar{u} \frac{\partial w}{\partial n} d\Gamma - \int_{\Gamma_2} u \frac{\partial w}{\partial n} d\Gamma = 0 \quad (4.10)$$

Una vez más, se integra por partes para volver a obtener el término $\nabla^2 u$ con el objetivo de ver la importancia de las condiciones de contorno en la ecuación. Integrando por partes una vez:

$$\int_{\Omega} \left(\frac{\partial^2 w}{\partial x^2} u + \frac{\partial^2 w}{\partial y^2} u - bw \right) d\Omega + \int_{\Gamma} \frac{\partial w}{\partial n} u d\Gamma + \int_{\Gamma_1} q w d\Gamma + \int_{\Gamma_2} \bar{q} w d\Gamma - \int_{\Gamma_1} \bar{u} \frac{\partial w}{\partial n} d\Gamma - \int_{\Gamma_2} u \frac{\partial w}{\partial n} d\Gamma = 0 \quad (4.11)$$

Nótese que al dividir la integral sobre Γ en las dos partes correspondientes, se anula el término $\int_{\Gamma_2} u \frac{\partial w}{\partial n} d\Gamma$. Dicho esto, se vuelve a aplicar integración por partes:

$$\int_{\Omega} (\nabla^2 u \cdot w - bw) d\Omega - \int_{\Gamma_2} qw d\Gamma + \int_{\Gamma_1} \frac{\partial w}{\partial n} u d\Gamma + \int_{\Gamma_2} \bar{q}w d\Gamma - \int_{\Gamma_1} \bar{u} \frac{\partial w}{\partial n} d\Gamma = 0 \quad (4.12)$$

que puede ser reescrita como:

$$\int_{\Omega} (\nabla^2 u \cdot w - bw) d\Omega - \int_{\Gamma_2} (q - \bar{q})w d\Gamma + \int_{\Gamma_1} (u - \bar{u}) \frac{\partial w}{\partial n} d\Gamma = 0 \quad (4.13)$$

En la ecuación descrita anteriormente no se ha hecho ningún tipo de simplificación, es decir, dicha ecuación satisface tanto la ecuación diferencial en todo el dominio como las condiciones de contorno en la frontera. Por supuesto, aunque toda la formulación ha sido presentada para el caso 2D, el mismo resultado se puede obtener al trabajar con un problema 3D con variables (x, y, z) .

4.4. Ecuación Integral de Contorno para la Ecuación de Laplace

Una vez descritos los primeros pasos, el objetivo será resolver la ecuación de Laplace (dos o tres dimensiones) en un dominio Ω . Esto es:

$$\nabla^2 u = 0 \text{ en } \Omega \quad (4.14)$$

Con las siguientes condiciones de contorno:

1. Condiciones tipo Dirichlet $\rightarrow u = \bar{u}$ en Γ_1
2. Condiciones tipo Neumann $\rightarrow q = \frac{\partial u}{\partial n} = \bar{q}$ en Γ_2 .

Como se verá mas adelante, también se han implementado en este Proyecto condiciones mixtas de la forma

$$\alpha u + \beta q = \gamma \quad (4.15)$$

en las que los parámetros α , β y γ son conocidos. Dichas condiciones se pueden ver como combinación de una condición de Dirichlet y una condición de Neumann, por lo que son fácilmente aplicables.

Retomando la ecuación 4.10, imponiendo la ecuación de Laplace $b = 0$ y renombrando $w = u^*$ y $\frac{\partial w}{\partial n} = q^*$ se obtiene:

$$\int_{\Omega} (\nabla^2 u^*) \cdot u d\Omega = - \int_{\Gamma_1} qu^* d\Gamma - \int_{\Gamma_2} \bar{q}u^* d\Gamma + \int_{\Gamma_1} \bar{u}q^* d\Gamma + \int_{\Gamma_2} uq^* d\Gamma \quad (4.16)$$

El objetivo es transformar la formula anterior en una ecuación integral de contorno, para ello se usa un tipo especial de función u^* llamada solución fundamental.

Se propone una solución fundamental u^* que satisface la ecuación de Laplace y representa el campo generado por una 'carga' concentrada actuando en el punto i . El efecto de esta 'carga' se

propaga desde el punto i hasta el infinito sin ninguna condición de contorno. Esta solución puede ser escrita como:

$$\nabla^2 u^* + \Lambda^i = 0 \quad (4.17)$$

Donde Λ^i representa la función Delta de Dirac, cuyas características son:

$$\Lambda^i = \left\{ \begin{array}{ll} 0 & \text{si } x \neq x_i \\ \infty & \text{si } x = x_i \end{array} \right\}.$$

$$\int_{-\infty}^{+\infty} \Lambda^i = 1$$

$$\int_{-\infty}^{+\infty} \Lambda^i \cdot f(x) = f(x_i)$$

Al introducir estas propiedades en el primer miembro de la ecuación 4.16:

$$\int_{\Omega} (\nabla^2 u^*) \cdot u d\Omega = \int_{\Omega} u (-\Lambda^i) d\Omega = -u^i \quad (4.18)$$

La ecuación 4.16 se reescribe:

$$u^i + \int_{\Gamma_1} \bar{u}q^* d\Gamma + \int_{\Gamma_2} uq^* d\Gamma = \int_{\Gamma_1} qu^* d\Gamma + \int_{\Gamma_2} \bar{q}u^* d\Gamma \quad (4.19)$$

Como se aprecia, para cada punto P^i interior a Ω se puede formular una ecuación distinta.

Para un problema 3D, la solución fundamental u^* viene dada por:

$$u^* = \frac{1}{4\pi r} \quad (4.20)$$

y para el caso bidimensional

$$u^* = \frac{1}{2\pi} \log \left(\frac{1}{r} \right) \quad (4.21)$$

donde r es la distancia desde el punto P^j , o 'Punto Objetivo', hasta el punto considerado P^i llamado 'Punto Fuente', que es donde se aplica la función delta.

Se ha definido que la función Λ^i se aplica sobre un punto interior al dominio Ω , pero ¿que ocurre si dicho punto de aplicación esta sobre la frontera Γ ?. Para responder la pregunta se debe plantear una pequeña deformación de la frontera como aparece en la figura 4.3 para el caso bidimensional y en la figura 4.4 para el caso tridimensional.

Se observa que la frontera se deforma artificialmente con una semicircunferencia (caso 2D) o una semiesfera (caso 3D) centrada en el punto de estudio situado en el contorno Γ . Dicha semicircunferencia o semiesfera tiene un radio ϵ , que, con el paso al límite se irá reduciendo hasta 0, quedando la

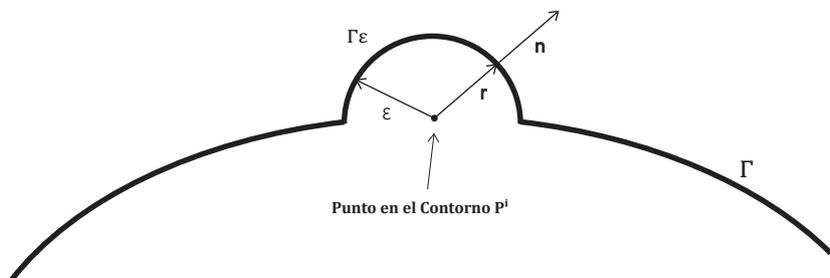


Figura 4.3: Deformación de la frontera en el caso bidimensional

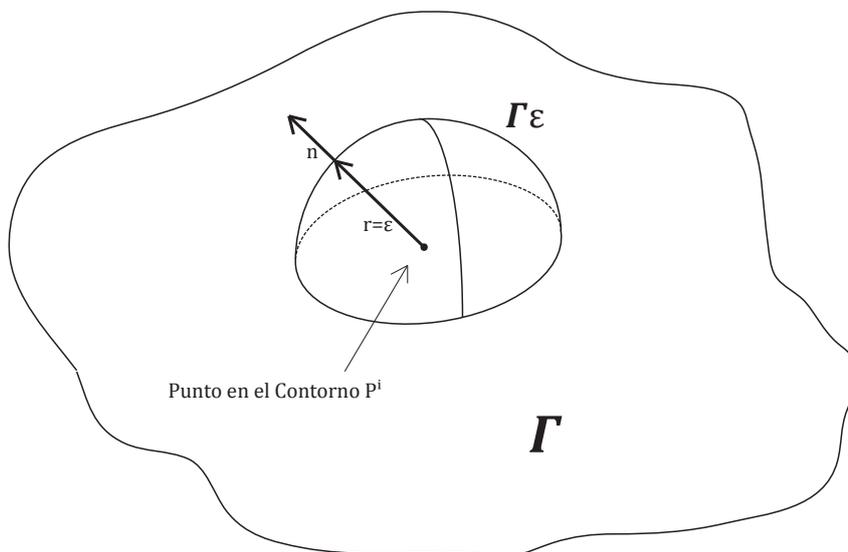


Figura 4.4: Deformación de la frontera en el caso tridimensional

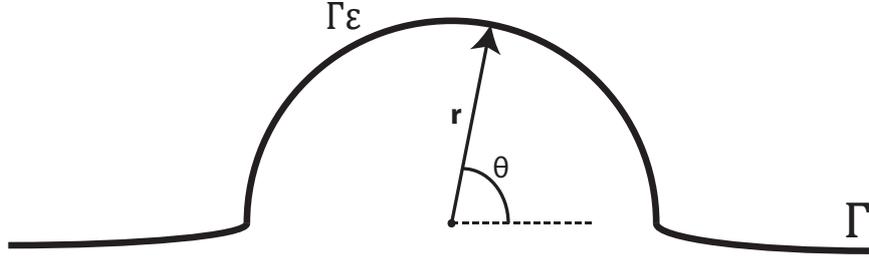


Figura 4.5: Cambio de coordenadas en el caso bidimensional

frontera Γ en su estado inicial. Nótese que, para estos casos, $n \equiv r$. A continuación, cada integral en el contorno de la ecuación 4.16 será evaluada teniendo en cuenta la nueva semiesfera o semicircunferencia, cuyo dominio es llamado Ω_ϵ y su contorno Γ_ϵ para una mayor claridad. Volviendo a unir ambos contornos $\Gamma_1 + \Gamma_2 = \Gamma$, se cumple que:

$$\int_{\Gamma} d\Gamma = \lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma - \Gamma_\epsilon} d\Gamma + \int_{\Gamma_\epsilon} d\Gamma \right\} \quad (4.22)$$

Para la primera integral:

$$\lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma - \Gamma_\epsilon} d\Gamma \right\} = \int_{\Gamma} d\Gamma \quad (4.23)$$

De modo que solo queda evaluar las integrales respecto de Γ_ϵ . Para el caso 3D:

$$\lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma_\epsilon} q \cdot u^* d\Gamma \right\} = \lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma_\epsilon} q \frac{1}{4\pi\epsilon} d\Gamma \right\} = \lim_{\epsilon \rightarrow 0} \left\{ q \frac{2\pi\epsilon^2}{4\pi\epsilon} \right\} = 0 \quad (4.24)$$

$$\lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma_\epsilon} u \cdot q^* d\Gamma \right\} = \lim_{\epsilon \rightarrow 0} \left\{ - \int_{\Gamma_\epsilon} u \frac{1}{4\pi\epsilon^2} d\Gamma \right\} = \lim_{\epsilon \rightarrow 0} \left\{ -u \frac{2\pi\epsilon^2}{4\pi\epsilon^2} \right\} = -\frac{1}{2}u^i \quad (4.25)$$

Se debe aclarar que los valores de u y q en el paso al límite son valores constantes y valen u^i y q^i respectivamente. Para el caso 2D, las integrales son algo más complicadas y se deben resolver con un cambio de variable como se puede apreciar en la figura 4.5. Basándose en este cambio, las ecuaciones para el caso bidimensional son:

$$\begin{aligned} \lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma_\epsilon} q \cdot u^* d\Gamma \right\} &= \lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma_\epsilon} q \frac{1}{2\pi} \log \left(\frac{1}{r} \right) d\Gamma \right\} = \\ &= \lim_{\epsilon \rightarrow 0} \left\{ q \frac{1}{2\pi} \int_0^\alpha \log \left(\frac{1}{\epsilon} \right) \epsilon d\theta \right\} = \lim_{\epsilon \rightarrow 0} \left\{ q \frac{1}{2\pi} \cdot \epsilon \cdot \log \left(\frac{1}{\epsilon} \right) \cdot \alpha \right\} = 0 \end{aligned} \quad (4.26)$$

$$\lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma_\epsilon} u \cdot q^* d\Gamma \right\} = \lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma_\epsilon} -u \frac{1}{2\pi r} d\Gamma \right\} = \lim_{\epsilon \rightarrow 0} \left\{ \int_0^\alpha -u \frac{1}{2\pi \epsilon} \cdot \epsilon d\theta \right\} = -u \frac{1}{2\pi} \cdot \alpha \quad (4.27)$$

Suponiendo que el punto P^i está situado en una zona diferenciable de la frontera Γ , esto es, que no esté en una esquina, se cumple que $\alpha = \pi$.

$$u \frac{1}{2\pi} \cdot \alpha = -\frac{1}{2}u \quad (4.28)$$

Como se puede observar, tanto en el caso bidimensional como en el tridimensional se ha llegado al mismo resultado, la ecuación integral de contorno:

$$u^i + \int_{\Gamma} uq^* d\Gamma = \int_{\Gamma} qu^* d\Gamma \text{ para } P^i \in \Omega \quad (4.29)$$

$$\frac{1}{2}u^i + \int_{\Gamma} uq^* d\Gamma = \int_{\Gamma} qu^* d\Gamma \text{ para } P^i \in \Gamma \quad (4.30)$$

4.5. Método de los Elementos de Contorno

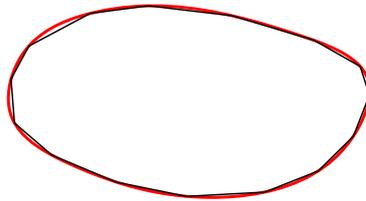
Una vez obtenida la ecuación 4.30, se procede a discretizarla con el fin de encontrar un sistema de ecuaciones con el que se puedan calcular los valores de u y q en el contorno. Se asume, por simplicidad, el caso 2D, en el que la frontera se divide en N segmentos o elementos como se muestra en la figura 4.6. Los puntos donde se quieren conocer valores de u o q se llaman nodos. Estos nodos se pueden tomar en el centro de los segmentos, por lo tanto, se considera que en cada elemento el valor de u y q es constante, estos segmentos son los llamados **Elementos Constantes**. Otra opción es tomar un nodo en cada extremo de cada segmento, de esta forma, se considera que hay una variación lineal de u y q en cada elemento, estos son los llamados **Elementos Lineales**. Un caso más complejo es considerar tres nodos por elemento, uno en cada extremo y otro en el centro, de esta forma se considera una variación cuadrática del valor de u y q en cada elemento, estos son los llamados **Elementos Cuadráticos**.

Para la formulación propuesta a continuación se considera el caso de elementos constantes, que es el caso más sencillo. Como se ha dicho, se asume que los valores de u y q son constantes a lo largo de cada elemento¹. Dicho esto, la ecuación 4.30 se reduce a:

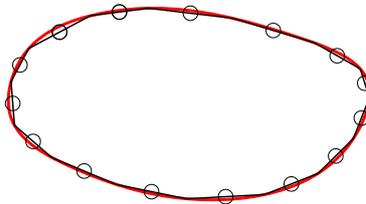
$$\frac{1}{2}u^i + \sum_{j=1}^N \int_{\Gamma_j} uq^* d\Gamma = \sum_{j=1}^N \int_{\Gamma_j} qu^* d\Gamma \quad (4.31)$$

Donde Γ_j es el j -ésimo elemento de la frontera y N es el número total de elementos resultantes de discretizar el contorno. El punto i , según se ha definido en el apartado anterior, es un punto perteneciente a Γ . Al ser los valores de u y q constantes en cada elemento, se pueden sacar fuera de la integral

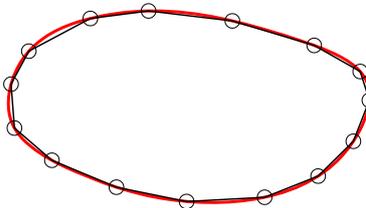
¹Más adelante, se tratará el caso de elementos lineales



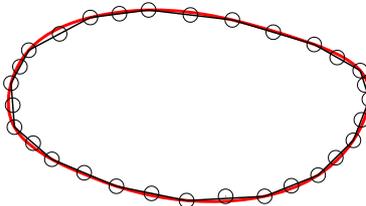
División en Elementos



Elementos Constantes



Elementos Lineales



Elementos Cuadráticos

Figura 4.6: División del Contorno en Elementos y Tipos de Elementos

$$\frac{1}{2}u^i + \sum_{j=1}^N u^j \cdot \left(\int_{\Gamma_j} q^* d\Gamma \right) = \sum_{j=1}^N q^j \cdot \left(\int_{\Gamma_j} u^* d\Gamma \right) \quad (4.32)$$

Nótese que se obtienen dos integrales:

$$\int_{\Gamma_j} q^* d\Gamma$$

$$\int_{\Gamma_j} u^* d\Gamma$$

que relacionan el nodo i con cada nodo j . Se recuerda que:

$$u^* = \frac{1}{2\pi} \log \left(\frac{1}{r} \right) \text{ para el caso 2D} \quad (4.33)$$

$$u^* = \frac{1}{4\pi r} \text{ para el caso 3D} \quad (4.34)$$

Siendo r

$$r = \|P^j - P^i\| = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2} \quad (4.35)$$

Donde P^j es el 'Punto o Nodo Objetivo', P^i el 'Punto o Nodo Fuente' y (x_k, y_k, z_k) son las coordenadas del nodo P^k . Si se toma la nomenclatura (n_x^k, n_y^k, n_z^k) como las componentes del vector normal a la frontera Γ en el punto P^k :

$$q^* = \frac{\partial u^*}{\partial n} = \frac{\partial u^*}{\partial r} \frac{\partial r}{\partial n} = \frac{-1}{2\pi r} \frac{n_x^j \cdot (x_j - x_i) + n_y^j \cdot (y_j - y_i)}{r} \text{ para el caso 2D} \quad (4.36)$$

$$q^* = \frac{\partial u^*}{\partial n} = \frac{\partial u^*}{\partial r} \frac{\partial r}{\partial n} = \frac{-1}{4\pi r^2} \frac{n_x^j \cdot (x_j - x_i) + n_y^j \cdot (y_j - y_i) + n_z^j \cdot (z_j - z_i)}{r} \text{ para el caso 3D} \quad (4.37)$$

Como se ha mencionado anteriormente, por cada elemento en la frontera se puede formular una ecuación de forma equivalente a 4.30. Es decir, tendríamos N ecuaciones de la forma:

$$\sum_{j=1}^N H^{ij} u^j = \sum_{j=1}^N G^{ij} q^j \quad (4.38)$$

Donde:

$$H^{ij} = \left\{ \begin{array}{ll} \frac{1}{2} + \int_{\Gamma_j} q^* d\Gamma & \text{si } i = j \\ \int_{\Gamma_j} q^* d\Gamma & \text{si } i \neq j \end{array} \right\}. \quad (4.39)$$

$$G^{ij} = \int_{\Gamma_j} u^* d\Gamma \quad (4.40)$$

Resultando un sistema tal que:

$$HU = GQ \quad (4.41)$$

Donde H y G son matrices cuadradas de orden N , en las cuales, la fila i representa los valores de $\int_{\Gamma_j} q^* d\Gamma$ y $\int_{\Gamma_j} u^* d\Gamma$ respectivamente con el nodo i como 'Punto Fuente'. Los vectores U y Q son vectores de N elementos cuyos valores son u^i y q^i respectivamente. A simple vista, puede parecer que se ha obtenido N ecuaciones y $2N$ incógnitas (dos incógnitas por nodo). Sin embargo, al aplicar las condiciones de contorno, las incógnitas se reducen hasta N ya que, en cada nodo de la frontera se conoce el valor de \bar{u} , el valor de \bar{q} o una combinación de ambas de la forma descrita en la ecuación 4.15. Introduciendo estos datos y reordenando el sistema se puede obtener un sistema de N ecuaciones de la forma:

$$Ax = F \quad (4.42)$$

Dicho sistema se puede resolver con algún algoritmo de resolución directo como el Método de Eliminación de Gauss o mediante un método iterativo como es el GMRES. Este último es el que se ha implementado en este proyecto y será explicado en los próximos capítulos. Así, el vector X contendrá los valores desconocidos a priori de u y q para cada nodo del contorno. Gracias a estos datos, ya se está en disposición de calcular los valores de u o q en cualquier punto del interior de Ω aplicando la ecuación 4.29, que, discretizada, resulta:

$$u^i = \sum_{j=1}^N G^{ij} q^j - \sum_{j=1}^N \bar{H}^{ij} u^j \quad (4.43)$$

Donde

$$\bar{H}^{ij} = \int_{\Gamma_j} q^* d\Gamma \quad (4.44)$$

En este caso el 'Punto Fuente' es siempre el mismo, es decir, el punto interior del que se quiere conocer el valor de u . Para calcular los valores de la derivada de u respecto x o y . La fórmula a aplicar es la siguiente:

$$q_x^i = \frac{\partial u}{\partial x} = \int_{\Gamma} q \left(\frac{\partial u^*}{\partial x} \right)^i d\Gamma - \int_{\Gamma} u \left(\frac{\partial q^*}{\partial x} \right)^i d\Gamma \quad (4.45)$$

$$q_y^i = \frac{\partial u}{\partial y} = \int_{\Gamma} q \left(\frac{\partial u^*}{\partial y} \right)^i d\Gamma - \int_{\Gamma} u \left(\frac{\partial q^*}{\partial y} \right)^i d\Gamma \quad (4.46)$$

Para el caso bidimensional:

$$\frac{\partial u^*}{\partial x} = \frac{\partial u^*}{\partial r} \frac{\partial r}{\partial x} = \frac{1}{2\pi r} \frac{-(x^j - x^i)}{r} \quad (4.47)$$

$$\frac{\partial u^*}{\partial y} = \frac{\partial u^*}{\partial r} \frac{\partial r}{\partial y} = \frac{1}{2\pi r} \frac{-(y^j - y^i)}{r} \quad (4.48)$$

$$\frac{\partial q^*}{\partial x} = \frac{\partial q^*}{\partial r} \frac{\partial r}{\partial x} = \frac{1}{\pi r^4} (x^j - x^i) \cdot \left[n_x^j (x^j - x^i) + n_y^j (y^j - y^i) \right] - \frac{n_x^j}{2\pi r^2} \quad (4.49)$$

$$\frac{\partial q^*}{\partial y} = \frac{\partial q^*}{\partial r} \frac{\partial r}{\partial y} = \frac{1}{\pi r^4} (y^j - y^i) \cdot \left[n_x^j (x^j - x^i) + n_y^j (y^j - y^i) \right] - \frac{n_y^j}{2\pi r^2} \quad (4.50)$$

Para el caso tridimensional:

$$\frac{\partial u^*}{\partial x} = \frac{\partial u^*}{\partial r} \frac{\partial r}{\partial x} = \frac{-1}{4\pi r^2} \frac{-(x^j - x^i)}{r} \quad (4.51)$$

$$\frac{\partial u^*}{\partial y} = \frac{\partial u^*}{\partial r} \frac{\partial r}{\partial y} = \frac{-1}{4\pi r^2} \frac{-(y^j - y^i)}{r} \quad (4.52)$$

$$\frac{\partial u^*}{\partial z} = \frac{\partial u^*}{\partial r} \frac{\partial r}{\partial z} = \frac{-1}{4\pi r^2} \frac{-(z^j - z^i)}{r} \quad (4.53)$$

$$\frac{\partial q^*}{\partial x} = \frac{\partial q^*}{\partial r} \frac{\partial r}{\partial x} = \frac{-3}{4\pi r^5} (x^j - x^i) \cdot \left[n_x^j (x^j - x^i) + n_y^j (y^j - y^i) + n_z^j (z^j - z^i) \right] + \frac{n_x^j}{4\pi r^3} \quad (4.54)$$

$$\frac{\partial q^*}{\partial y} = \frac{\partial q^*}{\partial r} \frac{\partial r}{\partial y} = \frac{-3}{4\pi r^5} (y^j - y^i) \cdot \left[n_x^j (x^j - x^i) + n_y^j (y^j - y^i) + n_z^j (z^j - z^i) \right] + \frac{n_y^j}{4\pi r^3} \quad (4.55)$$

$$\frac{\partial q^*}{\partial z} = \frac{\partial q^*}{\partial r} \frac{\partial r}{\partial z} = \frac{-3}{4\pi r^5} (z^j - z^i) \cdot \left[n_x^j (x^j - x^i) + n_y^j (y^j - y^i) + n_z^j (z^j - z^i) \right] + \frac{n_z^j}{4\pi r^3} \quad (4.56)$$

Como se comprueba, hay un cambio de signo al derivar r respecto de x , y o z , ya que se deriva respecto del 'Punto Fuente' y no respecto del 'Punto Objetivo'.

4.6. Evaluación de Singularidades

Como el lector habrá podido prever, en los elementos de la diagonal de G^{ij} y H^{ij} ocurre que el 'Punto Objetivo' es el mismo que el 'Punto Fuente', por lo que $r = 0$. Según el método aproximado propuesto, en estos casos tiene lugar una singularidad ya que $u^*(r = 0) = \infty$. Por tanto, se debe aplicar otro tipo de método de integración para esos elementos en concreto. Para el caso bidimensional con elementos constantes, se puede evaluar la integral exacta. Si se tiene en cuenta el esquema de la figura 4.7.

La formulación sería la siguiente:

$$H^{ii} = \int_{\Gamma_i} q^* d\Gamma = \int_{\Gamma_i} \frac{\partial u^*}{\partial r} \frac{\partial r}{\partial n} d\Gamma \equiv 0 \quad (4.57)$$

Como se puede apreciar en la figura 4.7, el vector r y el vector n son perpendiculares, por lo que $\frac{\partial r}{\partial n} = 0$.

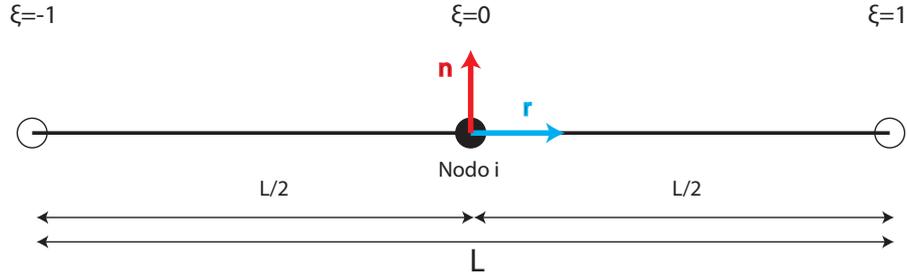


Figura 4.7: Esquema para la evaluación de singularidades en 2D

$$G^{ii} = \int_{\Gamma_i} u^* d\Gamma = \frac{1}{2\pi} \int_{\Gamma_i} \log\left(\frac{1}{r}\right) d\Gamma = \frac{1}{2\pi} \cdot 2 \cdot \int_0^{L/2} (-\log(r)) dr = \frac{-1}{\pi} \cdot [r \log(r) - r]_0^{L/2} = \frac{-L}{2\pi} \left(\log\left(\frac{L}{2}\right) - 1 \right) \quad (4.58)$$

Para el caso tridimensional con elementos constantes y triangulares, también se puede evaluar la integral exacta. El procedimiento a seguir es dividir el triángulo en tres subtriángulos cuyo punto común sea el Nodo (ver figura 4.8). Para resolver la integral hay que hacer un cambio a coordenadas polares de la siguiente forma:

$$dS = r dr d\theta$$

$$G^{ii} = \int_{\Gamma_i} u^* dS = \sum_{k=1}^3 \int_0^{\theta_k} \int_0^{R(\theta)} \frac{1}{4\pi r} r dr d\theta = \frac{1}{4\pi} \sum_{k=1}^3 \int_0^{\theta_k} R(\theta)_k d\theta \quad (4.59)$$

Dado que se conocen las coordenadas de los tres vértices del triángulo y las del nodo, es posible calcular los ángulos internos y los lados mediante trigonometría sencilla. El resultado de la integral se muestra a continuación (para conocer más detalles acerca del procedimiento, se puede consultar [8], [7] o [10]).

$$\int_0^{\theta_k} R(\theta)_k d\theta = R(0)_k \cdot \sin(A_k) \cdot \left[\log\left(\tan\left(\frac{A_k + \theta_k}{2}\right)\right) - \log\left(\tan\left(\frac{A_k}{2}\right)\right) \right] \quad (4.60)$$

La nomenclatura usada en la ecuación anterior queda especificada en la figura 4.9.

Para casos 2D y 3D con elementos no constantes es preferible usar reglas de cuadratura de Gauss, siempre y cuando ningún punto de colocación caiga en el centro del elemento (ya que en este caso se produciría una singularidad). Se recomienda consultar el Apéndice A de [6]; en él hay un compendio completo para cuadratura sobre elementos lineales (para BEM 2D), y de superficie (BEM 3D).

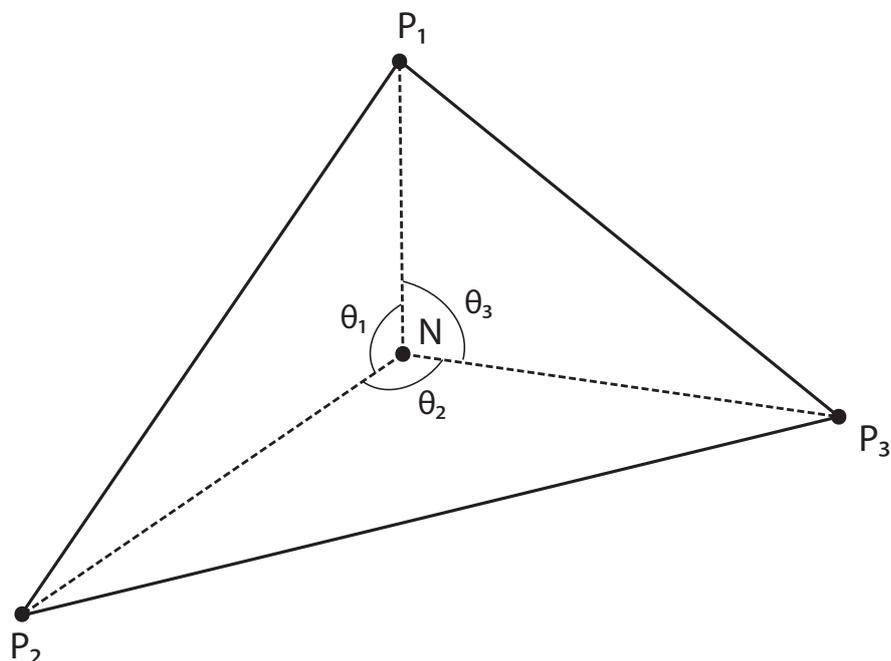


Figura 4.8: Evaluación de Singularidades con elementos triangulares constantes.

4.7. Elementos Lineales

Hasta ahora, toda la formulación que se ha considerado, se ha tratado con elementos constantes. Sin embargo, como se ha mencionado anteriormente, existen otros tipos de elementos, como los lineales o cuadráticos, que hacen más complejas las expresiones pero dan resultados más fiables y precisos. En este capítulo se explicará la aplicación de elementos lineales para el caso 2D y los cambios que se introducen en la formulación.

La ecuación integral de contorno es:

$$c^i u^i + \sum_{j=1}^N \int_{\Gamma_j} u q^* d\Gamma = \sum_{j=1}^N \int_{\Gamma_j} q u^* d\Gamma \quad (4.61)$$

Donde, como se puede comprobar, se ha sustituido $\frac{1}{2}u^i$ por $c^i u^i$ ya que, para elementos lineales, en la discretización del contorno Γ , los nodos están en esquinas (ver figura 4.11).

$$c^i = \frac{\theta}{2\pi} ; \text{ Donde } \theta \text{ se introduce en radianes} \quad (4.62)$$

En el caso estudiado en apartados anteriores, las variables u y q de 4.61, se podían sacar fuera de las integrales porque se suponía que eran valores constantes en cada elemento. En este caso, en cada elemento se supone una variación lineal del valor de u o q . Para ayudar a resolver las integrales,

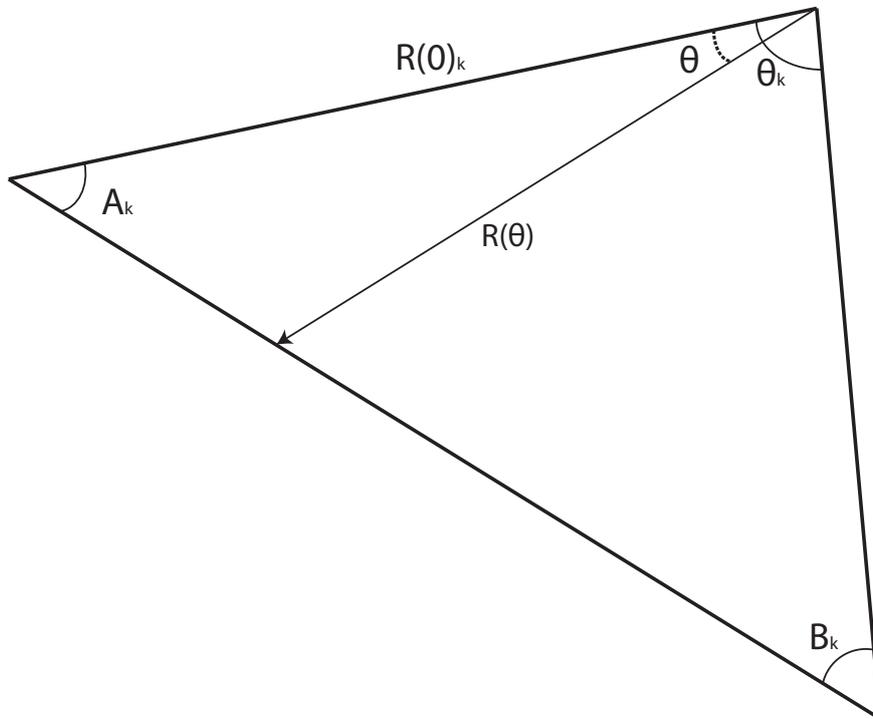


Figura 4.9: Esquema para la resolución de la integral obtenida en el estudio de la singularidad

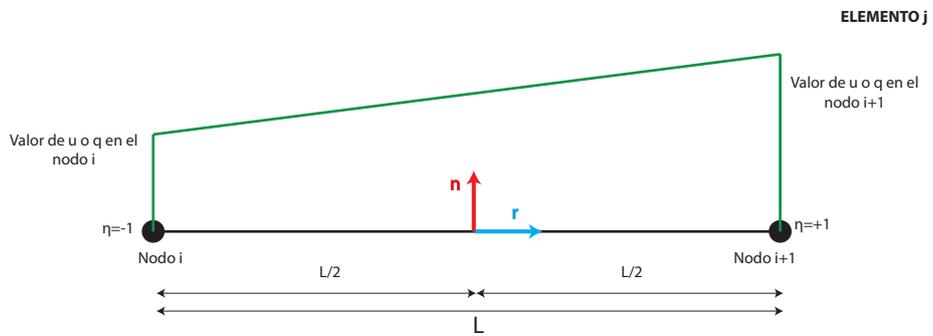


Figura 4.10: Esquema de los Elementos Lineales

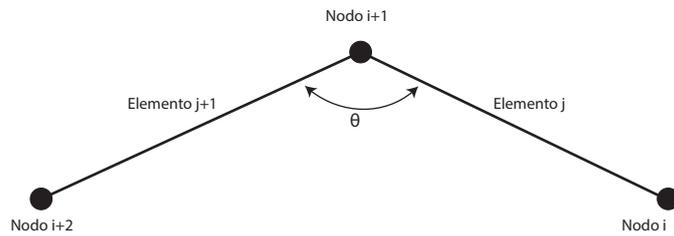


Figura 4.11: Ángulo entre Elementos Lineales

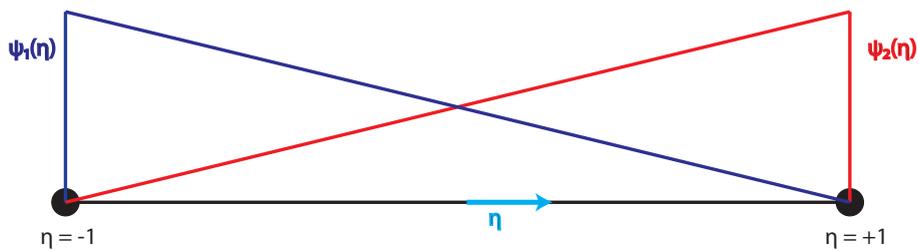


Figura 4.12: Funciones de Interpolación

se introducen unas funciones $\psi_1(\eta)$ y $\psi_2(\eta)$ que no son más que funciones de interpolación para representar la variación de u o q en cada elemento. Teniendo en cuenta las figuras 4.10 y 4.12.

$$\psi_1(\eta) = \frac{1}{2}(1 - \eta) \quad \psi_2(\eta) = \frac{1}{2}(1 + \eta) \quad \eta \in [-1, 1] \quad (4.63)$$

Así, se obtiene

$$u^j(\eta) = \psi_1 \cdot u^1 + \psi_2 \cdot u^2 \quad (4.64)$$

$$q^j(\eta) = \psi_1 \cdot q^1 + \psi_2 \cdot q^2 \quad (4.65)$$

Y las integrales quedan:

$$\int_{\Gamma_j} u q^* d\Gamma = \int_{\Gamma_j} q^* (\psi_1 \cdot u^1 + \psi_2 \cdot u^2) d\Gamma = h_1^{ij} \cdot u^1 + h_2^{ij} \cdot u^2 \quad (4.66)$$

$$\int_{\Gamma_j} q u^* d\Gamma = \int_{\Gamma_j} u^* (\psi_1 \cdot q^1 + \psi_2 \cdot q^2) d\Gamma = g_1^{ij} \cdot q^1 + g_2^{ij} \cdot q^2 \quad (4.67)$$

Siendo

$$h_1^{ij} = \int_{\Gamma_j} \psi_1 q^* d\Gamma \quad (4.68)$$

$$h_2^{ij} = \int_{\Gamma_j} \psi_2 q^* d\Gamma \quad (4.69)$$

$$g_1^{ij} = \int_{\Gamma_j} \psi_1 u^* d\Gamma \quad (4.70)$$

$$g_2^{ij} = \int_{\Gamma_j} \psi_2 u^* d\Gamma \quad (4.71)$$

En estos casos, el nodo 2 del elemento j es el mismo que el nodo 1 del elemento $j + 1$, por ello, el potencial u^2 del elemento j es igual al potencial u^1 del elemento $j + 1$. Sin embargo, q^2 del elemento j no tiene porque ser igual a q^1 del elemento $j + 1$, ya que la normal del elemento j no es la misma que la normal del elemento $j + 1$ ². Con esta observación, por lo general, en un problema con elementos lineales se tienen $3N$ incógnitas, siendo N el número de nodos en el contorno Γ . Esto es, N incógnitas del potencial u^i en cada nodo y $2N$ incógnitas de $\frac{\partial u^i}{\partial n}$ en cada nodo y para cada dirección de los elementos a los que pertenece dicho nodo.

Finalmente, reordenando, se tiene un sistema de N ecuaciones de la forma:

$$\sum_{j=1}^N H^{ij} u^j = \sum_{j=1}^{2N} G^{ij} q^j \quad (4.72)$$

Donde H^{ij} es una matriz $N \times N$ es de la forma:

$$H^{ij} = \left\{ \begin{array}{ll} c^i + h_1^{ij} + h_2^{i(j-1)} & \text{si } i = j \\ h_1^{ij} + h_2^{i(j-1)} & \text{si } i \neq j \end{array} \right\}. \quad (4.73)$$

Y G^{ij} es una matriz $N \times 2N$ de la forma:

$$G^{ij} = \begin{bmatrix} g_1^{ij} & g_2^{ij} \end{bmatrix} \quad (4.74)$$

Dado que se obtienen N ecuaciones y $3N$ incógnitas, se necesitan $2N$ datos para que el sistema sea resoluble, que pueden ser:

- Los datos del potencial u^i en todos los nodos y N datos de $\frac{\partial u}{\partial n}$.
- Los datos de $\frac{\partial u}{\partial n}$ en todos los nodos y para ambos elementos de cada nodo, es decir, $2N$ datos de $\frac{\partial u}{\partial n}$.

Hay un caso particular, en el que solo se conocen los datos de u^i en todos los nodos, que también es resoluble. La técnica consiste en duplicar los nodos situados en las esquinas, esto es, donde $\theta \neq \pi$, y

²En el caso de que $\theta = \pi$, sí se produce esta igualdad

llevarlos a una determinada distancia de dicha esquina. De esta forma, se pueden formular las ecuaciones necesarias para que el sistema sea resoluble. Para profundizar en esta técnica es recomendable la lectura del capítulo 2.7 de [6].

4.8. Conclusiones

A lo largo de este capítulo se ha explicado el Método de los Elementos de Contorno y los distintos usos de elementos que se pueden usar, así como una profundización en los elementos constantes y lineales. Para la implementación de elementos de mayor orden, se puede consultar bibliografía especializada como [6].

Sin embargo, ante la complejidad extra que añade el uso de elementos lineales y de mayor orden, para este proyecto se ha preferido el uso de elementos constantes más pequeños para conseguir el extra de precisión que se consigue con elementos de mayor orden. Además, en la mayor parte de la bibliografía consultada, la metodología usada iba encaminada en la misma dirección.

CAPÍTULO 5

Ecuación de Mild-Slope y el Método de los Elementos de Contorno de Doble Reciprocidad

5.1. Ecuación de Mild-Slope

En esta sección se obtendrá la ecuación de Mild-Slope siguiendo los mismos pasos que en [36].

Al igual que en el capítulo anterior, se supone que la pendiente de la onda es pequeña, esto es, $kA \ll 1$. Se desacopla el término temporal de la función potencial, quedando:

$$\sigma(x, y, t) = \eta(x, y)e^{-i\sigma t} \quad (5.1)$$

$$\Phi(x, y, z, t) = \phi(x, y, z)e^{-i\sigma t} \quad (5.2)$$

Las ecuaciones 2.27, 2.28, 2.29 y 2.30 que rigen el problema de ondas del oleaje se pueden volver a reformular quedando de la siguiente forma:

$$\frac{\partial^2 \phi}{\partial z^2} + \nabla^2 \phi = 0 \quad \text{en } -h(x, y) \leq z \leq 0 \quad (5.3)$$

$$\frac{\partial \phi}{\partial z} + \nabla \phi \cdot \nabla h = 0 \quad \text{en } z = -h(x, y) \quad (5.4)$$

$$\frac{\partial \phi}{\partial z} - \frac{\sigma^2}{g} \phi = 0 \quad \text{en } z = 0 \quad (5.5)$$

Donde, ahora, $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$

De igual manera, la solución se puede escribir como:

$$\phi(x, y, z) = \frac{ig\eta(x, y)}{\sigma} f(z, h) \quad (5.6)$$

$$f(z, h) = \frac{\cosh(k(z+h))}{\cosh(kh)} \quad (5.7)$$

Por supuesto, σ debe cumplir la ecuación de dispersión ($\sigma^2 = gk \tanh(kh)$)

La ecuación $f(z, h)$ cumple las siguientes condiciones:

$$\frac{\partial^2 f}{\partial z^2} + k^2 f = 0 \quad \text{en } -h(x, y) \leq z \leq 0 \quad (5.8)$$

$$\frac{\partial f}{\partial z} = 0 \quad \text{en } z = -h(x, y) \quad (5.9)$$

$$\frac{\partial f}{\partial z} - \frac{\sigma^2}{g} f = 0 \quad \text{en } z = 0 \quad (5.10)$$

Empleando la fórmula de integración por partes, esto es:

$$\int_{-h}^0 \left[f \left(\frac{\partial^2 \phi}{\partial z^2} - k^2 \phi \right) - \phi \left(\frac{\partial^2 f}{\partial z^2} - k^2 f \right) \right] dz = \left[f \frac{\partial \phi}{\partial z} - \phi \frac{\partial f}{\partial z} \right]_{-h}^0 \quad (5.11)$$

se puede llegar a obtener la siguiente igualdad:

$$\int_{-h}^0 f(k^2 \phi + \nabla^2 \phi) dz = [f \nabla h \cdot \nabla \phi]_{-h} \quad (5.12)$$

Sustituyendo las expresiones de $\nabla \phi$ y $\nabla^2 \phi$ resultantes de la ecuación 5.6, se obtiene:

$$\begin{aligned} \int_{-h}^0 \left[f^2 \nabla^2 \eta + 2f \frac{\partial f}{\partial h} \nabla \eta \cdot \nabla h + \eta f \frac{\partial^2 f}{\partial h^2} (\nabla h)^2 + \eta f \frac{\partial f}{\partial h} \nabla^2 h + k^2 \eta f^2 \right] dz = \\ = -\nabla h \cdot \nabla \eta f^2|_{-h} - \eta (\nabla h)^2 f \frac{\partial f}{\partial h}|_{-h} \end{aligned} \quad (5.13)$$

A partir de esta ecuación hay autores como [4] que desprecian los términos que acompañan a $(\nabla h)^2$ y $\nabla^2 h$ porque suponen que el término $\frac{\nabla h}{kh} \ll 1$. Otros, sin embargo, sí los tienen en cuenta, como es el caso de [32]. En este proyecto se hará una comparación entre los resultados teniendo en cuenta esos términos y despreciándolos para un caso en el la hipótesis de $\frac{\nabla h}{kh} \ll 1$ no es cierta.

Transformando la ecuación 5.13 (para más información consultar [32]) se obtiene:

$$\nabla \cdot (CC_g \nabla \eta) + \left[k^2 CC_g + u_1 \nabla^2 h + u_2 (\nabla h)^2 \right] \eta = 0 \quad (5.14)$$

Donde C es la celeridad de onda, C_g es la celeridad de grupo y u_1 y u_2 son dos funciones que dependen de k y h .

$$C = \sqrt{\frac{g}{k} \tanh(kh)} \quad (5.15)$$

$$C_g = \frac{C}{2} \left(1 + \frac{2kh}{\sinh(2kh)} \right) \quad (5.16)$$

$$u_1 = g \frac{1 - \tanh^2(kh)}{4(K + \sinh(K))} (\sinh(K) - K \cosh(K)) \quad (5.17)$$

$$\begin{aligned} u_2 = g \frac{k(1 - \tanh^2(kh))}{12(K + \sinh(K))} [K^4 + 4K^3 \sinh(K) - 9\sinh(K) \sinh(2K) + \\ + 3K(K + 2\sinh(K))(\cosh^2(K) - 2\cosh(K) + 3)] \end{aligned} \quad (5.18)$$

$$K = 2kh \quad (5.19)$$

La ecuación 5.14, es la llamada Ecuación Modificada de Mild-Slope. La ecuación de Mild-Slope original es la ecuación 5.14 despreciando los términos $u_1 \nabla^2 h$ y $u_2 (\nabla h)^2$, esto es:

$$\nabla \cdot (CC_g \nabla \eta) + k^2 CC_g \eta = 0 \quad (5.20)$$

Nótese que, para fondo constante, se obtiene la siguiente ecuación de Helmholtz.

$$\nabla^2 \eta + k^2 \eta = 0 \quad (5.21)$$

5.2. Método de los Elementos de Contorno de Doble Reciprocidad

El método de los elementos de contorno de doble reciprocidad se implementa para la resolución de ecuaciones diferenciales no homogéneas, es decir, ecuaciones del tipo:

$$\nabla^2 u + k^2 u = R(x, y, u, u_x, u_y) \quad (5.22)$$

En primer lugar, se planteará la solución del problema de la ecuación de Helmholtz mediante el método de los elementos de contorno y, posteriormente, se atajará el problema completo.

5.2.1. Ecuación de Helmholtz

Como se ha expuesto anteriormente, la ecuación de Helmholtz tiene la siguiente apariencia:

$$\nabla^2 u + k^2 u = 0 \quad (5.23)$$

Basándose en el mismo procedimiento explicado en el capítulo anterior, se llega a la siguiente ecuación:

$$c_\epsilon u + \int_{\Gamma} (uq^* - qu^*) d\Gamma = 0 \quad (5.24)$$

$$c_\epsilon = \begin{cases} 0 & \text{si el 'Punto Fuente' y el 'Punto Objetivo' no son el mismo} \\ \frac{1}{2} & \text{si el 'Punto Fuente' y el 'Punto Objetivo' son el mismo} \end{cases}.$$

Donde la función u^* , en este caso, es la solución del siguiente problema:

$$\nabla^2 u^* + k^2 u^* = -\Lambda^i \quad (5.25)$$

Por tanto, la ecuación u^* tiene la siguiente apariencia:

$$u^* = \frac{i}{4} H_0(kr) \quad (5.26)$$

Donde k es el número de onda y r es la distancia desde el 'Punto Objetivo' hasta el 'Punto Fuente'. La función $H(x)$ es la función de Hankel.

5.2.2. Ecuación Completa

La solución por el método de los elementos de contorno del problema completo tiene la siguiente apariencia:

$$-c_\epsilon u + \int_{\Gamma} (qu^* - uq^*) d\Gamma = \int_{\Omega} Ru^* d\Omega \quad (5.27)$$

Donde aparece una integral en el dominio que se debe eliminar.

El método de los elementos de contorno con doble reciprocidad, propuesto por Nardini y Brebbia en [37], consiste en usar una combinación lineal de funciones que aproximen la función R en el dominio, esto es

$$R \approx \sum_{j=1}^{n+l} \alpha_j f_j(x) \quad (5.28)$$

Como se puede apreciar, se debe obligar a que la combinación lineal de funciones de interpolación dé valores exactos en n puntos del contorno del dominio y en l puntos del interior del dominio. Solo quedaría averiguar los valores de las distintas α , para ello se debe resolver el siguiente sistema de ecuaciones:

$$\begin{pmatrix} R(x_1) \\ R(x_2) \\ \vdots \\ R(x_N) \end{pmatrix} = \begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1N} \\ f_{21} & f_{22} & \cdots & f_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ f_{N1} & f_{N2} & \cdots & f_{NN} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{pmatrix} \quad (5.29)$$

Donde $N = n + l$ y $f_{ij} = f_j(x_i)$. En forma matricial, el sistema de ecuaciones anterior se expresa:

$$\mathbf{R} = \mathbf{F}\boldsymbol{\alpha} \quad (5.30)$$

Si la matriz \mathbf{F} es invertible, el vector $\boldsymbol{\alpha}$ se puede obtener de la siguiente forma:

$$\boldsymbol{\alpha} = \mathbf{F}^{-1}\mathbf{R} \quad (5.31)$$

Estas funciones de interpolación f se eligen de tal forma que se conozca la solución al siguiente problema:

$$\nabla^2 \hat{u}_j + k^2 \hat{u}_j = f_j \quad (5.32)$$

Aplicando la segunda identidad de Green al segundo miembro de 5.27:

$$\begin{aligned} \int_{\Omega} Ru^* d\Omega &\approx \sum_{j=1}^{n+l} \alpha_j \int_{\Omega} f_j(x) u^* d\Omega \\ &= \sum_{j=1}^{n+l} \alpha_j \int_{\Omega} [\nabla^2 \hat{u}_j + k^2 \hat{u}_j] u^* d\Omega \\ &= \sum_{j=1}^{n+l} \alpha_j \left[-c_{\epsilon} \hat{u}_j + \int_{\Gamma} (\hat{q}_j u^* - \hat{u}_j q^*) d\Gamma \right] \end{aligned} \quad (5.33)$$

Sustituyendo en la ecuación 5.27 se obtiene:

$$c_{\epsilon} u - \int_{\Gamma} (qu^* - uq^*) d\Gamma = \sum_{j=1}^{n+l} \alpha_j \left[c_{\epsilon} \hat{u}_j - \int_{\Gamma} (\hat{q}_j u^* - \hat{u}_j q^*) d\Gamma \right] \quad (5.34)$$

5.2.3. Elección de la Función de Interpolación

La exactitud de la solución del método de los elementos de contorno de doble reciprocidad depende en gran medida de la elección de una función de interpolación adecuada. Las funciones de interpolación más famosas son las funciones llamadas 'RBF' que son:

$$\begin{aligned} f_j(x) &= f(r_j) \\ r_j &= \|x - x_j\| \end{aligned} \quad (5.35)$$

Partridge y Brebbia en [29] demostraron que se obtenían unos resultados satisfactorios para funciones de la forma:

$$f_j(x) = \sum_{m=0}^s r_j^m \quad (5.36)$$

Es más, estos autores reseñaron que con $s = 1$, es decir, $f_j = 1 + r_j$ era suficiente.

Más recientemente, Muleshkov en [38] demostró que con los splines de orden n :

$$f_j(x) = r_j^{2n} \log(r_j) \quad (5.37)$$

se obtenían los mejores resultados vistos hasta la fecha en cuanto convergencia y exactitud.

Sin embargo, en este proyecto se usarán funciones tipo 'RBF' al igual que [4] y otros. Más concretamente, se empleará la siguiente función de interpolación:

$$f_j(x) = 1 + r_j^2 + r_j^3 \quad (5.38)$$

La razón de elegir concretamente esta función es que en [39] se demostró que dicha función no introducía singularidades en la matriz F al contrario que la función $f_j = 1 + r_j$ que sí las introducía.

Como se ha mencionado anteriormente, se debe conocer la solución del problema 5.32. Dicha solución viene proporcionada en los apéndices de [4]. Para un problema del tipo:

$$\nabla^2 \hat{u}_j + k^2 \hat{u}_j = r^j \quad (5.39)$$

la solución es:

$$\begin{aligned} \hat{u}_j &= \frac{-\pi}{2} J_0(kr) \int_0^r t^{j+1} Y_0(kt) dt + \frac{\pi}{2} Y_0(kr) \int_0^r t^{j+1} J_0(kt) dt \\ &= \frac{-\pi}{2} J_0(kr) I_{j+1}(Y_0(kr)) + \frac{\pi}{2} Y_0(kr) I_{j+1}(J_0(kr)) \end{aligned} \quad (5.40)$$

Donde $I_{j+1}(Y_0(kr))$ y $I_{j+1}(J_0(kr))$ son las fórmulas recursivas que se pueden obtener de [40]. Estas fórmulas recursivas tienen la siguiente forma:

$$\begin{aligned} I_{j+1}(Y_0(kr)) &= \int_0^r t^{j+1} Y_0(kt) dt = \frac{r^j}{k} \left[r Y_1(kr) + \frac{j}{k} Y_0(kr) \right] - \frac{j^2}{k^2} I_{j-1}(Y_0(kr)) \\ I_1(Y_0(kr)) &= \frac{r}{k} Y_1(kr) \\ I_0(Y_0(kr)) &= r \left[Y_0(kr) + \frac{\pi}{2} \left(\hat{H}_0(kr) Y_1(kr) - \hat{H}_1(kr) Y_0(kr) \right) \right] \end{aligned} \quad (5.41)$$

De manera similar se puede obtener la expresión de $I_{j+1}(J_0(kr))$.

En la ecuación anterior, $\hat{H}_0(kr)$ y $\hat{H}_1(kr)$ son las funciones de Struve de orden 0 y de orden 1 respectivamente. $J_a(x)$ y $Y_a(x)$ son la primera y segunda función de Bessel, respectivamente, de orden a .

Nótese también las siguientes igualdades:

$$\hat{u}_0 = \frac{1}{k^2} \quad (5.42)$$

$$\frac{\partial \hat{u}_0}{\partial r} = 0 \quad (5.43)$$

$$\frac{\partial \hat{u}_j}{\partial r} = \frac{\pi}{2} k J_1(kr) I_{j+1}(Y_0(kr)) - \frac{\pi}{2} k Y_1(kr) I_{j+1}(J_0(kr)) \quad (5.44)$$

5.2.4. Evaluación de Singularidades

Al igual que en el capítulo anterior, para este problema también aparecen singularidades en la función u^* y q^* cuando el 'Punto Objetivo' coincide con el 'Punto Fuente'. En este caso, se evaluará la integral analítica en el segmento (se desarrollará la solución para el problema bidimensional y elementos constantes).

$$u^* = \frac{i}{4} H_0(kr) \quad (5.45)$$

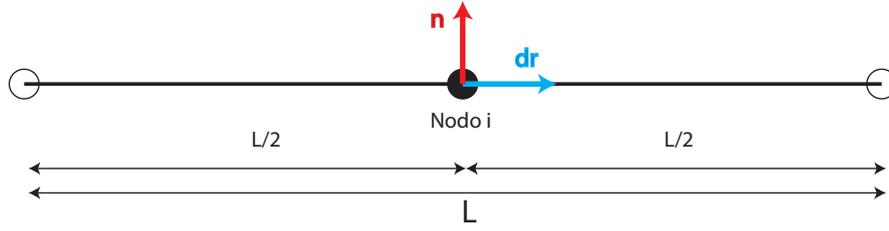


Figura 5.1: Esquema para la evaluación de singularidades

$$q^* = \frac{\partial u^*}{\partial n} = \frac{\partial u^*}{\partial r} \frac{\partial r}{\partial n} = \frac{-ik}{4} H_1(kr) \cdot \frac{n_x^j \cdot (x_j - x_i) + n_y^j \cdot (y_j - y_i)}{r} \quad (5.46)$$

Al igual que en el capítulo anterior, el vector dr y el vector normal son perpendiculares, ver figura 5.1, por tanto ocurre:

$$\int_{-L/2}^{L/2} q^* dr = 0 \quad (5.47)$$

Por otro lado, para u^* :

$$\int_{-L/2}^{L/2} u^* dr = 2 \cdot \int_0^{L/2} u^* dr = \frac{2i}{4k} \int_0^{kL/2} H_0(t) dt \quad (5.48)$$

Donde $t = kr$, para resolver la integral, sirve de apoyo el Capítulo 11 de [40], donde consta la siguiente igualdad:

$$\int_0^x H_0(t) dt = xH_0(x) + \frac{\pi x}{2} \left[\hat{H}_0(x)H_1(x) - \hat{H}_1(x)H_0(x) \right] \quad (5.49)$$

Basándose en la anterior igualdad, se puede obtener la integral de forma analítica:

$$\int_{-L/2}^{L/2} u^* dr = \frac{i}{2k} \left(\frac{kL}{2} H_0\left(\frac{kL}{2}\right) + \frac{\pi kL}{4} \left[\hat{H}_0\left(\frac{kL}{2}\right)H_1\left(\frac{kL}{2}\right) - \hat{H}_1\left(\frac{kL}{2}\right)H_0\left(\frac{kL}{2}\right) \right] \right) \quad (5.50)$$

5.3. Aplicación del Método de los Elementos de Contorno de Doble Reciprocidad a la Ecuación de Mild-Slope

5.3.1. Resolución de la Ecuación de MildSlope con Fondo Variable

En este apartado se dará solución a un problema en el que una región con fondo variable se ve excitada con un tren monocromático de olas que se propaga con un ángulo de incidencia θ , el cual se puede expresar como:

$$\eta^I(x, y) = A e^{ik_0(x \cos \theta + y \sin \theta)} \quad (5.51)$$

Como se habrá podido observar, cuando hay que estudiar un fondo variable, se tiene una ecuación del tipo 5.14 para cada punto del dominio ya que los términos CC_g , u_1 , $\nabla^2 h$, u_2 y $(\nabla h)^2$ varían con la profundidad.

Para poder aplicar el método de los elementos de contorno de doble reciprocidad, la ecuación 5.14 se debe transformar en esta otra:

$$\begin{aligned}\nabla^2(G\eta) + k_0^2(G\eta) &= R(x) \\ G &= CC_g \\ R(x) &= (k_0^2 - k^2)G\eta + \nabla G \cdot \nabla \eta + \eta \nabla^2 G - u_1 \nabla^2 h \eta - u_2 (\nabla h)^2 \eta\end{aligned}\quad (5.52)$$

Donde k_0 es el número de onda en el contorno del dominio¹.

A partir de este punto se utilizará la siguiente nomenclatura:

- n = número de elementos de contornos interiores, por ejemplo islas.
- m = número de elementos de contorno exteriores, es decir, donde está presente el potencial u oleaje entrante.
- l = número de nodos interiores donde se interpolarán las funciones $f(r)$.
- Γ_o = contorno exterior.
- Γ_i = contorno interior.
- Ω_o = región exterior.
- Ω_i = región interior, donde se va a llevar a cabo el estudio de oleaje

Siguiendo con el procedimiento, la solución por el método de los elementos de contorno viene dada por:

$$\begin{aligned}c_\epsilon^{(i)} G(\epsilon) \eta(\epsilon) - \int_{\Gamma_o + \Gamma_i} \frac{\partial G}{\partial n} \eta \eta^* d\Gamma + \int_{\Gamma_o + \Gamma_i} G(\eta q^* - q \eta^*) d\Gamma &= - \int_{\Omega_i} R \eta^* d\Omega \\ q &= \frac{\partial \eta}{\partial n} \\ q^* &= \frac{\partial \eta^*}{\partial n} \\ c_\epsilon^{(i)} &= \left\{ \begin{array}{ll} \frac{1}{2} & \text{si } \epsilon \in \Gamma_o + \Gamma_i \\ 1 & \text{si } \epsilon \in \Omega_i \end{array} \right\}.\end{aligned}\quad (5.53)$$

Como se ha mencionado anteriormente, en la región exterior Ω_o se asume que la profundidad es constante, por lo tanto, en dicha región exterior, se cumple la siguiente ecuación:

$$\nabla^2 \eta_s + k_0^2 \eta_s = 0 \quad (5.54)$$

Cuya solución es:

$$\begin{aligned}-c_\epsilon^{(o)} \eta_s(\epsilon) + \int_{\Gamma_o} \left(\frac{\partial \eta_s}{\partial n'} \eta^* - \eta_s \frac{\partial \eta^*}{\partial n'} \right) d\Gamma + \int_{\Gamma_\infty} \left(\frac{\partial \eta_s}{\partial n'} \eta^* - \eta_s \frac{\partial \eta^*}{\partial n'} \right) d\Gamma &= 0 \\ c_\epsilon^{(o)} &= \left\{ \begin{array}{ll} \frac{1}{2} & \text{si } \epsilon \in \Gamma_o \\ 0 & \text{si } \epsilon \in \Omega_i + \Gamma_i \end{array} \right\}.\end{aligned}\quad (5.55)$$

Donde n' es el vector normal a la región Ω_o , con sentido hacia fuera de dicha región. Por consiguiente, el vector n y n' tiene sentidos opuestos en el contorno Γ_o .

¹Para este tipo de problemas, se supone que el fondo en la región exterior a la zona de estudio, así como el contorno exterior de la zona de estudio son planos y con la misma profundidad.

Aplicando la condición de radiación, esto es:

$$\lim_{r \rightarrow \infty} \sqrt{r} \left(\frac{\partial \eta}{\partial r} - ik\eta \right) = 0 \quad (5.56)$$

La segunda integral de la ecuación 5.55 desaparece. Finalmente, si se tiene en cuenta que:

$$\left\{ \begin{array}{l} \eta = \eta_s + \eta^I \\ q = - \left(\frac{\partial \eta_s}{\partial n'} + \frac{\partial \eta^I}{\partial n'} \right) \end{array} \right\} \quad (5.57)$$

se obtiene:

$$\int_{\Gamma_o} (\eta q^* - q \eta^*) d\Gamma = c_\epsilon^{(o)} \eta(\epsilon) - c_\epsilon^{(o)} \eta^I(\epsilon) + \int_{\Gamma_o} (\eta^I q^* - q^I \eta^*) d\Gamma \quad (5.58)$$

La expresión anterior, junto con la resultante en 5.33, se sustituyen en la ecuación 5.53 teniendo así la forma final de la solución:

$$\begin{aligned} c_\epsilon G(\epsilon) \eta(\epsilon) - \int_{\Gamma_o + \Gamma_i} \frac{\partial G}{\partial n} \eta \eta^* d\Gamma + \int_{\Gamma_i} G(\eta q^* - q \eta^*) d\Gamma = \\ = \sum_{j=1}^{n+m+l} \alpha_j \left[c_\epsilon^{(i)} \hat{\eta}_j(\epsilon) + \int_{\Gamma_o + \Gamma_i} (\hat{\eta}_j q^* - \hat{q}_j \eta^*) d\Gamma \right] \\ + G_o \left[c_\epsilon^{(o)} \eta^I(\epsilon) - \int_{\Gamma_o} (\eta^I q^* - q^I \eta^*) d\Gamma \right] \end{aligned} \quad (5.59)$$

Donde:

$$c_\epsilon = \left\{ \begin{array}{l} \frac{1}{2} \quad \text{si } \epsilon \in \Gamma_i \\ 1 \quad \text{si } \epsilon \in \Gamma_o \\ 1 \quad \text{si } \epsilon \in \Omega_i \end{array} \right\}. \quad (5.60)$$

Además, $\hat{\eta}_j$ son las soluciones al problema:

$$\nabla^2 \hat{\eta}_j + k^2 \eta_j = f_j \quad (5.61)$$

y, por último, $\hat{q}_j = \frac{\partial \hat{\eta}_j}{\partial n}$.

Resolviendo este sistema de ecuaciones en los que, al introducir las condiciones de contorno, solo se consigue un vector de incógnitas, se obtienen los valores de η en los puntos de los contornos Γ_o y Γ_i y en los l puntos interiores (que se han usado para las funciones de interpolación). Con estos valores de η se deduce la altura de ola compleja en cada instante de tiempo con la siguiente fórmula:

$$\zeta(\mathbf{x}, t) = \eta(\mathbf{x}) e^{i\sigma t} = [Real(\eta) \cos(\sigma t) + Im(\eta) \sen(\sigma t)] + i [Im(\eta) \cos(\sigma t) - Real(\eta) \sen(\sigma t)] \quad (5.62)$$

Y, por consiguiente, resultan la altura de ola física y el run-up, que es la máxima elevación de la superficie que puede darse:

$$\zeta_{Fisica}(\mathbf{x}, t) = Real(\eta) \cos(\sigma t) + Im(\eta) \sen(\sigma t) \quad (5.63)$$

$$run - up(\mathbf{x}, t) = Abs(\zeta_{Fisica}(\mathbf{x}, t)) = \sqrt{(Real(\eta))^2 + (Im(\eta))^2} \quad (5.64)$$

5.3.2. Discretización Numérica

Las integrales del sistema de ecuaciones 5.59 se deben discretizar para poder implementar la formulación en un algoritmo. Dicha discretización da como resultado el siguiente sistema de ecuaciones equivalente:

$$\begin{aligned}
c_\epsilon G(\epsilon)\eta(\epsilon) - \sum_{k=1}^{n+m} \left(\frac{\partial G}{\partial n} \eta \right) |_{x_k} g_{\epsilon k} + \sum_{k=1}^n (G\eta) |_{x_k} h_{\epsilon k} - \sum_{k=1}^n (Gq) |_{x_k} g_{\epsilon k} = \\
= \sum_{j=1}^{n+m+l} \alpha_j \left[c_\epsilon^{(i)} \hat{\eta}_j(\epsilon) + \sum_{k=1}^{n+m} (\hat{\eta}_j |_{x_k} h_{\epsilon k} - \sum_{k=1}^{n+m} \hat{q}_j |_{x_k} g_{\epsilon k}) \right] \\
+ G_o \left[c_\epsilon^{(o)} \eta^I(\epsilon) - \sum_{k=n+1}^{n+m} (\eta^I |_{x_k} h_{\epsilon k}) + \sum_{k=n+1}^{n+m} (q^I |_{x_k} g_{\epsilon k}) \right] \quad (5.65)
\end{aligned}$$

Donde, para elementos constantes:

$$g_{\epsilon k} = \int_{Elementok} \eta^*(\epsilon, x) d\Gamma = \frac{i}{4} H_0(kr) \quad (5.66)$$

$$h_{\epsilon k} = \int_{Elementok} q^*(\epsilon, x) d\Gamma = \frac{-ik}{4} H_1(kr) \cdot \frac{n_{x_k}^x \cdot (x_k^x - \epsilon^x) + n_{x_k}^y \cdot (x_k^y - \epsilon^y)}{r} \quad (5.67)$$

siendo r es la distancia entre el 'Punto Fuente', llamado en este caso ϵ , y el 'Punto Objetivo', llamado en este caso x_k

Como el lector habrá podido imaginar, toda esta formulación tiene un inconveniente, y es que, para obtener el vector \mathbf{R} es necesario hallar las incógnitas η . Este problema se puede solucionar mediante dos vías, la primera consiste en suponer unos valores de η y resolver, mediante cálculo iterativo, el problema. La segunda vía, que es la que se ha adoptado para este proyecto, es cambiar la expresión del vector α por su expresión equivalente $\mathbf{F}^{-1}\mathbf{R}$. Haciendo varias operaciones (consultar Apéndice A de [4]) resulta el siguiente sistema matricial:

$$[\mathbf{C} - (\mathbf{G}_i + \mathbf{G}_o)\mathbf{G}_n + \mathbf{G}_i\mathbf{H}_i - \mathbf{S}] \mathbf{U} - \mathbf{G}_i\mathbf{G}_i\mathbf{Q} = \mathbf{G}_o \left[(\mathbf{C}^{(o)} - \mathbf{H}_o)\mathbf{U}^I + \mathbf{G}_o\mathbf{Q}^I \right] \quad (5.68)$$

$$\mathbf{S} = \left[(\mathbf{C}^{(i)} + \mathbf{H}_i + \mathbf{H}_o) \hat{\mathbf{U}} - (\mathbf{G}_i + \mathbf{G}_o) \hat{\mathbf{Q}} \right] \mathbf{F}^{-1} \left[\mathbf{K}^2\mathbf{G} + \nabla^2\mathbf{G} + (\mathbf{G}_x\mathbf{F}_x + \mathbf{G}_y\mathbf{F}_y) \mathbf{F}^{-1} - \mathbf{U1} - \mathbf{U2} \right] \quad (5.69)$$

Donde:

$$\begin{aligned}
\mathbf{C} &= \text{diag}(0.5G|_{x_1}, \dots, 0.5G|_{x_n}, G|_{x_{n+1}}, \dots, G|_{x_{n+m+1}}) \\
\mathbf{G}_n &= \left(\frac{\partial G}{\partial n} \Big|_{x_1}, \dots, \frac{\partial G}{\partial n} \Big|_{x_{n+m}}, 0, \dots, 0 \right) \\
\mathbf{C}^{(0)} &= \text{diag}(\underbrace{0, \dots, 0}_n, \underbrace{0.5, \dots, 0.5}_m, \underbrace{0, \dots, 0}_l) \\
\mathbf{C}^{(i)} &= \text{diag}(\underbrace{0.5, \dots, 0.5}_n, \underbrace{0.5, \dots, 0.5}_m, \underbrace{1, \dots, 1}_l) \\
\mathbf{U} &= (\eta_{x_1}, \dots, \eta_{x_{n+m+1}})^T \\
\mathbf{Q} &= (\eta_{x_1}, \dots, \eta_{x_{n+m}}, \underbrace{0, \dots, 0}_l)^T \\
\mathbf{U}^I &= (\eta_{x_1}^I, \dots, \eta_{x_{n+m+1}}^I)^T \\
\mathbf{Q}^I &= (\eta_{x_1}^I, \dots, \eta_{x_{n+m}}^I, \underbrace{0, \dots, 0}_l)^T \\
\mathbf{F}_x &= \left[\frac{\partial f_j}{\partial x} \Big|_{x_i} \right]_{(n+m+1) \times (n+m+1)} \\
\mathbf{F}_y &= \left[\frac{\partial f_j}{\partial y} \Big|_{x_i} \right]_{(n+m+1) \times (n+m+1)} \\
\mathbf{G} &= \text{diag}(G|_{x_1}, \dots, G|_{x_{n+m+1}}) \\
\mathbf{G}_x &= \text{diag}\left(\frac{\partial G}{\partial x} \Big|_{x_1}, \dots, \frac{\partial G}{\partial x} \Big|_{x_{n+m+1}}\right) \\
\mathbf{G}_y &= \text{diag}\left(\frac{\partial G}{\partial y} \Big|_{x_1}, \dots, \frac{\partial G}{\partial y} \Big|_{x_{n+m+1}}\right) \\
\nabla^2 \mathbf{G} &= \text{diag}(\nabla^2 G|_{x_1}, \dots, \nabla^2 G|_{x_{n+m+1}}) \\
\mathbf{K}^2 &= \text{diag} \left((k_0^2 - k^2|_{x_1}), \dots, (k_0^2 - k^2|_{x_n}), \underbrace{0, \dots, 0}_m, (k_0^2 - k^2|_{x_{n+m+1}}), \dots, (k_0^2 - k^2|_{x_{n+m+1}}) \right) \\
\mathbf{U1} &= \text{diag} \left(u_1 \nabla^2 h|_{x_1}, \dots, u_1 \nabla^2 h|_{x_{(n+m+1)}} \right) \\
\mathbf{U2} &= \text{diag} \left(u_2 (\nabla h)^2|_{x_1}, \dots, u_2 (\nabla h)^2|_{x_{(n+m+1)}} \right) \\
\hat{\mathbf{U}} &= \begin{pmatrix} \hat{\eta}_{1,1} & \cdots & \hat{\eta}_{1,(n+m+1)} \\ \vdots & \ddots & \vdots \\ \hat{\eta}_{(n+m+1),1} & \cdots & \hat{\eta}_{(n+m+1),(n+m+1)} \end{pmatrix} \\
\hat{\mathbf{Q}} &= \begin{pmatrix} \hat{q}_{1,1} & \cdots & \hat{q}_{1,(n+m+1)} \\ \vdots & \ddots & \vdots \\ \hat{q}_{(n+m),1} & \cdots & \hat{q}_{(n+m),(n+m+1)} \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
\mathbf{H}_i &= \begin{pmatrix} h_{1,1} & \cdots & h_{1,n} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ h_{(n+m+1),1} & \cdots & h_{(n+m+1),n} & 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix} \\
\mathbf{H}_o &= \begin{pmatrix} 0 & \cdots & 0 & h_{1,n+1} & \cdots & h_{1,n} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & h_{(n+m+1),(n+1)} & \cdots & h_{(n+m+1),(n+m)} & 0 & \cdots & 0 \end{pmatrix} \\
\mathbf{G}_i &= \begin{pmatrix} g_{1,1} & \cdots & g_{1,n} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ g_{(n+m+1),1} & \cdots & g_{(n+m+1),n} & 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix} \\
\mathbf{G}_o &= \begin{pmatrix} 0 & \cdots & 0 & g_{1,n+1} & \cdots & g_{1,n} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & g_{(n+m+1),(n+1)} & \cdots & g_{(n+m+1),(n+m)} & 0 & \cdots & 0 \end{pmatrix}
\end{aligned}$$

De todas las ecuaciones presentadas, solo quedan conocer los valores, de $\frac{\partial G}{\partial x}$, $\frac{\partial G}{\partial y}$ y $\nabla^2 G$. Dichos valores dependen de la profundidad del punto considerado, y de las pendientes y curvaturas en dicho punto. Las expresiones para calcular estos valores vienen facilitadas en los Apéndices B y C de [4].

5.4. Ecuación de Mild-Slope para Fondo Constante

El hecho de considerar un fondo constante implica una simplificación de la formulación expuesta en los apartados anteriores. En este apartado se obtendrá la solución a este problema por una vía alternativa a la expuesta anteriormente. Este problema es interesante para el estudio de dársenas o puertos donde el fondo es cuasiconstante e interesa conocer la agitación interior ante un oleaje entrante. Este apartado está basado en [3].

Se diferenciarán dos regiones, una interior, donde se estudiará la altura de ola, y otra exterior, desde donde proviene un potencial determinado. Así el potencial se define:

$$\phi = \begin{cases} \phi_w + \phi_s^e & \text{En la Región Exterior} \\ \phi_s^{(i)} & \text{En la Región Interior} \end{cases}. \quad (5.70)$$

Donde, ϕ_w es el potencial entrante y ϕ_s^e y $\phi_s^{(i)}$ son los potenciales dispersos de la región exterior e interior respectivamente.

Se le llamará Γ a la frontera que comparten ambas regiones. Las condiciones impuestas del problema son, además de las mencionadas en 5.3, 5.4 y 5.5, las siguientes:

$$\begin{aligned}
\phi_s^{(i)} &= \phi_w + \phi_s^{(e)} \quad \text{en } \Gamma \\
\frac{\partial \phi_s^{(i)}}{\partial n} &= \frac{\partial \phi_w}{\partial n} - \frac{\partial \phi_s^{(e)}}{\partial n'}
\end{aligned} \quad (5.71)$$

Donde n' es el vector normal de la región exterior con sentido hacia fuera de la misma. Por tanto, n y n' tienen sentidos opuestos.

La condición de reflexión parcial en los contornos es derivada de la condición de radiación 5.56 y tiene el siguiente aspecto:

$$\frac{\partial \phi}{\partial n} - \alpha k \phi = 0 \quad (5.72)$$

En la ecuación anterior, α es el coeficiente de transmisión, y, de la forma más simplificada, se calcula como sigue:

$$\alpha = i \frac{1 - K_r}{1 + K_r} \quad (5.73)$$

$$K_r \in [0, 1]$$

Donde K_r es el coeficiente de reflexión.

Para la solución del problema se supondrá que el potencial en un punto es provocado por una distribución de fuentes puntuales de ondas a lo largo del contorno Γ (para la región exterior) ó $\Gamma + S$ (para la región interior), donde S son los contornos interiores que no coinciden con los contornos de la región exterior. De esta forma, la expresión de los potenciales queda:

$$\phi_s^{(e)}(x) = \frac{1}{4\pi} \int_{\Gamma} f^{(e)}(\xi) G(x, \xi) dS \quad (5.74)$$

$$\phi_s^{(i)}(x) = \frac{1}{4\pi} \int_{\Gamma+S} f^{(i)}(\xi) G(x, \xi) dS$$

Donde $f^{(e)}$ y $f^{(i)}$ representan las funciones de distribución de intensidad de los potenciales exteriores e interiores respectivamente. x es el 'Punto Objetivo' y ξ es el 'Punto Fuente'. La función $G(x, \xi)$, en este caso, es:

$$G(x, \xi) = i\pi H_0(kr) \quad (5.75)$$

$$r = \|x - \xi\|$$

Las expresiones para la derivada del potencial son las siguientes:

$$\frac{\phi_s^{(e)}(x)}{\partial n'} = -\frac{1}{2} f^{(e)}(x) + \frac{1}{4\pi} \int_{\Gamma} f^{(e)}(\xi) \frac{\partial G(x, \xi)}{\partial n'} dS \quad (5.76)$$

$$\frac{\phi_s^{(i)}(x)}{\partial n} = -\frac{1}{2} f^{(i)}(x) + \frac{1}{4\pi} \int_{\Gamma+S} f^{(i)}(\xi) \frac{\partial G(x, \xi)}{\partial n} dS$$

Ya se tienen todas las herramientas para obtener el sistema de ecuaciones. De las ecuaciones 5.71 y 5.72 se tiene que:

$$\begin{aligned} & \frac{1}{4\pi} \int_{\Gamma+S} f^{(i)}(\xi) G(x, \xi) dS - \frac{1}{4\pi} \int_{\Gamma} f^{(e)}(\xi) G(x, \xi) dS = \phi_w(x) \quad \text{Para } x \in \Gamma \\ & -\frac{1}{2} f^{(i)}(x) + \frac{1}{4\pi} \int_{\Gamma+S} f^{(i)}(\xi) \frac{\partial G(x, \xi)}{\partial n} dS - \\ & -\frac{1}{2} f^{(e)}(x) - \frac{1}{4\pi} \int_{\Gamma} f^{(e)}(\xi) \frac{\partial G(x, \xi)}{\partial n} dS = \frac{\partial \phi_w}{\partial n}(x) \quad \text{Para } x \in \Gamma \quad (5.77) \\ & -\frac{1}{2} f^{(i)}(x) + \frac{1}{4\pi} \int_{\Gamma+S} f^{(i)}(\xi) \frac{\partial G(x, \xi)}{\partial n} dS - \alpha(x) \frac{k}{4\pi} \int_{\Gamma+S} f^{(i)}(\xi) G(x, \xi) dS = 0 \quad \text{Para } x \in S \\ & \frac{\partial G}{\partial n} = -i\pi k H_1(kr) \frac{n_x(x_x - \xi_x) + n_y(x_y - \xi_y)}{r} \end{aligned}$$

Si, a partir de las ecuaciones anteriores, se forma un sistema de ecuaciones lineales del tipo $AX = B$, el contenido de dichas matrices y vectores es el siguiente:

$$X_j = \left\{ \begin{array}{ll} f_j^{(i)} & \text{Para } j = 1, \dots, N1 \\ f_j^{(e)} & \text{Para } j = N1 + 1, \dots, N \end{array} \right\}. \quad (5.78)$$

$$B_i = \left\{ \begin{array}{ll} 0 & \text{Para } i = 1, \dots, NS \\ 4\pi\phi_w & \text{Para } i = NS + 1, \dots, N1 \\ 2\frac{\partial\phi_w}{\partial n} & \text{Para } i = N1 + 1, \dots, N \end{array} \right\}. \quad (5.79)$$

$$A_{ij} = \left\{ \begin{array}{ll} P_{ij} & \text{Para } i = 1, \dots, NS; j = 1, \dots, N1 \\ 0 & \text{Para } i = 1, \dots, NS; j = N1 + 1, \dots, N \\ G_{ij} & \text{Para } i = NS + 1, \dots, N1; j = 1, \dots, N1 \\ -G_{ij} & \text{Para } i = NS + 1, \dots, N1; j = N1 + 1, \dots, N \\ -\delta_{ij} + \frac{1}{2\pi}G_{ij}^n & \text{Para } i = N1 + 1, \dots, N; j = 1, \dots, N1 \\ -\delta_{ij} - \frac{1}{2\pi}G_{ij}^n & \text{Para } i = N1 + 1, \dots, N; j = N1 + 1, \dots, N \end{array} \right\}. \quad (5.80)$$

$$P_{ij} = -\delta_{ij} + \frac{1}{2\pi} [G_{ij}^n - k\alpha(x_i)G_{ij}] \quad (5.81)$$

Si se le llama NS al número de elementos en los que se divide el contorno S y $N\Gamma$ al número de elementos en los que se divide el contorno Γ , $N1 = NS + N\Gamma$ y $N = NS + 2N\Gamma$.

$$G_{ij} = \int G(x_i, \xi_j) dS_j$$

$$G_{ij}^n = \int \frac{\partial G(x_i, \xi_j)}{\partial n} dS_j$$

Para la evaluación de singularidades hay que tener en cuenta las igualdades expuestas en el capítulo presente y, además, que ocurren singularidades no solo cuando $i = j$, sino también en otros casos. Esto es debido a que los elementos del contorno Γ se repiten dos veces en el sistema de ecuaciones.

Una vez obtenida la solución del sistema de ecuaciones, esto es, $f_j^{(i)}$ y $f_j^{(e)}$, el potencial en cualquier punto del interior se consigue mediante la siguiente fórmula:

$$\begin{aligned} \phi_s^{(i)}(x) &= \sum_{j=1}^{NS+N\Gamma} f_j^{(i)} G(x, \xi_j) S_j \\ \phi_s^{(e)}(x) &= \sum_{j=1}^{N\Gamma} f_j^{(e)} G(x, \xi_j) S_j \end{aligned} \quad (5.82)$$

Donde S_j es la longitud del segmento j del contorno.

Finalmente solo queda por definir ϕ_w , de la siguiente forma:

$$\phi_w = A [\text{Cos}(k \cdot x \cdot \cos(\theta) + k \cdot y \cdot \sin(\theta)) + i \cdot \text{Sin}(k \cdot x \cdot \cos(\theta) + k \cdot y \cdot \sin(\theta))] \quad (5.83)$$

A = Altura de Ola

θ = Ángulo de incidencia del oleaje respecto el Eje X

De este modo, la solución obtenida mediante la ecuación 5.82 será directamente la elevación de la superficie compleja.

CAPÍTULO 6

Generalized Minimum Residual Method

6.1. Introducción

Como se ha explicado en capítulos anteriores, a diferencia de otros métodos como el Método de Elementos Finitos en el que la matriz de coeficientes que se obtiene es simétrica, lo que facilita mucho su resolución; la matriz de coeficientes resultante del sistema de ecuaciones que se obtiene al aplicar el método de los elementos de contorno es llena y no simétrica. El método más extendido de resolución de sistemas de ecuaciones generales es el método de 'Eliminación de Gauss', cuyo coste computacional es de aproximadamente $\frac{2}{3} \cdot N^3$ flops [41], siendo N el número de ecuaciones del sistema. Este coste resulta demasiado elevado cuando se trabaja con un sistema de un número elevado de ecuaciones¹. Por ello ha sido necesario buscar y aplicar un método de resolución especial.

El método general de los residuos mínimos o 'GMRES Method' fue propuesto por Saad and Schultz en 1986 [14] con el objetivo de resolver sistemas de ecuaciones en las que la matriz de coeficientes fuese simétrica (o no fuese Hermitiana en el caso de trabajar con números complejos) y 'llena'. A lo largo de este capítulo se explicará en que consiste dicho método y sus principales ventajas. Este capítulo contiene partes de [15] que es el documento que ha ayudado al autor a comprender este método.

6.2. Subespacios de Krylov

Los métodos basados en los subespacios de Krylov se han desarrollado para la resolución de grandes sistemas de ecuaciones lineales donde la matriz de coeficientes es no singular y de tipo SPARSE (como es este caso).

$$A \cdot x = b \tag{6.1}$$

Siendo A la matriz de coeficientes, x el vector de incógnitas y b el vector de términos independientes.

Los métodos basados en subespacios de Krylov tratan de encontrar un vector solución x_k que minimize el residuo: $r = b - A \cdot x$. Dada una primera aproximación de la solución, x_0 , y calculando su correspondiente vector de residuos r_0 , por iteraciones sucesivas se obtiene que:

$$x_1 = x_0 + r_0$$

$$x_2 = x_1 + r_1 = x_0 + r_0 + (b - A \cdot x_1) = x_0 + r_0 + (b - A \cdot x_0 - A \cdot r_0) = x_0 + 2r_0 - A \cdot r_0$$

$$x_3 = x_2 + r_2 = x_0 + 2r_0 + A \cdot r_0 - A^2 \cdot r_0$$

¹Para un sistema de ecuaciones de 30000 ecuaciones habría que realizar $18 \cdot 10^{12}$ flops, lo que equivale a aproximadamente 10 horas si se lograra una velocidad de 500 MFlops/s

⋮

$$x_k = x_{k-1} + r_{k-1}$$

Así, se define que una base del k -ésimo subespacio de Krylov cuya nomenclatura es $K^k(A, r_0)$, es:

$$K_k(A, r_0) = \{r_0, Ar_0, A^2r_0, \dots, A^{k-1}r_0\} \quad (6.2)$$

Se observa que la base del k -ésimo subespacio de Krylov está formada por k vectores de orden N , de esta forma, para la k -ésima iteración:

$$x_k \in x_0 + K_k(A, r_0) \quad (6.3)$$

En general, la base $\{r_0, Ar_0, A^2r_0, \dots, A^{k-1}r_0\}$ es una mala elección para el subespacio de Krylov, $K_k(A, r_0)$. Así pues, es necesario aplicar algún tipo de ortogonalización.

6.3. Técnicas de Proyección

Dado el sistema 6.1, tal que la matriz A es real y no singular, sean L y K dos subespacios de \mathbb{R}^n de dimensión k ; una técnica de proyección sobre el subespacio K y ortogonal a L , es un proceso para encontrar una solución \tilde{x} aproximada del sistema anterior, imponiendo la condición de que \tilde{x} pertenezca a K y que el nuevo vector residuo sea ortogonal a L . Esto es:

$$\text{Encontrar } \tilde{x} \in K, \text{ tal que } b - A \cdot \tilde{x} \perp L \quad (6.4)$$

Se redefine el problema. Sea x_0 una aproximación inicial, \tilde{x} debe estar en el espacio afín $x_0 + K$,

$$\text{Encontrar } \tilde{x} \in x_0 + K, \text{ tal que } b - A \cdot \tilde{x} \perp L \quad (6.5)$$

Expresando la solución aproximada por $\tilde{x} = x_0 + \delta$ y el vector residuo inicial es $r_0 = b - A \cdot x_0$; tendrá que cumplirse que:

$$b - A \cdot (x_0 + \delta) \perp L \rightarrow r_0 - A \cdot \delta \perp L \quad (6.6)$$

Es decir, la solución aproximada \tilde{x} de 6.1 queda definida como,

$$\tilde{x} = x_0 + \delta, \delta \in K \text{ tal que } \langle r_0 - A \cdot \delta, w \rangle = 0, \forall w \in L \quad (6.7)$$

La representación matricial de la condición de ortogonalidad del residuo, expresada en la ecuación 6.7, se hace de la siguiente forma: Sea $V = [v_1, v_2, \dots, v_k]$ la matriz $n \times k$, cuyos vectores columnas forman una base del subespacio K , y sea $W = [w_1, w_2, \dots, w_k]$ la matriz $n \times k$, cuyos vectores columnas forman una base del subespacio L . Si la solución aproximada de 6.1 se escribe como:

$$x = x_0 + V \cdot y \quad (6.8)$$

por 6.7,

$$\langle r_0 - A \cdot V \cdot y, w \rangle = 0, \forall w \in L \quad (6.9)$$

o lo que es lo mismo,

$$W^T \cdot A \cdot V \cdot y = W^T \cdot r_0 \quad (6.10)$$

Por tanto, si la matriz $n \times n$ $W^T \cdot A \cdot V$ es no singular, la solución aproximada vendrá dada por:

$$x = x_0 + V \cdot \left(W^T \cdot A \cdot V \right)^{-1} \cdot W^T \cdot r_0 \quad (6.11)$$

donde, si A es simétrica definida positiva y $L = K$; o bien si A es no singular y $L = A \cdot K$, entonces la matriz $W^T \cdot A \cdot V$ es no singular para las bases V y W de K y L respectivamente (ver [42]). El caso verdaderamente interesante en este proyecto es el segundo (A no singular y $L = A \cdot K$). La solución aproximada \tilde{x} es el resultado de un método de proyección oblicua sobre K , ortogonalmente a L , si y solo si, minimiza la norma euclídea del vector residuo $b - A \cdot x$ sobre $x \in x_0 + K$,

$$R(\tilde{x}) = \min R(x) \text{ con } x \in x_0 + K, \quad (6.12)$$

siendo,

$$R(\tilde{x}) = \|b - A \cdot x\| \quad (6.13)$$

La condición necesaria y suficiente para que el resultado de minimizar $R(x)$ sea \tilde{x} , es que $b - A \cdot \tilde{x}$ sea ortogonal a todos los vectores de la forma $v = A \cdot y$, donde $y \in K$, es decir:

$$(b - A \cdot \tilde{x}, v) = 0 \forall v \in A \cdot K \quad (6.14)$$

siendo la condición de Petrov-Galerkin que define la solución aproximada \tilde{x} .

6.4. Método de Arnoldi

Este método, introducido en 1951 [43], se basa en reducir una matriz densa a una matriz de Hessenberg cuyos autovalores, obtenidos en un número de pasos menor que n , proporcionan una aproximación exacta de algunos autovalores de la matriz original, a la vez que genera un sistema de vectores ortonormados.

Algoritmo 1 Algoritmo de Arnoldi

Elegir un vector $v_1 / \|v_1\| = 1$

for $j = 1, \dots, k$ **do**

for $i = 1, \dots, j$ **do**

$$H_{ij} = \langle A \cdot v_j, v_i \rangle$$

$$w_j = A \cdot v_j - H_{ij} \cdot v_i$$

end for

$$H_{j+1,j} = \|w_j\|$$

if $H_{j+1,j} = 0$ **then**

 Parar

end if

$$v_{j+1} = \frac{1}{H_{j+1,j}} \cdot w_j$$

end for

Los vectores v_1, v_2, \dots, v_k obtenidos en el algoritmo de Arnoldi forman una base ortogonal del subespacio de Krylov,

$$K_k(A; v_1) = \text{C.L.} \{v_1, Av_1, A^2v_1, \dots, A^{k-1}v_1\} \quad (6.15)$$

Estos vectores v_j ($j = 1, 2, \dots, k$) son ortonormados por construcción. Cada vector del subespacio $K_k(A; v_1)$ es de la forma $q_{j-1}(A)v_1$, donde q_{j-1} es un polinomio de grado $(j-1)$. Se puede demostrar que para $j = 1$, $v_1 = q_0(A) \cdot v_1$ y $q_0(A) = I$, siendo I la matriz identidad. Asumiendo que el resultado es válido para todo entero $i \leq j$ y considerando v_{j+1} , se tiene que:

$$h_{j+1,1}v_{j+1} = Av_j - \sum_{i=1}^j h_{ij}v_i = Aq_{j-1}(A)v_1 - \sum_{i=1}^j h_{ij}q_{i-1}(A)v_1 \quad (6.16)$$

Por tanto, v_{j+1} puede ser expresado como $q_j(A)v_1$, donde q_j es de grado j .

Llamando V_k a la matriz $n \times k$, cuyas columnas son los vectores v_j con $j = 1, \dots, k$; \overline{H}_k a la matriz $(k+1) \times k$ de Hasseberg, cuyas entradas no nulas h_{ij} son obtenidas en el algoritmo de Arnoldi 1; y H_k la matriz obtenida eliminando la última fila de \overline{H}_k , es inmediato comprobar que:

$$AV_k = V_k H_k + w_k e_k^T = V_{k+1} \overline{H}_k \quad (6.17)$$

$$V_k^T AV_k = H_k \quad (6.18)$$

En la práctica, el algoritmo básico de Arnoldi se sustituye por el Modificado Gram-Schmidt siendo éste último más eficiente.

Algoritmo 2 Algoritmo de Arnoldi Modificado Gram-Schmidt

Elegir un vector $v_1 / \|v_1\| = 1$

for $j = 1, \dots, k$ **do**

$w_j = Av_j$

for $i = 1, \dots, j$ **do**

$H_{ij} = \langle A \cdot v_j, v_i \rangle$

$w_j = A \cdot v_j - H_{ij} \cdot v_i$

end for

$H_{j+1,j} = \|w_j\|$

if $H_{j+1,j} = 0$ **then**

 Parar

end if

$v_{j+1} = \frac{1}{H_{j+1,j}} \cdot w_j$

end for

6.5. Método de Mínimo Residuo Generalizado (GMRES)

El GMRES es un método de proyección basado, como se ha dicho anteriormente, en que $K = K_k$ y $L = A \cdot K_k$, donde K_k es un subespacio de Krylov de dimensión k , lo que equivale a minimizar la norma del residuo. Así, el desarrollo del algoritmo consiste en encontrar un vector x de $x_0 + K_k$ tal que,

$$x = x_0 + V_k \cdot y \quad (6.19)$$

imponiendo la condición de mínimo para:

$$J(y) = \|b - A \cdot x\| \quad (6.20)$$

y como:

$$b - A \cdot x = b - A \cdot (x_0 + V_k y) = r_0 - AV_k y \quad (6.21)$$

Teniendo en cuenta la ecuación 6.17 y $v_1 = r_0 / \|r_0\|$ obtenido del algoritmo de Arnoldi y llamando $\beta = \|r_0\|$, entonces:

$$b - Ax = \beta v_1 - V_{k+1} \overline{H_k} y \quad (6.22)$$

pero, $v_1 = V_{k+1} e_1$, con $e_1 \in \mathfrak{R}^{k+1}$, por tanto

$$b - Ax = V_{k+1} (\beta e_1 - \overline{H_k} y) \quad (6.23)$$

y la ecuación 6.20 queda:

$$J(\mathbf{y}) = \|\mathbf{V}_{k+1}(\beta\mathbf{e}_1 - \overline{\mathbf{H}}_k\mathbf{y})\| \quad (6.24)$$

Como las columnas de la matriz \mathbf{V}_{k+1} son ortonormales por construcción, podemos simplificar la anterior expresión:

$$J(\mathbf{y}) = \|(\beta\mathbf{e}_1 - \overline{\mathbf{H}}_k\mathbf{y})\| \quad (6.25)$$

El algoritmo GMRES busca el único vector de $\mathbf{x}_0 + K_k$ que minimiza $J(\mathbf{y})$.

Algoritmo 3 Algoritmo GMRES

Aproximación inicial \mathbf{d} y $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$

Definir la matriz de $(k+1) \times k$ elementos $\overline{\mathbf{H}}_k = \{\mathbf{H}\}_{1 \leq i \leq k+1, 1 \leq j \leq k}$. Poner $\overline{\mathbf{H}}_k = 0$

for $j = 1, \dots, k$ **do**

$\mathbf{w}_j = \mathbf{A}\mathbf{v}_j$

for $i = 1, \dots, j$ **do**

$\mathbf{H}_{ij} = \langle \mathbf{A} \cdot \mathbf{v}_j, \mathbf{v}_i \rangle$

$\mathbf{w}_j = \mathbf{A} \cdot \mathbf{v}_j - \mathbf{H}_{ij} \cdot \mathbf{v}_i$

end for

$\mathbf{H}_{j+1,j} = \|\mathbf{w}_j\|$

if $\mathbf{H}_{j+1,j} = 0$ **then**

 Poner $k = j$ y parar

end if

$\mathbf{v}_{j+1} = \frac{1}{\mathbf{H}_{j+1,j}}\mathbf{w}_j$

end for

Hallar \mathbf{y}_k que minimiza $\|(\beta\mathbf{e}_1 - \overline{\mathbf{H}}_k\mathbf{y})\|$;

Determinar $\mathbf{x}_0 + \mathbf{V}_k\mathbf{y}_k$ siendo $\mathbf{V}_k = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k]$

Calcular $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$

Este algoritmo resulta impracticable cuando k es muy grande, ya que conlleva un elevado coste computacional y de almacenamiento, por lo se procede a usar las técnicas *restart* y de truncamiento.

Algoritmo 4 Algoritmo Restarted GMRES

1: Aproximación inicial \mathbf{x}_0 , $\beta = \|\mathbf{r}_0\|$, y $\mathbf{v}_1 = \mathbf{r}_0/\beta$;

2: Generar la base de Arnoldi y la matriz $\overline{\mathbf{H}}_k$ usando el algoritmo de Arnoldi;

3: Iniciar con \mathbf{v}_1 ;

4: Hallar \mathbf{v}_1 ;

5: Hallar \mathbf{y}_k que minimiza $\|(\beta\mathbf{e}_1 - \overline{\mathbf{H}}_k\mathbf{y})\|$ y $\mathbf{x}_k = \mathbf{x}_0 + \mathbf{V}_k\mathbf{y}_k$;

6: Si se satisface entonces parar. En caso contrario volver a definir $\mathbf{x}_0 = \mathbf{x}_k$ y volver al paso 1.

Con este algoritmo se realizan las mismas operaciones descritas en 3 con la salvedad de que solo se ejecutan m pasos, por tanto las matrices y vectores definidos son más pequeños, con el consiguiente ahorro computacional y de memoria. Cuando se llega al paso m se calcula el vector solución

(aunque el vector residuos r tenga una norma mayor que la tolerancia impuesta) y se vuelve a empezar de nuevo con este vector como primera aproximación. Este algoritmo es el que finalmente se ha introducido en el código creado con unas pequeñas modificaciones que se verán en el siguiente apartado.

6.6. GMRES Implementado y sus particularidades

Como se indicaba anteriormente, se ha implementado el algoritmo 'Restarted GMRES de m pasos' con una pequeña modificación para que pudiese aplicarse a sistemas de ecuaciones con números complejos. El código implementado, obtenido de [44], es el siguiente:

Algoritmo 5 Restarted GMRES implementado

```

1:  $r = b - Ax_0$  para una aproximación inicial  $x_0$ 
2:  $x = x_0$ 
3: for  $j = 1, 2, \dots$  do
4:    $\beta = \|r\|_2$ 
5:    $v_1 = r/\beta$ 
6:    $\hat{b} = \beta e_1$ 
7:   for  $i = 1, 2, \dots, m$  do
8:      $w = Av_i$ 
9:     for  $k = 1, \dots, i$  do
10:       $h_{k,i} = v_k^* w$ 
11:       $w = w - h_{k,i} v_k$ 
12:    end for
13:     $h_{i+1,i} = \|w\|_2$ 
14:     $v_{i+1} = w/h_{i+1,i}$ 
15:     $r_{1,i} = h_{1,i}$ 
16:    for  $k = 2, \dots, i$  do
17:       $\gamma = c_{k-1} r_{k-1,i} + \overline{s_{k-1}} h_{k,i}$ 
18:       $r_{k,i} = -s_{k-1} r_{k-1,i} + c_{k-1} h_{k,i}$ 
19:       $r_{k-1,i} = \gamma$ 
20:    end for
21:     $\delta = \sqrt{|r_{i,i}|^2 + |h_{i+1,i}|^2}$ 
22:    if  $|r_{i,i}| < |h_{i+1,i}|$  then
23:       $\mu = r_{i,i}/h_{i+1,i}$ 
24:       $\tau = \overline{\mu}/|\mu|$ 
25:    else
26:       $\mu = h_{i+1,i}/r_{i,i}$ 
27:       $\tau = \mu/|\mu|$ 
28:    end if

```

```

29:    $c_i = |r_{i,i}| / \delta$ 
30:    $s_i = |h_{i+1,i}| \tau / \delta$ 
31:    $r_{i,i} = c_i r_{i,i} + \bar{s}_i h_{i+1,i}$ 
32:    $\hat{b}_{i+1} = -s_i \hat{b}_i$ 
33:    $\hat{b}_i = c_i \hat{b}_i$ 
34:    $\rho = |\hat{b}_{i+1}|$  ( $= \|b - Ax_{(j-1)m+i}\|_2$ )
35:   if  $\rho \leq \text{tol}$  then
36:      $n_r = i$ 
37:      $y_{n_r} = \hat{b}_{n_r} / r_{n_r, n_r}$ 
38:     Ir a línea 36
39:   end if
40: end for
41:  $n_r = m$ 
42:  $y_{n_r} = \hat{b}_{n_r} / r_{n_r, n_r}$ 
43: for  $k = n_r - 1, \dots, 1$  do
44:    $y_k = (\hat{b}_{n_k} - \sum_{i=k+1}^{n_r} r_{k,i} y_i) / r_{k,k}$ 
45: end for
46:  $x = x + \sum_{i=1}^{n_r} y_i v_i$ 
47: if  $\rho \leq \text{tol}$  then
48:   Solución Encontrada  $\rightarrow$  Salir
49: end if
50:  $r = b - Ax$ 
51: end for

```

De este código, se deben aclarar varias líneas: En primer lugar, v^* y \bar{a} significan vector transpuesto conjugado y complejo conjugado respectivamente. Como se puede apreciar, el algoritmo anterior está preparado para números complejos. Se debe recordar que cuando se trata con números complejos, el producto vectorial se modifica de manera que:

$$\langle a, b \rangle = \sum \bar{a}_i \cdot b_i \quad (6.26)$$

De igual manera, la norma $|v|$ y la norma al cuadrado $\|v\|_2$ de un vector se formulan de la siguiente forma:

$$|v| = \sqrt{\sum \bar{v}_i \cdot v_i} \quad (6.27)$$

$$\|v\|_2 = \sum \bar{v}_i \cdot v_i \quad (6.28)$$

En la línea 6 aparece e_1 , que es un vector de m elementos, todos ellos 0 menos el primer término que es un 1. Por otra parte, se debe aclarar la correspondiente parte del algoritmo que va desde la línea 17 hasta la línea 33. Como se ha mencionado en 4, se debe hallar y_k que minimiza $\|(\beta e_1 - \bar{H}_k y)\|$. Para conseguir este objetivo, lo más habitual es aplicar transformaciones al vector $\beta e_1 - \bar{H}_k y$ que no modifiquen la norma del resultado y que transformen la matriz de Hessemberg en una matriz triangular superior. Para ello se aplican rotaciones matriciales en un plano [45].

Por ello, ha sido necesario buscar vías para acelerar estos cálculos que serán explicadas en los siguientes capítulos.

En la siguiente gráfica (Figura 6.1) se observa una comparativa del tiempo de computación entre el método GMRES implementado y el Método de Eliminación de Gauss para sistemas de diferentes tamaños, advirtiéndose como el método de eliminación de Gauss es proporcional a N^3 mientras que el método del GMRES es proporcional a N^2 .

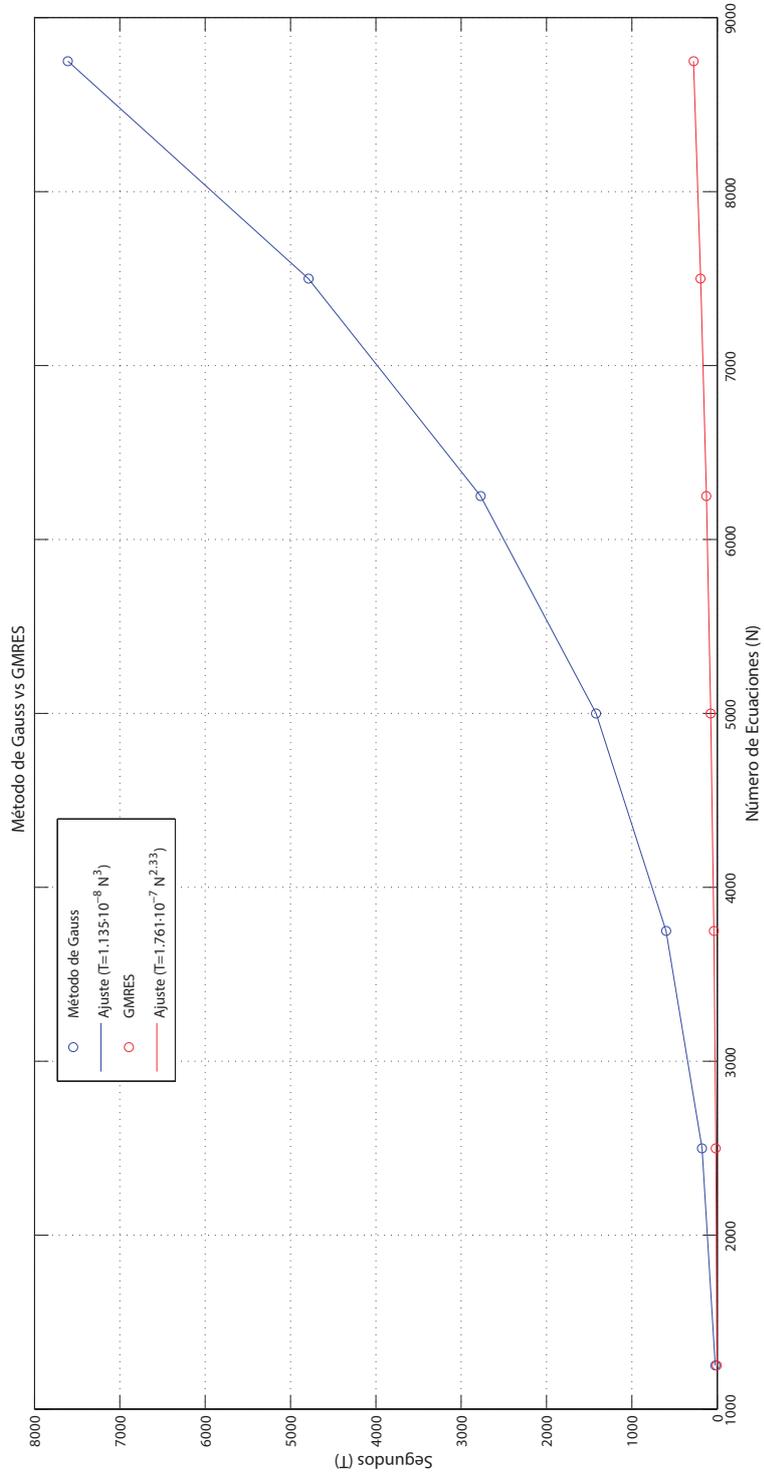


Figura 6.1: Comparativa entre el Método de Eliminación de Gauss y el GMRES

CAPÍTULO 7

CUDA y la Programación Paralela

7.1. Introducción

Hasta hace algunos años, el software siempre había sido escrito para el procesamiento en serie. El objetivo del código era ser ejecutado en una máquina con una sola CPU, donde el problema se descomponía en series discretas de instrucciones que se ejecutaban una tras otra, es decir, sólo se podía ejecutar una solución a la vez.

Sin embargo, poco a poco se ha ido implantando la programación paralela debido a las limitaciones físicas que impiden el aumento de la frecuencia de los procesadores simples. A diferencia de la anterior, en la programación paralela el problema se divide en varias partes con el objetivo de ser ejecutadas en distintas unidades de procesamiento.

De esta forma, un código basado en diferentes estructuras que no dependen unas de otras pero que se deben ejecutar para llevar a cabo un fin, pueden resolverse en un tiempo mucho menor debido a que cada unidad de procesamiento puede ocuparse de una estructura independiente. De igual modo, gracias a la programación paralela, si el código en cuestión implica 'loops' en los que cada paso

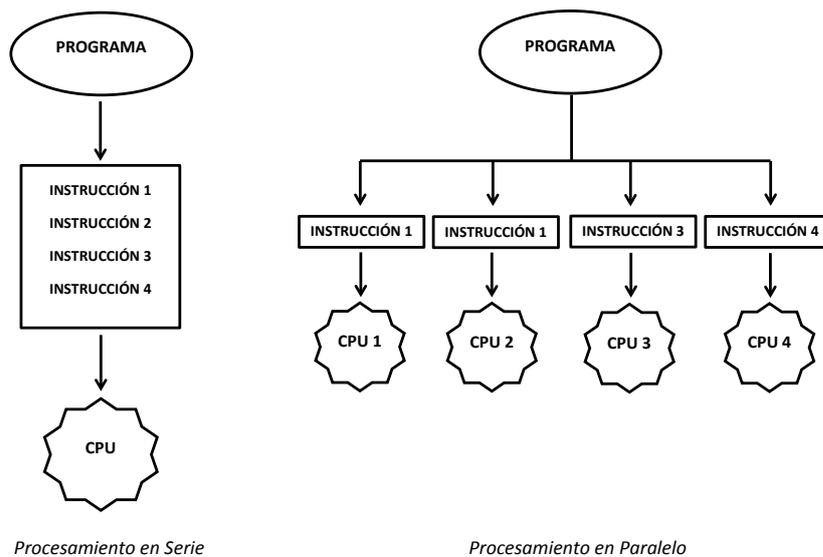


Figura 7.1: Procesamiento en Serie vs Procesamiento en Paralelo

del mismo no dependa del paso anterior, se pueden dividir los pasos del 'loop' entre los distintos núcleos disponibles, disminuyendo así, al igual que antes, el tiempo de computación. Sin embargo, como se verá a lo largo de este capítulo, no todo es tan sencillo como aparenta. Muestra de ello es que en la gran mayoría de los casos la computación paralela resulta más lenta que la computación secuencial.[16].

Cuando se habla de computación paralela, se puede hablar básicamente de dos arquitecturas principales:

1. Las Arquitecturas Multi-Núcleo; que se basan simplemente en la computación del código mediante varios CPU en una misma máquina (actualmente se encuentran fácilmente multiprocesadores con 8 núcleos, sin embargo se prevé que en pocos años se llegue a 128 núcleos).
2. Las Arquitecturas basadas en Placas Gráficas (GPUs). Este capítulo está dedicado a este tipo de arquitectura, más concretamente a la programación de tarjetas gráficas NVIDIA mediante CUDA (Compute Unified Device Architecture).

7.2. ¿Qué es una GPU?

La unidad de procesamiento gráfico o GPU (acrónimo del inglés Graphics Processing Unit) es un coprocesador dedicado al procesamiento de gráficos u operaciones de coma flotante, para aligerar la carga de trabajo del procesador central en aplicaciones como los videojuegos y/o aplicaciones 3D interactivas. De esta forma, mientras gran parte de lo relacionado con los gráficos se procesa en la GPU, la unidad central de procesamiento (CPU) puede dedicarse a otro tipo de cálculos (como la inteligencia artificial o los cálculos mecánicos en el caso de los videojuegos).

Básicamente, una GPU se encarga de transformar una gran cantidad de elementos virtuales en la imagen plana que aparece en nuestras pantallas. Se recomienda la lectura del capítulo 6.2.2 de [16], en el que se puede profundizar en la labor que desempeña una GPU.

Dicho todo esto, ¿qué diferencia una CPU de una GPU?. Entre otras muchas, las diferencias básicas son:

- La Frecuencia del Reloj: en las CPU actuales, la frecuencia oscila alrededor de 3 GHz ($3 \cdot 10^9$ cambios de estado por segundo) mientras que en las GPU la frecuencia de reloj oscila entre los 800 MHz ($8 \cdot 10^8$ cambios de estado por segundo). Es decir, la relación es de casi 4 veces más del lado de la CPU
- El Número de Núcleos de Procesamiento: mientras que las CPU actuales tienen a lo sumo 6 u 8 núcleos, una GPU puede llegar a tener varios miles de núcleos de procesamiento.
- La Arquitectura: normalmente, las CPU tienen una arquitectura tipo Von Neumann mientras que la arquitectura de las GPU se basa en el Modelo Circulante. Este modelo facilita el procesamiento en paralelo, y la gran segmentación que posee la GPU para sus tareas.

Como ejemplo ilustrativo, mediante unos simples cálculos, se puede saber el potencial de un procesador gráfico. Suponiendo que es necesario obtener una secuencia de video (ya sea por un videojuego o simplemente por el uso de un programa de diseño 3D) con una resolución FullHD. Cada imagen tiene 1920x1080 píxeles, además los monitores actuales tienen una frecuencia de muestreo de unos 60 Hz, por lo tanto, cada segundo, la tarjeta gráfica debe calcular 60 imágenes de 1920x1080 píxeles, lo que hace un total de:

$$1920 \cdot 1080 = 2073600 \text{ píxeles} \approx 2 \text{ Mpx/imagen} \quad (7.1)$$

$$2073600 \cdot 60 \approx 1.24 \cdot 10^8 \text{ píxeles calculados/segundo} \quad (7.2)$$

7.3. CUDA y sus Principales Elementos

CUDA (acrónimo en inglés de Compute Unified Device Architecture) son una serie de herramientas basadas en el lenguaje C concebidas por NVIDIA para otorgar al usuario la posibilidad de codificar sus códigos, de tal manera que puedan usar las GPU para el procesamiento en paralelo. NVIDIA dispone de productos especialmente creados para trabajar con CUDA, como las tarjetas gráficas TESLA con 32 GB de memoria RAM dedicada que pueden alcanzar más de 500 Gflops ($5 \cdot 10^{11}$ operaciones de coma flotante por segundo). A continuación se describen brevemente los elementos más importantes de la computación sobre GPUs:

- Hilos: un hilo o thread es la secuencia más pequeña de instrucciones que puede ser manejada independientemente por un sistema de computación.
- Madejas: una madeja o warp es la cantidad mínima de datos que se puede ejecutar a la vez por un multiprocesador CUDA. Una madeja se compone de 32 hilos.
- Bloques: los bloques son un conjunto de hilos, normalmente se definen los bloques como una matriz bidimensional o tridimensional de hilos, como se detalla más adelante. El tamaño máximo de hilos que puede albergar un bloque depende de la GPU que se utilice pero generalmente es de 512 hilos, aunque las nuevas GPU pueden trabajar con bloques de hasta 1024 hilos. Cada bloque es ejecutado por un multiprocesador de la GPU.
- Mallas: una malla o grid es un conjunto de bloques que, al igual que antes, se definen como una matriz bidimensional o tridimensional de hilos. El número máximo de bloques que puede albergar una malla es de 65535 bloques por lo general.
- Kernel: se le llama kernel al conjunto de instrucciones que deben llevar a cabo cada hilo de una malla definida.

En lo referente a los tipos de memorias que usa una GPU, para la implementación de CUDA en este proyecto se han utilizado 2 tipos de memoria como se verá en las próximas páginas:

- Memoria Global: es la memoria a la que tienen acceso todos los hilos de todos los bloques que son lanzados por un kernel. Es de mayor tamaño pero con el inconveniente de que los accesos de lectura y escritura a la misma son lentos.
- Memoria Compartida: es una memoria de baja capacidad y ultra rápida situada dentro de cada multiprocesador de la GPU. Dicha memoria es solo accesible desde cada bloque, y a su vez cada bloque tiene distinta información almacenada en su memoria compartida. De esta forma, un hilo no puede acceder a la memoria compartida de un bloque que no sea el suyo.

Para la implementación de CUDA se deben usar una serie de rutinas y funciones que se explican a continuación.

- `cudaMemGetInfo`: es una función que nos permite conocer la cantidad total de memoria disponible en nuestra GPU.
- `cudaMalloc`: es la función que nos permite reservar una cantidad dada (en bytes) de memoria RAM de nuestra GPU.

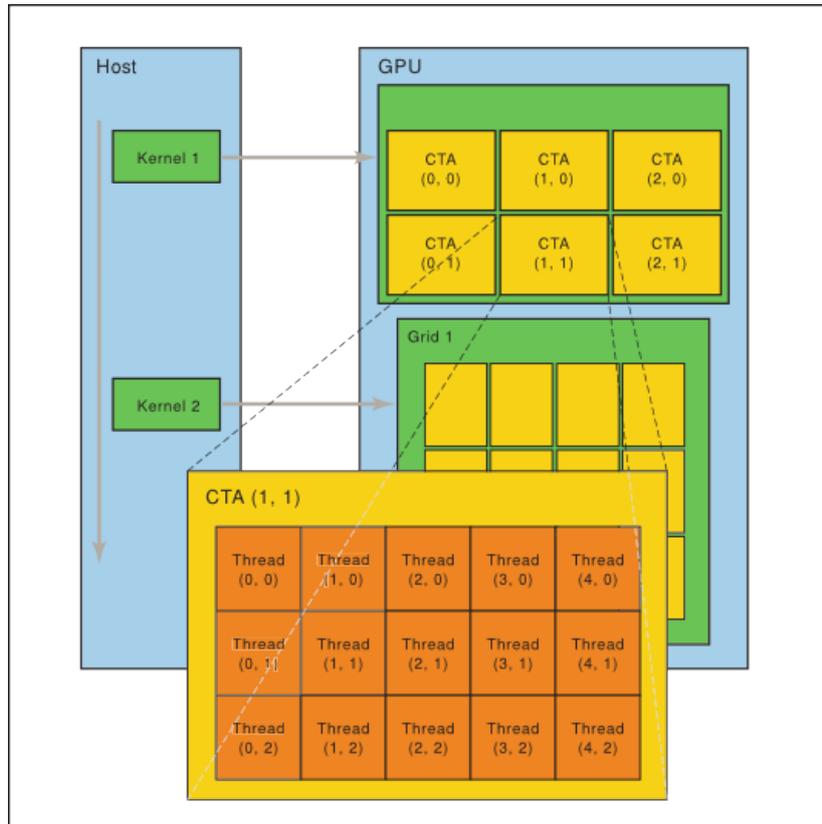


Figura 7.2: Treads, Blocks, Grid y Kernel (Fuente: www.nvidia.com)

- `cudaMemcpy`: es una función que nos permite copiar datos de la memoria de la GPU a la memoria de la máquina y viceversa.
- `cudaFree`: es la función que nos permite liberar un espacio de memoria reservado en la memoria RAM de nuestra GPU.

Una vez mencionadas las principales funciones y características de CUDA y para una mayor claridad, se explicará la estructura de un programa escrito en CUDA con la ayuda de un código sencillo a modo de ejemplo:

```
using namespace std;
#include <cstdlib>
#include <iostream>
#include <math.h>
#include <string.h>
#include <cmath>

__global__ void sumacuda (double *Agpu, double *Bgpu, double *Cgpu, int N) {

    const int idx = (blockIdx.x * blockDim.x) + threadIdx.x;
    const int idy = (blockIdx.y * blockDim.y) + threadIdx.y;
    const int tid = ((gridDim.x * blockDim.x) * idy) + idx;
    if (tid < N) {
        Cgpu[tid] = Agpu[tid] + Bgpu[tid];
    }
}
```

```

};
}

void sumarm(double *A, double *B, double *C, int n) {

    double *Agpu;
    double *Bgpu;
    double *Cgpu;

    cudaMalloc((void **)&Agpu, n * sizeof(double));
    cudaMemcpy(Agpu, A, n*sizeof(double), cudaMemcpyHostToDevice);
    cudaMalloc((void **)&Bgpu, n * sizeof(double));
    cudaMemcpy(Bgpu, B, n*sizeof(double), cudaMemcpyHostToDevice);
    cudaMalloc((void **)&Cgpu, n * sizeof(double));

    const int nT=512;
    dim3 dimGrid((n+nT-1)/nT);

    sumacuda <<<dimGrid,nT>>> (Agpu,Bgpu,Cgpu,n);

    cudaMemcpy(&C[0], Cgpu, n*sizeof(double), cudaMemcpyDeviceToHost);

    cudaFree(Agpu);
    cudaFree(Bgpu);
    cudaFree(Cgpu);
    return;
};

```

Este pequeño programa define una función en la CPU que toma tres vectores y llama a la GPU para que los sume. En el caso en el que se tengan matrices, lo ideal es almacenarlas mediante un puntero simple¹ y no mediante uno doble, ya que el puntero simple reserva un espacio de memoria secuencial mientras que el puntero doble reserva 'filas' no consecutivas de memoria. La consecuencia inmediata de esto es una disminución en la velocidad de lectura y escritura en la memoria y por lo tanto, la ralentización del programa. Este principio es aplicable tanto a la memoria RAM de la CPU como a la memoria global de la GPU.

En dicho código, después de una pequeña cabecera, se define en primer lugar un kernel con las instrucciones que ejecutará cada hilos lanzado a la GPU. Los kernel se definen en el lenguaje C como una función cualquiera salvo por el prefijo `__global__`. Esta función no tiene parámetro de salida (void); y como entrada, se definen tres vectores de números reales de doble precisión que serán la matriz A, la matriz B y la matriz resultado C. También se define un entero N que es el número de elementos de la matriz.

Como se ha podido verificar en la Figura 7.2, los hilos tienen una numeración² independiente en cada bloque, por ello, para que cada hilo acceda a un elemento diferente de los punteros, se definen las tres primeras líneas del kernel que le dan a cada hilo una numeración global. Las funciones que se utilizan para este objetivo son:

- `threadIdx.x`: devuelve, según la numeración local de cada bloque, el número de hilo en la dirección X.
- `threadIdx.y`: devuelve, según la numeración local de cada bloque, el número de hilo en la dirección Y. (el primer elemento tiene índice 0).

¹Almacenar la matriz en un vector de dimensión NxN, una fila detrás de otra, siendo N la dimensión de una matriz cuadrada

²Tanto la numeración de los hilos como la de los bloques empieza (como es habitual en C) por 0 para el primer elemento.

- `blockIdx.x`: devuelve el número de bloque, dentro del grid definido, según la dirección X.
- `blockIdx.y`: devuelve el número de bloque, dentro del grid definido, según la dirección Y.
- `blockDim.x`: devuelve el total de hilos (en la dirección X) lanzados dentro de cada bloque.
- `blockDim.y`: devuelve el total de hilos (en la dirección Y) lanzados dentro de cada bloque.
- `gridDim.x`: devuelve el total de bloques lanzados, dentro del grid definido, según la dirección X.

Una vez que cada hilo tiene almacenado un entero con su numeración global, se impondrá una condición mediante la cual se asegura que solamente ejecuten la suma los hilos cuya numeración sea menor que la dimensión del vector. De otra forma, habría hilos que accederían a espacios de memoria no reservados con el consiguiente error. Como se explica más adelante, no se pueden lanzar bloques con distinto número de hilos, por lo que, en la mayoría de los casos se lanzarán más hilos que elementos contenidos en la matriz, de aquí este pequeño paso tan importante.

Por último, cada hilo que pase la criba impuesta leerá un elemento del vector A, leerá el correspondiente elemento del vector B, los sumará y lo almacenará en el vector C.

En lo correspondiente a la rutina que se ejecuta en la CPU, en primer lugar hay que tener en cuenta que todo parámetro que se lance en el kernel hay que definirlo previamente en CPU, de aquí el motivo por el que se definen los tres punteros a los vectores que se almacenarán en la GPU. El paso siguiente es reservar espacio en la memoria de la GPU y copiar los vectores a la misma gracias a las funciones definidas anteriormente (`cudaMalloc` y `cudaMemcpy`). Este paso es muy delicado, ya que la copia de datos supone el mayor porcentaje de tiempo de ejecución del programa y es clave para que un programa con CUDA implementado llegue a ser más lento que un programa secuencial. La prueba está en que para matrices pequeñas (5x5, 10x10 etc. . .) este programa puede resultar hasta varias decenas de veces más lento que uno secuencial.

A continuación, en el programa de ejemplo, se definen el número de hilos por bloque. Es muy aconsejable que este número sea múltiplo de una madeja, es decir, que sea divisible por 32. Consecuentemente, aprovechando que en la mayoría de las GPU el número máximo de hilos por bloque es 512 y que éste número cumple con la recomendación, se define éste por defecto. Para definir el número de bloques en el grid se debe hacer la operación que se encuentra en el código para asegurar de que se lanzan un total de hilos superior al número de elementos. Por consiguiente, si se tuviera por ejemplo un vector de 513 elementos, la operación resultaría:

$$\frac{513 + 512 - 1}{512} = 2 \quad (7.3)$$

Se lanzarían dos bloques de 512 hilos cada uno. ¿Qué pasaría si se tuviera un vector de 512 elementos?:

$$\frac{512 + 512 - 1}{512} = 1.998 \quad (7.4)$$

En este caso, al estar almacenado en un entero, los decimales se truncan, por lo que se lanza un bloque de 512 hilos. De igual modo, pasaría para cualquier vector con $N \leq 512$, siendo N el número de elementos.

Definido ya el número de bloques y de hilos por bloque, se lanza el kernel. Para ello se indica el nombre del mismo seguido de <<< N° Bloques por Grid, N° Treads por Bloque >>>³ y de las variables de entrada que son requeridas.

Ya ejecutado el kernel, el vector o matriz resultado está en la memoria de la GPU, por lo que, debe copiarse a la memoria RAM para su posterior manejo. Para ello, al igual que antes, se recurre al comando `cudaMemcpy`.

Por último se debe liberar la memoria reservada para cada variable en la GPU con el comando `cudaFree`.

7.4. Implementación de CUDA en el Software

En este proyecto se ha utilizado CUDA, para acelerar la rutina que resuelve el sistema matricial de ecuaciones ($A \cdot X = B$) mediante el método de mínimos cuadrados (GMRES) explicado anteriormente. Más concretamente, se ha implementado para reducir el tiempo de ejecución de las multiplicaciones matriz-vector, ya que este tipo de operaciones se realiza muy a menudo en el *solver* programado. El código desarrollado presenta algunas optimizaciones adicionales que serán explicadas a lo largo del apartado.

Como es sabido, una operación matricial de una matriz A de dimensiones $N \times N$ por un vector v de dimensiones $N \times 1$ da como resultado otro vector a de dimensiones $N \times 1$ cuyos elementos vienen determinados por la siguiente expresión:

$$a_i = \sum_{j=1}^N A_{ij} \cdot v_j \quad (7.5)$$

Para implementar este algoritmo en CUDA han surgido varias cuestiones a resolver que se enumeran a continuación:

1. Se debe tener en cuenta que las dimensiones de la matriz pueden ser mayores a 512 elementos por fila, luego no se puede definir un bloque por fila y/o un hilo para cada elemento como en el código propuesto anteriormente.
2. Una matriz cuadrada de orden N de números complejos de doble precisión ocupa un espacio en memoria de $N^2 \cdot 16$ bytes⁴, es por esto que debe tenerse un especial cuidado, ya que si la GPU disponible tiene una memoria dedicada de 2GB por ejemplo, el orden máximo de la matriz que se puede introducir será de:

$$\sqrt{\frac{2 \text{ GB} \cdot 1024^3 \text{ Bytes/GB}}{16 \text{ Bytes/N}^\circ \text{ Complejo}}} \approx 11585 \quad (7.6)$$

Este resultado no es del todo correcto ya que en la memoria de la GPU también se debe almacenar el vector ' v ' por el que se multiplica la matriz y el vector resultado. Además, nunca se tiene la totalidad de la memoria de la GPU disponible. Por ello el problema se agrava. Así, cuando

³Esta notación es solo en el caso de que definamos bloques y grid unidimensionales, en caso contrario debemos ayudarnos del tipo de variable '`dim3`' ver [16].

⁴Un número real en doble precisión ocupa en memoria 8 bytes, por lo tanto un número complejo (en el que se deben almacenar 2 números de doble precisión) ocupa 16 bytes.

se quiera multiplicar una matriz de orden superior al indicado anteriormente, se debe dar un tratamiento especial.

3. Se debe minimizar el acceso a la memoria global de la GPU para disminuir el tiempo de ejecución.

Para el primer problema, se pueden plantear dos soluciones o incluso una mezcla de ambas. La primera solución es definir un grid bidimensional en el que se tenga $\frac{N+512-1}{512}$ columnas de bloques y N filas de bloques; de esta forma se seguiría asignando una sola operación a cada hilo. Otra solución es seguir con el grid unidimensional y que cada hilo de cada bloque computase varios elementos de una misma fila, de tal forma que el salto entre elemento y elemento sería igual al número de hilos por bloque. La solución adoptada en este caso ha sido una combinación de ambas. Por un lado se ha asignado más de un bloque por fila de la matriz⁵ y por otro a cada hilos se le ha asignado el cómputo de más de un elemento de la matriz⁶. Con este procedimiento se ha observado que, dependiendo del tamaño de la matriz, el algoritmo es más rápido si se le asigna un número de bloques por fila u otro. Así, se ha observado que para matrices pequeñas, el algoritmo tarda menos cuando se definen pocos bloques por fila (en torno a 1 o 2 bloques por fila para matrices de orden 1500) mientras que para matrices más grandes (en torno a orden 15000) el programa se ejecuta más rápido para 8 bloques que para 6 ó 10. Queda evidenciado que el número idóneo de bloques por fila depende del orden de la matriz a introducir.

En cuanto a la segunda cuestión, gracias a la función 'cudaMemGetInfo' nombrada anteriormente, se puede conocer la memoria de la GPU disponible. A esta memoria se le debe restar un valor igual a $2 \cdot N \cdot 16$ bytes, que es lo que ocupa el vector 'v' por el que se debe multiplicar la matriz y el vector resultado de esa operación. La memoria restante es la disponible para almacenar la matriz. En el caso de que el espacio que ocupa dicha matriz sea mayor que la memoria restante, el procedimiento a seguir ha sido dividir la matriz en varios bloques de K filas de tal manera que el número de copias de la memoria RAM a la memoria de la GPU sea lo más pequeño posible. Dado que se tiene almacenado el vector 'v' y el vector resultado en la GPU, simplemente los pasos a seguir son copiar bloques de filas de la matriz A a la GPU e ir calculando poco a poco el vector resultado.

El tercer problema se ha solventado declarando en la 'Memoria Compartida' un vector de T números complejos (siendo T el número de hilos por bloque), de tal forma que cada hilo de cada bloque guarda sus operaciones en la 'Memoria Compartida' con la correspondiente reducción de acceso a la 'Memoria Global' que ello conlleva. Una vez realizadas todas las operaciones habría que sumar todos los elementos del vector declarado en la 'Memoria Compartida' de cada bloque. Para ello, se ha implementado un sencillo pero efectivo algoritmo de reducción (ver [16]). Una vez realizado esto y teniendo almacenado un valor por bloque, si se ha definido varios bloques por fila, el siguiente paso es sumar los valores de todos estos bloques para obtener uno de los elementos del vector resultado.

7.5. Conclusiones

Durante la etapa de aprendizaje de CUDA, se ha percibido que esta herramienta es muy potente pero se debe tener una tarjeta gráfica medianamente potente para que los programas optimizados con CUDA obtengan ventaja respecto a los programas no optimizados. El gran 'Talón de Aquiles' de CUDA es la copia de datos a la memoria de la GPU. Tanto es así que en todas las pruebas realizadas

⁵Se ha programado una pequeña función que asigna un máximo de 8 bloques por fila (o menor según la dimensión de la matriz).

⁶Nótese que debe ponerse un 'filtro' para que los hilos no den más saltos de lo necesario e intenten leer espacios de memoria no ocupados por elementos de la matriz.

en un portátil con una GPU 'NVIDIA GEFORCE GT 630M' con 2GB de memoria dedicada y un procesador 'Intel Core i7', a lo sumo se ha llegado a igualar el tiempo de ejecución de la CPU con una ejecución en serie, resultando en la mayoría de los casos más lento el ejecutar el programa con CUDA (ejecución paralela) que en ejecución en serie. Esto se debe a que para un número pequeño de operaciones (multiplicar una matriz de orden 10 por un vector), la copia de los datos resulta nefasta (hablando en términos de tiempo de ejecución) y CUDA pierde toda su ventaja⁷. Por otro lado, para un número de operaciones grande (multiplicar una matriz de orden 20000 por un vector), la memoria dedicada de nuestra GPU suele resultar insuficiente, teniendo que recurrir a varias copias a la memoria de la GPU y llegando a la misma conclusión de antes.

Sin embargo, en pruebas realizadas en un servidor con una CPU Intel(R) Core(TM) i7-3930K CPU @ 3.20GHz y una tarjeta gráfica GeForce GTX 760, se ha apreciado una extraordinaria mejora de los programas que implementaban CUDA. A modo de ejemplo, en la figura 7.3 se puede observar la diferencia de tiempo de computación entre un programa que calcula una multiplicación matriz-vector de números complejos en la CPU de forma secuencial y otro que lo hace de forma paralela con CUDA.

Por último, también se presenta en la figura 7.4 una comparativa entre el método GMRES (expuesto en el capítulo anterior) y el mismo método optimizado con CUDA, dicha gráfica compara la resolución de un sistema N ecuaciones.

⁷A modo de ejemplo, en un pequeño programa que multiplicaba escalarmente dos vectores de dimensión 1000, el código para CPU en serie llegaba a tardar 40 veces menos que el código de ejecución escrito para CUDA.

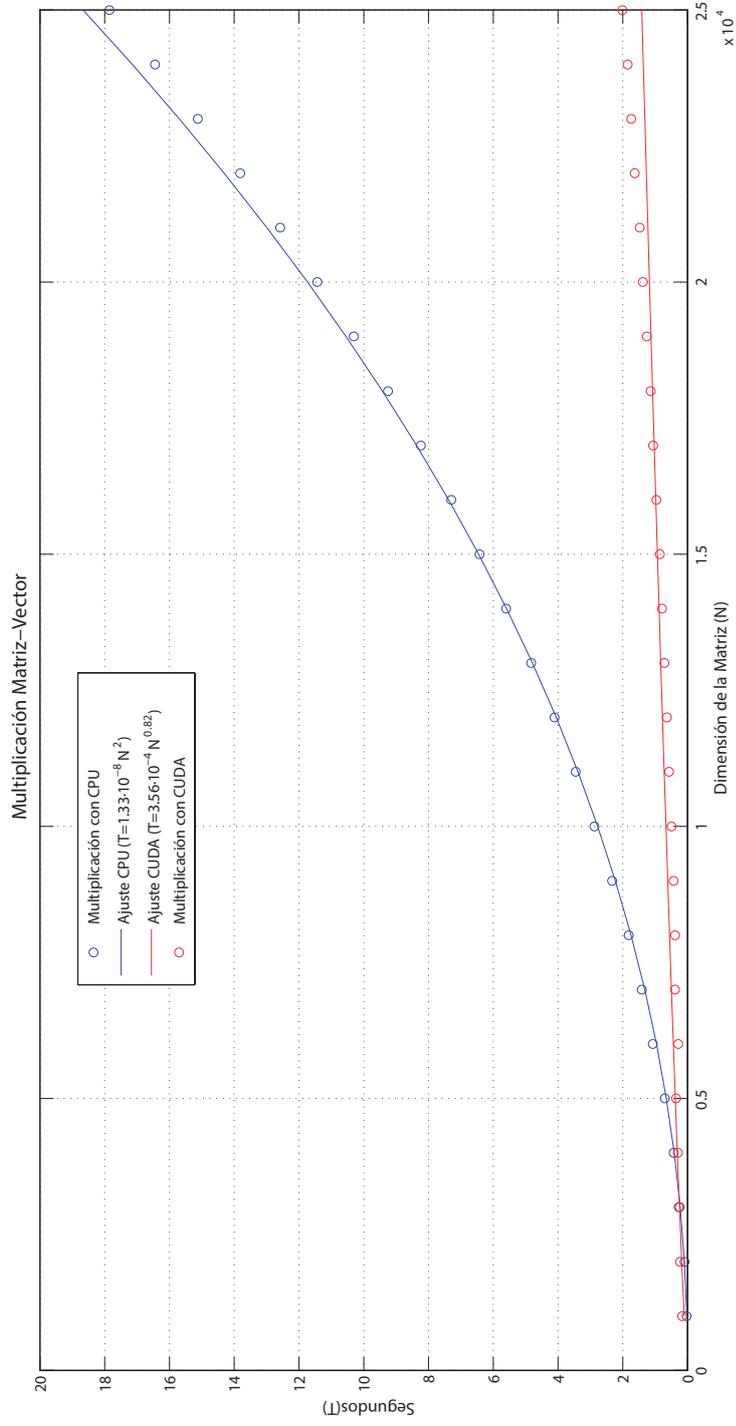


Figura 7.3: Comparativa entre un programa con CUDA implementado y un Programa Secuencial al ejecutar una Multiplicación Matriz-Vector

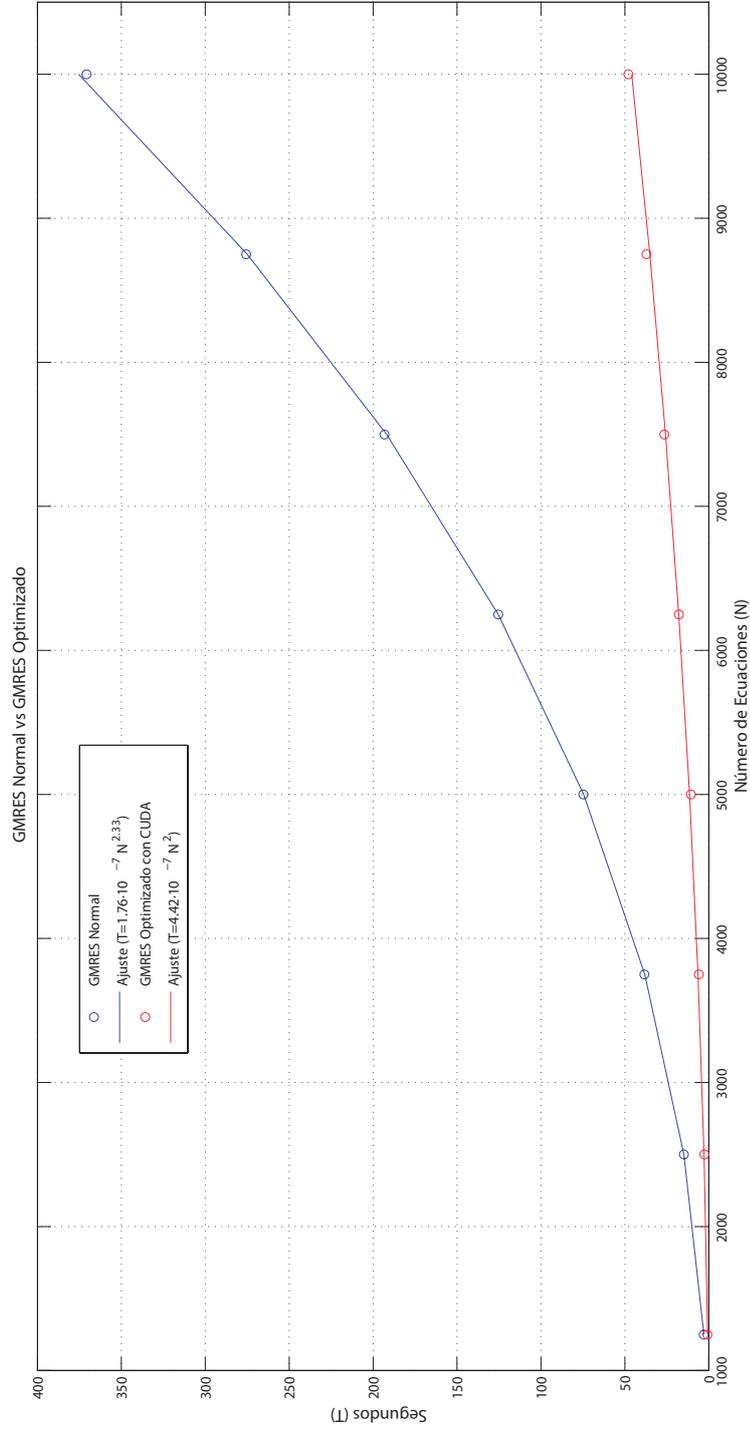


Figura 7.4: Comparativa entre el método GMRES normal y el método GMRES optimizado con CUDA

Parte II

Programas Desarrollados y Pruebas Numéricas de Validación

CAPÍTULO 8

Preproceso y Postproceso

8.1. Introducción

El Preproceso de cualquier programa de resolución numérico conlleva tanto la creación de la geometría de estudio como el mallado de la misma. Por otra parte, el Postproceso abarca toda la metodología utilizada para el visionado del 'output' del algoritmo y la obtención de resultados, gráficas y tablas. En este proyecto, para la realización de estas dos tareas, se ha utilizado el software de licencia libre 'SALOME', cuya información viene detallada en su dominio WEB¹. Este software incorpora un módulo para la creación de CAD o geometrías y un módulo para el mallado de las mismas, entre otros. Finalmente, para el Postproceso, se usará un software, también de libre licencia, llamado 'ParaView', que se puede instalar de forma independiente o junto al paquete SALOME. Estos tres módulos van a ser brevemente explicados en los siguientes apartados.

Bien es cierto que, los módulos del Preproceso solo han sido utilizados para los programas que requerían mallas y geometrías complicadas. Para los casos más sencillos, generalmente casos 2D, la generación de la malla se hace en el propio programa como se podrá ver en los siguientes capítulos.

8.2. Módulo Geometry

En la figura 8.1 se puede apreciar una captura de pantalla del inicio del módulo. Se observa que hay bastantes herramientas para crear tanto objetos 2D como 3D. Además, en la pestaña 'Operations' se pueden encontrar todo tipo de funciones para cortar, fusionar o copiar objetos, entre otras muchas. En la figura 8.2 se muestran algunas de las geometrías sencillas que se pueden hacer con este software. Adicionalmente, con este módulo se pueden crear grupos de sólidos, caras, o aristas dentro de un mismo elemento. Esta característica es de vital importancia ya que pueden constituirse grupos en función de la condición que se le debe asignar a cada contorno. En los próximos capítulos, cuando se expliquen las geometrías de estudio, se profundizará más en la composición de elementos con este módulo.

8.3. Módulo MESH

Este módulo tiene la misión de crear mallas a partir de las geometrías resultantes del módulo Geometry. La figura 8.3 muestra una captura de pantalla del inicio del módulo.

El módulo Mesh cuenta con repertorio bastante extenso de algoritmos de mallado; como el mallado por tetraedros y mallado por hexaedros para mallas 3D, mallado por triángulos y mallado por paralelogramos para mallas 2D, el algoritmo Netgen etc . . . Además se pueden construir submallas si el usuario quisiese tener un tamaño de malla más fino en unas zonas y más grueso en otras. En la figura 8.4 se aprecia el mallado del toroide creado en el módulo Mesh y mostrado en la figura 8.2. Los grupos creados en el módulo Geometry se pueden heredar en la malla y, por consiguiente, exportar para ser leídos por un programa externo.

¹<http://www.salome-platform.org/>

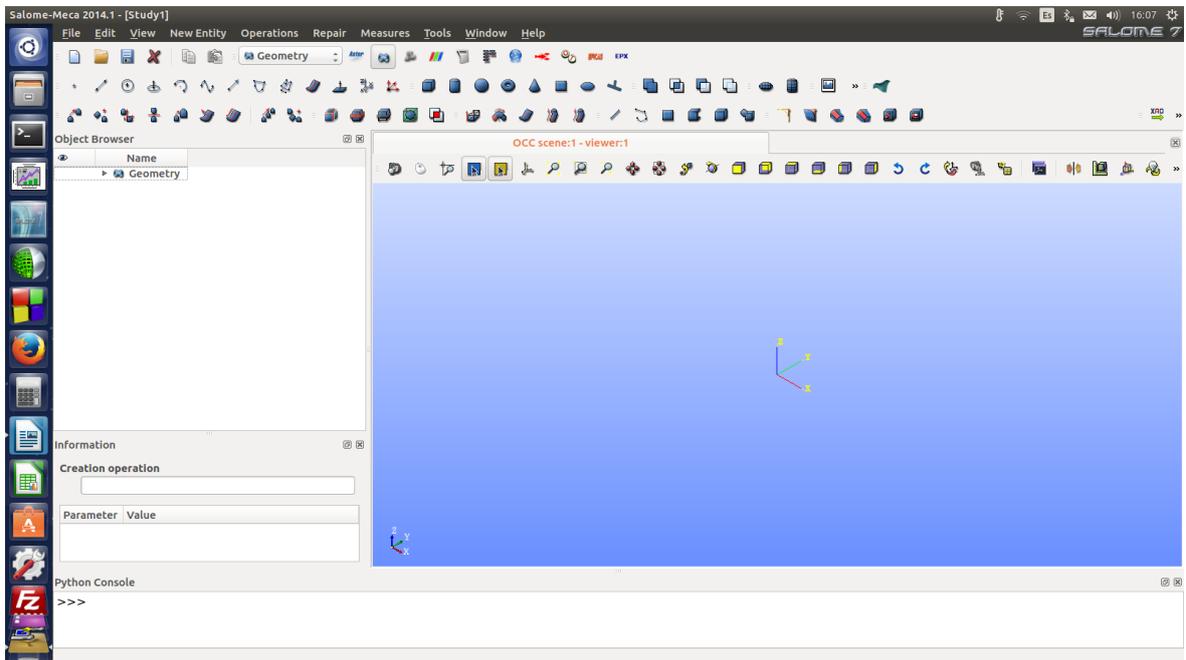


Figura 8.1: Captura de pantalla del inicio del Módulo Geometry

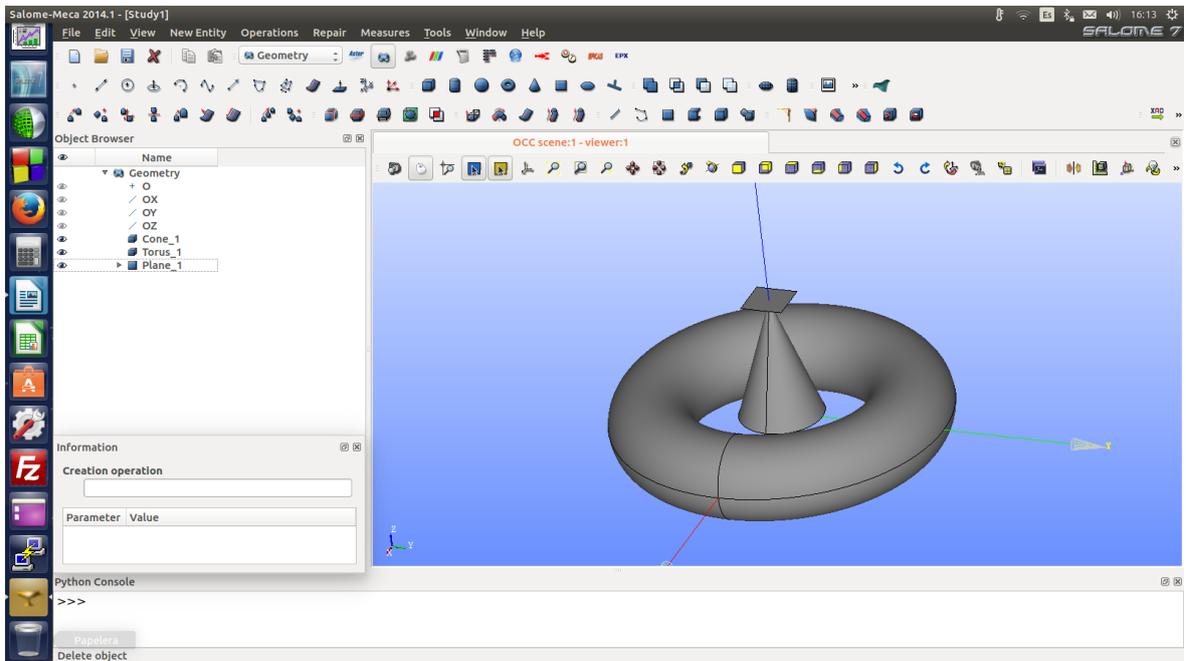


Figura 8.2: Captura de pantalla de figuras hechas con el Módulo Geometry

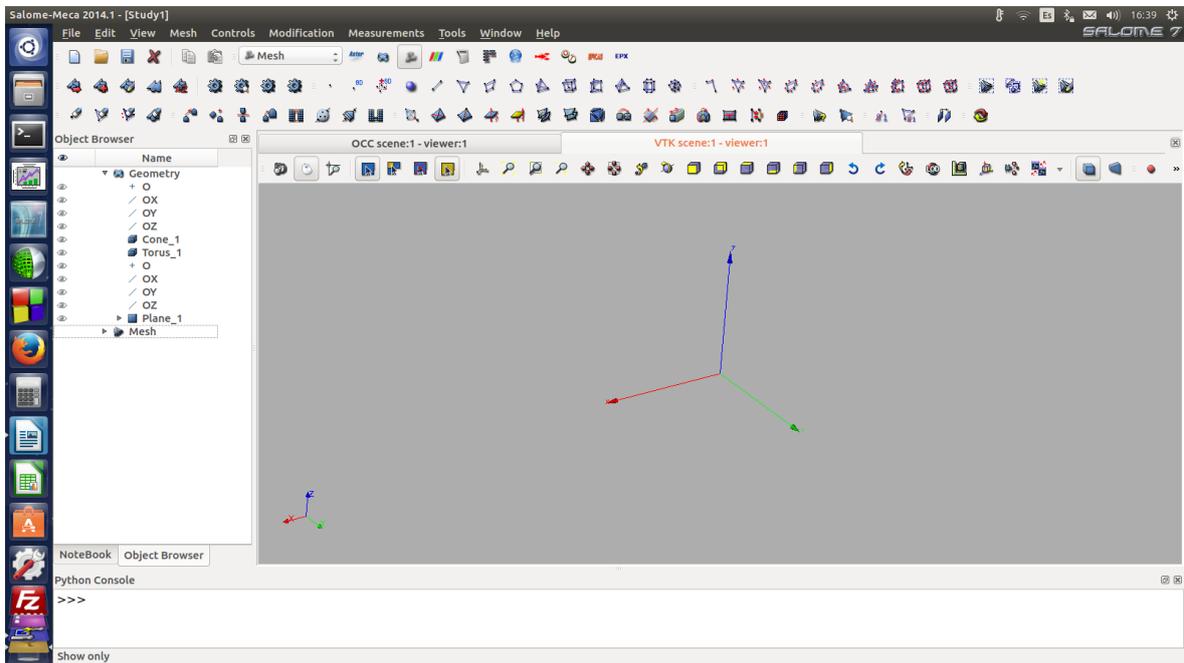


Figura 8.3: Captura de pantalla del inicio del Módulo MESH

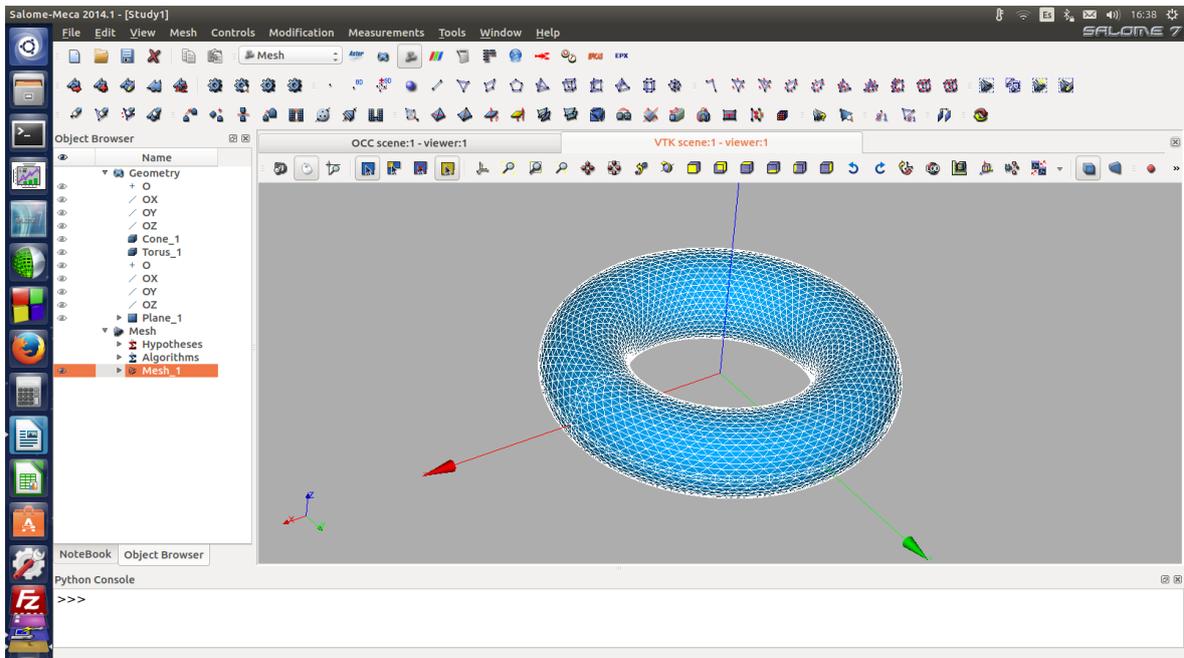


Figura 8.4: Captura de pantalla de la malla creada a partir de un toroide

8.4. Exportación de la Malla

Una vez creada la malla deseada con la ayuda del Módulo MESH, es necesario encontrar una vía para que dicha malla pueda ser leída por el programa implementado, que resolverá un problema en el dominio introducido. En este proyecto se ha usado el formato de archivo 'UNV', cuya estructura de datos se presenta a continuación:

- **Cabecera:**

Donde se muestran los datos del sistema de unidades así como de los ejes cartesianos. A continuación se muestra un ejemplo:

```

-1
164
  1 SI: Meter (newton)      2
1.0000000000000000E+0    1.0000000000000000E+0    1.0000000000000000E+0
2.7314999999999998E+2
-1
-1
2420
  1
SMESH_Mesh
  1      0      0
Global Cartesian Coordinate System
1.0000000000000000E+0    0.0000000000000000E+0    0.0000000000000000E+0
0.0000000000000000E+0    1.0000000000000000E+0    0.0000000000000000E+0
0.0000000000000000E+0    0.0000000000000000E+0    1.0000000000000000E+0
0.0000000000000000E+0    0.0000000000000000E+0    0.0000000000000000E+0

```

- **Datos de los Nodos**

A continuación se muestra un ejemplo:

```

  1      1      1      11
0.0000000000000000E+00    3.0000000000000000E+02    0.0000000000000000E+00
  2      1      1      11
0.0000000000000000E+00    0.0000000000000000E+00    0.0000000000000000E+00
  3      1      1      11
2.0000000000000000E+02    3.0000000000000000E+02    0.0000000000000000E+00
  4      1      1      11
2.0000000000000000E+02    0.0000000000000000E+00    0.0000000000000000E+00
  5      1      1      11
2.0000000000000000E+02    0.0000000000000000E+00    4.0000000000000000E+02

```

Como se puede apreciar, por cada nodo hay dos líneas, en la primera aparece la numeración del nodo y en la segunda sus coordenadas.

- **Datos de los Elementos**

En este apartado se definen tanto los tetraedros como los triángulos y los bordes. A continuación se muestra un ejemplo:

```

34      11      2      1      7      2
0       1      1
31      8
35      11      2      1      7      2
0       1      1
8       32

```

36	11	2	1	7	2
0	1	1			
32	6				
37	41	2	1	7	3
11	33	1			
38	41	2	1	7	3
10	33	35			
39	41	2	1	7	3
33	11	3			
40	41	2	1	7	3
34	2	9			

Dependiendo del elemento, puede haber dos o más filas para definirlo. El primer dígito es la numeración del elemento, y el último indica el número de nodos que lo componen. En el caso de ser 2, se estaría definiendo un borde, si fuese 3; se estaría definiendo un triángulo; y si fuese 4, se estaría definiendo un tetraedro. En la siguientes líneas constan los números identificativos de los nodos que forman el elemento en cuestión.

■ Grupos de Elementos

Por último se definen los grupos de elementos formados en el Módulo Geometry e importados al Módulo MESH. A continuación se muestra un ejemplo:

base	1	0	0	0	0	0	0	16
	8	37	0	0	8	38	0	0
	8	39	0	0	8	40	0	0
	8	41	0	0	8	42	0	0
	8	43	0	0	8	44	0	0
	8	45	0	0	8	46	0	0
	8	47	0	0	8	48	0	0
	8	49	0	0	8	50	0	0
	8	51	0	0	8	52	0	0
	2	0	0	0	0	0	0	128

En el ejemplo anterior, se define un grupo llamado 'base' que tiene 16 elementos (este dato consta en la primera fila). Posteriormente, a la derecha de cada 8, se nombra uno de los elementos que compone el grupo. Así, se definen 2 elementos por fila. La última fila empieza por un 2 y su último elemento es un 128; esto indica que en las próximas filas se va a definir el segundo grupo que está formado por 128 elementos.

La exportación se ha llevado a cabo mediante una función del Módulo Mesh (explicado anteriormente) y la lectura del archivo se ha completado gracias a las funciones 'ifstream' de C++.

8.5. Módulo ParaView

ParaView es un software multiplataforma de licencia libre para el análisis y visionado de experimentos numéricos, en la figura 8.5 se puede ver una captura de pantalla del programa. ParaView dispone de múltiples herramientas para el visionado de datos en una malla, como son secciones por planos o representación de datos a lo largo de una línea. Con este software también es posible un análisis a lo largo del tiempo, introduciendo archivos de datos distintos para diferentes instantes de tiempo. Toda la información relacionada con este programa está disponible en su dominio WEB². En la figura 8.6 se muestran algunos de los resultados que se pueden obtener con ParaView.

²<http://www.paraview.org/>

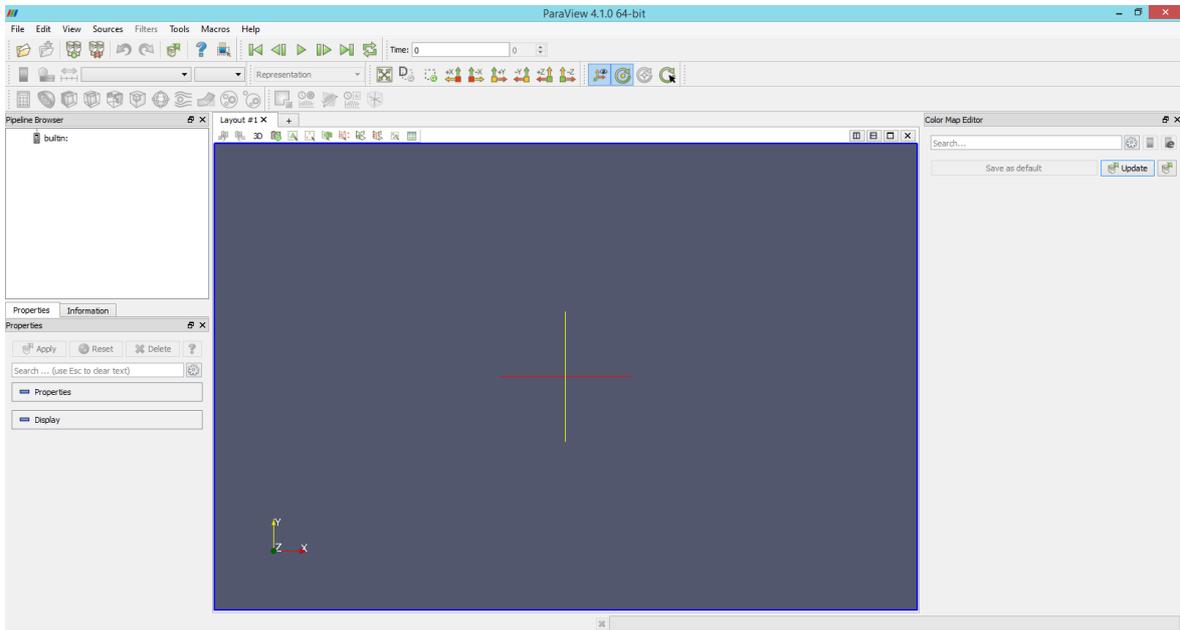


Figura 8.5: Captura de pantalla del inicio de ParaView

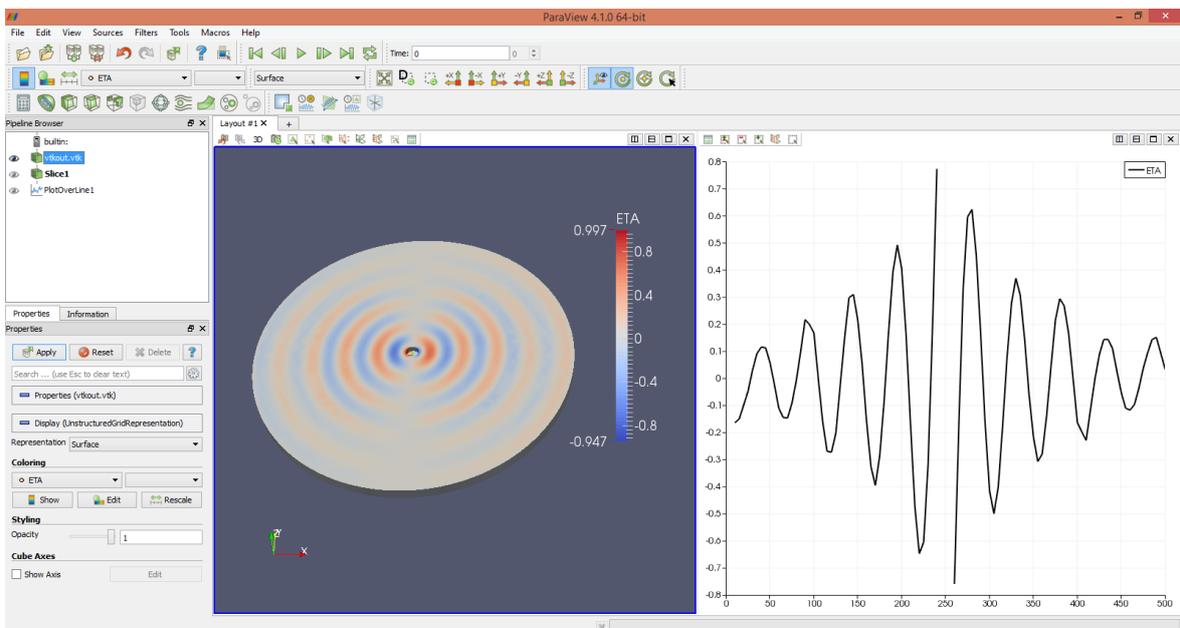


Figura 8.6: Captura de pantalla de resultados obtenidos con ParaView

8.6. Exportación de Resultados para el PostProceso

Los resultados numéricos obtenidos del algoritmo programado deben reflejarse en un archivo que sea compatible con ParaView para hacer posible el visionado y tratamiento de dichos datos.

ParaView trabaja con un formato llamado 'VTK' que tiene una estructura singular y que se expondrá brevemente en este apartado. Para más información, se puede hacer uso de una pequeña guía disponible en la dirección WEB <http://www.vtk.org/VTK/img/file-formats.pdf> . Las partes del archivo principales son las siguientes:

■ Cabecera

```
# vtk DataFile Version 3.0
BEM3D vtk-output
ASCII
```

En dicha cabecera aparece la versión compatible con la que el archivo debe ser leído, el nombre del archivo y, por último, el formato en el que está escrito (que puede ser en formato ASCII o binario).

■ Tipo de Geometría

```
DATASET UNSTRUCTURED_GRID
```

En esta línea debe estar la palabra 'DATASET' seguida del tipo de geometría que se va a definir. Ésta puede ser 'Structured Points', 'Structured Grid', 'Rectilinear Grid', 'Unstructured Grid' o 'Field'. Cada tipo tiene sus particularidades y marcan la forma de componer el resto del archivo. En este proyecto solo se ha utilizado el tipo 'Unstructured Grid', que sirve para definir combinaciones arbitrarias de todos los tipos de elementos posibles.

■ Puntos

```
POINTS 4729 DOUBLE
250 -6.12323e-14 0
249.8 10.0024 0
249.2 19.9888 0
248.2 29.9432 0
246.804 39.8496 0
245.012 49.6922 0
242.827 59.4552 0
```

En la primera línea debe estar la palabra 'POINTS' seguida del número de puntos que se van a declarar y de la precisión de las coordenadas de cada punto. A continuación debe haber tantas líneas como puntos se quieran declarar y, en cada línea, las tres coordenadas de cada punto.

■ Elementos

```
CELLS 23667 110767
3 0 1 314
3 1 2 315
3 3 4 316
3 5 6 317
4 3870 3724 3830 3827
4 523 2840 954 2763
4 1398 3870 3831 1349
```

En la primera línea debe estar la palabra 'CELLS' seguida del número de elementos que se van a definir y del número de datos que serán leídos³. A continuación debe constar, para cada

³Si, por ejemplo, se van a declarar x triángulos y y tetraedros, el segundo número debe tener un valor de $(4 \cdot x + 5 \cdot y)$

elemento, todos los nodos que lo forman. Se comprueba que el primer dígito de cada línea indica la cuantía de nodos que se deben leer y, seguido de este, la numeración (al primer nodo se le asigna el número 0) de los nodos.

- **Tipo de Elementos**

```
CELL_TYPES 23667
5
5
5
5
10
10
10
```

En la primera línea debe estar la palabra 'CELL_TYPES' seguida del número de elementos declarados anteriormente. Las siguientes líneas solo tienen una cifra que indica el tipo de elemento que se ha definido. Por ejemplo, para vértices aislados se debe poner 1, para líneas 3, para triángulos 5, para tetraedros 10 y para pirámides de cinco vértices el 14, entre otros.

- **Datos de los Puntos y Elementos**

```
POINT_DATA 4729
SCALARS Abs(u) double
LOOKUP_TABLE default
0.534979
0.537372
0.540219
VECTORS Abs(q) double
0.0290332 0.000231671 4.05181e-05
0.0305182 0.0010377 5.41419e-05
0.0297389 0.0023489 2.40024e-05
CELL_DATA 23667
SCALARS Abs(u) double
LOOKUP_TABLE default
0.542572
0.550018
0.550323
```

En este apartado se asignan valores (ya sean escalares, vectores o tensores) a los puntos y/o elementos. Para declarar los valores de los puntos se debe poner 'POINT_DATA' seguido del número de puntos; y para declarar los valores de los elementos se debe poner 'CELL_DATA' seguido, al igual que antes, del número de elementos. La segunda línea debe empezar con la palabra 'SCALARS' si se van a declarar valores escalares o con la palabra 'VECTORS' si se van a declarar vectores. Seguido de esta primera palabra aparece el nombre del dato y finalmente la precisión de los valores. La línea que comienza con la palabra 'LOOKUP_TABLE' sirve para indicar la paleta de color con la que se quiere representar los datos. Después de estas tres primeras líneas, se deben escribir los valores o los componentes del vector para cada punto o elemento.

Para crear esta estructura y guardarla en un archivo se han utilizado las funciones 'ofstream' de C++.

CAPÍTULO 9

Test de Verificación para BEM2D

9.1. Descripción

BEM2D ha sido el primer programa desarrollado por el autor de este proyecto como toma de contacto. El programa resuelve problemas gobernados por la ecuación de Laplace mediante el método de los elementos de contorno (BEM) en cualquier dominio y con condiciones de contorno reales y no mixtas (la variable compleja y las condiciones mixtas se introdujeron en BEM3D).

El programa necesita dos archivos de entrada llamados 'datosentrada.dat' y 'datosreq.dat'. El archivo 'datosentrada.dat', es el que contiene los datos del contorno y en el que debe haber un primer dato entero, siendo el número de elementos en los que se divide el contorno. A continuación deben aparecer los datos de cada elemento (una línea de datos por elemento), de la siguiente forma:

- N° ENTERO: número de elemento.
- N° FLOTANTE: coordenada x del nodo (punto central del elemento).
- N° FLOTANTE: coordenada y del nodo.
- N° FLOTANTE: coordenada x del primer punto del elemento al que pertenece el nodo (X1).
- N° FLOTANTE: coordenada y del primer punto del elemento al que pertenece el nodo (Y1).
- N° FLOTANTE: coordenada x del segundo punto del elemento al que pertenece el nodo (X2).
- N° FLOTANTE: coordenada y del segundo punto del elemento al que pertenece el nodo (Y2).
- N° ENTERO: que debe ser 0 si el dato que conocemos en el elemento es de "u" y 1 si es de "q".
- N° FLOTANTE: el valor de "u" o "q" según el caso de cada elemento.

Es muy importante que estos nodos estén numerados de forma antihoraria si el contorno rodea al dominio y de forma horaria si el contorno está rodeado por el dominio (huecos, agujeros etc...). Por otro lado, el archivo 'datosreq.dat', que es el archivo que contiene los datos de los puntos internos donde se pretende conocer los valores del potencial y de sus derivadas, tiene una estructura similar. En primer lugar, debe haber un único número entero en la primera línea que debe ser el número de puntos internos, a continuación deben estar los datos de cada punto de la siguiente forma:

- N° ENTERO: número del nodo interno
- N° FLOTANTE: coordenada x del nodo interno
- N° FLOTANTE: coordenada y del nodo interno

El programa resuelve el problema mediante el procedimiento expuesto en el Capítulo 4. Primero calcula las matrices **G** y **H**, a continuación forma el sistema $\mathbf{AX} = \mathbf{B}$ y finalmente lo resuelve. El procedimiento que se sigue para la evaluación de singularidades es el mismo que el explicado en el apartado 4.6. Las integrales no singulares se han resuelto mediante una cuadratura de cuatro puntos. Las coordenadas ξ de los puntos de integración (ver figura 4.7) y los valores de los pesos están representados en la tabla 9.1.

Quedando la evaluación de las integrales:

Nodo	Coordenada ξ_k	w_k
1	-0.861136311594053	0.347854845137454
2	-0.339981043584856	0.652145154862546
3	0.339981043584856	0.652145154862546
4	0.861136311594053	0.347854845137454

Cuadro 9.1: Coordenadas de los Puntos de Integración y Funciones Peso para el caso bidimensional

$$\begin{aligned}
G_{ij} &= \int_{\Gamma_j} u^* d\Gamma = \frac{L}{4\pi} \sum_{k=1}^4 \log\left(\frac{1}{R_k}\right) \cdot w_k \\
H_{ij} &= \int_{\Gamma_j} q^* d\Gamma = \frac{L}{4\pi} \sum_{k=1}^4 \frac{-1}{R_k^2} \cdot \left[n_x^j \cdot (XCO_k - x_i) + n_y^j \cdot (YCO_k - y_i) \right] \cdot w_k \\
R_k &= \sqrt{(XCO_k - x_i)^2 + (YCO_k - y_i)^2} \\
XCO_k &= \frac{X2 - X1}{2} \cdot \xi_k + \frac{X2 + X1}{2} \\
YCO_k &= \frac{Y2 - Y1}{2} \cdot \xi_k + \frac{Y2 + Y1}{2} \\
L &= \text{Longitud del Elemento} = \sqrt{(X2 - X1)^2 + (Y2 - Y1)^2}
\end{aligned} \tag{9.1}$$

Finalmente, para la visualización de resultados, se ha implementado un algoritmo que crea un archivo en formato 'VTK', como se ha explicado en el capítulo anterior.

9.2. Prueba de Validación

Para realizar la validación del programa se ha resuelto, en un dominio cuadrado de 200 unidades de longitud, el siguiente problema de condiciones de contorno:

$$\begin{aligned}
\nabla^2 \phi(x, y) &= 0 \quad \text{en } \Omega \\
\phi(0, y) &= 0 \\
\phi(x, 0) &= 0 \\
\phi(x, 200) &= 0 \\
\phi(200, y) &= \sinh(\pi) \cdot \sin\left(\frac{\pi y}{200}\right)
\end{aligned} \tag{9.2}$$

La solución analítica a este problema es fácil de obtener:

$$\phi(x, y) = \sin\left(\frac{\pi y}{200}\right) \cdot \sinh\left(\frac{\pi x}{200}\right) \tag{9.3}$$

Para su introducción en el programa, cada lado del cuadrado se ha separado en 200 elementos de 1 unidad de longitud.

En la figura 9.1 se muestra una captura de pantalla de ParaView con el valor del potencial en todo el dominio.

Gracias a la herramienta de ParaView 'Plot Over Line' se pueden representar gráficas de resultados. La figura 9.2 muestra una gráfica del potencial a lo largo de la línea $y = 100$.

Para apreciar la exactitud de la solución, en la tabla 9.2, se muestra una comparativa entre los resultados obtenidos y los resultados teóricos para 11 puntos diferentes en la línea $y = 100$.

Como se ha demostrado, los resultados solo tienen un error de aproximadamente $5 \cdot 10^{-6}$, por lo que se puede afirmar que el algoritmo no tiene errores de programación y es válido.

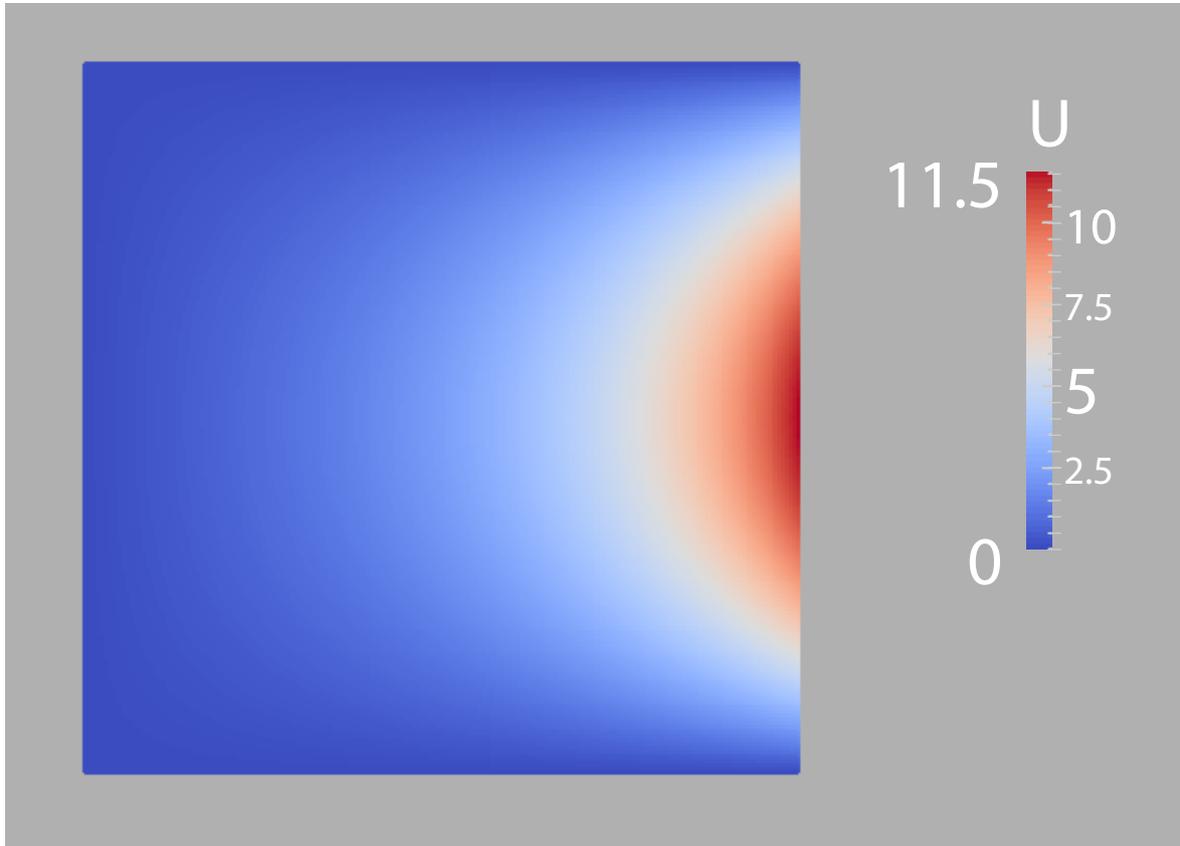


Figura 9.1: Captura de pantalla de los resultados obtenidos, se muestra el valor del potencial en todo el dominio.

Coordenada X	Solución Numérica	Solución Analítica	Error
10	0.157726	0.157726...	$2.5 \cdot 10^{-4} \%$
20	0.319351	0.319353...	$4.82 \cdot 10^{-4} \%$
40	0.670481	0.670484...	$4.47 \cdot 10^{-4} \%$
60	1.08833	1.08834...	$5.17 \cdot 10^{-4} \%$
80	1.61448	1.61449...	$4.98 \cdot 10^{-4} \%$
100	2.30129	2.30130...	$3.87 \cdot 10^{-4} \%$
120	3.21710	3.21711...	$4.06 \cdot 10^{-4} \%$
140	4.45304	4,45306...	$5.31 \cdot 10^{-4} \%$
160	6.13211	6,13214...	$5.00 \cdot 10^{-4} \%$
180	8.42139	8,42143...	$4.73 \cdot 10^{-4} \%$
190	9.86310	9,86315...	$5.32 \cdot 10^{-4} \%$

Cuadro 9.2: Precisión de los resultados

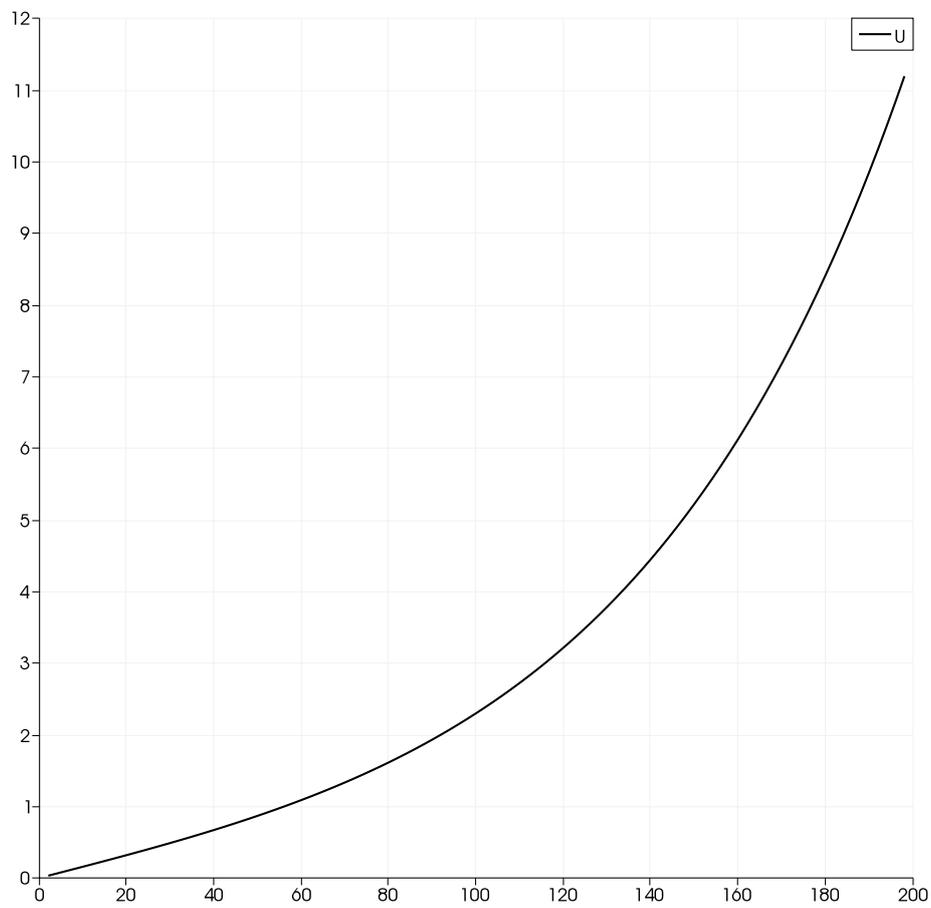


Figura 9.2: Resultados del potencial en la línea $y = 100$

9.3. Conclusiones

Este programa es interesante para resolver rápidamente problemas sencillos de potencial, y, aunque no tenga mucha utilidad desde el punto de vista ingenieril, ha cumplido su objetivo: servir como primera toma de contacto con los elementos de contorno y con la programación.

CAPÍTULO 10

Test de Verificación para BEM3D

10.1. Descripción

Este programa resuelve problemas gobernados por la ecuación de Laplace mediante el método de los elementos de contorno (BEM), en cualquier dominio y con condiciones de contorno de tipo Dirichlet, Neumann o Robin (mixtas). En este caso, sí se ha implementado la variable compleja, que es fundamental para la correcta resolución de los problemas de flotabilidad.

El programa necesita dos archivos de entrada llamados 'MESH.unv' y 'CONDICIONES.txt'.

El archivo 'MESH.unv' es el archivo de salida del programa 'SALOME' (explicado en el Capítulo 8) y contiene todos los elementos del dominio y del contorno. El algoritmo está preparado para recibir una malla de tetraedros, que aunque no se usarán para la resolución del problema (ya que solo interesan los elementos en el contorno), sirven para captar puntos interiores y calcular la solución en el interior¹. De esta forma no es necesario introducir un archivo adicional con los puntos interiores como ocurría en BEM2D. Nótese que al mallar el dominio con tetraedros, el contorno, automáticamente, se malla con triángulos, que es lo que se pretendía.

El archivo 'CONDICIONES.txt' debe contener las condiciones de contorno de todas las fronteras definidas como grupos en 'SALOME' y exportadas al archivo 'MESH.unv'. Si a la condición de contorno se le asigna la siguiente forma:

$$\alpha q + \beta u = \gamma \quad (10.1)$$

el archivo tiene el siguiente aspecto:

- N° Entero: número total de condiciones de contorno.
- Para cada condición de contorno:
 - Nombre de la condición de contorno (el mismo que se le asignó en 'SALOME').
 - Los valores de α, β y γ en una misma línea y separados por espacios.

A continuación se muestra un ejemplo:

```
5
Sup_Libre
(9.81,0.) (-1.,0.) (0.,0.)
Cond_Radiacion
(1.,0.) (0.,-1.) (0.,0.)
Fondo_Mar
(1.,0.) (0.,0.) (0.,0.)
Pared_Cilindro
(1.,0.) (0.,0.) (0.,0.)
Fondo_Cilindro
(1.,0.) (0.,0.) (0.,-1.)
```

¹El programa, una vez obtenidos los valores del potencial y su derivada en los contornos, calcula estos valores en los vértices de los tetraedros (siempre que no formen parte del contorno).

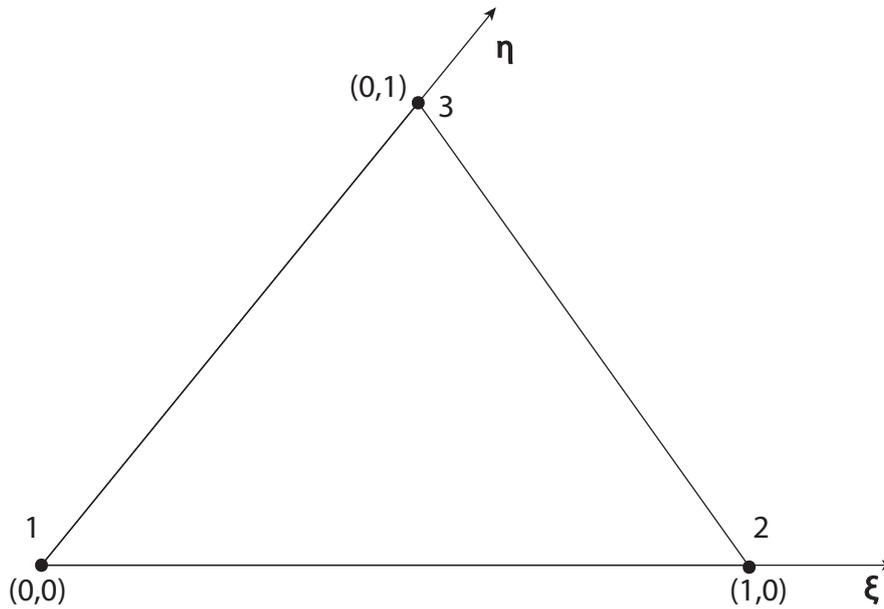


Figura 10.1: Cambio de Coordenadas para la integración.

Como se ha mostrado, para que sean reconocibles en C++, los valores complejos se deben escribir entre paréntesis y separando la parte real de la imaginaria con una coma. Finalmente, como se explicó en el Capítulo 8, el archivo .unv define cada triángulo por sus tres vértices; dichos vértices se posicionan de forma antihoraria para contornos exteriores y de forma horaria para contornos interiores. Así, con la formulación apropiada, el usuario puede estar seguro de que los vectores normales siempre tendrán sentido hacia fuera del dominio.

El procedimiento de resolución es el mismo que el explicado en el Capítulo 9. Para la evaluación de singularidades, el método aplicado es el mismo que el mencionado en el Capítulo 4. Las integrales no singulares se han resuelto con una cuadratura de Gauss de 7 puntos. Basándose en el cambio de coordenadas propuesto en la figura 10.1, cualquier punto en el plano del triángulo se puede definir como:

$$\begin{aligned}
 P(x, y, z) &= P(\xi, \eta) = (Q_x, Q_y, Q_z) \\
 Q_x &= q_{1x} + (q_{2x} - q_{1x}) \cdot \xi + (q_{3x} - q_{1x}) \cdot \eta \\
 Q_y &= q_{1y} + (q_{2y} - q_{1y}) \cdot \xi + (q_{3y} - q_{1y}) \cdot \eta \\
 Q_z &= q_{1z} + (q_{2z} - q_{1z}) \cdot \xi + (q_{3z} - q_{1z}) \cdot \eta
 \end{aligned}
 \tag{10.2}$$

Donde q_1 , q_2 y q_3 son las coordenadas reales de los vértices 1, 2 y 3 del triángulo (ver figura 10.1). Dada la ecuación 10.2, las coordenadas ξ , η y los pesos de los puntos de integración son los que constan en la tabla 10.1.

Quedando la evaluación de las integrales como sigue:

Nodo	Coordenada ξ_k	Coordenada η_k	w_k
1	0.3333333333333333	0.3333333333333333	0.2250000000000000
2	0.797426985353087	0.101286507323456	0.125939180544827
3	0.101286507323456	0.797426985353087	0.125939180544827
4	0.101286507323456	0.101286507323456	0.125939180544827
5	0.470142064105115	0.470142064105115	0.132394152788506
6	0.470142064105115	0.059715871789770	0.132394152788506
7	0.059715871789770	0.470142064105115	0.132394152788506

Cuadro 10.1: Coordenadas de los Puntos de Integración y Funciones Peso para el caso tridimensional

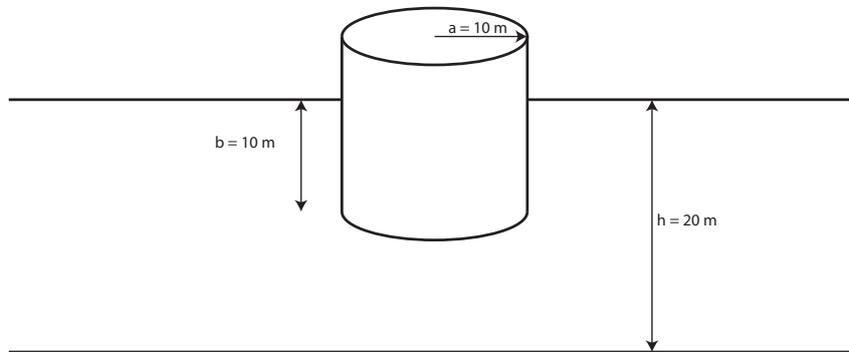


Figura 10.2: Geometría del problema de validación

$$G_{ij} = \int_{\Gamma_j} u^* d\Gamma = \frac{A}{4\pi} \sum_{k=1}^7 \frac{1}{R_k} \cdot w_k$$

$$H_{ij} = \int_{\Gamma_j} q^* d\Gamma = \frac{-A}{4\pi} \sum_{k=1}^7 \frac{1}{R_k^3} \cdot \left[n_x^j \cdot (Q_x^k - x_i) + n_y^j \cdot (Q_y^k - y_i) + n_z^j \cdot (Q_z^k - z_i) \right] \cdot w_k \quad (10.3)$$

$$R_k = \sqrt{(Q_x^k - x_i)^2 + (Q_y^k - y_i)^2 + (Q_z^k - z_i)^2}$$

$A =$ Area del Triángulo

Como ocurre con el programa BEM2D, la visualización de resultados se consigue mediante un archivo 'VTK' generado.

10.2. Prueba de Validación

Para la prueba de validación se ha calculado la diagonal de las matrices de masa añadida y de amortiguamiento hidrodinámico. La geometría del problema queda definida en la figura 10.2.

Los desplazamientos modales calculados son el 'Surge', 'Heave' y 'Pitch', que son los desplazamientos en el Eje X, desplazamiento en el Eje Z y giro alrededor del Eje Y respectivamente. El desplazamiento en el Eje Y es igual al desplazamiento modal 'Surge' y el giro en el Eje X es igual

al desplazamiento modal 'Pitch', debido a la simetría del problema. Por otra parte, el giro respecto al Eje Z no induce tensiones normales al fluido², por lo que el término de masa añadida y de amortiguamiento para este desplazamiento modal es 0.

Las condiciones de contorno de los tres problemas que se van a resolver están reflejadas en el Capítulo 3 y son:

Desplazamiento Surge

$$\begin{aligned}
\nabla^2 \phi &= 0 \\
g \frac{\partial \phi}{\partial z} - \sigma^2 \phi &= 0 \quad \text{en } z = 0 \\
\frac{\partial \phi}{\partial z} &= 0 \quad \text{en } z = -h \\
\frac{\partial \phi}{\partial z} &= 0 \quad \text{en } r \leq a \text{ y } z = -b \\
\frac{\partial \phi}{\partial r} &= n_x = \cos(\theta) \quad \text{en } r = a \text{ y } -b \leq z \leq 0 \\
\lim_{r \rightarrow \infty} \sqrt{r} \left(\frac{\partial \phi}{\partial r} - ik\phi \right) &= 0 \quad \text{Condición de radiación}
\end{aligned} \tag{10.4}$$

Desplazamiento Heave

$$\begin{aligned}
\nabla^2 \phi &= 0 \\
g \frac{\partial \phi}{\partial z} - \sigma^2 \phi &= 0 \quad \text{en } z = 0 \\
\frac{\partial \phi}{\partial z} &= 0 \quad \text{en } z = -h \\
\frac{\partial \phi}{\partial z} &= 1 \quad \text{en } r \leq a \text{ y } z = -b \\
\frac{\partial \phi}{\partial r} &= 0 \quad \text{en } r = a \text{ y } -b \leq z \leq 0 \\
\lim_{r \rightarrow \infty} \sqrt{r} \left(\frac{\partial \phi}{\partial r} - ik\phi \right) &= 0 \quad \text{Condición de radiación}
\end{aligned} \tag{10.5}$$

Desplazamiento Pitch

$$\begin{aligned}
\nabla^2 \phi &= 0 \\
g \frac{\partial \phi}{\partial z} - \sigma^2 \phi &= 0 \quad \text{en } z = 0 \\
\frac{\partial \phi}{\partial z} &= 0 \quad \text{en } z = -h \\
\frac{\partial \phi}{\partial z} &= n_5 \quad \text{en } r \leq a \text{ y } z = -b \\
\frac{\partial \phi}{\partial r} &= n_5 \quad \text{en } r = a \text{ y } -b \leq z \leq 0 \\
\lim_{r \rightarrow \infty} \sqrt{r} \left(\frac{\partial \phi}{\partial r} - ik\phi \right) &= 0 \quad \text{Condición de radiación}
\end{aligned} \tag{10.6}$$

Donde n_5 está definido en la ecuación 3.15.

²Este desplazamiento modal solo induce tensiones tangenciales, que no se transmiten al fluido debido a la hipótesis de fluido no viscoso.

Se ha creado una malla en 'SALOME' con aproximadamente 9000 elementos. Dicha malla está dividida en 5 grupos llamados 'Fondo_Mar', 'Sup_Libre', 'Cond_Radiacion', 'Pared_Cilindro' y 'Fondo_Cilindro'. La figura 10.3 muestra varias capturas de pantalla de 'SALOME' de dicha malla. En esta figura se distingue como se ha densificado la malla en las proximidades del cilindro; además, la condición de radiación se ha impuesto a una distancia de 250 metros del centro de dicho cilindro, que se ha supuesto, está lo suficientemente alejado en comparación con el radio del cilindro.

Las pruebas se han realizado para varias longitudes de onda, desde $k = 0.05$ hasta $k = 0.225$. Para estas longitudes de onda, se tienen desde unos 50 elementos constantes por longitud de onda (para $k = 0.05$) hasta unos 15 elementos por longitud de onda (para $k = 0.225$)³. Los resultados del movimiento 'Surge' se han comparado con los resultados analíticos, que se pueden encontrar en [23]. Dichos resultados se muestran en la figura 10.4. Como se puede apreciar, los datos ajustan bastante bien a los teóricos, excepto por el dato de amortiguamiento hidrodinámico para $k \cdot a = 1.75$, donde el resultado numérico es mayor al teórico. Para valores de $k \cdot a$ mayores de 2.25 no se han realizado pruebas ya que, a partir de este punto, el tamaño de elemento empieza a ser demasiado grande en comparación con la longitud de onda, y los resultados no son fiables.

Una vez verificado que el programa no contiene errores de programación⁴ se ha procedido a estudiar los otros dos desplazamientos modales restantes. Los resultados obtenidos se pueden ver en las figuras 10.5 y 10.6.

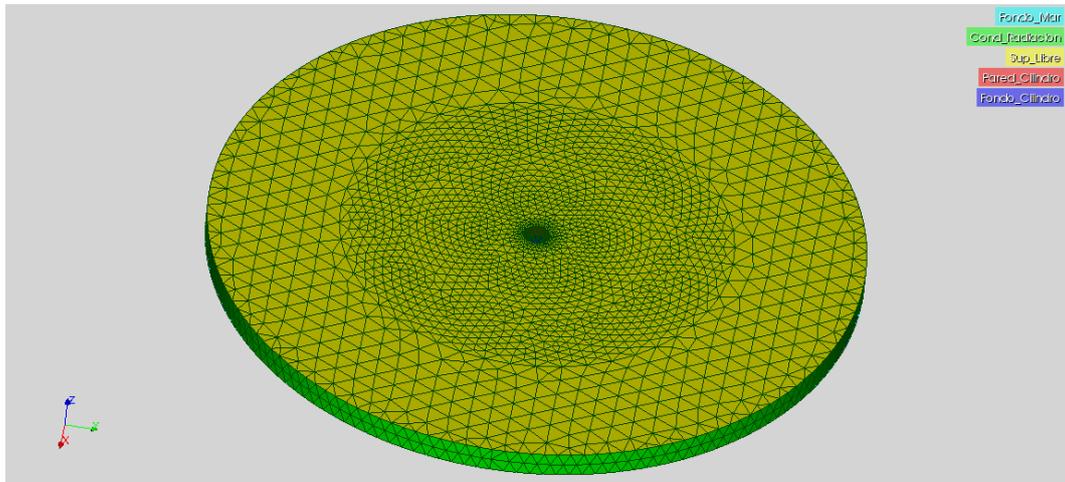
Finalmente, las figuras 10.7, 10.8 y 10.9, muestran capturas de pantalla de 'ParaView', donde aparecen las formas que adopta la superficie del mar como consecuencia de cada uno de los tres desplazamientos modales estudiados.

10.3. Conclusiones

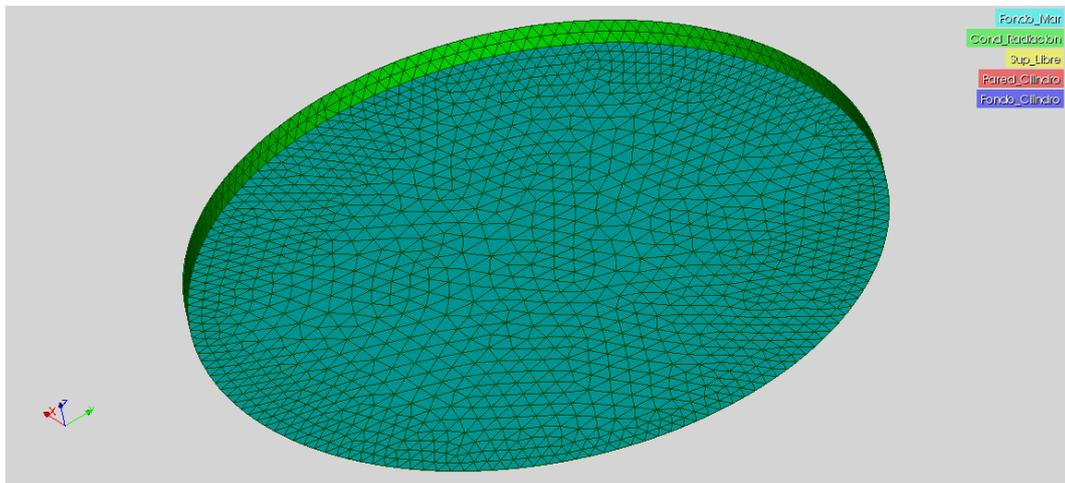
El programa presentado tiene mucho potencial en la ingeniería offshore ya que puede resolver (con más o menos exactitud) las matrices de masa añadida y de amortiguamiento hidrodinámico de cualquier estructura parcial o totalmente sumergida. En el estudio de estas estructuras offshore ante cargas dinámicas, como son el oleaje o la acción del viento, se obtienen, generalmente, resultados bastantes conservadores si no se tienen en cuenta dichos coeficientes mencionados anteriormente. Hoy en día, ciertos programas de cálculo de estructuras vienen preparados para que se les introduzcan las matrices de masa añadida y amortiguamiento hidrodinámico y, de esta forma, conocer mejor los esfuerzos críticos, así como hacer un mejor estudio a fatiga de la estructura.

³Por regla general, se deben tener alrededor de 20 elementos constantes por longitud de onda para que los resultados sean fiables.

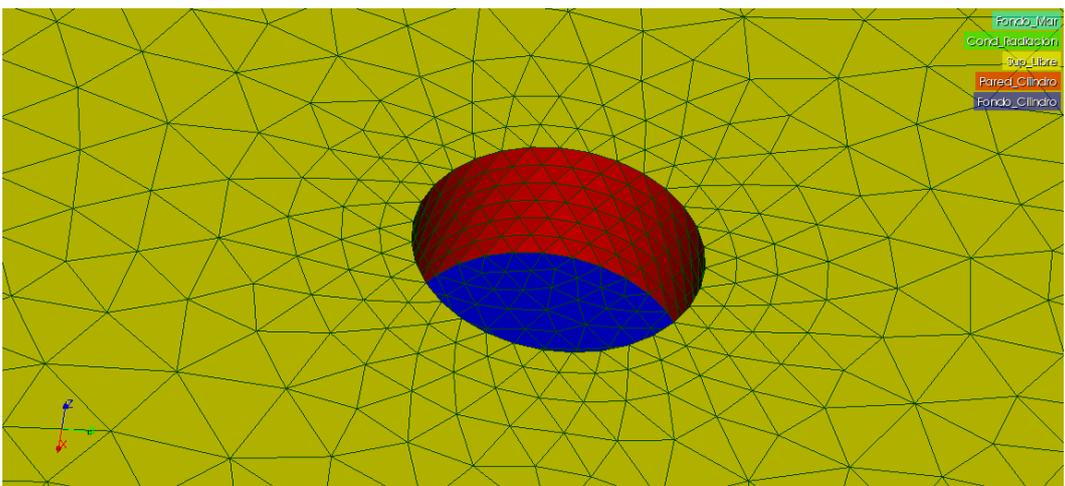
⁴Las diferencias de los resultados numéricos con los teóricos se pueden deber a que la malla introducida sea más o menos adecuada y/o a que se han considerado elementos constantes y no de mayor orden.



(a) Vista Superior



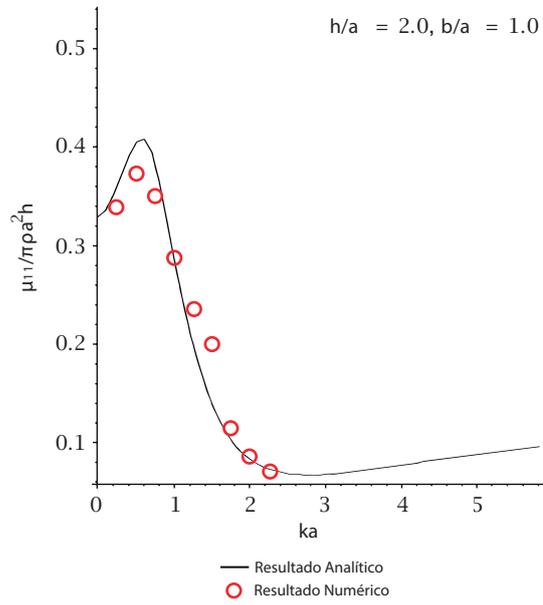
(b) Vista Inferior



(c) Mallado del Cilindro

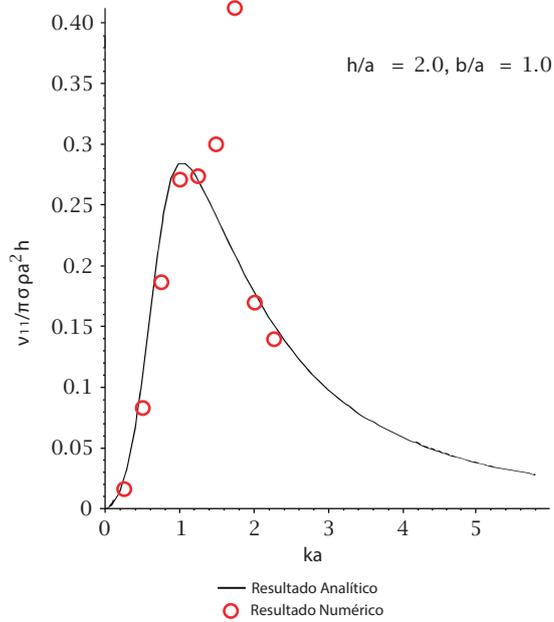
Figura 10.3: Malla para la prueba de validación de BEM3D.

Masa Añadida (Surge Motion)



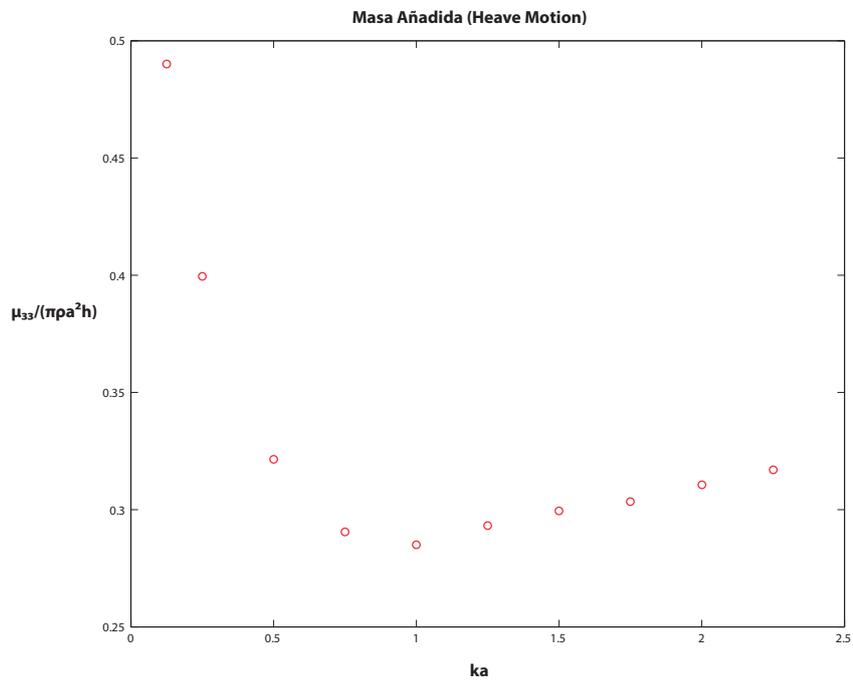
(a) Desplazamiento Surge - Masa Añadida

Amortiguamiento por Radiación (Surge Motion)

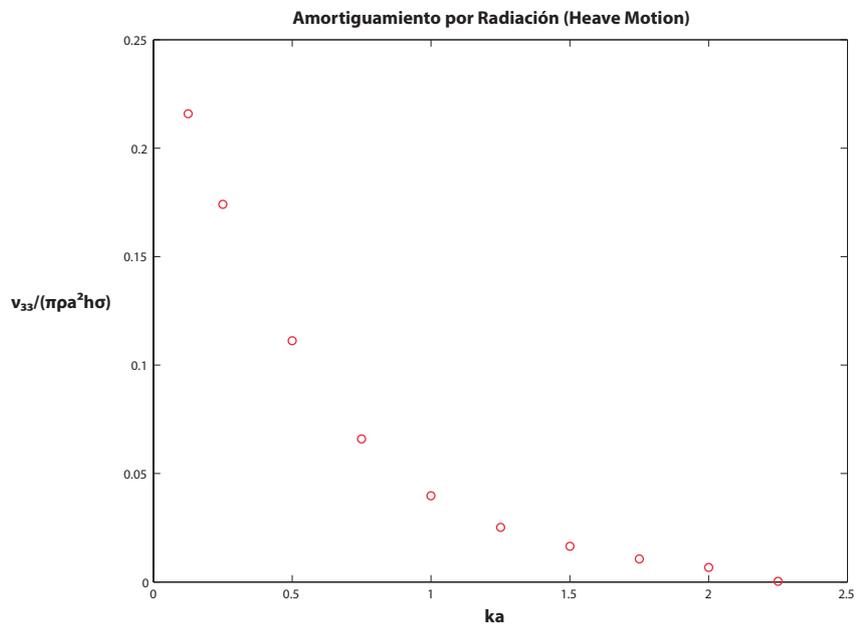


(b) Desplazamiento Surge - Amortiguamiento Hidrodinámico

Figura 10.4: Resultados del Movimiento Surge

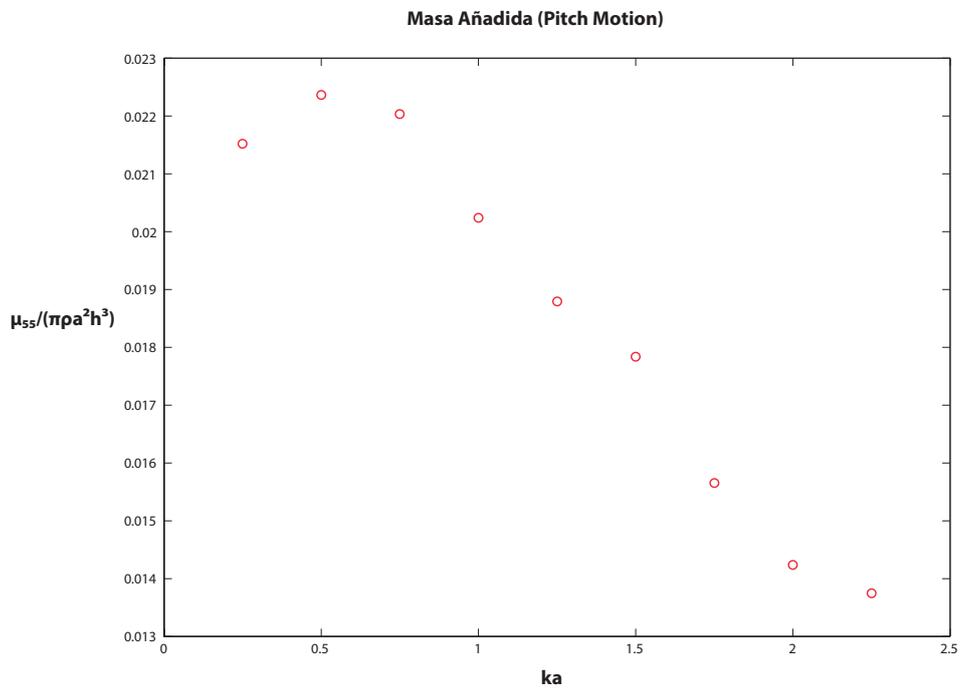


(a) Desplazamiento Heave - Masa Añadida

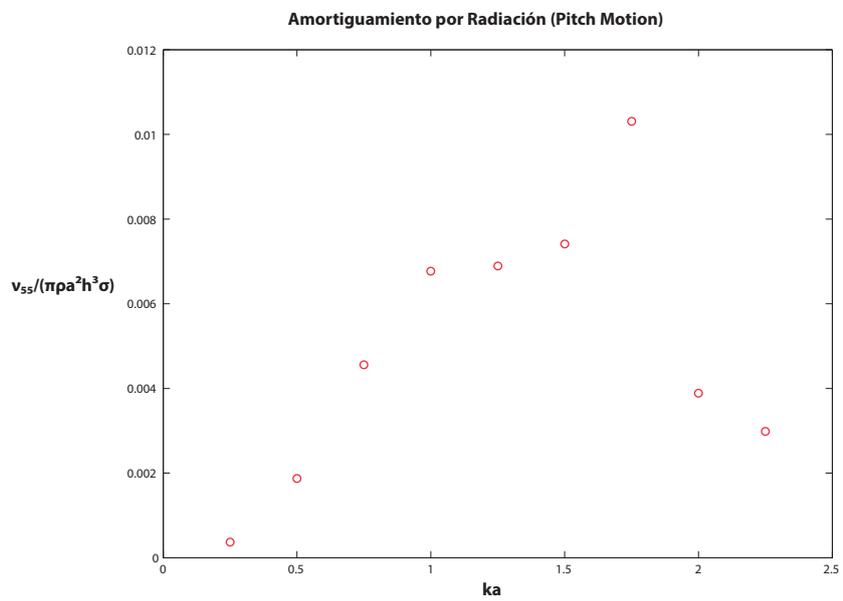


(b) Desplazamiento Heave - Amortiguamiento Hidrodinámico

Figura 10.5: Resultados del Movimiento Heave



(a) Desplazamiento Pitch - Masa Añadida



(b) Desplazamiento Pitch - Amortiguamiento Hidrodinámico

Figura 10.6: Resultados del Movimiento Pitch

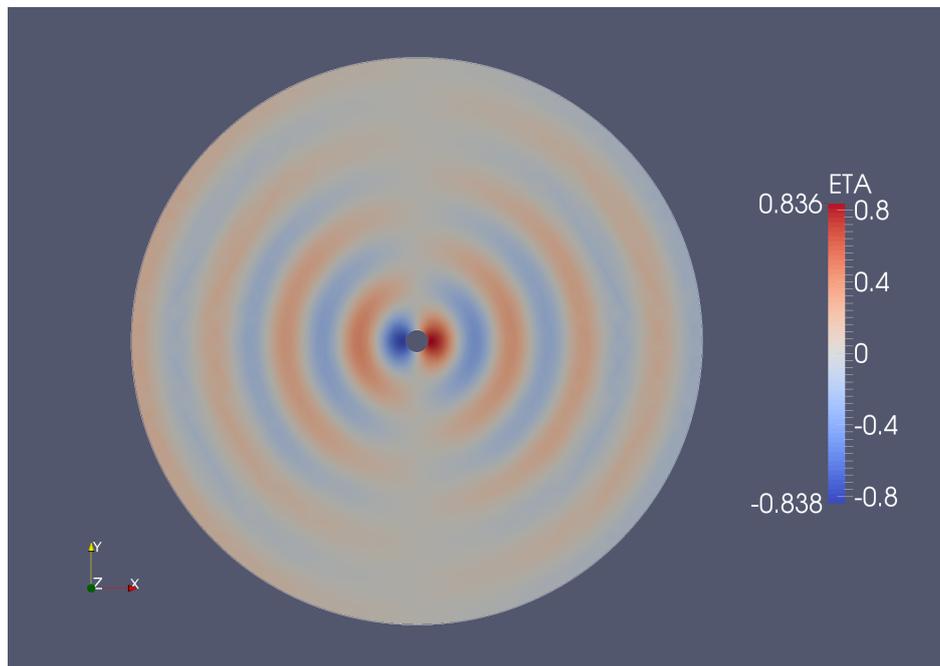


Figura 10.7: Superficie del Mar debido al desplazamiento modal Surge, $k=0.1$ y Amplitud = 1 metro (Unidades en metros)

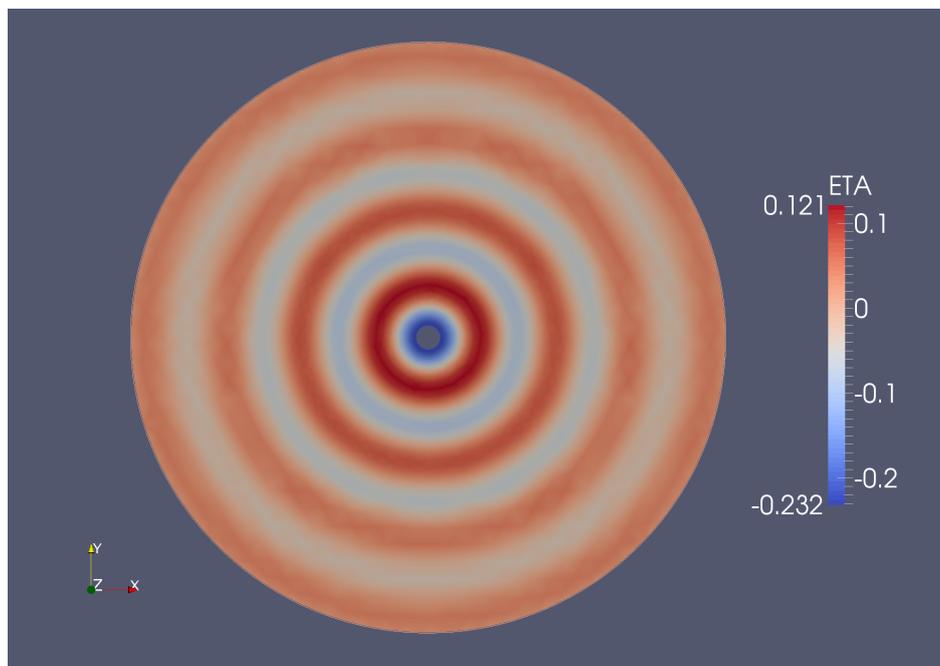


Figura 10.8: Superficie del Mar debido al desplazamiento modal Heave, $k=0.1$ y Amplitud = 1 metro (Unidades en metros)

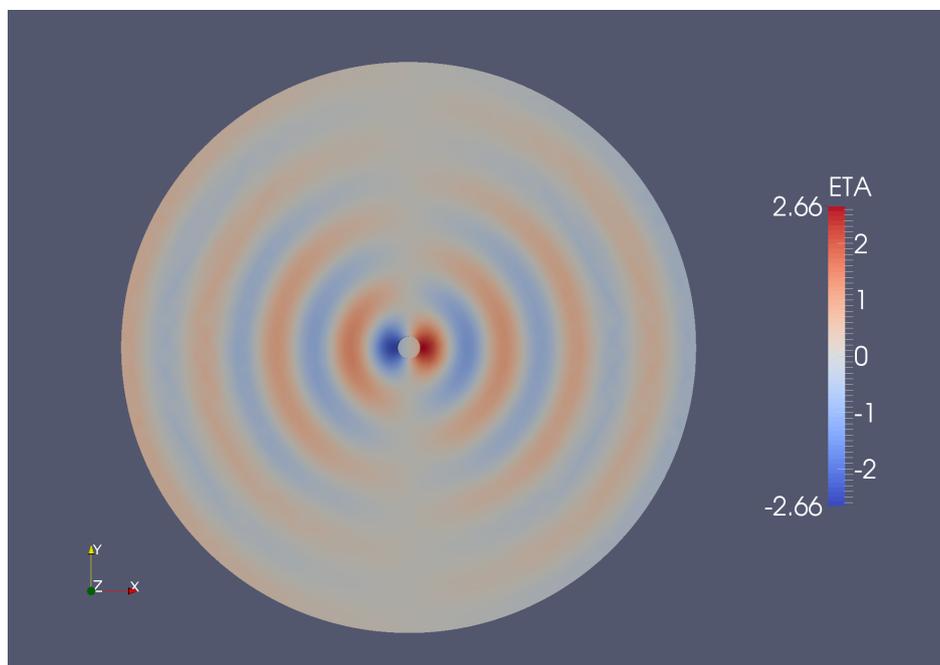


Figura 10.9: Superficie del Mar debido al desplazamiento modal Pitch, $k=0.1$ y Amplitud = 1 radian (Unidades en metros)

CAPÍTULO 11

Test de Verificación para Mild-Slope con Fondo Constante

11.1. Descripción

Este programa, como su propio nombre indica, resuelve la ecuación de MildSlope para dominios con fondo constante. Está concebido expresamente para calcular la agitación interior en dársenas y puertos (que tengan un fondo cuasiconstante) ya que se tiene en cuenta los efectos de la reflexibilidad de los contornos. El programa está creado según lo expuesto en el apartado 5.4.

Al ser un programa sencillo, no se ha implementado una entrada desde 'SALOME', sino a través de archivos de texto. Los archivos de entrada requeridos son tres, 'datosext.dat', 'datosint.dat' y 'puntosint.dat'.

El archivo 'datosext.dat' contendrá los datos relacionados con el potencial entrante y el borde de contacto entre el dominio exterior y el dominio interior. A continuación se expone un ejemplo:

```
20
1.0
5.7
0.0
1
50
-150.0 25.0
-150.0 -25.0
```

En la primer línea debe aparecer la profundidad del fondo en metros. La segunda línea contiene la altura de ola entrante. En la tercera debe estar el periodo de onda en segundos y en la cuarta el ángulo de incidencia del potencial entrante. Las líneas restantes definen el contorno llamado Γ en el apartado 5.4. El número 1 indica que solo se definirá un contorno; el siguiente dígito (en este caso 50) señala el número de segmentos en los que se dividirá el contorno Γ . Las dos líneas restantes contienen las coordenadas x e y de los puntos inicial y final de dicha frontera Γ^1 .

El archivo 'datosint.dat' debe contener los datos de todos los contornos llamados S en el apartado 5.4. A continuación se expone un ejemplo:

```
5
125
-150.0 -25.0
-150.0 -150.0
0.0
300
-150.0 -150.0
150.0 -150.0
0.0
300
150.0 -150.0
```

¹Se deben escribir los puntos en sentido antihorario

```

150.0  150.0
0.0
300
150.0  150.0
-150.0 150.0
0.0
125
-150.0 150.0
-150.0 25.0
0.0

```

En la primera línea consta el número de contornos diferentes que se van a definir. A continuación, para cada uno de dichos contornos, se define, en primer lugar los segmentos en los que se quiere dividir dicho contorno, las coordenadas x e y de los puntos inicial y final y, por último, el valor del coeficiente de reflexión K_r asignado a esa frontera.

Por último, el archivo 'puntosint.dat' define mallas rectangulares de puntos donde se calculará la solución interior. A continuación se expone un ejemplo:

```

1
298  298
1 1
-149.0 -149.0

```

El primer dígito (1 en este caso) muestra el número de mallas de puntos que se van a definir. A continuación, para cada malla, se define el número de puntos en el Eje X y Eje Y , después el intervalo (en metros) entre punto y punto en cada uno de los ejes y, por último, las coordenadas del punto situado en la esquina inferior izquierda de la malla. Así, para el ejemplo expuesto, se define una malla de 298 por 298 puntos (88804 puntos en total), con una separación de 1 metro entre cada uno de ellos y el siguiente (en cada eje) y cuyo primer punto tiene coordenadas (-149,-149).

La evaluación de singularidades se ha gestionado de la misma forma que la explicada en el Capítulo 5 y las integrales se han resuelto mediante una cuadratura de Gauss de 4 puntos al igual que en el programa explicado en el Capítulo 9.

Como resultado, el programa crea un archivo 'VTK' de ParaView con los datos del run-up en cada punto definido por el archivo 'puntosint.dat'.

11.2. Prueba de Validación

La prueba de validación se ha hecho introduciendo los tres archivos mostrados a lo largo del apartado anterior. Estos archivos definen un problema estudiado en [3] y que tiene la geometría mostrada en la figura 11.1. Ésta consiste en un recinto cuadrado de 300 metros de radio que tiene una entrada centrada de 25 metros en su lado izquierdo.

Por la frontera marcada como Γ entra un oleaje con un periodo de 5.7 segundos, la profundidad del fondo es de 20 metros y el ángulo de incidencia es 0 grados. Se estudiarán dos casos, uno en el que todos los bordes nombrados como S son totalmente absorbentes, es decir, $K_r = 0$; y otro, en el que todos los bordes son totalmente reflexivos, es decir $K_r = 1$. Este dato se recuerda que se cambia en el archivo 'datosint.dat'. Los resultados obtenidos de la altura de ola en cada punto y del run-up (ver ecuación 5.64) son los mostrados en la figura 11.2 para el caso con todos los bordes absorbentes y en la figura 11.4 para el caso con todos los bordes reflexivos.

Como era de esperar, para el caso totalmente reflexivo, se forma una onda estacionaria con una amplitud del doble que la amplitud del oleaje entrante. Si estos resultados se comparan con los que

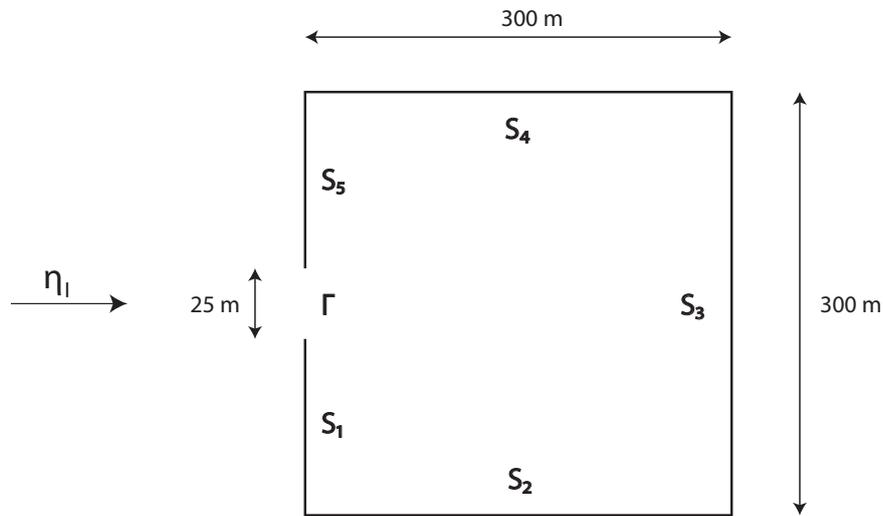


Figura 11.1: Geometría del problema de validación

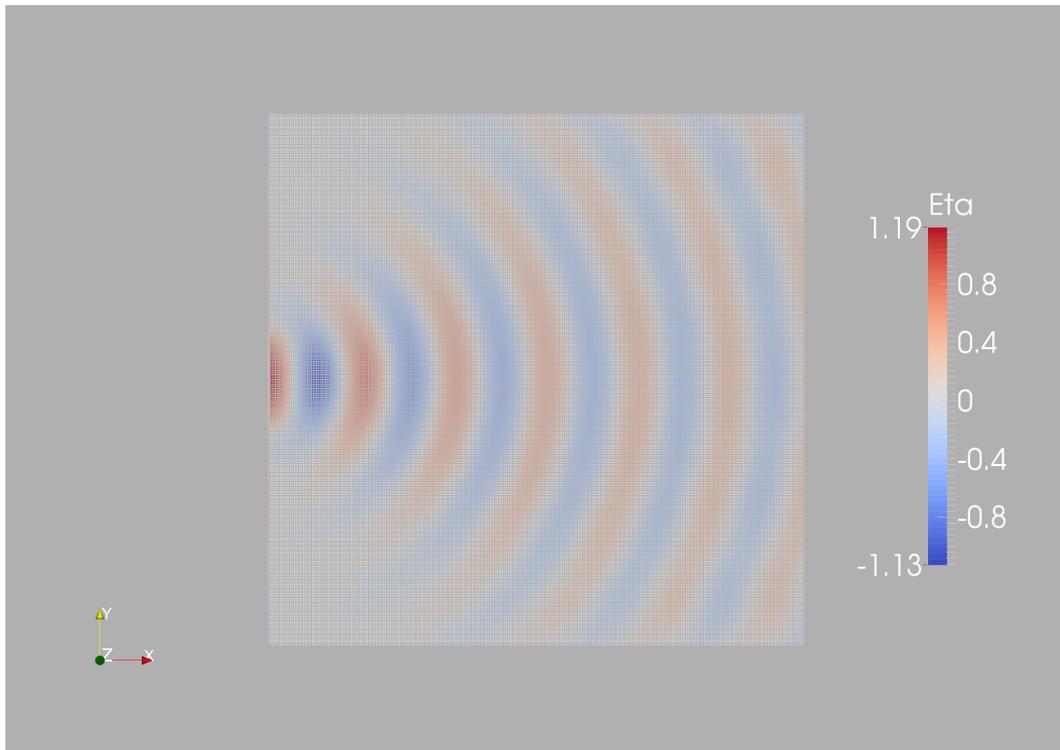
se muestran en las figuras 11.3 y 11.5, obtenidos de [3], se comprueba que los resultados son bastante parecidos.

Como prueba adicional, se ha estudiado el mismo problema con bordes totalmente absorbentes pero con un periodo de onda entrante de solo 2 segundos (alta frecuencia). Los resultados se pueden observar en la figura 11.6.

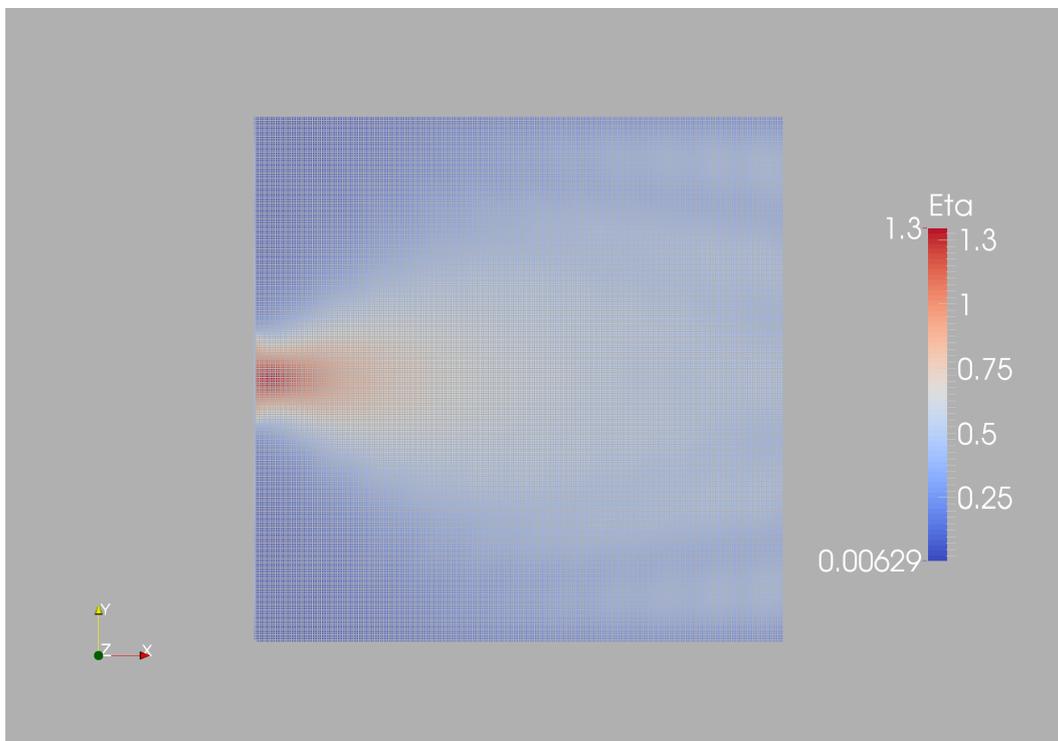
Como era de esperar, las ondas de más alta frecuencia son más penetrantes en el medio pero menos difusivas, obteniéndose mayor superficie con poca agitación pero más agitación en la parte central de la dársena.

11.3. Conclusiones

Esta herramienta podría ser bastante útil para el estudio de la ubicación óptima para la construcción de cajones portuarios, ya que los puertos, generalmente, tienen la mayor parte de su fondo dragado a una profundidad cuasiconstante. El algoritmo presentado permite modelizar todos los muelles, diques y obstáculos verticales que se pueden encontrar en un puerto, además de asignarle un coeficiente de reflexión. Este coeficiente será más cercano a 1 en muelles y diques verticales de hormigón y será más cercano a 0 cuando se presenten diques de escollera, que absorben más energía. Aunque este programa se hizo con el objetivo del estudio de cajones portuarios, es obvio que sirve para más aspectos de la ingeniería portuaria como son estudios del oleaje o estudios de agitación interior.



(a) Altura de Ola instantánea



(b) Run-Up

Figura 11.2: Resultados para el caso con todos los bordes absorbentes

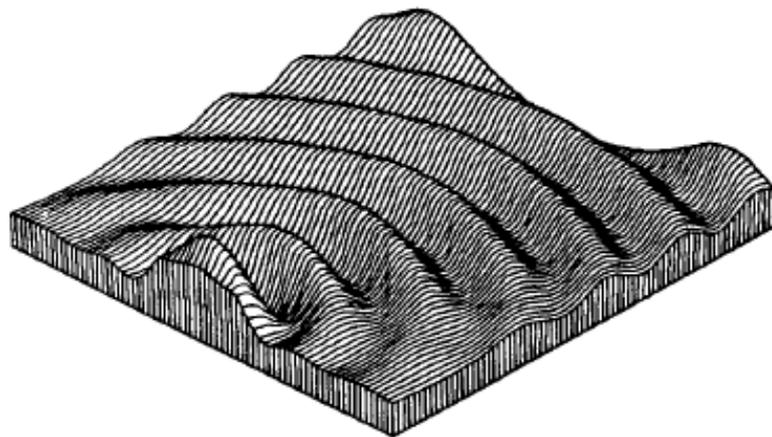
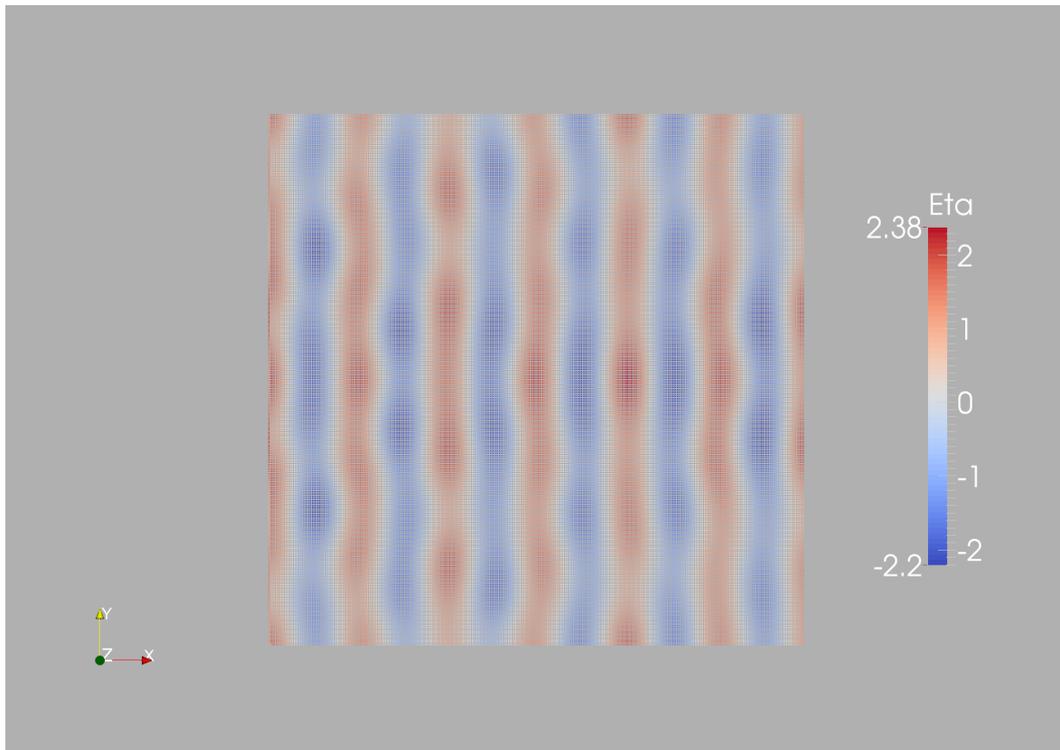
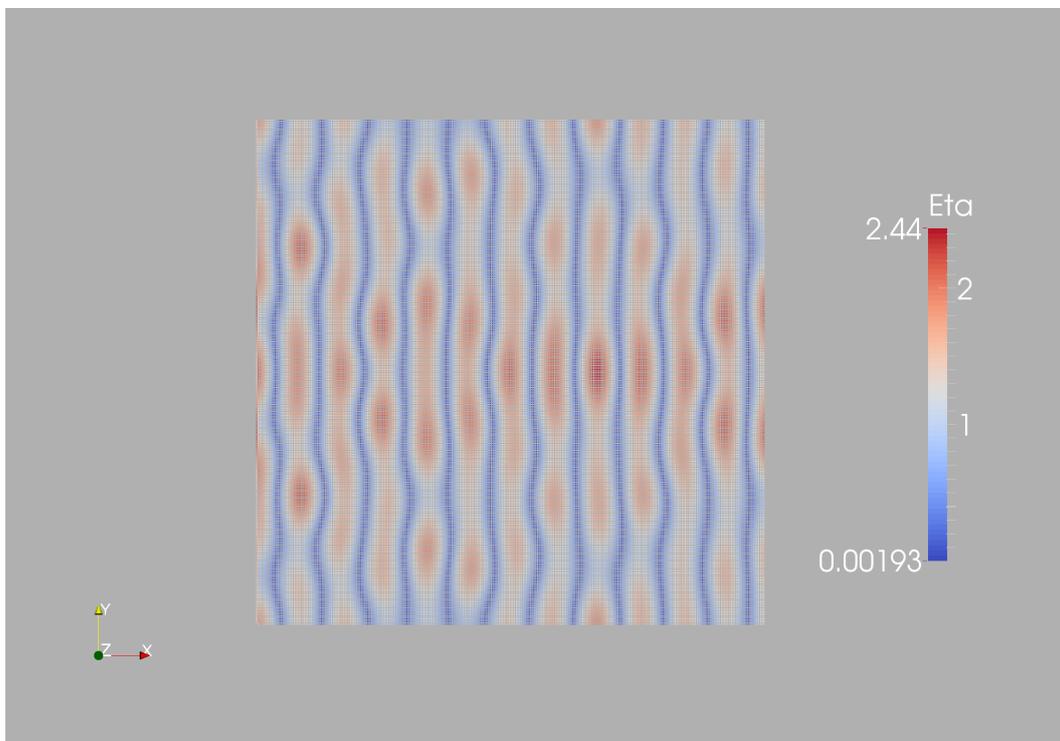


Figura 11.3: Solución para bordes totalmente absorbentes obtenida en [3]



(a) Altura de Ola instantánea



(b) Run-Up

Figura 11.4: Resultados para el caso con todos los bordes reflexivos

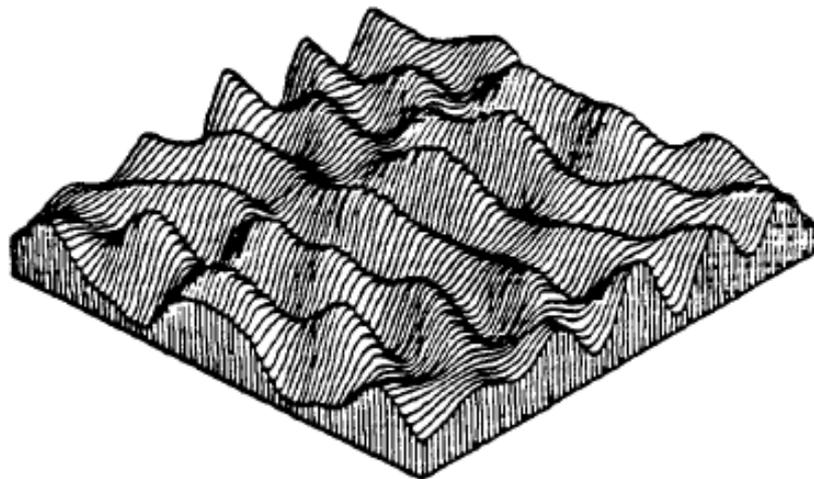
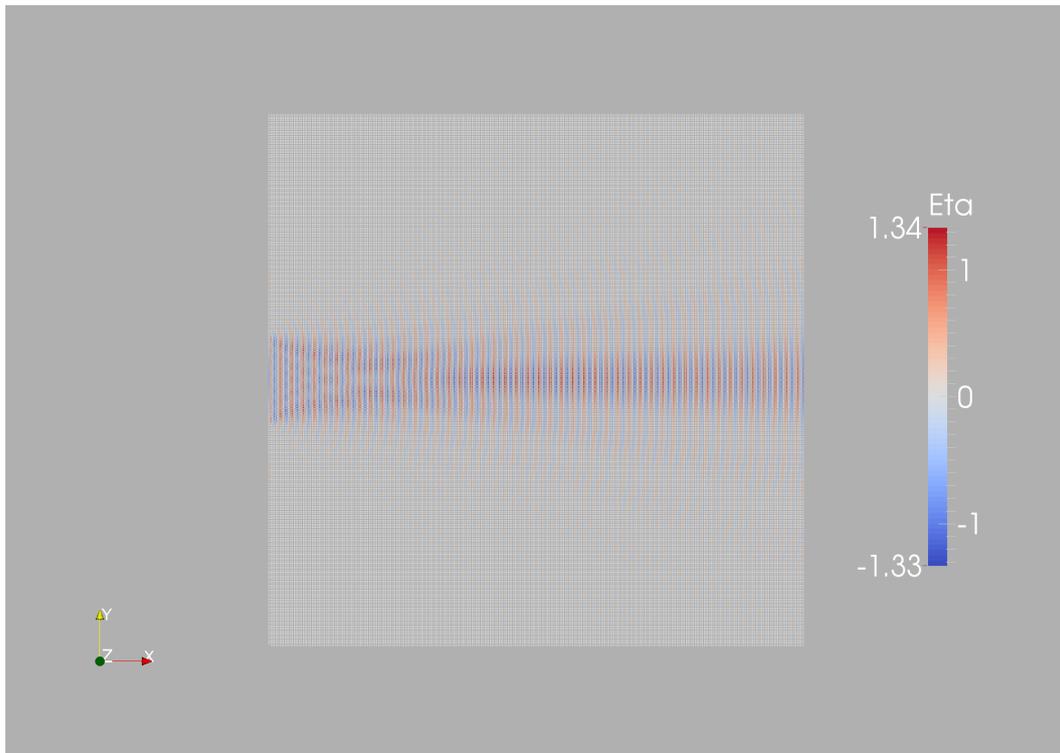
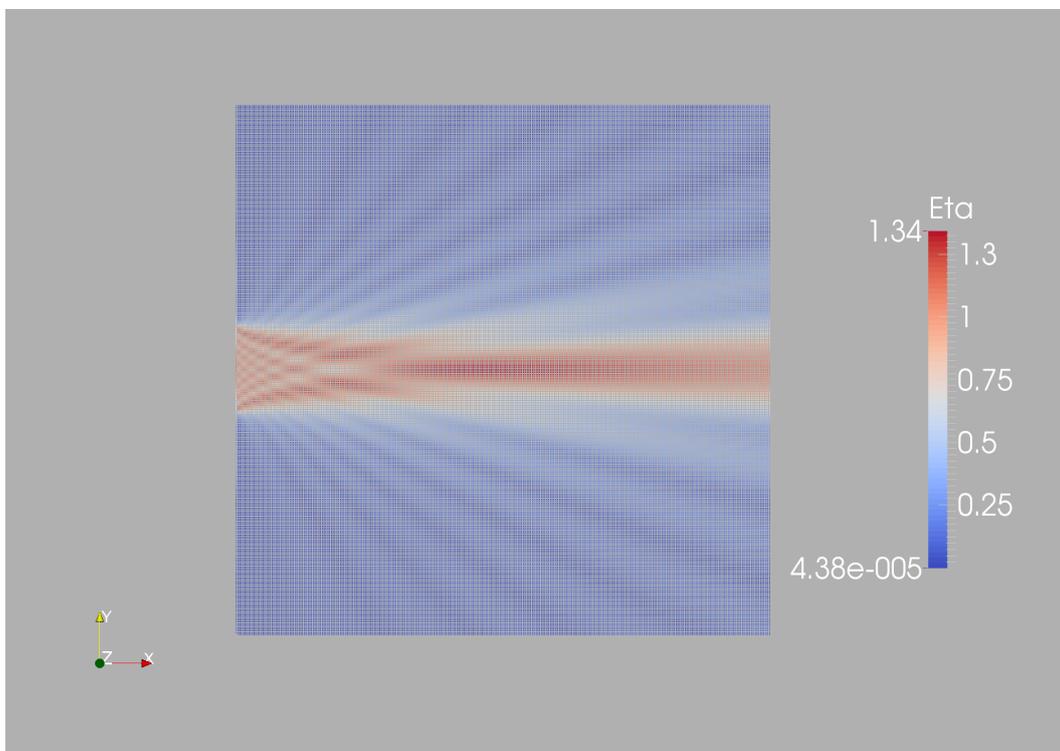


Figura 11.5: Solución para bordes totalmente reflexivos obtenida en [3]



(a) Altura de Ola instantánea



(b) Run-Up

Figura 11.6: Resultados para Alta Frecuencia ($T=2\text{seg}$)

CAPÍTULO 12

Test de Verificación para Mild-Slope con Fondo Variable

12.1. Descripción

A diferencia del programa expuesto en el capítulo anterior, éste algoritmo resuelve la ecuación de Mild-Slope en un dominio con fondo variable. El programa está creado según lo explicado en el Capítulo 5.

La entrada de este programa está pensada para la introducción de una malla de triángulos que modelice el fondo del dominio. Esto se puede conseguir gracias al comando 'Smoothing Surface' de 'SALOME', el cual, a partir de un listado de puntos, es capaz de crear una malla que se adapte a dichos puntos.

Además de este archivo que se debe llamar 'MESH.unv', debe haber otros dos que tienen que llamarse 'Oleaje.txt' y 'Condiciones'.

El archivo 'Oleaje.txt' contiene tres líneas con un número en cada línea, estos números deben corresponder con:

- Periodo del Oleaje Entrante
- Profundidad del Fondo en el Dominio Exterior
- Altura de Ola del Oleaje Entrante

Por otra parte, el archivo 'Condiciones.txt' debe contener dos líneas por cada contorno en el problema¹. En la primera línea debe constar el nombre del grupo tal y como se ha definido en 'SALOME' y, en la segunda línea, el valor del coeficiente de reflexión K_r de dicha frontera.

Una vez dicho esto, para las pruebas de validación realizadas, debido a su simplicidad, los datos se han introducido a través del propio código.

La evaluación de singularidades se ha gestionado de la misma forma que la explicada en el Capítulo 5 y las integrales, en este caso, se han resuelto aplicando cuadratura de Gauss en un solo punto.

Como resultado, el programa crea un archivo 'VTK' de ParaView con los datos del run-up en cada punto definido por el archivo 'MESH.unv'.

12.2. Prueba de Validación 1 - Isla Homma

Esta prueba de validación ha sido utilizada en muchos artículos tales como [4], [30] y [31]. Consiste en el estudio de una isla, cuyas características geométricas se detallan en la figura 12.1, ante ondas de largo periodo producidas por tsunamis.

La malla de nodos utilizada queda definida en la figura 12.2. En ésta se definen anillos de 60 nodos para $r = a + c_i \cdot (b - a)$, donde c_i adquiere los valores 0.10, 0.25, 0.40, 0.55, 0.65, 0.80.

Todos los autores nombrados anteriormente hicieron sus ensayos con elementos cuadráticos y no constantes como es el caso de este proyecto, por lo tanto se espera que los resultados obtenidos difieran de los obtenidos por otros autores. Aunque se podría pensar que este problema se puede

¹En 'SALOME' se deben hacer grupos de líneas que formarán las fronteras del problema. Habría que hacer tantos grupos como contornos, con diferente coeficiente de reflexión.

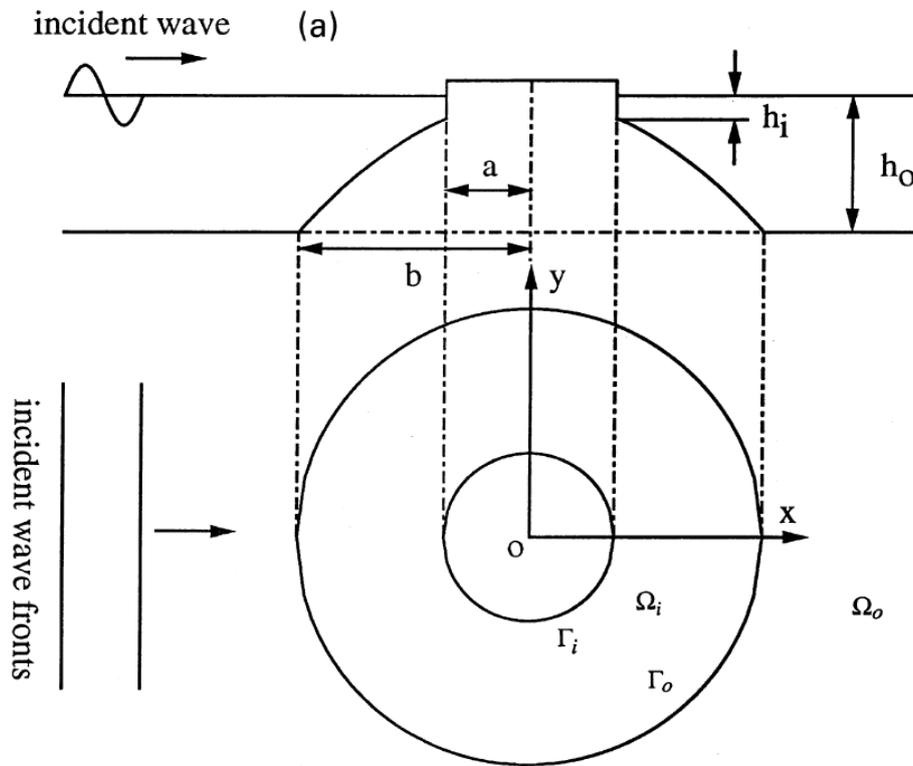


Figura 12.1: Geometría de la Isla Homma ($a = 10000m$, $b = 30000m$, $h_o = 4000m$ y $h_i = 444m$)

solventar incrementando el número de elementos, no es así, es más, al incrementar el número de elementos los resultados obtenidos difieren en mayor cantidad de los obtenidos por otros autores.

En la figura 12.3 constan los resultados obtenidos en [4] para varios periodos de onda. Por otra parte, en la figura 12.4 se muestran los resultados para elementos constantes. Para una mejor comparación, en la figura 12.5 se aprecian los resultados obtenidos por el autor y los obtenidos por Huan-Wen Liu para $T=410$ segundos. Se observa que, los zonas de máxima y mínima amplificación se producen en los mismos puntos aproximadamente, sin embargo, los picos son más pronunciados con elementos constantes.

Para este ejemplo, en el que las pendientes son muy pequeñas, el hecho de incorporar o no las dos componentes llamadas $U1$ y $U2$ en el Capítulo 5 no supone cambios en los resultados.

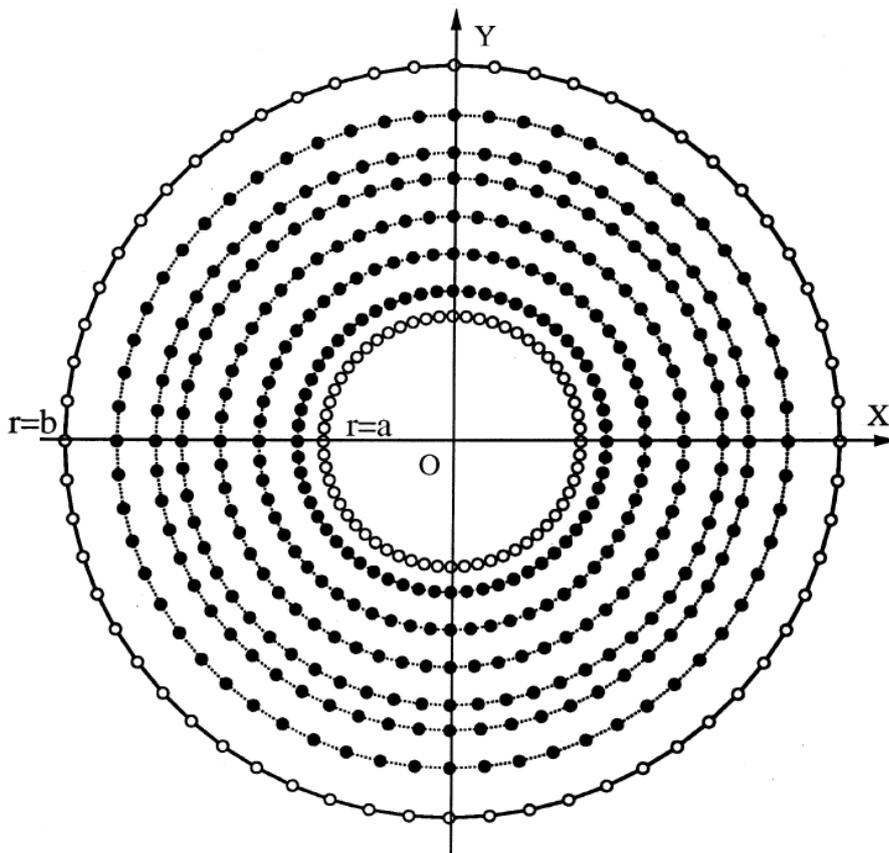


Figura 12.2: Definición de los nodos utilizados para la Isla Homma

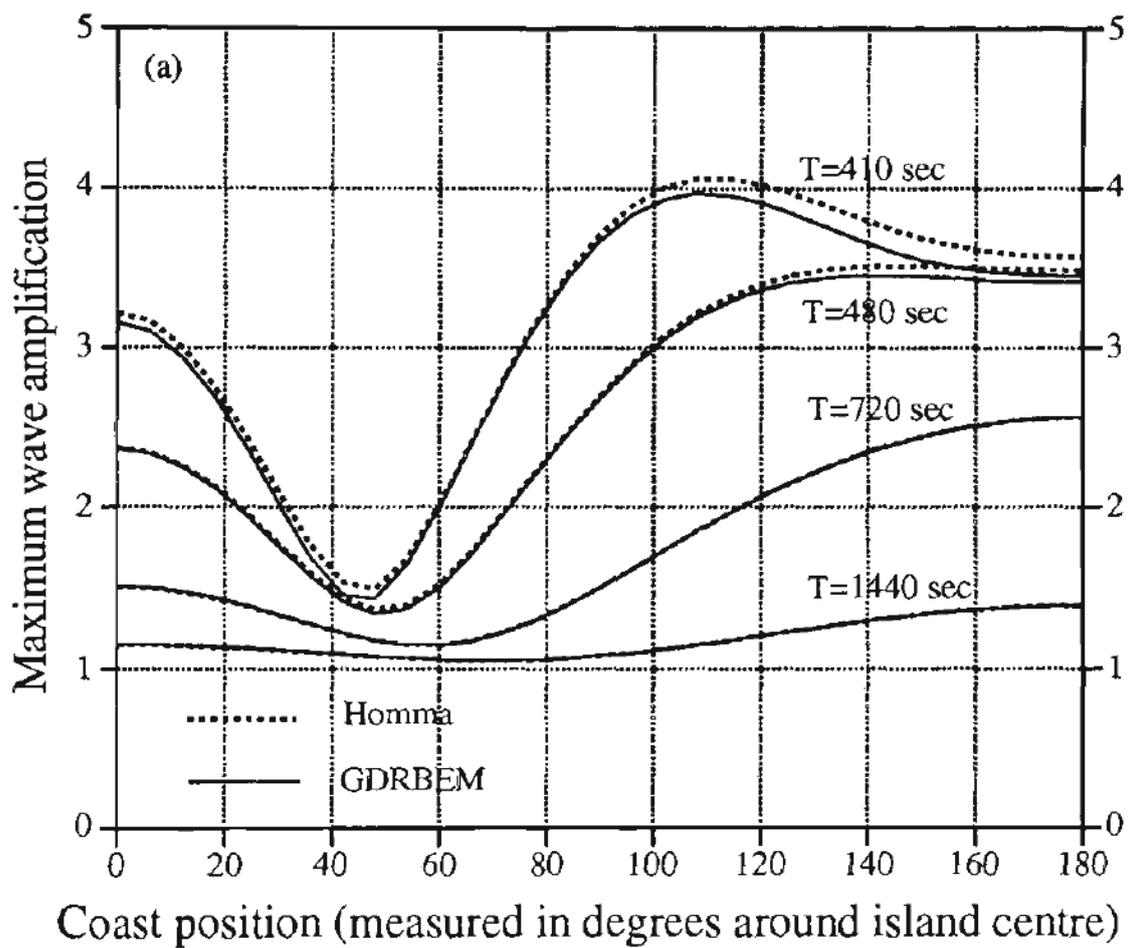


Figura 12.3: Resultados de la prueba de la Isla Homma obtenidos en [4].

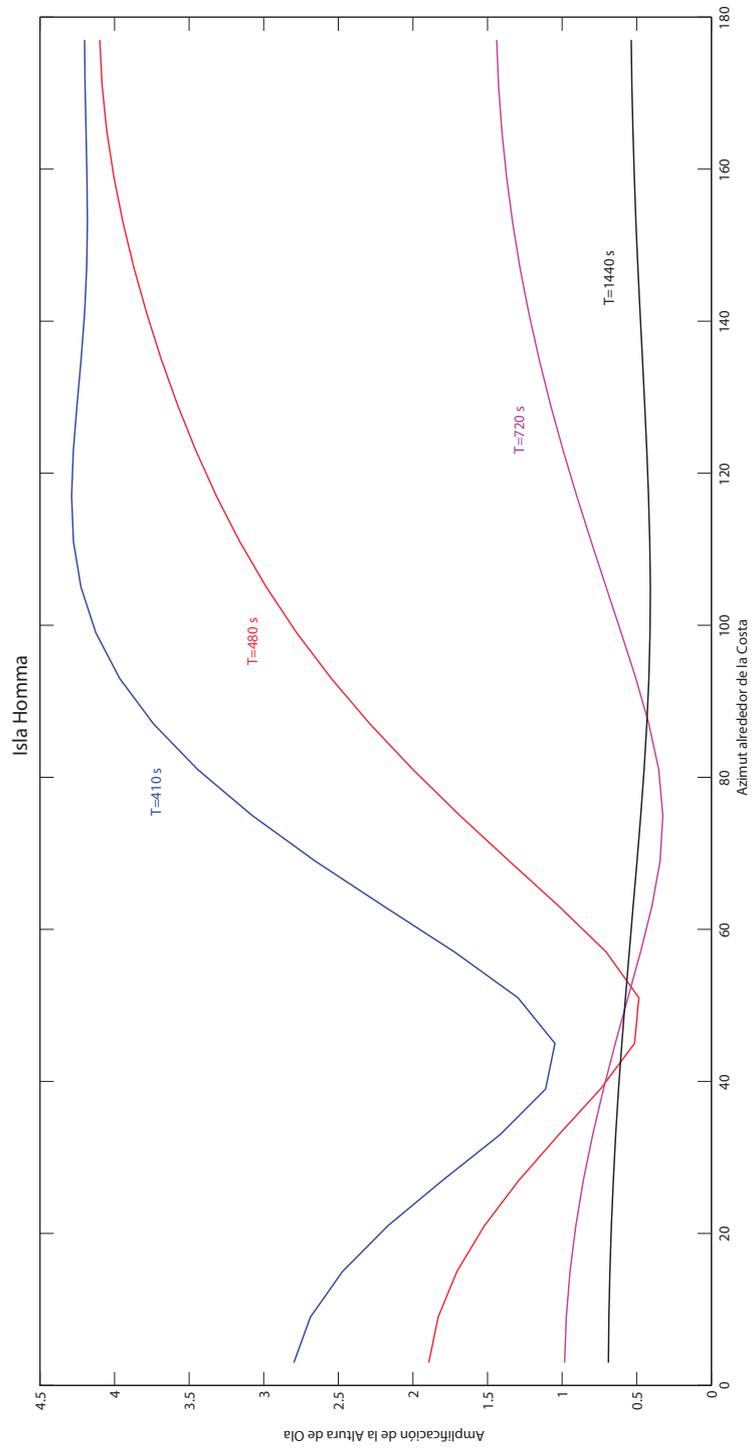


Figura 12.4: Resultados de la prueba de la Isla Homma con elementos constantes.

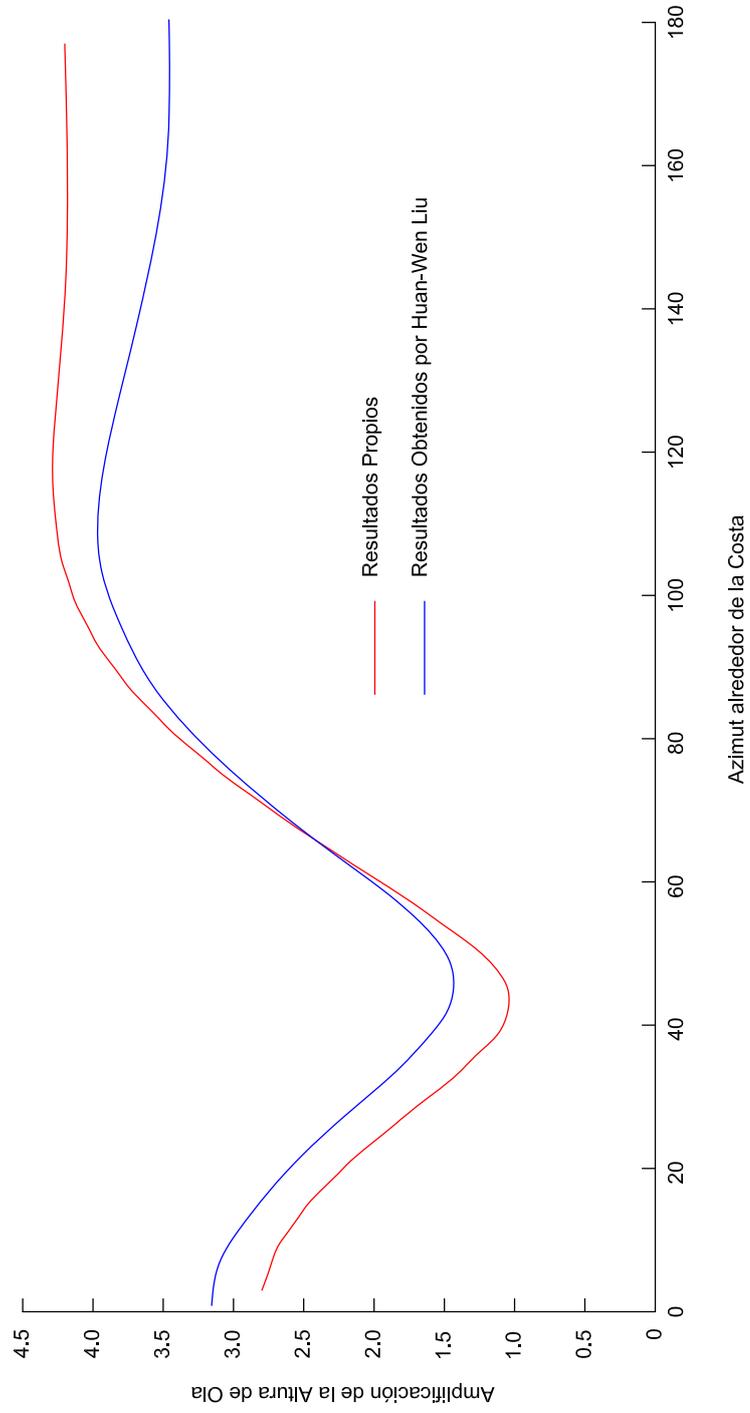


Figura 12.5: Comparación de los resultados obtenidos por el autor y los obtenidos por Huan-Wen Liu para $T=410$ segundos.

12.3. Prueba de Validación 2 - Experimento de Kyung Doug Suh

Esta prueba está basada en un experimento realizado por Kyung Doug Suh. Dicho experimento, que viene detallado en [46], se realizó en un tanque de olas. En él se situó un banco de arena de 0.45 m cuya profundidad, dependiente del radio, viene dada por la siguiente fórmula:

$$h = 0.3 - 0.18 \cdot \left[1 - \left(\frac{r}{0.45} \right)^2 \right] \quad (12.1)$$

Kyung Doug Suh estudió la diferencia entre los resultados experimentales obtenidos en el tanque de oleaje y los resultados al resolver el mismo problema mediante el método de los elementos finitos. Analizó los resultados para varias longitudes de onda, las correspondientes a $k \cdot h = 1$, $k \cdot h = 2$ y $k \cdot h = 3$. La altura de ola fue común para todos los casos e igual a 0.03 metros.

En este proyecto se comparará los resultados obtenidos para $k \cdot h = 3$. Los distintos resultados, mediante diferentes métodos, obtenidos por Kyung Doug Suh se muestran en la figura 12.6.

Por otra parte, en la figura 12.7 se detallan los resultados obtenidos para el intervalo de X/R entre -1 y 1. En pos de una mejor comparación, en la figura 12.8 se aprecian los resultados obtenidos por el autor y los resultados experimentales obtenidos por Kyung Doug Suh. Para los resultados correspondientes a $X/R = -1$ y $X/R = 1$ se observa un valor mayor al obtenido por K.D.Suh. Esto puede ser debido a que, en la malla introducida, la pendiente en los puntos más alejados del centro del banco de arena es 0 y los puntos en cuestión son los primeros en tener una pendiente no nula; siendo este salto bastante pronunciado y ocasionando un pico no deseado en los resultados. El resto de datos sí se ajustan de manera más o menos adecuada a lo esperado. Además, se observa que los resultados obtenidos teniendo en cuenta los términos U1 y U2 son mejores. A cambio de lo que ocurría en la prueba anterior, en este caso, las pendientes son más pronunciadas, por lo que éstos términos tienen mayor importancia.

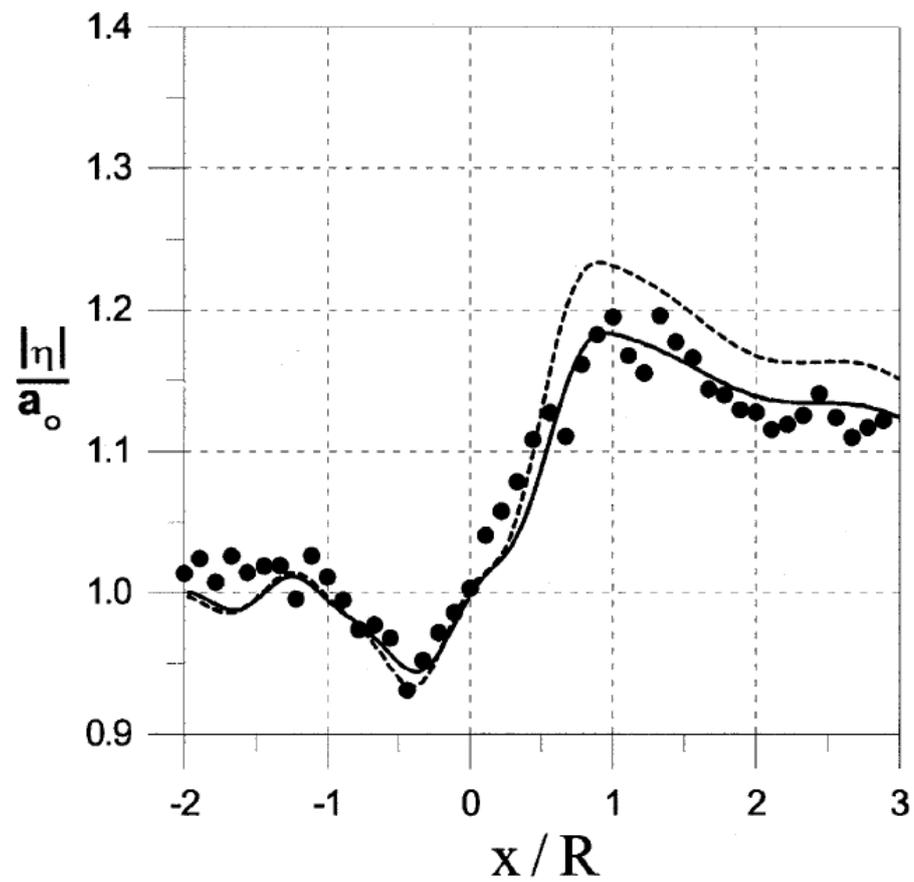


Figura 12.6: Resultados del experimento de Kyung Doug Suh para $k \cdot h = 3$. Puntos = Resultados Experimentales; Línea Continua = Resultados con el Modelo MMSE; Línea Discontinua = Resultados con el Modelo MSE

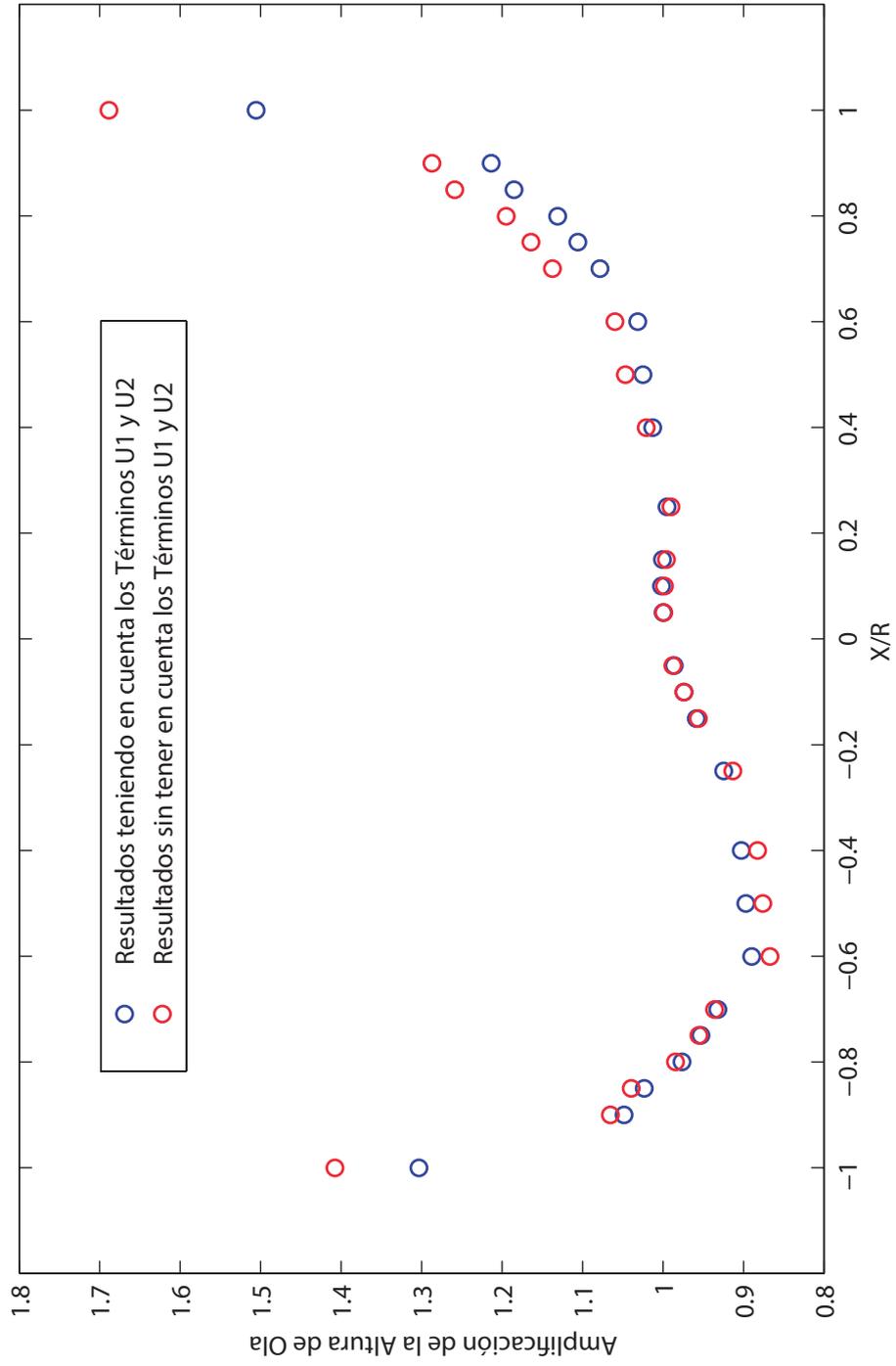


Figura 12.7: Resultados obtenidos de la prueba con el Experimento de Kyung Doug Suh.

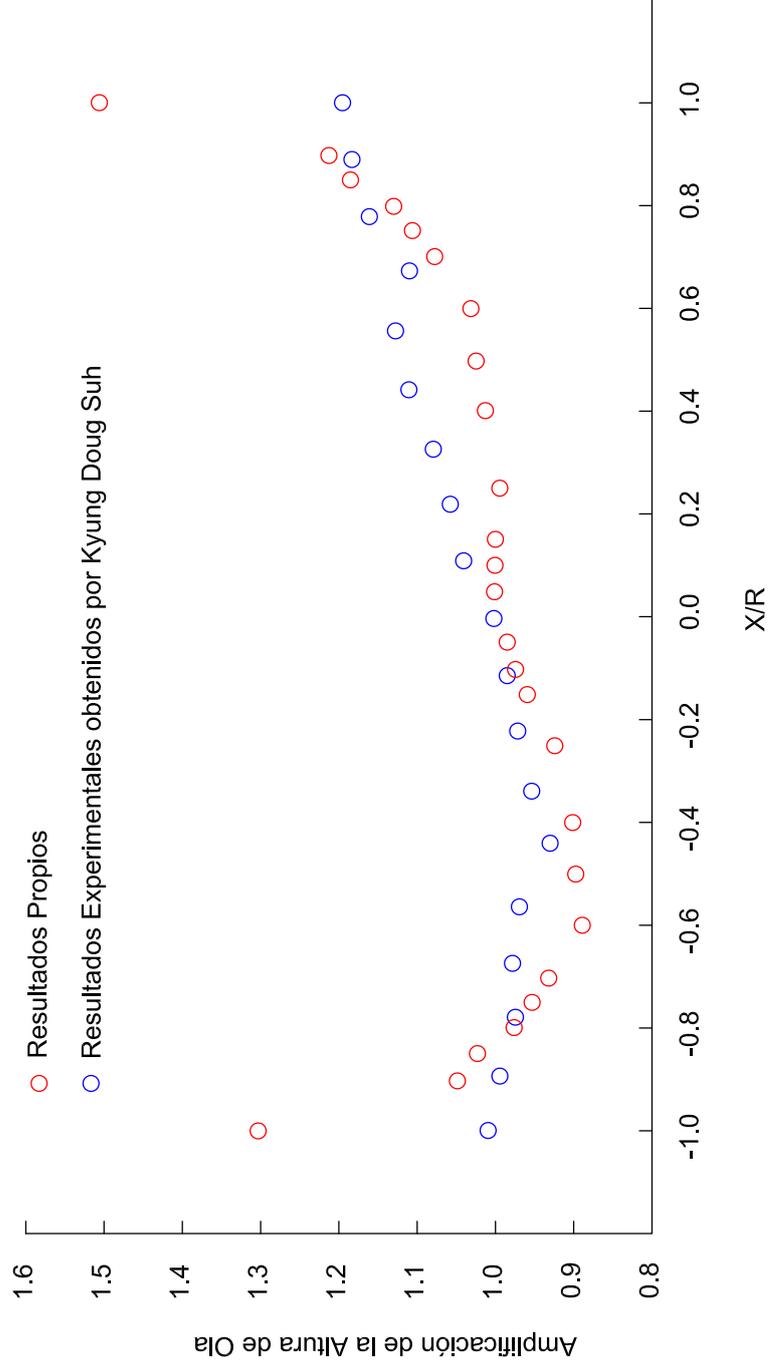


Figura 12.8: Comparación de los resultados obtenidos por el autor y los resultados experimentales obtenidos por Kyung Doug Suh

12.4. Conclusiones

Como se ha podido comprobar en ambas pruebas, este método de resolución tiene ventajas e inconvenientes. Por un lado, con este método se reduce la dimensionalidad del problema de R^3 a R^2 ya que se estudian dominios tridimensionales a través de nodos y contornos lineales. Huan-Wen Liu, en su tesis doctoral [4], ya mencionó que el mismo problema que él resolvió con 960 elementos cuadráticos, lo resolvió Houston en 1981 con la supercomputadora CRAY-1 y requirió 21120 elementos finitos. Como se podrá imaginar el lector, el coste computacional disminuye bastante para el algoritmo presentado.

Por otro lado, los aspectos negativos que se concluyen en este proyecto son la extrema sensibilidad de los resultados a la malla de nodos usada. El hecho de poner demasiados nodos o poner menos, incrementa el error cometido, siendo necesarias varias pruebas con diferentes mallas para converger a la solución real. En segundo lugar, se debe mencionar que los elementos constantes dan resultados muy pobres y es necesario implementar elementos de mayor orden. Por último, otro gran factor a estudiar son las funciones de interpolación; es posible que, dependiendo del problema de estudio, el uso de unas funciones u otras den mejores resultados.

CAPÍTULO 13

Conclusiones, Aportaciones y Desarrollo Futuro

13.1. Conclusiones

Las conclusiones finales que se pueden obtener de este proyecto son las siguientes:

- La aplicación del GMRES supone una disminución del tiempo de computación con respecto al tradicional método de Gauss. Dicha reducción es proporcional a $N^{0.7}$, siendo N el número de ecuaciones del sistema.
- La introducción de algoritmos paralelos procesados en la GPU y programados en CUDA da pie a una reducción adicional en los tiempos de computación, que en este caso es proporcional a $N^{0.3}$, siendo N el número de ecuaciones del sistema.
- Las pruebas de verificación del BEM3D demuestran un buen ajuste con los resultados teóricos, salvo desviaciones puntuales que pueden ser debidas a errores introducidos por la malla.
- Las pruebas de verificación del programa que resuelve la ecuación de Mild-Slope con fondo constante demuestran unos buenos resultados para elementos constantes, no siendo necesaria la introducción de elementos de mayor orden.
- Las pruebas de verificación del programa que resuelve la ecuación de Mild-Slope con fondo variable con elementos constantes muestran la necesidad de introducir elementos de mayor orden para la obtención de mejores resultados, sobre todo para oleaje de alta frecuencia.

13.2. Aportaciones

Las aportaciones que se proporcionan en este documento se enumeran seguidamente:

- Códigos en C++ para la resolución de problemas potenciales en 2D y 3D así como algoritmos para resolver problemas de propagación y difracción de ondas oceánicas con fondo variable y fondo constante.
- Algoritmo GMRES escrito en C++, como alternativa al método de Gauss, para la resolución de sistema lineales de ecuaciones.
- Código escrito en CUDA-C para la aceleración del GMRES, más concretamente para la aceleración de las multiplicaciones matriz-vector.
- Algoritmo escrito en C++ para la lectura de ficheros 'UNV', los cuales contienen la información de la malla de estudio.
- Algoritmo escrito en C++ para la exportación de resultados en ficheros 'VTK' (fase de postproceso). Dicho fichero puede ser leído por el software libre 'ParaView'.

13.3. Desarrollo Futuro

Los aspectos relacionados con este proyecto que se podrían investigar en el futuro son:

- Desarrollo de códigos en C++ que incorporen clases, de tal forma que, por ejemplo, no sea necesario un programa para resolver problemas potenciales 3D y otro para problemas potenciales

2D; sino que, dependiendo de la malla introducida se resuelva en 2D ó 3D (automáticamente se elegiría la función de Green adecuada, la evaluación de singularidades correspondiente etc. . .).

- Implementación de CUDA-C en otras partes del código, como en el caso del cálculo de las integrales que forman las matrices G y H propias del Método de los Elementos de Contorno.
- Desarrollo de un código que implemente el 'Fast Multipole Method' en conjunto con CUDA.
- Implementación de CUDA para varias GPUs con el objetivo de tener más memoria para el almacenamiento de matrices. De esta forma se reducirían los tiempos de copia de datos y, por consiguiente, el tiempo de computación total.

Bibliografía

- [1] Chiang C. Mei et al. *Theory and Applications of Ocean Surface Waves. Part 1: Linear Aspects*. World Scientific, 2005.
- [2] Boris Stok y Nikilaj Mole. Coupling FEM and BEM for computationally efficient solutions of multi-physics and multi-domain problems. *Engineering Computations*, 22:711–738, 2005.
- [3] Michael Isaacson y Shiqin Qu. Waves in a harbour with partially reflecting boundaries. *Coastal Engineering*, 14:193–214, 1990.
- [4] Huan-Wen Liu. *Numerical modelling of the propagation of ocean waves*. PhD thesis, University of Wollongong, 2001.
- [5] Walter Ritz. Über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen physik. *Journal für die Reine und Angewandte Mathematik*, 135:1–61, 1909.
- [6] C.A. Brebbia y J.Dominguez. *Boundary Elements: An Introductory Course*. WITPRESS Computational Mechanics Publications, 1998.
- [7] Stephen Kirkup. *The Boundary Element Method in Acoustics*. Integrated Sound Software, 2007.
- [8] Stephen Kirkup. *Solution of exterior acoustic problems by the boundary element method*. PhD thesis, Brighton Polytechnic, 1989.
- [9] Gernot Beer et al. *The Boundary Element Method with programming*. SpringerWienNewYork, 2008.
- [10] Alok Sutradhar et al. *Symmetric Galerkin Boundary Element Method*. Springer, 2008.
- [11] Y.J. Liu y N. Nishimura. The fast multipole boundary element method ofr potential problems: A tutorial. *Engineering Analysis with Boundary Elements*, 30:371–381, 2006.
- [12] Yijun Liu. *Fast Multipole Boundary Element Method. Theory and Applications in Engineering*. Cambridge, 2009.
- [13] I. Benedetti et al. A fast 3d dual boundary element method based on hierarchical matrices. *International Journal of Solids and Structures*, 45:2355–2376, 2008.
- [14] Y.Saad y M. Schultz. *GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems*. SIAM J. Sci. Stat. Comput, 1986.
- [15] M. Dolores García León. *Estrategias para la resolución de grandes sistemas de ecuaciones lineales. Métodos de Cuasi-Mínimo Residuo Modificados*. PhD thesis, Universidad de Las Palmas de Gran Canaria - Departamento de Matemáticas, 2003.
- [16] Jason Sanders y Edward Kandrot. *CUDA BY EXAMPLE*. Addison-Wesley, 2010.
- [17] Shoji Hamada. Gpu-accelerated indirect boundary element method for voxel model analyses with fast multipole method. *Computer Physics Communications*, 182:1161–1168, 2011.
- [18] Toru Takahashi y Tsuyoshi Hamada. Gpu-accelerated boundary element method ofr helmholtz equation in three dimensions. *International Journal for Numerical Methods in Engineering*, 80:1295–1321, 2009.
- [19] C.M. Linton y P. McIver. *Mathematical Techniques for Wave/Structure Interactions*. Champman and Hall/CRC, 2001.
- [20] Turgut 'Sarp' Sarpkaya. *Wave forces on Offshore Structures*. Cambridge, 2010.

- [21] M. Rahman y D.D.Bhatta. Evaluation of added mass and damping coefficient of an oscillating circular cylinder. *Appl. Math. Modelling*, 17, 1993.
- [22] Ronald W. Yeung. Added mass and damping of a vertical cylinder in finite-depth waters. *Applied Ocean Research*, 3, 1981.
- [23] Dambaru D. Bhatta. Surge Motion on a floating cylinder in water of finite depth. *International Journal of Mathematics and Mathematical Sciences*, 2003:3643–3656, 2003.
- [24] D. D. Bhatta et M.Rahman. On scattering and radiation problem for a cylinder in water of finite depth. *International Journal of Engineering Science*, 41:931–967, 2003.
- [25] Y. Drobyshevski. Hydrodynamic coefficients of a floating, truncated vertical cylinder in shallow water. *Ocean Engineering*, 31:269–304, 2004.
- [26] Simon Bogner et Ulrich Rude. Simulation of floating bodies with the lattice boltzmann method. *Computers and Mathematics with Applications*, 65:901–913, 2013.
- [27] C.F.Liu et al. A 3d time-domain method for predicting the wave-induced forces and motions of a floating body. *Ocean Engineering*, 38:2142–2150, 2011.
- [28] Michael H. Meylan. The time-dependent motion of a floating elastic or rigid body in two dimensions. *Applied Ocean Research*, 46:54–61, 2014.
- [29] P.W. Partridge y C.A. Brebbia. *The dual reciprocity boundary element method*. Computational Mechanics Publications, 1992.
- [30] Song-Ping Zhu et al. A general DRBEM model for wave refraction and diffraction. *Engineering Analysis with Boundary Elements*, 24:377–390, 2000.
- [31] Sung-Shan Hsiao et al. DRBEM Solution of Extended Mild-Slope Equation fo Waves around a Circular Island on a Polynomial Shoal. *Proceedings of the Nineteenth International Offshore and Polar Engineering Conference*, 2009.
- [32] P.G. Chamberlain y D. Porter. The modified mild-slope equation. *Fluid Mechanics*, 291:393–407, 1995.
- [33] Huanwen LIU. A modified GDRBEM model for wave scattering. *International Conference on Estuaries and Coasts*, 2003.
- [34] Kyung Doug Suh et al. Numerical Modeling of Mild Slope Equation with Finite Volume Method. *WSEAS TRANSACTIONS on MATHEMATICS*, 7, 2008.
- [35] Asuncin Baquerizo et al. *Fundamentos del movimiento oscilatorio*. Universidad de Granada, 2004.
- [36] R. Smith y T. Sprinks. Scattering of surface waves by a conical island. *Fluid Mechanics*, 72:373, 1975.
- [37] A.J. Nowak y C.A. Brebbia. The multiple-reciprocity method. A new approach for transforming B.E.M. Domain integrals to the boundary. *Engineering Analysis with Boundary Elements*, 6:164–168, 1989.
- [38] A.S. Muleshkov et al. Particular solutions of Helmholtz-type operators using higher order polyharmonic splines. *Computational Mechanics*, 23:411–419, 1999.
- [39] Y.L. Zhang y S.P. Zhu. On the choice of interpolation functions used in the dual-reciprocity boundary-element method. *Engineering Analysis with Boundary Elements*, 13:387–396, 1994.
- [40] Abramowitz y Stegun. *Handbook of Mathematical Functions. With formulas, graphs, and mathematical tables*. Dover, 1965.
- [41] Lloyd N. Trefethen y David Bau III. *Numerical Linear Algebra*. SIAM, 1997.
- [42] Y.Saad. *Iterative Methods for Sparse Linear Systems*. PWS PUBLISHING COMPANY, 1996.
- [43] W.E. Arnoldi. *The Principle of Minimized Iteration in the Solution of the Matrix Eigenvalue Problem*. Quart.Appl.Math., 1951.

- [44] Henk van der Vorst. *Iterative Krylov Methods for Large Linear System*. Cambridge, 2003.
- [45] J.París et al. Métodos Semi-Iterativos para Grandes Sistemas de Ecuaciones Lineales: GMRES.
- [46] Kyung Doug Suh et al. Experimental verification of horizontal two-dimensional modified mild-slope equation model. *Coast Engineering*, 44:1–12, 2001.



**E.T.S. INGENIERÍA DE CAMINOS, CANALES Y PUERTOS
UNIVERSIDAD DE GRANADA**