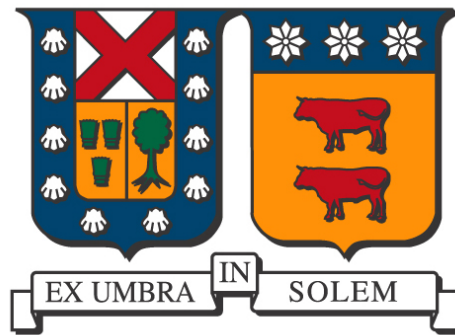


**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA**

**DEPARTAMENTO DE INGENIERÍA MECÁNICA**

**VALPARAÍSO - CHILE**



**EMULACIÓN NUMÉRICO-COMPUTACIONAL DEL  
COMPORTAMIENTO DEL CAMPO MAGNÉTICO  
ELECTROESTÁTICO EN EL BORDE DE MICROHILOS  
FERROMAGNÉTICOS.**

**RENÉ EDUARDO VELÁSQUEZ CÁRCAMO**

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO  
CIVIL MECÁNICO MENCIÓN ENERGÍA**

**PROFESOR GUÍA: Ph. D. Ing. CHRISTOPHER COOPER VILLAGRÁN**

**PROFESOR CORREFERENTE: Ph. D. Ing. ALEJANDRO PACHECO SANJUAN**

**NOVIEMBRE - 2018**



# Agradecimientos

Al culminar la etapa universitaria en la entrega de este trabajo es inevitable detenerse a agradecer a cada una de las personas que me acompañó durante el trayecto, durante cada parte de él. En primer lugar a mi familia por darme su apoyo en las decisiones tomadas. A mis amigos y compañeros con quienes compartí momentos inolvidables durante la época universitaria. No puedo dejar afuera a la gente que me ayudó a sacar este trabajo adelante, al profesor Christopher Cooper y a mi compañero Milán Ungerer, a quienes pude recurrir en momentos de profunda duda. A todos, muchas gracias.

# Contenido

<b>1</b>	<b>Introducción.</b>	<b>1</b>
<b>2</b>	<b>Fundamentos teóricos matemáticos.</b>	<b>2</b>
2.1	Cálculo vectorial. . . . .	2
2.2	Cálculo integral. . . . .	5
2.3	Ecuaciones Diferenciales. . . . .	5
2.3.1	Ecuación de Laplace. . . . .	5
2.3.2	Ecuación de Helmholtz. . . . .	6
<b>3</b>	<b>Fundamentos teóricos eléctricos.</b>	<b>7</b>
3.1	Campo eléctrico y campo magnético. . . . .	7
3.1.1	Campo eléctrico. . . . .	7
3.1.2	Campo magnético. . . . .	8
3.1.3	Leyes de Maxwell. . . . .	9
<b>4</b>	<b>Método de elementos de borde.</b>	<b>12</b>
<b>5</b>	<b>Biblioteca bempp.</b>	<b>19</b>
5.1	Estructura. . . . .	19
5.1.1	Algoritmo y explicación. . . . .	20
<b>6</b>	<b>Planteamiento del problema</b>	<b>24</b>
<b>7</b>	<b>Condiciones del problema.</b>	<b>25</b>
7.1	Creación de mallas. . . . .	31
7.2	Desarrollo del problema. . . . .	33
<b>8</b>	<b>Conclusiones.</b>	<b>44</b>
<b>9</b>	<b>Anexos</b>	<b>45</b>
9.1	Anexo I: Código 1. . . . .	45

# Indice de figuras

2.1	Representación y suma de vectores . . . . .	2
2.2	Campo vectorial. (Fuente: [2]) . . . . .	3
2.3	Campo vectorial. (Fuente: [4]) . . . . .	4
3.1	Líneas de campo eléctrico. (Fuente: [7]) . . . . .	7
3.2	Campo magnético por una carga puntual. (Fuente: [7]) . . . . .	8
3.3	Campo magnético por una corriente eléctrica. (Fuente: [7]) . . . . .	9
3.4	Líneas de campo magnético. (Fuente: [7]) . . . . .	10
3.5	Líneas de campo magnético. (Fuente: [7]) . . . . .	11
3.6	Propagación de onda electromagnética. (Fuente: [7]) . . . . .	11
4.1	Diferencias en la discretización en BEM y FEM en 3D. . . . .	13
4.2	Puntos de frontera para los casos 2D y 3D (Fuente:[5]) . . . . .	14
4.3	Diferentes tipos de elementos de frontera. (Fuente:[5]) . . . . .	15
4.4	Diferencias en la discretización en BEM y FEM en 2D (Fuente:[5]). . . . .	17
4.5	Elementos de borde y puntos internos para el estudio . . . . .	18
5.1	Modulos de bempp (Fuente:[8]) . . . . .	20
5.2	Algoritmo resolución bempp . . . . .	21
5.3	Intersección de elementos lineales continuos (Fuente:[5]) . . . . .	22
7.1	Situación del problema . . . . .	25
7.2	Onda incidente perturbada. . . . .	26
7.3	Micro-hilo no tiene efecto sobre la onda. . . . .	27
7.4	Cilindro en SolidWorks . . . . .	31
7.5	Malla vista completa . . . . .	31
7.6	Malla detalle tapa . . . . .	32
7.7	Malla detalle elementos . . . . .	32
7.1	Malla con ajustes predeterminados en <i>plotly</i> . . . . .	34
7.2	Malla con eje $z$ no 'comprimido' en <i>plotly</i> . . . . .	34
7.3	Solución situacion 1 . . . . .	37
7.4	Solución situacion 1 . . . . .	39

## Resumen

**Palabras claves:** Método de Elementos de Borde, BEM, BEM++, Micro-hilos, Sensor, Dispersión Múltiple.

## Abstract

**Keywords:** Boundary Element Method, BEM, BEM++, Micro-wires, Sensor, Multiple Scattering.

## Glosario

- **Dispersión electromagnética:** proceso en que una onda electromagnética incidente se desvía de su trayectoria a causa de pasar por una no uniformidad. Principalmente se distinguen los procesos de reflexión y refracción.
- **Número de onda:** se puede definir como el número de ondas que existen en una determinada distancia. Frecuencia espacial.
- **Índice de refracción:** número adimensional que describe cómo se propaga una onda electromagnética en un medio.
- **Material dieléctrico:** material que exhibe polarización frente un campo eléctrico.
- **Material conductor:** material que permite fluir sus electrones frente a un campo eléctrico. La capacidad de fluir libremente depende de la resistencia y conductividad del material.
- **Microondas:** forma de radiación electromagnética con longitudes de onda desde 1[m] a 1[mm], o con frecuencias desde los 300[MHz] hasta los 30[GHz].
- **Permitividad:** medida de resistencia encontrada cuando se forma un campo eléctrico en un medio.
- **Permeabilidad (magnética):** medida de la capacidad de un material para formar un campo magnético en él.
- **Permitividad relativa:** representación de la permitividad de un medio como un cociente de la permitividad absoluta y la permitividad del vacío.
- **Permeabilidad relativa:** representación de la permeabilidad de un medio como un cociente de la permeabilidad absoluta y la permeabilidad del vacío.
- **Compósito:** material compuesto por dos o más materiales con propiedades muy diferentes, que cuando se encuentran combinados presentan características distintas a las de los materiales individuales.
- **SEM:** un microscopio de escáner por electrones (scanning electron microscope) es una clase de microscopio que produce imágenes de una muestra a través de escaneo por un haz enfocado de electrones.
- **Polarización:** En ondas es la habilidad de oscilar en más de una dirección. Para el caso de materiales (dieléctricos) significa la redistribución de electrones al interior de este producto de un campo eléctrico.
- **Impedancia:** es el cociente complejo entre el voltaje y la corriente generada. Caso similar a la resistencia eléctrica con la diferencia que la resistencia sólo tiene magnitud y la impedancia magnitud y fase.
- **Magneto impedancia:** cambio de la impedancia producto de un campo magnético.
- **Giant Magnetoimpedance (GMI):** corresponde a grandes variaciones en la impedancia eléctrica exhibidas por algunos materiales en función de un campo magnético externo.



- **Anisotropía magnética:** dependencia direccional de las propiedades magnéticas de un material.
- **Impedancia de superficie:** corresponde a la impedancia presentada por un conductor en su superficie como aproximación a su funcionamiento interno.
- **Magneto estricción:** propiedad de los materiales ferromagnéticos que produce su cambio de forma o dimensiones durante el proceso de magnetización. Estos cambios serán producidos producto de un campo magnético hasta alcanzar su punto de saturación.
- **Análisis numérico:** estudio de algoritmos para aproximar problemas matemáticos complejos.
- **Boundary Element Method (BEM):** método numérico para aproximar ecuaciones diferenciales por medio de una formulación integral en el borde del dominio.
- **Python:** lenguaje de programación de alto nivel para uso variado.
- **BEM++:** librería abierta para Python que tiene como finalidad resolver problemas a través del método de elementos de borde.

# Capítulo 1

## Introducción.

Durante los últimos años el desarrollo de nuevos métodos de resolución de ecuaciones diferenciales ha llamado la atención de la comunidad científica, siendo el método de elementos finitos el más conocido, utilizado y estudiado; este método realiza una discretización volumétrica completa del cuerpo de estudio, por lo que es muy útil para geometrías complejas. Sin embargo, este método no es el más eficiente en todos los casos.

Dentro de la gama de métodos con los que es posible resolver ecuaciones diferenciales parciales, el método de elementos de borde (conocido también como método de elementos de contorno o de frontera), BEM por sus siglas en inglés, es un método particularmente eficiente cuando tratamos con algo de homogeneidad dentro del cuerpo de estudio, esto usualmente sucede en mecánica de fluidos, acústica, electromagnetismo y algunos problemas de mecánica.

Si comparamos ambos métodos vemos una considerable cantidad de software que utilizan el método de elementos finitos para la resolución de problemas, sin embargo, BEM está aún 'en pañales', en etapa de desarrollo, por lo que los avances en la implementación del método, mejoras en la velocidad de resolución, mayor precisión a menor costo computacional, entre otros factores, son objetivos a lograr dentro del corto y mediano plazo.

## Capítulo 2

# Fundamentos teóricos matemáticos.

### 2.1 Cálculo vectorial.

La palabra vector significa 'que conduce', físicamente hablando es un elemento que posee magnitud, dirección y sentido. Siendo la magnitud la longitud del vector, la dirección la orientación de la flecha, el sentido indica hacia el lado donde se dirige el vector. Su expresión geométrica es la de una recta.

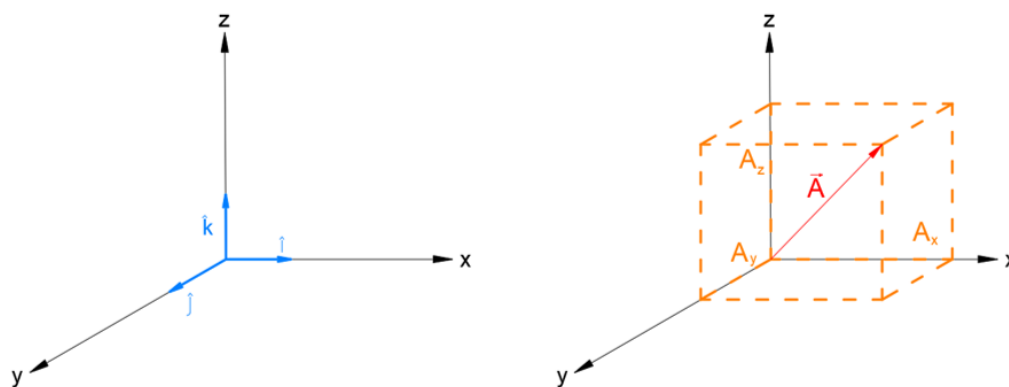


Figura 2.1: Representación y suma de vectores

Además de los vectores también existen los escalares para asignar magnitudes, estos solo poseen magnitud. Los vectores pueden sumarse y restarse, es importante notar que se debe tomar en cuenta la dirección y sentido.

El sistema de referencia para ubicar el vector es usualmente el sistema de coordenadas cartesianas, con ejes denotados como  $x$ ,  $y$  y  $z$ . Para mayor facilidad de escritura se asignan vectores unitarios a cada uno de los ejes, siendo  $\hat{i}$ ,  $\hat{j}$  y  $\hat{k}$

Ahora denotemos un vector  $\vec{r} \in \mathbb{R}^3$ , usando coordenadas cartesianas lo podemos denotar como  $\vec{r} = (x, y, z) = x\hat{i} + y\hat{j} + z\hat{k}$ .

Existen operaciones algebraicas posibles con los vectores, estas son:

Sea el vector  $\vec{A} = (A_x, A_y, A_z)$  y el vector  $\vec{B} = (B_x, B_y, B_z)$ :

- Suma de vectores  $\vec{A}$  y  $\vec{B}$ :

$$\vec{A} + \vec{B} = (A_x + B_x)\hat{i} + (A_y + B_y)\hat{j} + (A_z + B_z)\hat{k}$$

- Resta de vectores  $\vec{A}$  y  $\vec{B}$ :

$$\vec{A} - \vec{B} = (A_x - B_x)\hat{i} + (A_y - B_y)\hat{j} + (A_z - B_z)\hat{k}$$

- Multiplicación de un escalar  $k$  por un vector  $\vec{A}$ :

$$k\vec{A} = kA_x\hat{i} + kA_y\hat{j} + kA_z\hat{k}$$

- Producto punto entre vectores  $\vec{A}$  y  $\vec{B}$ :

$$\vec{A} \cdot \vec{B} = A_xB_x + A_yB_y + A_zB_z$$

- Producto cruz entre vectores  $\vec{A}$  y  $\vec{B}$ :

$$\vec{A} \times \vec{B} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ A_x & A_y & A_z \\ B_x & B_y & B_z \end{vmatrix}$$

Sea  $\Omega$  un espacio abierto en  $\mathbb{R}^3$  Llamaremos campo vectorial sobre  $\Omega$  a toda función

$$F : \Omega \subseteq \mathbb{R}^3 \rightarrow \mathbb{R}^3$$

Escribiendo en coordenadas cartesianas esto nos queda:

$$\vec{F}(x, y, z) = \vec{F}_1(x, y, z)\hat{i} + \vec{F}_2(x, y, z)\hat{j} + \vec{F}_3(x, y, z)\hat{k}$$

Entonces podríamos definir el campo vectorial como un campo que asocia un vector a cada punto del espacio.

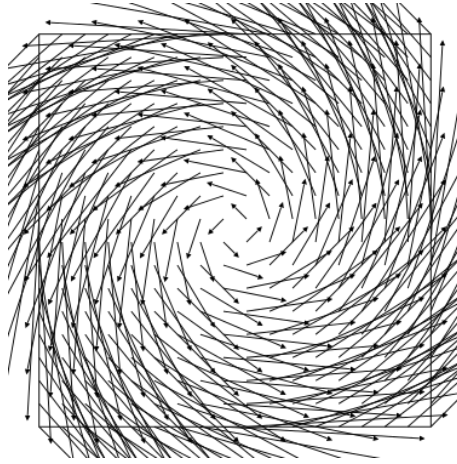


Figura 2.2: Campo vectorial. (Fuente: [2])

Para graficar estos campos usualmente se escalan los vectores en cada punto de manera que al mirar el campo completo el dibujo sea entendible, es decir escalamos los vectores para que quepan todos en el dibujo. Otra buena manera de graficar estos campos de forma que sean más fácil de interpretar por parte del lector es coloreando las líneas y establecer una escala de colores para poder interpretar de mejor manera.

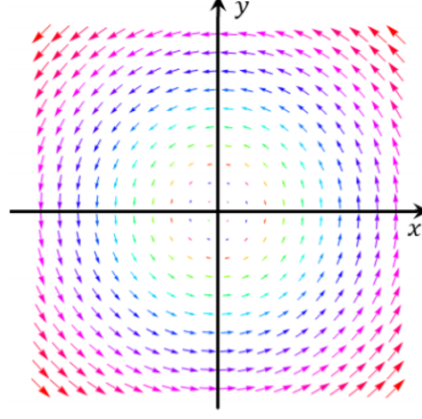


Figura 2.3: Campo vectorial. (Fuente: [4])

Es necesario también saber las operaciones posibles entre campos vectoriales, las fundamentales son 2. El gradiente y el rotacional del campo. Según las definiciones de citecamposvectoriales2: Sea  $f : A \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$  un campo vectorial. Suponiendo las condiciones necesarias de derivación definimos divergencia y rotor respectivamente como:

$$\text{div } F(x, y, z) = \frac{\partial F_1}{\partial x}(x, y, z) + \frac{\partial F_2}{\partial y}(x, y, z) + \frac{\partial F_3}{\partial z}(x, y, z)$$

$$\text{rot } F(x, y, z) = \left( \frac{\partial F_3}{\partial y} - \frac{\partial F_2}{\partial z}, \frac{\partial F_1}{\partial z} - \frac{\partial F_3}{\partial x}, \frac{\partial F_2}{\partial x} - \frac{\partial F_1}{\partial y} \right) = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ F_1 & F_2 & F_3 \end{vmatrix}$$

Otro tipo de notación también es utilizada, ocupando el simbolo de gradiente como un operador:

$$\nabla = \left( \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)$$

Entonces, reescribimos la divergencia y el rotor utilizando el producto punto y cruz con nuestro operador de la forma:

$$\nabla \cdot F = \left( \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right) \cdot (F_1, F_2, F_3) = \frac{\partial F_1}{\partial x} + \frac{\partial F_2}{\partial y} + \frac{\partial F_3}{\partial z} = \text{div } F$$

$$\nabla \times F = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ F_1 & F_2 & F_3 \end{vmatrix} = \left( \frac{\partial F_3}{\partial y} - \frac{\partial F_2}{\partial z}, \frac{\partial F_1}{\partial z} - \frac{\partial F_3}{\partial x}, \frac{\partial F_2}{\partial x} - \frac{\partial F_1}{\partial y} \right) = \text{rot } F$$

Este operador se puede aplicar sobre si mismo, este 'nuevo' operador se conoce como el 'operador laplaciano' y corresponde a la divergencia del gradiente, para representarlo se utiliza  $\Delta$  o  $\nabla^2$ :

$$\nabla \cdot (\nabla \cdot F) = \nabla^2 \cdot F = \frac{\partial^2 F_1}{\partial x^2} + \frac{\partial^2 F_2}{\partial y^2} + \frac{\partial^2 F_3}{\partial z^2}$$

Existen dos propiedades importantes que debemos tener en cuenta:

$$\nabla \times (\nabla \cdot F) = 0 \quad (2.1)$$

$$\nabla \cdot (fV) = f \cdot (\nabla V) - V \cdot \nabla f \quad (2.2)$$

Otro concepto de cálculo vectorial que debemos tener claro son los campos vectoriales conservativos los cuales son de vital importancia en el campo físico. Los campos vectoriales que pueden ser definidos bajo el gradiente de una función escalar  $f$  son llamados conservativos. Esto es  $\vec{F} = \vec{\nabla} f$ , además la función escalar  $f$  es conocida como la función potencial del campo  $\vec{F}$ .

En otras palabras, esto es, siendo el campo vectorial  $\vec{F}(x, y, z) = \vec{F}_1(x, y, z)\hat{i} + \vec{F}_2(x, y, z)\hat{j} + \vec{F}_3(x, y, z)\hat{k}$  la función potencial  $f(x, y, z)$  del campo  $\vec{F}$  está definida como:

$$\frac{\partial f}{\partial x}(x, y, z) = F_1(x, y, z), \quad \frac{\partial f}{\partial y}(x, y, z) = F_2(x, y, z), \quad \frac{\partial f}{\partial z}(x, y, z) = F_3(x, y, z)$$

## 2.2 Cálculo integral.

En BEM resolveremos ecuaciones diferenciales, en donde a las funciones diferenciales se les aplica muchas veces el concepto de integral. Si bien en este documento no nos adentraremos en demasía en el tema, los teoremas que se presentan a continuación serán de gran utilidad para resolver la problemática planteada:

- Teorema fundamental del cálculo: Sea  $f$  una función escalar.

$$\int_a^b \frac{df(x)}{dx} dx = f(b) - f(a) \quad (2.1)$$

- Teorema fundamental del gradiente: Sea  $F$  una función vectorial.

$$\int_a^b \nabla F ds = F(b) - F(a) \quad (2.2)$$

- Teorema de Gauss:

$$\int_{\Omega} (\nabla \cdot F) d\Omega = \oint_{\Gamma} n \cdot F d\Gamma \quad (2.3)$$

- Integración por partes:

$$\int u dv = uv - \int v du \quad (2.4)$$

## 2.3 Ecuaciones Diferenciales.

En este documento investigaremos el comportamiento de dos ecuaciones de gran importancia en el modelamiento físico de varios fenómenos, por lo que es necesario tener un poco de entendimiento sobre su comportamiento matemático.

### 2.3.1 Ecuación de Laplace.

Esta es una ecuación de derivadas parciales de tipo elíptico. Es un caso particular de la ecuación de Helmholtz, sin embargo, la ecuación de Laplace junto con la ecuación de Poisson son los dos modelos más simples de las ecuaciones en derivadas parciales (EDP) de tipo elípticas.

La ecuación de Laplace en se puede escribir de distintas formas: Escrita con el operador nábla  $\nabla$

$$\nabla^2 \phi = 0 \quad (2.5)$$

Escrita en derivadas, para el caso tridimensional, en coordenadas cartesianas:

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} = 0 \quad (2.6)$$

Esta ecuación es la encargada de modelar el comportamiento de, por ejemplo, el potencial eléctrico en una región sin cargas.

### **2.3.2 Ecuación de Helmholtz.**

Esta ecuación, por otro lado, está compuesta por:

$$(\nabla^2 + k^2)\phi = 0 \quad (2.7)$$

## Capítulo 3

# Fundamentos teóricos eléctricos.

### 3.1 Campo eléctrico y campo magnético.

Es importante tener claros los conceptos de campo eléctrico y campo magnético antes de adentrarnos en el problema. En esta sección veremos la definición de ambos campos, sus propiedades y un esquema ilustrativo, para ayudar a entender de mejor manera estas ideas.

#### 3.1.1 Campo eléctrico.

Como definición el campo eléctrico es 'Campo físico que describe la interacción entre 2 cuerpos o sistemas con propiedades de naturaleza eléctrica'. Está representado por la letra  $E$ , también se podría definir como la fuerza eléctrica  $F_e$  por unidad de carga  $q$  o bien:

$$E = \frac{F}{q}$$

El campo eléctrico estático puede representarse esquemáticamente a través de líneas, conocidas como **líneas de campo**.

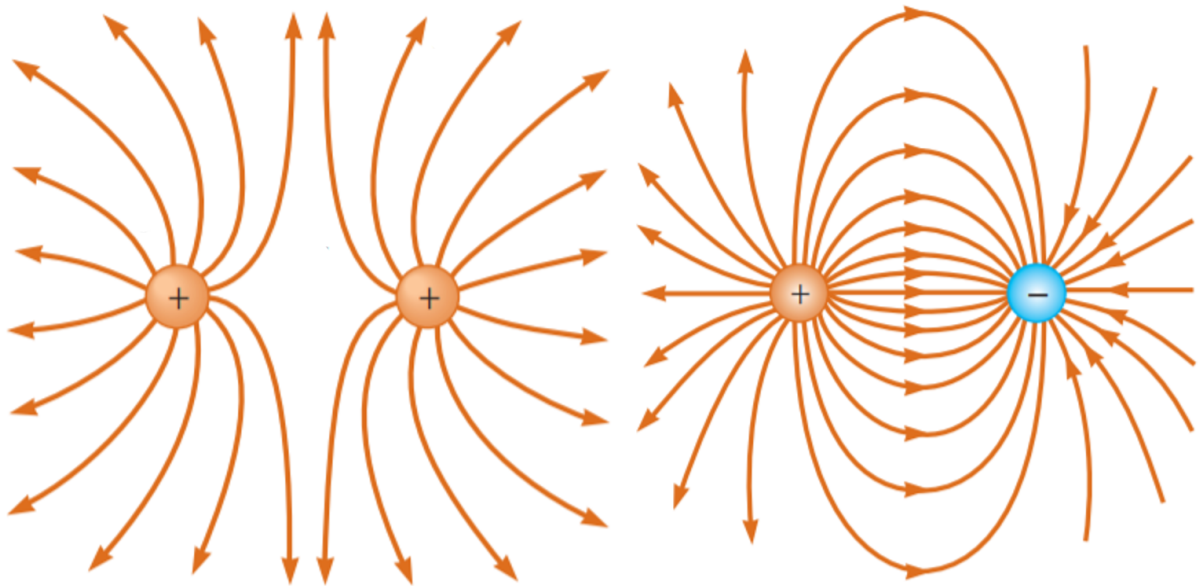


Figura 3.1: Líneas de campo eléctrico. (Fuente: [7])



Las líneas de campo eléctrico poseen las siguientes características:

- El vector de campo eléctrico en cualquier punto es tangente a la línea que pasa por dicho punto.
- Las líneas de campo no se cruzan.
- El número de líneas que salen de una carga positiva o entran en una carga negativa es proporcional a dicha carga.
- Las líneas de campo no pueden cortarse. De lo contrario en el punto de corte existirían dos vectores campo eléctrico distintos.
- A grandes distancias de un sistema de cargas, las líneas están igualmente espaciadas y son radiales, comportándose el sistema como una carga puntual.

### 3.1.2 Campo magnético.

El campo magnético, al igual que el campo eléctrico es una magnitud vectorial. Es generado por cargas en movimiento, estas pueden ser puntuales o un conjunto de cargas o en otras palabras, una corriente eléctrica. Su unidad de medida es el tesla ( $T$ ).

El movimiento de una carga puntual produce un campo magnético de la forma:

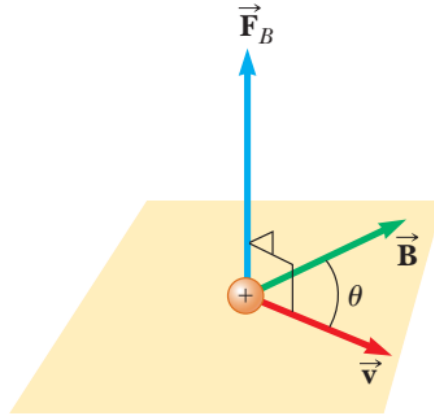


Figura 3.2: Campo magnético por una carga puntual. (Fuente: [7])

El campo magnético viene dado por:

$$\vec{B} = \frac{\mu_0}{4\pi} \frac{q}{r^2} \vec{v} \times \vec{u}_r$$

Donde  $q$  es la carga puntual que crea el campo,  $v$  es la velocidad de  $q$ ,  $r$  es la distancia desde  $q$  hasta  $P$ ,  $P$  es el punto donde estamos calculando el campo magnético,  $u_r$  es un vector unitario que va desde  $q$  hasta  $P$ ,  $\mu_0$  es una constante denominada permeabilidad del espacio libre. Su valor en el Sistema Internacional es  $\mu_0 = 4\pi 10^{-7} Tm/A$ .

Por otro lado, una corriente eléctrica son muchas cargas puntuales moviéndose en conjunto, por lo que también generan un campo magnético. En este caso el cálculo del campo será con elementos infinitesimales:

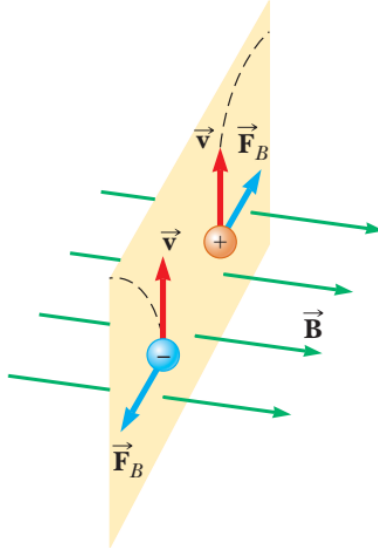


Figura 3.3: Campo magnético por una corriente eléctrica. (Fuente: [7])

En donde  $I$  es la intensidad de la corriente, dada por la fórmula  $I = q n v_d A$ , con  $n$  siendo la cantidad de cargas,  $A$  la sección del hilo y  $v_d$  la velocidad del desplazamiento.

$$d\vec{B} = \frac{\mu_0}{4\pi} \frac{I d\vec{l} \times \vec{u}_r}{r^2}$$

### 3.1.3 Leyes de Maxwell.

Como vimos anteriormente existen campos eléctricos y campos magnéticos, el conjunto de ambos campos se conoce como campo electromagnético. A través de este campo, los elementos que interactúan entre sí por medio de la electricidad o el magnetismo se comportan siguiendo ciertas reglas de como los elementos perturban al campo, o como el campo se perturba a sí mismo. Estas reglas son conocidas como las ecuaciones o leyes de Maxwell.

En un principio eran 8, luego se condensaron en 4 y se pueden escribir como:

$$\vec{\nabla} \cdot \vec{E} = \frac{\rho}{\epsilon_0} \quad (3.1)$$

$$\vec{\nabla} \cdot \vec{B} = 0 \quad (3.2)$$

$$\vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad (3.3)$$

$$\vec{\nabla} \times \vec{B} = \mu_0 \vec{J} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t} \quad (3.4)$$

Para entender un poco más las ecuaciones, pasaremos a explicarlas. La ecuación (3.1) también conocida como la **Ley de Gauss** nos dice básicamente que las cargas positivas serán fuentes y que las cargas negativas serán sumideros, tal cual como se ve en la imagen 3.1, en otras palabras, cargas del mismo signo repelen y cargas opuestas se atraen, notar que  $\rho$  es la densidad de carga y  $\epsilon_0$  es la permitividad eléctrica en el vacío. Esta ley también nos indica que el campo eléctrico decae con la distancia, lo hace a razón de  $1/r^2$ . La ecuación (3.2), algo así como la Ley de Gauss

del magnetismo, indica que no hay fuentes o sumideros de campo magnético, lo que no impide que hayan elementos que generen campos magnéticos, como vimos anteriormente. ¿Por qué es importante esto? Porque nos indica que el campo magnético siempre debe cerrarse sobre sí mismo. De forma práctica esto se ve al cortar un imán por la mitad, si bien en un inicio solo existe un polo sur y otro norte, al cortarlo en 2 cada trozo tiene su polo sur y norte respectivamente. Es decir, los monopolos no existen. La ecuación (3.3), también conocida como la **Ley de Faraday** nos dice que si un campo magnético cambia en el tiempo este 'activa' el campo eléctrico, por lo tanto no solo las cargas e imanes influyen en los campos sino que estos influyen entre sí, lo que nos lleva directamente a la ecuación (3.4) o la **Ley de Ampère-Maxwell** que también nos dice que si el campo eléctrico cambia en el tiempo o cargas moviéndose, es decir una corriente eléctrica denotada en la ecuación como  $\vec{J}$ , esto activará el campo magnético. Las constantes utilizadas en las ecuaciones son:

- Permitividad eléctrica en el vacío

$$\varepsilon_0 = 8.85 \times 10^{-12} \left[ \frac{C^2}{N \cdot m^2} \right]$$

- Permisividad magnética en el vacío

$$\mu_0 = 4\pi \times 10^{-7} \left[ \frac{N}{A^2} \right]$$

Combinando estas 4 leyes es válido decir que todos los fenómenos electromagnéticos que percibimos se pueden explicar.

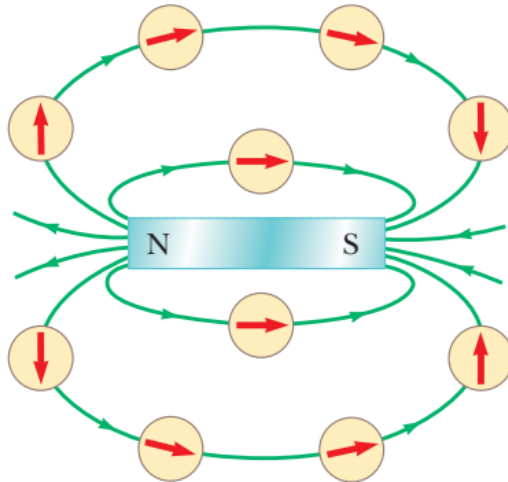


Figura 3.4: Líneas de campo magnético. (Fuente: [7])

Las ecuaciones descritas con anterioridad son dadas para el medio vacío. Para el caso en que los elementos se encuentren en un medio es necesario adaptar las propiedades del mismo. Se pueden encontrar una nueva relación para  $E$  y  $B$  a través de dos parámetros ya conocidos, como permitividad eléctrica y permisividad magnética. La relación es de la forma:

$$\vec{E} = \frac{\vec{D}}{\varepsilon}$$

$$\vec{B} = \mu \vec{H}$$

En donde  $D$  se define como la densidad el flujo eléctrico y  $H$  como la intensidad del campo magnético. Se dice que si la relación entre  $E/D$  y  $B/H$  estamos en presencia de un medio lineal. Esto permite que  $\varepsilon$  y  $\mu$  se representen de forma matricial. Si esta matriz se puede diagonalizar, es decir que que solo existan en la diagonal se habla de un medio isótropo y si además los elementos no son iguales se habla de un medio anisótropo. Un elemento isótropo es un elemento en el que sus cualidades físicas no dependen de la dirección en la que son examinadas, por el contrario en el anisótropo su dirección varía sus propiedades físicas.



Figura 3.5: Líneas de campo magnético. (Fuente: [7])

Es necesario notar que una onda electromagnética puede propagar a su vez un campo electro-magnético, el cual se puede descomponer de un campo eléctrico y uno magnético, los cuales son perpendiculares entre si y entre la dirección de propagación. Tal como se muestra en la figura siguiente:

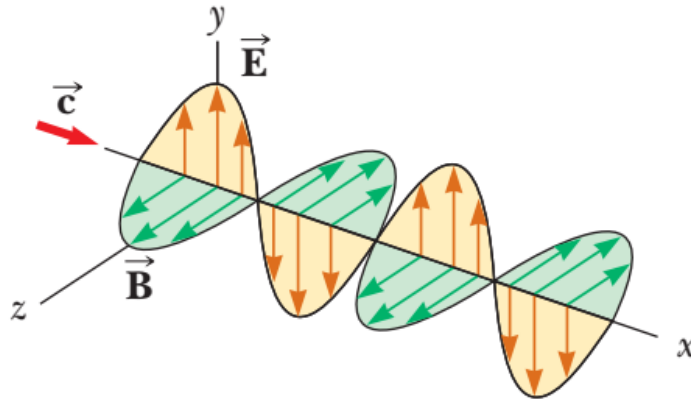


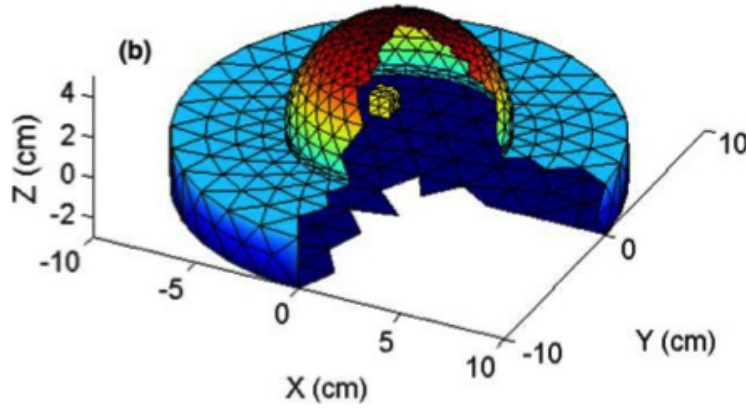
Figura 3.6: Propagación de onda electromagnética. (Fuente: [7])

## Capítulo 4

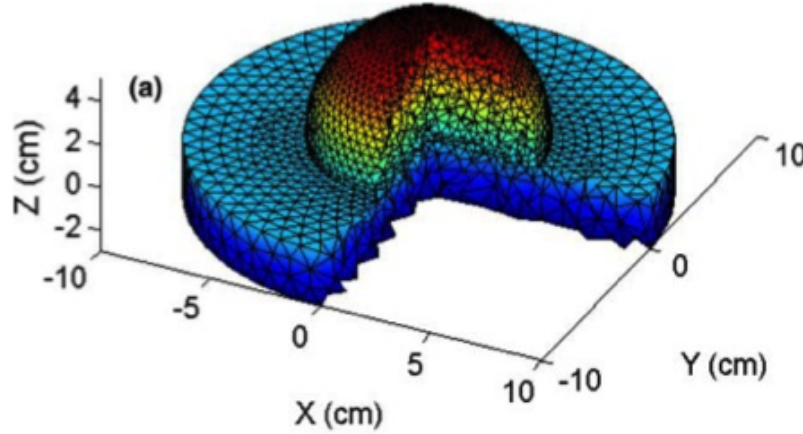
# Método de elementos de borde.

El método de elementos de borde o BEM por sus siglas en inglés Boundary Element Method, es en esencia un método numérico para la resolución de ecuaciones diferenciales, es muy utilizado en mecánica de fluidos, acústica, electromagnética, entre otras áreas. Cuando leemos esta definición es inevitable pensar en el método de elementos finitos o FEM, el cual es el modelo más clásico de la resolución de este tipo de ecuaciones, pero ¿en qué se diferencian?

Como bien sabemos, las ecuaciones diferenciales rigen una gran cantidad de fenómenos físicos. Para resolver estas ecuaciones diferenciales se realiza un proceso llamado discretización, el cual nos entrega resultados aproximados y consiste en dividir el elemento en estudio en elementos infinitesimales y basándose en las expansiones de Taylor, poder encontrar el resultado buscado. Es aquí justamente donde encontramos la primera gran diferencia entre ambos métodos:



(a) Discretización utilizada en BEM.



(b) Discretización utilizada en FEM.

Figura 4.1: Diferencias en la discretización en BEM y FEM en 3D.

Como observamos en la imagen, el elemento infinitesimal de estudio en FEM es un pequeño elemento de 3 dimensiones, puede ser un cubo, una piramide. Mientras que en BEM, el elemento es en 2D, triángulos, cuadrados, etc. Cabe destacar que mientras más elementos se utilicen para dividir el objeto de estudio, más preciso será el cálculo.

El BEM utiliza condiciones de borde o de frontera para ajustar los valores de borde a las ecuaciones diferenciales. Estas condiciones suelen ser:

$$\begin{aligned} \text{Condiciones 'esenciales' del tipo } u &= \bar{u} \text{ en } \Gamma_1 \\ \text{Condiciones 'naturales' del tipo } q &= \bar{q} \text{ en } \Gamma_2 \end{aligned} \quad (4.1)$$

Estas condiciones usualmente son conocidas como condiciones de Dirichlet, las condiciones 'esenciales', mientras que las condiciones 'naturales' son conocidas como condiciones de Neumann. Pueden existir condiciones de frontera más complejas tal como la combinación de las dos anteriores.

La ecuación integral que se usa como punto de inicio para el método es:

$$\nabla^2 u = 0 \quad (4.2)$$

Si a esta ecuación se le aplica una integral por el volumen  $\Omega$  y además se multiplica por una función  $w$  conveniente:

$$\int_{\Omega} \nabla^2 u(r') w(r - r') d\Omega(r') = 0 \quad (4.3)$$

Notar que estamos integrando sobre  $r'$  y  $r$  no está restringida. Utilizando la identidad mostrada en la ecuación (2.2) se puede descomponer de la forma :

$$\int_{\Omega} w(r - r') \nabla \cdot (\nabla u(r')) d\Omega(r') = \int_{\Omega} \nabla \cdot u(r') (\nabla w(r - r')) d\Omega(r') - \int_{\Omega} \nabla u(r') \nabla w(r - r') d\Omega(r') \quad (4.4)$$

En el primer término del lado derecho de la ecuación (4.4) podemos utilizar el teorema de Gauss, indicado en la ecuación (2.3), mientras que en el segundo término volvemos a utilizar la identidad (2.2), lo que nos entrega<sup>1</sup>:

$$\int_{\Omega} \nabla^2 u w d\Omega = \int_{\Gamma} n \cdot u \nabla w d\Gamma - \int_{\Omega} \nabla \cdot (w \nabla u) d\Omega - \int_{\Omega} v \nabla \cdot \nabla u d\Omega \quad (4.5)$$

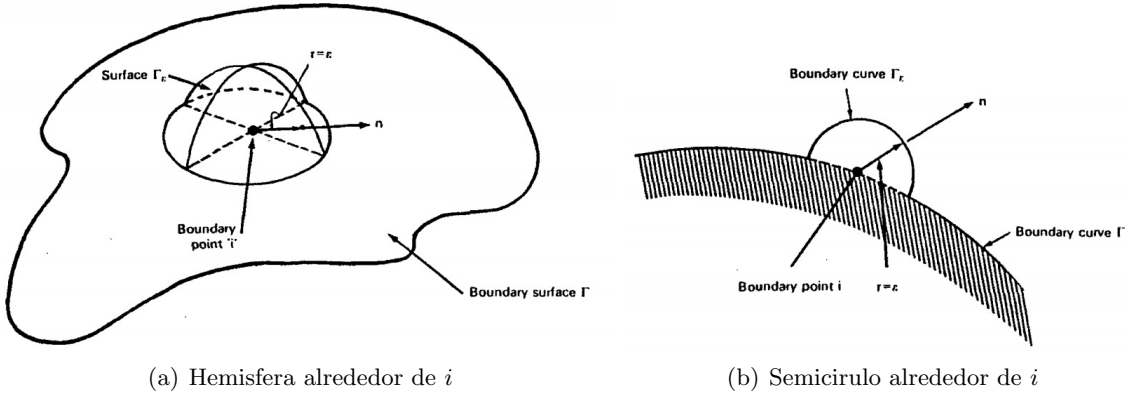


Figura 4.2: Puntos de frontera para los casos 2D y 3D (Fuente:[5])

Nuevamente podemos utilizar el teorema de Gauss con el segundo término del lado derecho de la ecuación. Además, igualamos la ecuación a cero tal como en (4.2):

$$0 = \int_{\Gamma} n \cdot u \nabla w d\Gamma - \int_{\Gamma} n \cdot w \nabla u d\Gamma - \int_{\Omega} u \nabla^2 w d\Omega \quad (4.6)$$

Ahora llamaremos a nuestra función  $w$  conveniente la solución fundamental de la ecuación, la cual tiene por segunda dervada la función Delta Dirac, la que si se encuentra bajo el dominio de la solución de la función tiene por valor 1, como este es el caso se puede afirmar que el tercer término de la ecuación tiene por valor  $u(r)$ . Sin embargo también hay que recordar que la función  $w$  está evaluada en  $(r - r')$  por lo que hay una singularidad que ocurre cuando  $r \rightarrow r'$  que se debe estudiar.

<sup>1</sup>Para facilidad de notación solo se utilizará  $u$  y  $w$

Esta singularidad, que está ilustrada en la figura 4.2, genera un término libre de valor  $-1/2 u(r)$ . Finalmente podemos escribir la ecuación como:

$$0 = \int_{\Gamma} n \cdot u \nabla w \, d\Gamma - \int_{\Gamma} n \cdot w \nabla u \, d\Gamma + \frac{1}{2}u \quad (4.7)$$

Tomando como nueva notación a  $w = u^*$ . Notar que las divergencias están multiplicadas por un vector normal, por lo que el término afectado por el operador  $\nabla$  se transforma en la derivada en la dirección normal. Reescribiendo llegamos a:

$$\frac{1}{2}u^i + \int_{\Gamma} u q^* \, d\Gamma = \int_{\Gamma} q u^* \, d\Gamma \quad (4.8)$$

En donde  $u$  es una función potencial,  $q$  es su derivada con respecto a la normal.  $u^*$  y  $q^*$  son las soluciones fundamentales de ambas funciones. El superíndice  $i$  indica el punto de 'anclaje' o centro del círculo o hemiesfera de interés. Estos puntos  $i$  serán considerados como 'nodos'. Al momento de discretizar la ecuación (4.8) nos queda de la siguiente forma:

$$\frac{1}{2}u^i + \sum_{j=1}^N \left( \int_{\Gamma_j} q^* \, d\Gamma \right) u^j = \sum_{j=1}^N \left( \int_{\Gamma_j} u^* \, d\Gamma \right) q^j \quad (4.9)$$

Las integrales mostadas, se llamarán para simplificar:

$$\hat{H}^{ij} = \int_{\Gamma_j} q^* \, d\Gamma \quad \text{y} \quad G^{ij} = \int_{\Gamma_j} u^* \, d\Gamma \quad (4.10)$$

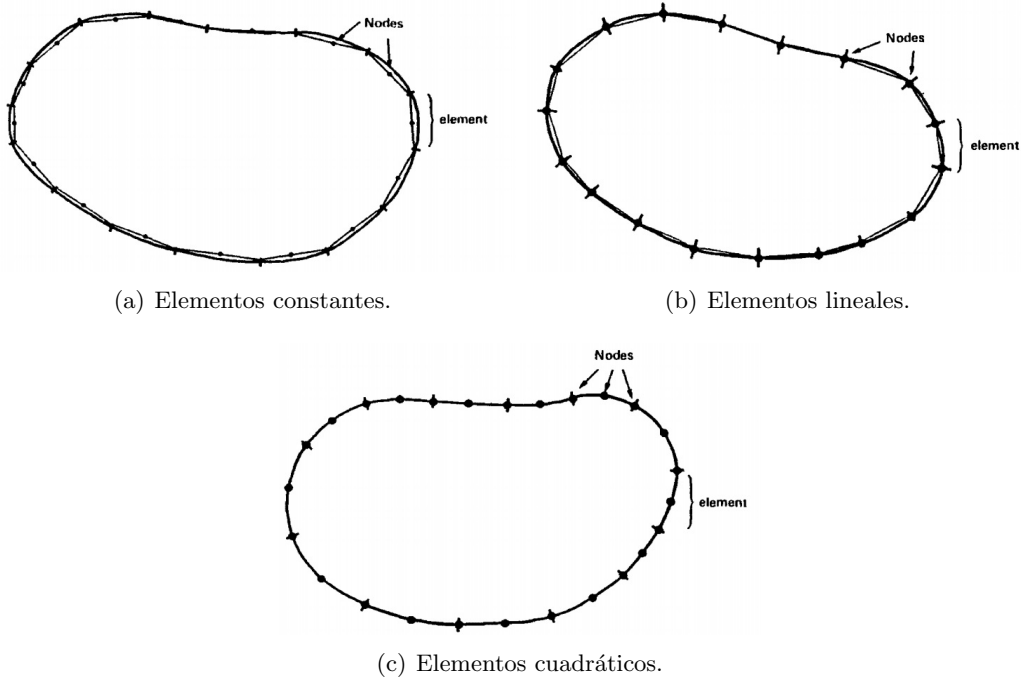


Figura 4.3: Diferentes tipos de elementos de frontera. (Fuente:[5])



$$\left. \begin{array}{l} \text{Si } i \neq j \text{ entonces } \hat{H}^{ij} \\ \text{Si } i = j \text{ entonces } \hat{H}^{ij} + \frac{1}{2} \end{array} \right\} = H^{ij}$$

Entonces la ecuación (4.9) puede ser escrita como:

$$\sum_{j=1}^N H^{ij} u^j = \sum_{j=1}^N G^{ij} q^j \quad (4.11)$$

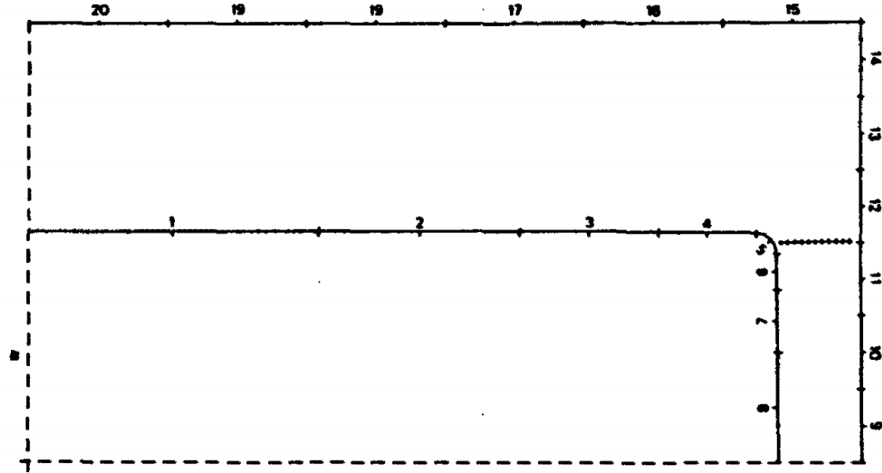
Esta serie de ecuaciones puede ser expresada en forma de matriz como:

$$HU = GQ \quad (4.12)$$

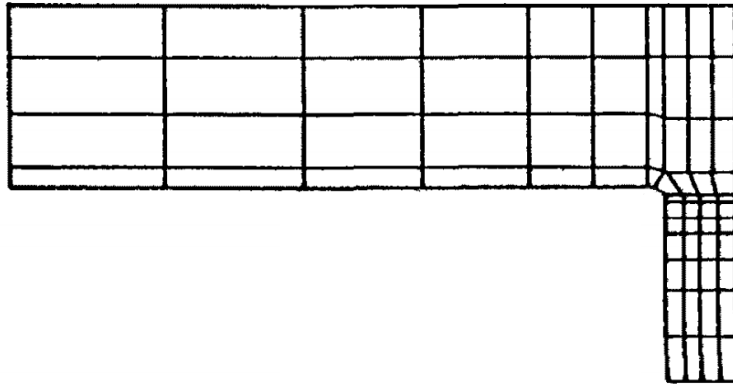
Podemos llevar todos los términos conocidos al lado izquierdo y los valores iniciales conocidos (como los ejemplificados en la ecuación (4.1)), reescribiendola como:

$$AX = F \quad (4.13)$$

Entonces, si nuestras condiciones iniciales son solo de Dirichlet, el vector  $X$  estaría compuesto solo por 'valores desconocidos' de  $q$  y viceversa. Si fuera una mezcla de ambas condiciones, el vector  $X$  sería una mezcla de  $u$ 's y  $q$ 's, evidentemente luego deberían reordenarse los valores y dejar un solo vector de  $u$ 's y otro distinto para  $q$ 's. Esto es consecuencia de la formulación mixta de elementos de frontera y entrega una importante ventaja respecto a elementos finitos.



(a) Discretización utilizada en BEM.



(b) Discretización utilizada en FEM.

Figura 4.4: Diferencias en la discretización en BEM y FEM en 2D (Fuente:[5]).

Es necesario hacer un par de alcances, para comprender mejor lo explicado anteriormente: Las integrales mostradas en la ecuación (4.10) son resueltas mediante una cuadratura de Gauss, la cual es una aproximación de la integral. En el problema a resolver en este informe, se utiliza una cuadratura con 4 nodos.

Es importante indicar que BEM es aplicable a problemas que puedan ser modelados por una función de Green, ya sea Laplace, Helmholtz o Helmholtz modificado. Es necesario hacer notar que las funciones de Green son principalmente utilizadas para modelar ecuaciones diferenciales con condiciones de contorno dadas, que es justamente como hemos definido los problemas de BEM.

¿Qué sucede con los puntos internos? Como dijimos en un principio, el comportamiento en el interior del cuerpo de estudio es homogéneo, por lo tanto es posible hacer una buena aproximación o predicción de como se comportará la función potencial estudiada en el cuerpo en los puntos interiores. Gráficamente los puntos interiores serían algo como:

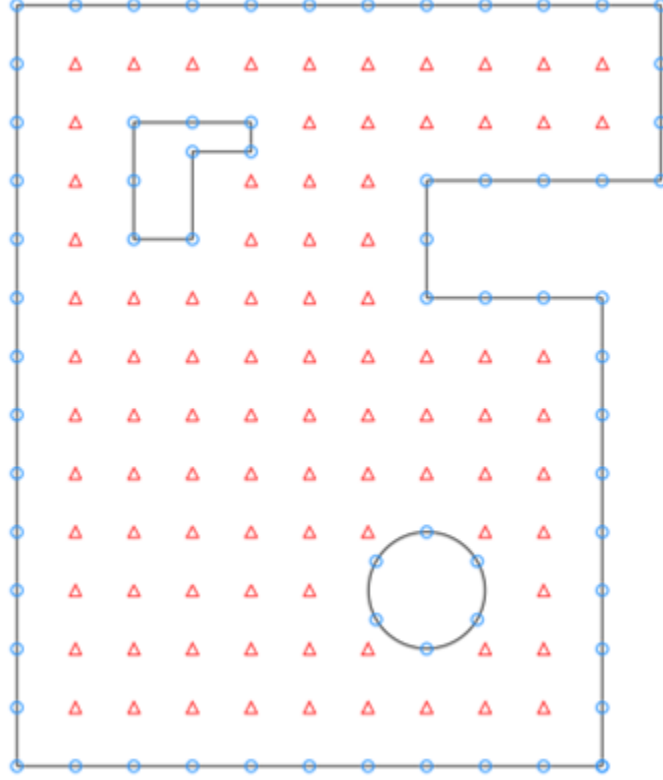


Figura 4.5: Elementos de borde y puntos internos para el estudio

Para calcular el potencial es necesario utilizar la ecuación que se muestra a continuación. Es necesario hacer notar que los coeficientes  $H^{ij}$  y  $G^{ij}$  son recalculados para cada punto interno.

$$u^i = \sum_{j=1}^N G^{ij} q^j - \sum_{j=1}^N \hat{H}^{ij} u^j \quad (4.14)$$

El cálculo para la derivada en ambas direcciones en los puntos internos también se puede realizar, pero su formulación es como sigue:

$$\begin{aligned} (q_{x_1})^i &= \left( \frac{\partial u}{\partial x_1} \right)^i = \sum_{j=1}^N \left( \int_{\Gamma} \frac{\partial u^*}{\partial x_1} d\Gamma \right) q^j - \sum_{j=1}^N \left( \int_{\Gamma} \frac{\partial q^*}{\partial x_1} d\Gamma \right) u^j \\ (q_{x_2})^i &= \left( \frac{\partial u}{\partial x_2} \right)^i = \sum_{j=1}^N \left( \int_{\Gamma} \frac{\partial u^*}{\partial x_2} d\Gamma \right) q^j - \sum_{j=1}^N \left( \int_{\Gamma} \frac{\partial q^*}{\partial x_2} d\Gamma \right) u^j \end{aligned} \quad (4.15)$$

## Capítulo 5

# Biblioteca bempp.

Bempp es una plataforma computacional gratuita de elementos de frontera se puede utilizar para resolver problemas de acústica o electroestática, entre otros. Utiliza una interfaz de python de fácil uso, en nuestro caso utilizamos *Docker*<sup>1</sup> *Images* para poder resolver nuestro problema.

Como se vio en la sección anterior, para resolver problemas con BEM es necesario resolver sistemas compuestos por matrices de gran tamaño, las cuales a su vez contienen elementos que son necesario aproximar, es por esto que la existencia de esta librería abierta facilita la tarea.

### 5.1 Estructura.

La librería bempp tiene una estructura basada en la imagen 5.1, la cual tiene 5 grandes ejes y es explicado con mayor detalle en [8]:

- Grid: Responsable del manejo de la malla, apoyado en la librería 'Dune-FoamGrid'.
- Fiber (**F**ast **I**ntegration **B**oundary **E**lement **R**outine): Es un elemento vital dentro de la librería. Encargado de evaluaciones las integrales de elementos de borde en cada elemento, sin importar su conectividad. Además de realizar la integración como tal. Este modulo es independiente del resto por lo que podría ser utilizado de manera particular en otros códigos de BEM.
- Space: Responsable del espacio de funciones y sus derivadas. También actúa como administrador de grados de libertad, usando los conocimientos de la conectividad entre elementos y las propiedades de continuidad del espacio de funciones.

---

<sup>1</sup>Software que crea contenedores virtuales para que puedan ejecutarse bajo cualquier máquina, independiente del sistema operativo que tenga.

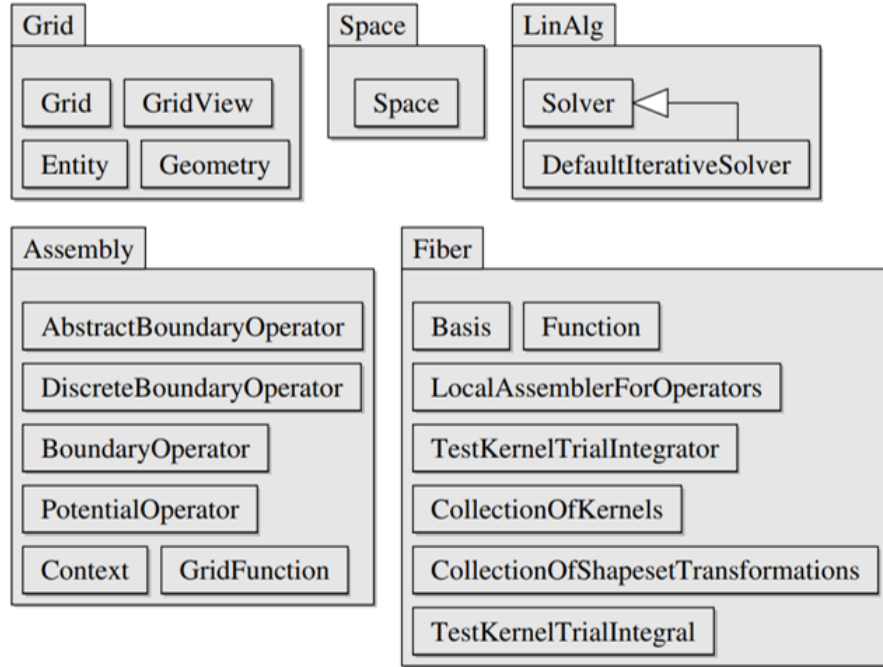


Figura 5.1: Módulos de bempp (Fuente:[8])

- Assembly: Es el módulo más grande de la librería. Define los operadores integrales y los espacios de funciones en la malla. También almacena las matrices de las integrales discretizadas que son formadas en el módulo Fiber.
- LinAlg: Módulo que almacena distintos tipos de *solver's* lineales.

### 5.1.1 Algoritmo y explicación.

Antes de comenzar a desmenuzar la resolución del problema o el código utilizado es más importante definir que es **bempp**[3].

El algoritmo para la resolución de problemas a través de bempp es más bien simple pero es necesario tener los conceptos claros o no podremos ejecutarlo de forma correcta. La forma de resolución es la que se muestra a continuación. Es necesario indicar que esta resolución resolverá los parámetros de borde, para resolver los elementos internos simplemente se debe realizar una nueva malla con puntos internos y utilizar la solución calculada.

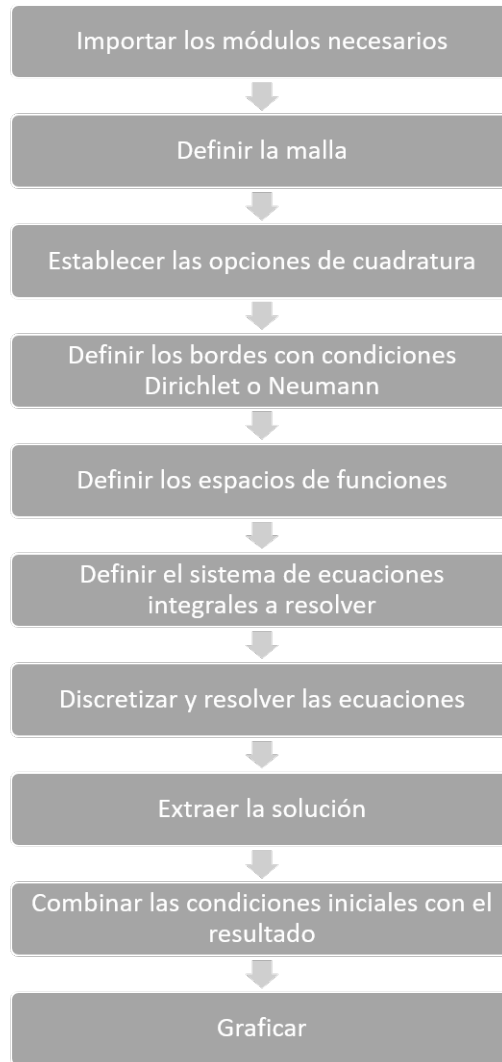


Figura 5.2: Algoritmo resolución bempp

Al observar el esquema se nos hace evidente definir los conceptos y además mostrar como se ejecutó lo descrito en el código:

**1. Importar los módulos necesarios.**

Módulos: También conocidas como bibliotecas o librerías, añaden funciones adicionales a nuestro lenguaje de programación de uso: Python. Para resolver el problema se utilizaron 3: numpy, scipy.sparse.linalg (para resolver sistemas lineales) y por supuesto, bempp.

**2. Definir la malla.**

Malla - Grid: El mallado en esta plataforma puede realizarse de manera bastante simple, siempre y cuando la figura sea común (esferas, cubos, etc...). Estas también pueden ser importadas desde el formato Gmsh.

**3. Establecer las opciones de cuadratura.**

Cuadratura - Quadrature: Como vimos en el código manual, hay que utilizar una cuadratura Gaussiana para resolver las ecuaciones integrales, en este punto se precisa de cuantos puntos

se hará esa cuadratura. Para asimilar esta forma de resolución lo más posible a la anterior utilizaremos nuevamente 4 puntos.

#### 4. Definir los espacios de funciones.

Conocido como espacio de función o Function spaces. Como sabemos los nodos podían ser tomados para medios constantes, lineales o polinomiales, dependiendo del medio en que queramos definir nuestros nodos, nuestros espacios de función deben ser definidos en el orden deseado. En la plataforma bempp es importante definir esto correctamente. Como se señala en el algoritmo es necesario haber generado con anterioridad la malla, ya que esto es un input necesario. Los tipos de 'Function spaces' que nos interesan son escalares y son utilizados para resolver problemas de Laplace y Helmholtz:

- **Continuos Polynomial "P"**: Se refiere a que el 'espacio' entre nodos debe tener continuidad, a modo de ejemplo se tiene la figura 5.3 la cual nos muestra una intersección de elementos lineales continuos. Este espacio no puede utilizarse con elementos constantes ya que todos deberían ser la misma constante lo cual carece de sentido, es por esto que el orden de este espacio va desde 1 (lineal) hasta 10.
- **Discontinuos Polynomial "DP"**: Caso contrario al anterior, en donde la intersección de los elementos no necesariamente debe tener una continuidad, es por esto que cuando se desea tener elementos constantes es necesario utilizar este tipo de espacio. El rango del orden va desde 0 (constante) hasta 10.
- **Dual spaces "DUAL"**: Este tipo de espacio está disponible solo para orden 0. Estos espacios forman un emparejamiento dual estable junto con funciones lineales continuas a trozos y son necesarios para ciertos preconditionadores de orden opuesto

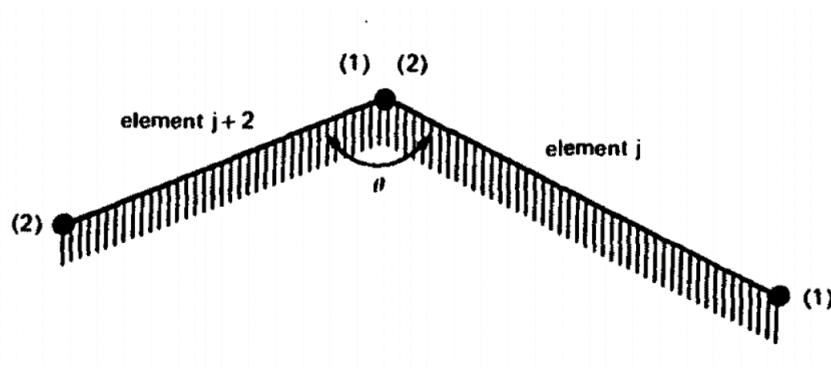


Figura 5.3: Intersección de elementos lineales continuos (Fuente:[5])

#### 5. Definir el sistema de ecuaciones integrales a resolver.

Operadores - Operators: Lo que en la ecuación (4.10) describimos como matrices  $H$  y  $G$ , en esta plataforma son conocidos como operadores ('Single Layer Operator' y 'Double Layer Operator' respectivamente) y no son los únicos que están disponibles. Se definen de la siguiente

forma:

$$\begin{aligned}
V &= \int_{\Gamma} g(x, y) d\Gamma && \text{Single Layer Operator} \\
K &= \int_{\Gamma} \frac{\delta g(x, y)}{\delta y} d\Gamma && \text{Double Layer Operator} \\
K' &= \int_{\Gamma} \frac{\delta g(x, y)}{\delta x} d\Gamma && \text{Adjoint Double Layer Operator} \\
H &= -\frac{\delta}{\delta x} \int_{\Gamma} \frac{\delta g(x, y)}{\delta y} d\Gamma && \text{Hypersingular Boundary Operator}
\end{aligned} \tag{5.1}$$

Es importante destacar que es necesario precisar el modulo en el que serán ocupados: Laplace, Helmholtz o Helmholtz modificado. Existen otro operadores importantes como el operador de matriz identidad. La función  $g(x, y)$  es la función de Green, la cual en todo este informe hemos definido como  $u$ .

Funciones de malla - Grid function: Se trata de la función dominante, la función potencial en la frontera del elemento de estudio. También se utiliza para definir la condiciones de frontera de nuestro problema. En nuestro caso, las Grid function utilizadas fueron las condiciones de Neumann y Dirichlet indicadas en el problema. Generación y resolución del sistema lineal: Una vez que se tiene definido todo lo anterior se procede a definir el lado izquierdo y derecho de la ecuación del problema, el cual posteriormente será resuelto usando *gmres* al igual que el código anterior.

## 6. Discretizar y resolver las ecuaciones.

Ahora que ya calculamos el potencial y su derivada en los bordes es hora de calcular el comportamiento en su interior. Para eso haremos una matriz de vectores y 'anclaremos' en un punto en el eje  $z$  para poder mostrar los resultados en un gráfico 2D al igual que en el código de Brebbia y Dominguez facilitando así la comparación de resultados. Por último se calculan los potenciales con las soluciones que obtuvimos en la parte anterior y se grafican.



## Capítulo 6

# Planteamiento del problema

## Capítulo 7

### Condiciones del problema.

La situación en la que nos enfrentamos en el problema es de una onda electromagnética incidente en nuestro microhilo. Esta onda tiene un campo magnético incidente  $B_i$  y un campo eléctrico incidente  $E_i$ . Denominaremos al volumen del micro-hilo como  $\Omega_1$  y al volumen externo como  $\Omega_2$  tal como se muestra en la figura 7.1.

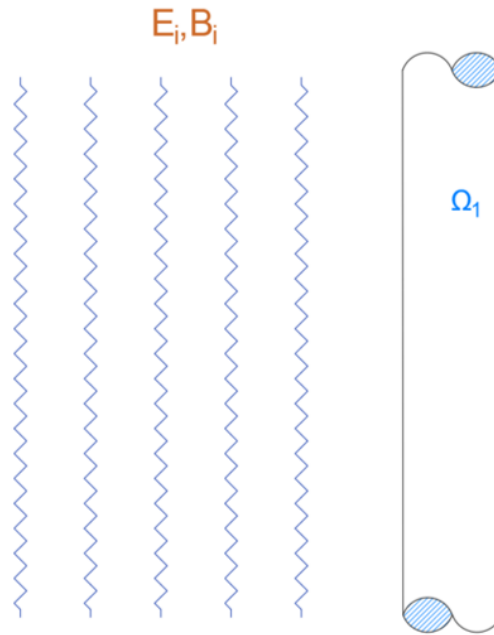


Figura 7.1: Situación del problema

Comenzaremos con las ecuaciones de Maxwell:

$$\begin{aligned} \nabla \cdot \mathbf{D} &= \rho & \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\ \nabla \cdot \mathbf{B} &= 0 & \nabla \times \mathbf{H} &= J_f + \frac{\partial \mathbf{D}}{\partial t} \end{aligned} \tag{7.1}$$

Reescribiendo la ecuación (7.1) en función solo de  $B$  y  $D$ :

$$\begin{aligned} \nabla \cdot \mathbf{D} &= \rho & \nabla \times \mathbf{D} &= -\varepsilon \frac{\partial \mathbf{B}}{\partial t} \\ \nabla \cdot \mathbf{B} &= 0 & \nabla \times \mathbf{B} &= \mu J_f + \mu \frac{\partial \mathbf{D}}{\partial t} \end{aligned} \quad (7.2)$$

Podemos suponer que una onda incidente en un entorno con propiedades electromagnéticas  $\mu_2$  y  $\varepsilon_2$  sufrirá de una perturbación si se encuentra con algún objeto con propiedades diferentes, tal como se plantea en nuestro problema.

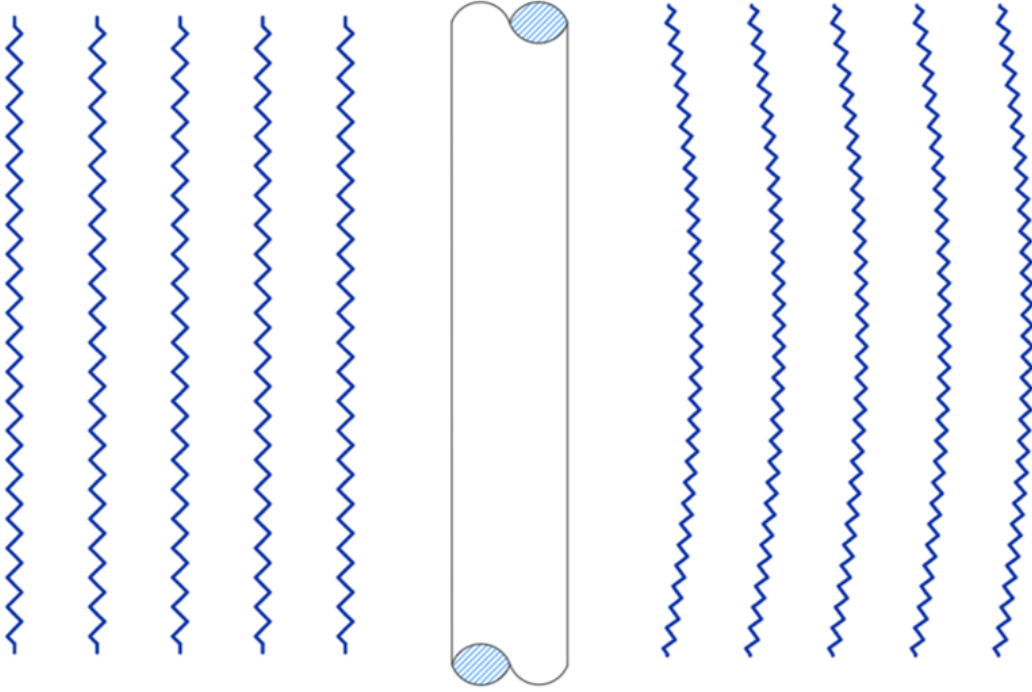


Figura 7.2: Onda incidente perturbada.

Si ahora aplicamos las ecuaciones de Maxwell a la situación ilustrada en la figura 7.2, obtenemos:

$$\begin{aligned} \left. \begin{aligned} \nabla \cdot D_1 &= \rho_1 & \nabla \times D_1 &= -\varepsilon_1 \frac{\partial B_1}{\partial t} \\ \nabla \cdot B_1 &= 0 & \nabla \times B_1 &= \mu_1 J_{f1} + \mu_1 \frac{\partial D_1}{\partial t} \end{aligned} \right\} & \text{En } \Omega_1 \\ \left. \begin{aligned} \nabla \cdot D_2 &= \rho_2 & \nabla \times D_2 &= -\varepsilon_2 \frac{\partial B_2}{\partial t} \\ \nabla \cdot B_2 &= 0 & \nabla \times B_2 &= \mu_2 J_{f2} + \mu_2 \frac{\partial D_2}{\partial t} \end{aligned} \right\} & \text{En } \Omega_2 \\ D_1 \cdot n &= D_2 \cdot n & \frac{B_1}{\mu_1} \cdot n &= \frac{B_2}{\mu_2} \cdot n \Big\} & \text{En } \Gamma \end{aligned} \quad (7.3)$$

Debemos tener en cuenta que las propiedades del microhilo tendrán un efecto sobre el campo incidente. En caso de que las propiedades del volumen  $\Omega_1$  sean iguales a los de  $\Omega_2$ , esto es  $\varepsilon_1 = \varepsilon_2$

y  $\mu_1 = \mu_2$ <sup>1</sup>. La situación descrita se puede ilustrar como en la imagen 7.3 y se puede representar en las ecuaciones de Maxwell de la forma:

$$\begin{aligned}
 & \left. \begin{aligned} \nabla \cdot D_i &= \rho_1 & \nabla \times D_1 &= -\varepsilon_2 \frac{\partial B_i}{\partial t} \\ \nabla \cdot B_i &= 0 & \nabla \times B_i &= \mu_2 J_{f1} + \mu_2 \frac{\partial D_i}{\partial t} \end{aligned} \right\} \text{En } \Omega_1 \\
 & \left. \begin{aligned} \nabla \cdot D_i &= \rho_2 & \nabla \times D_i &= -\varepsilon_2 \frac{\partial B_i}{\partial t} \\ \nabla \cdot B_i &= 0 & \nabla \times B_i &= \mu_2 J_{f2} + \mu_2 \frac{\partial D_i}{\partial t} \end{aligned} \right\} \text{En } \Omega_2 \\
 & \left. \begin{aligned} D_i \cdot n &= D_i \cdot n & \frac{B_i}{\mu_2} \cdot n &= \frac{B_i}{\mu_2} \cdot n \end{aligned} \right\} \text{En } \Gamma
 \end{aligned} \tag{7.4}$$

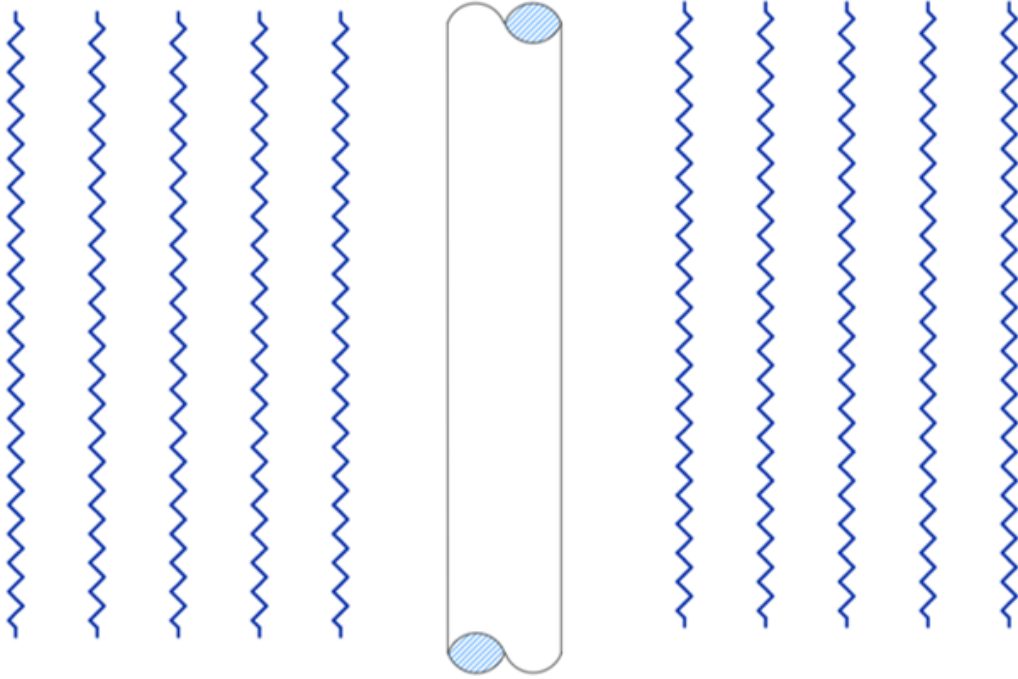


Figura 7.3: Micro-hilo no tiene efecto sobre la onda.

Sabemos que en la situación expuesta en la ecuación (7.3) el campo resultante, que es el expuesto en las ecuaciones puede verse como el campo incidente sumado al campo difundido producto del encuentro de la onda con al micro-hilo, por lo tanto los campos pueden ser representados como:

$$\begin{aligned}
 D &= D_i + D_d \\
 B &= B_i + B_d
 \end{aligned} \tag{7.5}$$

---

<sup>1</sup>Por conveniencia matemática dejaremos todo escrito en términos del campo  $\Omega_2$

Donde  $D_i$  y  $B_i$  es el desplazamiento eléctrico inducido y  $B_i$  es el campo magnético inducido. Es correcto asumir que en nuestro sistema no hay corrientes electricas ni densidad de corriente por lo tanto los terminos de  $\rho$  y  $\vec{J}$  pueden ser eliminados. Por lo tanto podemos reescribir la ecuación (7.3) como:

$$\begin{aligned}
& \left. \begin{aligned} \nabla \cdot D_{1d} + \nabla \cdot D_i &= 0 & \nabla \times D_{1d} + \nabla \times D_i &= -\varepsilon_1 \left( \frac{\partial B_{1d}}{\partial t} + \frac{\partial B_i}{\partial t} \right) \\ \nabla \cdot B_{1d} + \nabla \cdot B_i &= 0 & \nabla \times B_{1d} + \nabla \times B_i &= \mu_1 \left( \frac{\partial D_{1d}}{\partial t} + \frac{\partial D_i}{\partial t} \right) \end{aligned} \right\} & \text{En } \Omega_1 \\
& \left. \begin{aligned} \nabla \cdot D_{2d} + \nabla \cdot D_i &= 0 & \nabla \times D_{2d} + \nabla \times D_i &= -\varepsilon_2 \left( \frac{\partial B_{2d}}{\partial t} + \frac{\partial B_i}{\partial t} \right) \\ \nabla \cdot B_{2d} + \nabla \cdot B_i &= 0 & \nabla \times B_{2d} + \nabla \times B_i &= \mu_2 \left( \frac{\partial D_{2d}}{\partial t} + \frac{\partial D_i}{\partial t} \right) \end{aligned} \right\} & \text{En } \Omega_2 \\
& D_1 \cdot n = D_2 \cdot n \quad \frac{B_1}{\mu_1} \cdot n = \frac{B_2}{\mu_2} \cdot n \quad \left. \right\} & \text{En } \Gamma
\end{aligned} \tag{7.6}$$

Entonces si solo quisieramos saber como se comporta la onda difundida, solo es necesario restar la ecuación (7.4) a la ecuación (7.6):

$$\begin{aligned}
& \left. \begin{aligned} \nabla \cdot D_{1d} &= 0 & \nabla \times D_{1d} &= -\varepsilon_1 \frac{\partial B_{1d}}{\partial t} - (\varepsilon_1 - \varepsilon_2) \frac{\partial B_i}{\partial t} \\ \nabla \cdot B_{1d} &= 0 & \nabla \times B_{1d} &= \mu_1 \frac{\partial D_{1d}}{\partial t} + (\mu_1 - \mu_2) \frac{\partial D_i}{\partial t} \end{aligned} \right\} & \text{En } \Omega_1 \\
& \left. \begin{aligned} \nabla \cdot D_{1d} &= 0 & \nabla \times D_{1d} &= -\varepsilon_1 \frac{\partial B_{1d}}{\partial t} \\ \nabla \cdot B_{1d} &= 0 & \nabla \times B_{1d} &= \mu_1 \frac{\partial D_{1d}}{\partial t} \end{aligned} \right\} & \text{En } \Omega_2 \\
& D_{1d} \cdot n = D_{2d} \cdot n \quad \left( \frac{1}{\mu_1} B_{1d} \cdot n - \frac{1}{\mu_2} B_{2d} \cdot n \right) = \left( \frac{1}{\mu_2} - \frac{1}{\mu_1} \right) B_i \cdot n \quad \left. \right\} & \text{En } \Gamma
\end{aligned} \tag{7.7}$$

Se puede asumir una onda armónica con frecuencia  $\omega$  para representar el desplazamiento eléctrico y el campo magnético:

$$\begin{aligned}
B(x, t) &= B(x) e^{i\omega t} \\
D(x, t) &= D(x) e^{i\omega t}
\end{aligned}$$

La ecuación (7.7) queda:

$$\begin{aligned}
& \left. \begin{aligned} \nabla \cdot D_{1d} &= 0 & \nabla \times D_{1d} &= -\varepsilon_1 i\omega_d B_{1d} - (\varepsilon_1 - \varepsilon_2) i\omega_i B_i \\ \nabla \cdot B_{1d} &= 0 & \nabla \times B_{1d} &= \mu_1 i\omega_d D_{1d} + (\mu_1 - \mu_2) i\omega_i D_i \end{aligned} \right\} & \text{En } \Omega_1 \\
& \left. \begin{aligned} \nabla \cdot D_{1d} &= 0 & \nabla \times D_{1d} &= -\varepsilon_1 i\omega_d B_{1d} \\ \nabla \cdot B_{1d} &= 0 & \nabla \times B_{1d} &= \mu_1 i\omega_d D_{1d} \end{aligned} \right\} & \text{En } \Omega_2 \\
& D_{1d} \cdot n = D_{2d} \cdot n \quad \left( \frac{1}{\mu_1} B_{1d} \cdot n - \frac{1}{\mu_2} B_{2d} \cdot n \right) = \left( \frac{1}{\mu_2} - \frac{1}{\mu_1} \right) B_i \cdot n \quad \left. \right\} & \text{En } \Gamma
\end{aligned} \tag{7.8}$$

Al ser la onda incidente de una longitud de onda muy grande con respecto al cilindro es correcto asumir que  $\lambda_i \rightarrow \infty$ . Sabemos que por definición  $w = v/\lambda$ , por lo tanto  $w_i \rightarrow 0$ . Podemos escribir

las cantidades auxiliares:

$$\begin{aligned} b_s &= \sqrt{\varepsilon_2} B_s \\ d_s &= \sqrt{\mu_2} D_s \\ x' &= \frac{x}{d_x} \end{aligned} \quad (7.9)$$

A partir de esto podemos reescribir la ecuación (7.8) utilizando las nuevas cantidades auxiliares:

$$\begin{aligned} \left. \begin{aligned} \nabla \cdot d_{1d} &= 0 & \nabla \times d_{1d} &= \frac{\varepsilon_1}{\varepsilon_2} i \beta b_{1d} \\ \nabla \cdot b_{1d} &= 0 & \nabla \times b_{1d} &= \frac{\mu_1}{\mu_2} i \beta d_{1d} \end{aligned} \right\} & \text{En } \Omega_1 \\ \left. \begin{aligned} \nabla \cdot d_{1d} &= 0 & \nabla \times d_{1d} &= -i \beta b_{1d} \\ \nabla \cdot b_{1d} &= 0 & \nabla \times b_{1d} &= i \beta d_{1d} \end{aligned} \right\} & \text{En } \Omega_2 \\ d_{1d} \cdot n &= d_{2d} \cdot n & \left( \frac{1}{\mu_1} b_{1d} \cdot n - \frac{1}{\mu_2} b_{2d} \cdot n \right) &= \left( \frac{1}{\mu_2} - \frac{1}{\mu_1} \right) b_i \cdot n \end{aligned} \quad (7.10)$$

Con  $\beta = wd\sqrt{\varepsilon_2\mu_2}$ . Podemos expandir  $d$  y  $b$ , utilizando  $\beta$  como:

$$\begin{aligned} d &= d_s^{(0)} + \beta d_s^{(1)} + \beta^2 d_s^{(2)} \dots \\ b &= b_s^{(0)} + \beta b_s^{(1)} + \beta^2 b_s^{(2)} \dots \end{aligned} \quad (7.11)$$

Y ahora considerando los términos de orden cero de la ecuación (7.10), escribimos solo los términos del campo magnético:

$$\begin{aligned} \nabla \cdot b_{1d}^{(0)} &= 0 & \nabla \times b_{1d}^{(0)} &= 0 \\ \nabla \cdot b_{2d}^{(0)} &= 0 & \nabla \times b_{2d}^{(0)} &= 0 \\ \left( \frac{1}{\mu_1} b_{1d}^{(0)} \cdot n - \frac{1}{\mu_2} b_{2d}^{(0)} \cdot n \right) &= \left( \frac{1}{\mu_2} - \frac{1}{\mu_1} \right) b_i \cdot n \end{aligned} \quad (7.12)$$

Aplicando el operador  $\nabla$  a las ecuaciones en  $\Omega$  podemos concluir que:

- Por la identidad de la divergencia de un rotor:

$$\begin{aligned} \nabla \cdot (\nabla \times B_{1d}^{(0)}) &= 0 \\ \nabla \cdot (\nabla \times B_{2d}^{(0)}) &= 0 \end{aligned}$$

Podemos determinar que los campos  $B_{1d}$  y  $B_{2d}$  son conservativos basandonos en que el rotor en ambos casos es cero, esto quiere decir que hay una función potencial que puede describir el comportamiento del campo.

- Denominaremos como  $\psi$  a la función escalar potencial que describirá nuestros campos:

$$\nabla \cdot \varphi = B_d$$

- Nuevamente aplicamos el operador  $\nabla$  pero esta vez en los gradientes del campo y utilizando la nueva notación con  $\varphi$ , notar que estas ecuaciones se cumplen para el campo en  $\Omega_1$  y  $\Omega_2$  respectivamente:

$$\begin{aligned} \nabla^2 \cdot \varphi_{1d} &= 0 \\ \nabla^2 \cdot \varphi_{2d} &= 0 \end{aligned}$$

- Reescribimos las ecuaciones en el borde  $\Gamma$  y añadimos una más:

$$\left( \frac{1}{\mu_1} \frac{\partial \varphi_{1d}}{\partial n} - \frac{1}{\mu_2} \frac{\partial \varphi_{2d}}{\partial n} \right) = \left( \frac{1}{\mu_2} - \frac{1}{\mu_1} \right) \frac{\partial \varphi_i}{\partial n}$$

$$\varphi_{1d} = \varphi_{2d} \quad \text{En la frontera } \Gamma$$

Finalmente nos encontramos con una ecuación diferencial con condiciones de borde, justamente lo necesario para resolver este problema utilizando el método de elementos de borde.

$$\boxed{\begin{array}{l} \nabla^2 \cdot \varphi_{1d} = 0, \quad \nabla^2 \cdot \varphi_{2d} = 0 \quad \text{En } \Omega_1, \Omega_2 \\ \left( \frac{1}{\mu_1} \frac{\partial \varphi_{1d}}{\partial n} - \frac{1}{\mu_2} \frac{\partial \varphi_{2d}}{\partial n} \right) = \left( \frac{1}{\mu_2} - \frac{1}{\mu_1} \right) \frac{\partial \varphi_i}{\partial n} \quad \varphi_{1d} = \varphi_{2d} \quad \text{En la frontera } \Gamma \end{array}} \quad (7.13)$$

Y como se vio en la sección 4 la ecuación (7.13) puede representarse por una formulación de elementos de borde como:

$$\begin{aligned} \frac{1}{2} \varphi_{1d} + K \cdot \varphi_{1d} - V \left( \frac{\partial}{\partial n} \varphi_{1d} \right) &= 0 \\ \frac{1}{2} \varphi_{1d} - K \cdot \varphi_{1d} + V \left( \frac{\partial}{\partial n} \varphi_{2d} \right) &= 0 \end{aligned} \quad (7.14)$$

En la frontera  $\Gamma$

La primera ecuación presente en el conjunto mostrado en (7.14) es del mismo desarrollo que el mostrado en la sección citada. Sin embargo la segunda ecuación tiene los signos invertidos debido a que las normales apuntan en sentido contrario, *ergo* los signos de los términos en donde las normales tienen incidencia deben tener distinto signo para que ambas tengan la misma convención de signos. Si además a lo anterior se aplican las condiciones de borde de (7.13):

$$\begin{aligned} \frac{1}{2} \varphi_{1d} + K \cdot \varphi_{1d} - V \left( \frac{\partial}{\partial n} \varphi_{1d} \right) &= 0 \\ \frac{1}{2} \varphi_{2d} - K \cdot \varphi_{1d} + \mu_2 V \left( \frac{1}{\mu_1} \frac{\partial}{\partial n} \varphi_{1d} - \left( \frac{1}{\mu_2} - \frac{1}{\mu_1} \right) \frac{\partial}{\partial n} \varphi_i \right) &= 0 \end{aligned} \quad (7.15)$$

En la frontera  $\Gamma$

La forma matricial de la ecuación (7.15) es:

$$\begin{bmatrix} \frac{1}{2}I + K & -V \\ \frac{1}{2}I - K & \frac{\mu_2}{\mu_1} V \end{bmatrix} \begin{bmatrix} \varphi_{1d} \\ \frac{\partial}{\partial n} \varphi_{1d} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\mu_1 - \mu_2}{\mu_1} \frac{\partial \varphi_i}{\partial n} \end{bmatrix} \quad (7.16)$$

Donde  $I$  sería la matriz identidad. Usando los mismos pasos matemáticos pero teniendo como 'campos iniciales' el campo eléctrico  $E$  y la intensidad de campo magnético  $H$  podemos llegar a una ecuación que tiene la misma estructura pero distintos coeficientes:

$$\begin{bmatrix} \frac{1}{2}I + K & -V \\ \frac{1}{2}I - K & \frac{\varepsilon_1}{\varepsilon_2} V \end{bmatrix} \begin{bmatrix} \varphi_{1d} \\ \frac{\partial}{\partial n} \varphi_{1d} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\varepsilon_2 - \varepsilon_1}{\varepsilon_2} \frac{\partial \varphi_i}{\partial n} \end{bmatrix} \quad (7.17)$$

## 7.1 Creación de mallas.

Como se ha dicho en la sección 5 es necesario una malla de estudio, a la cual aplicaremos las condiciones del problema. Corresponde utilizar el formato aceptado por la librería, ".msh". La creación de la geometría de la malla se hace en el Software *SolidWorks* citeFulkerson SolidWorks en donde se exporta el archivo en formato *STEP*:

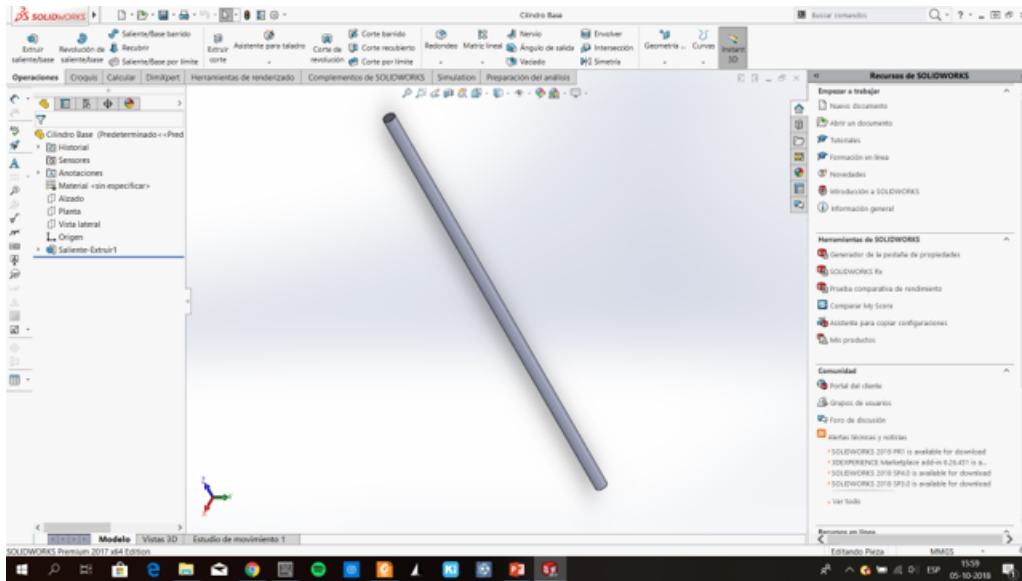


Figura 7.4: Cilindro en SolidWorks

Luego, este archivo en formato *STEP* se abre en el software *Gmsh*, el cual nos permitirá editar la forma de la malla, tamaño de los elementos, forma de los elementos, etc. Las mallas utilizadas tienen la forma de las imágenes que se muestran en 7.5, 7.6 y 7.7, cambiando el tamaño del elemento:

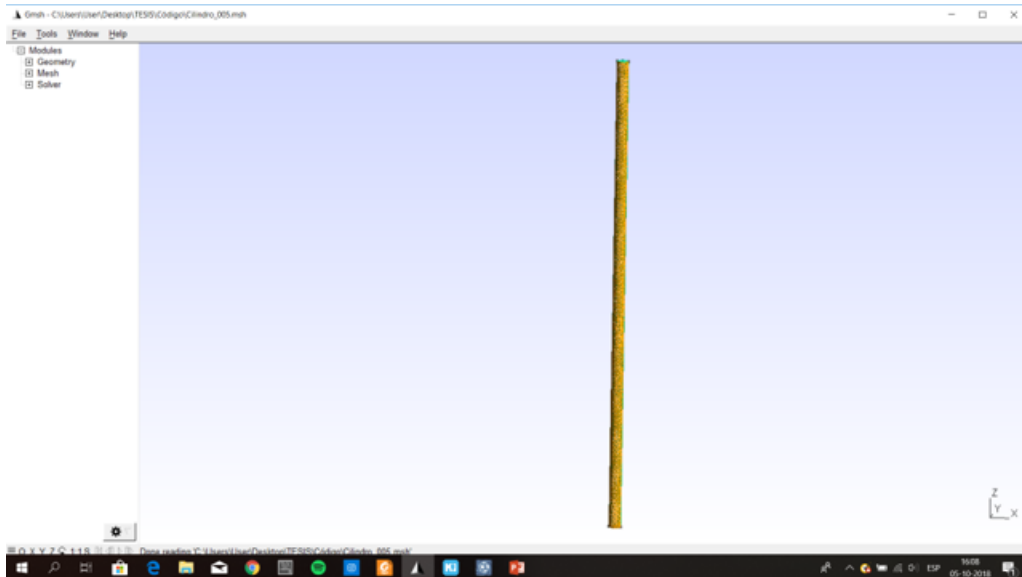


Figura 7.5: Malla vista completa



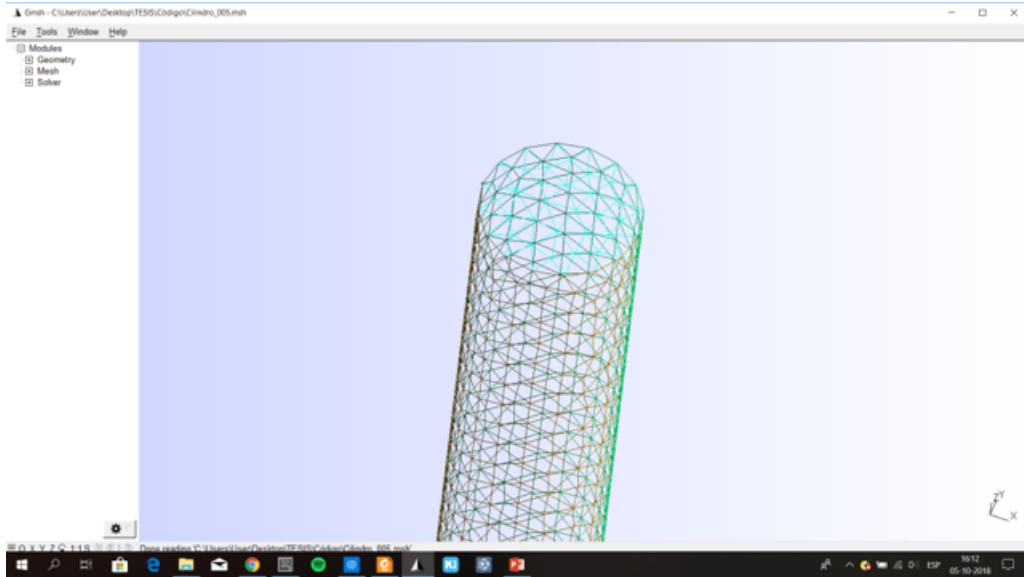


Figura 7.6: Malla detalle tapa

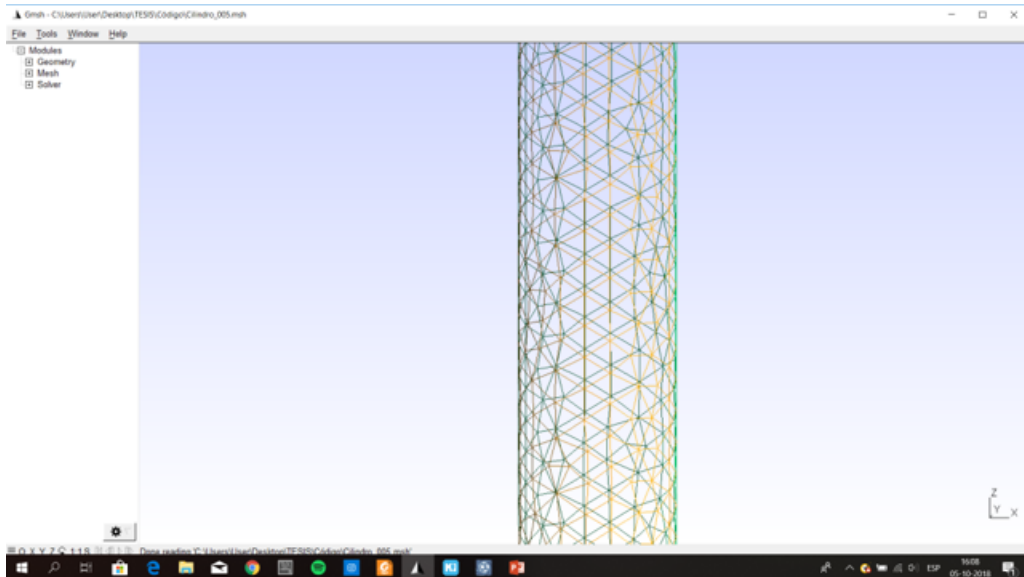


Figura 7.7: Malla detalle elementos

En el software *Gmsh* también es posible verificar la dirección de los vectores normales y tangentes, por medio de inspección visual. Esto no es suficiente, por lo que abrimos la malla en *MeshLab*[6] y verificamos no solo los puntos anteriores, sino también:

- Geometrías bien ubicadas.
- Vectores normales en la dirección deseada.
- Caras no duplicadas.

INSERTAR IMAGEN DE LA MALLA EN MESHLAB

## 7.2 Desarrollo del problema.

Como bien hablamos en capítulos anteriores, la resolución de la problemática planteada se realizará bajo la plataforma de *Jupyter Notebook*.

Es importante partir comenzando importando librerías que serán necesarias para la resolución de los problemas, obviamente tenemos a *bempp* y a *numpy*, una librería fundamental de python que contiene funciones de algebra lineal y arreglos de N-dimensiones que son utilizadas en todo método de resolución de ecuaciones diferenciales. Además de esto, se definen los nodos utilizados para realizar la cuadratura de Gauss.

```
# Import libraries
import bempp.api
import numpy as np
bempp.api.set_ipython_notebook_viewer()

# Set quadrature options
bempp.api.global_parameters.quadrature.near.double_order = 4
bempp.api.global_parameters.quadrature.medium.double_order = 4
bempp.api.global_parameters.quadrature.far.double_order = 4
```

Se define la malla, la cual es importada y creada como se explica en la subsección 7.1. Se imprime la cantidad de elementos de la malla, lo que nos permitirá saber el tamaño de la matriz que resolveremos. Graficaremos la malla utilizando el comando *plot*, el cual grafica bajo la librería *plotly*, librería de gráficas de python y que nos permite revisar la gráfica de nuestra malla a mayor profundidad. Sin embargo, por sus ajustes por defecto la malla puede verse algo rara, pero es solo un efecto óptico.

```
# Define grid
grid = bempp.api.import_grid("Cilindro_005.msh")

# Print out the number of elements
number_of_elements = grid.leaf_view.entity_count(0)
print("The grid has {0} elements.".format(number_of_elements))

# Plot the grid
grid.plot()
```

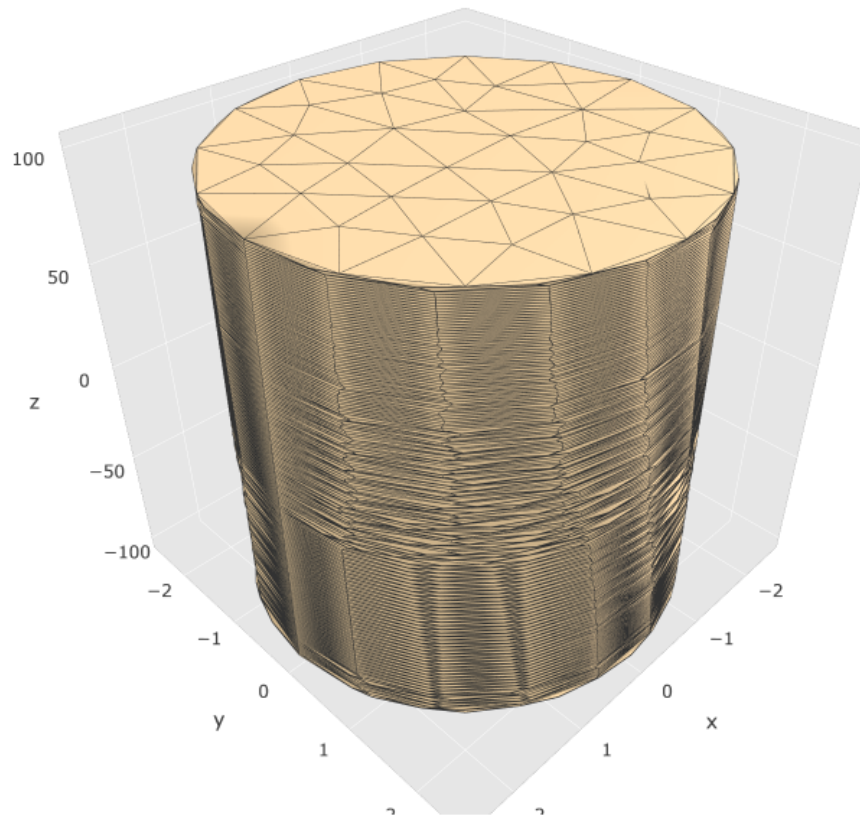


Figura 7.1: Malla con ajustes predeterminados en *plotly*

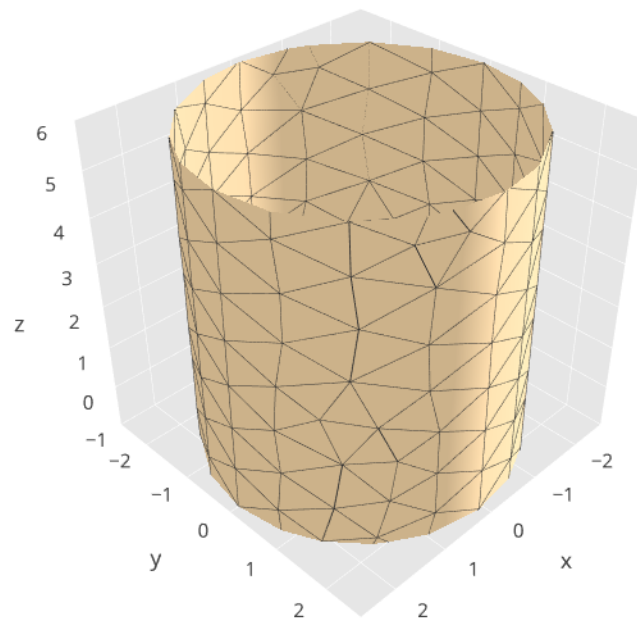


Figura 7.2: Malla con eje  $z$  no 'comprimido' en *plotly*

El siguiente paso a seguir es crear los espacios de funciones, en este caso se ha dejado como una variable el orden de los espacios. Se imprime el valor del grado de libertad del espacio. Se ingresan los datos del problemas.

```
# Set the Dirichlet and Neumann orders
order_neumann = 0
order_dirichlet = 0

# Create function spaces
global_neumann_space = bempp.api.function_space(grid, "DP", order_neumann)
global_dirichlet_space = bempp.api.function_space(grid, "DP", order_dirichlet)
NS = global_neumann_space
DS = global_dirichlet_space

# Print out the degrees of freedom
print("BEM dofs: {0}".format(NS.global_dof_count))

#Preamble

omega = 2.*np.pi*10.e9
e0 = 8.854*1e-12*1e-18
mu0 = 4.*np.pi*1e-7*1e6
mue = (1.)*mu0
ee = (16.)*e0
mui = (-2.9214+0.5895j)*mu0
ei = (82629.2677-200138.2211j)*e0
k = omega*np.sqrt(e0*mu0)
lam = 2*np.pi/k
```

Se generan los operadores que están presentes en la ecuación (7.16). Es importante hacer notar que el comando *blocked* genera una matriz capaz de almacenar operadores, este será el lado izquierdo de la ecuación.

```
#Operators
slp = bempp.api.operators.boundary.laplace.single_layer(NS,DS,DS)
dlp = bempp.api.operators.boundary.laplace.double_layer(DS,DS,DS)
id = bempp.api.operators.boundary.sparse.identity(DS,DS,DS)

#Formation of the left hand side matrix
blocked = bempp.api.BlockedOperator(2, 2)
blocked[0, 0] = 0.5 * id + dlp
blocked[0, 1] = -slp
blocked[1, 0] = 0.5 * id - dlp
blocked[1, 1] = ep1/ep2 * slp
```

El lado derecho de la ecuación está definido como una concatenación de dos funciones, que en el caso de la ecuación (7.16) estas funciones son 0 y  $\frac{\partial \varphi_i}{\partial n}$ , al definir  $\varphi = e^{jkx}$  se puede representar su derivada en la normal como se presenta a continuación. Además estas funciones, recordemos, viven

en un espacio de funciones que hemos definido más arriba, por lo que se debe definir la función dentro del espacio correspondiente para generar la 'función de malla':

```
#Definition of functions
def funcion1(x, n, domain_index, result):
    result[:] = ((ep2 - ep1) / ep2) * (1. * 1j * k * n[0] * np.exp(1j * k * x[0]))
    result[:] = ( Amp * n[0])

def cero(x, n, domain_index, result):
    result[:] = 0

#Formation functions in the space
funcion_fun = bempp.api.GridFunction(DS, fun=funcion1)
cero_fun = bempp.api.GridFunction(NS, fun=cero)
```

Para resolver el sistema de ecuaciones existen varias formas de hacerlo, con distinta rapidez, precisión, entre otros factores. En la mayoría podemos ver como converge la solución, la cantidad de iteraciones realizadas para llegar a esta y la precisión de la misma, que en este caso viene dado por el ajuste *tol*. La discretización de ambos lados de la ecuación viene por el comando *use\_strong\_form*, lo cual puede realizarse separadamente, pero aumenta la cantidad de capacidad computacional necesaria.

```
#Solving the matrix system
sol, info, it_count = bempp.api.linalg.gmres(blocked, [cero_fun, funcion_fun],
                                             use_strong_form=True, return_iteration_count=True, tol=1e-3)

#Print number of iterations
print("The linear system was solved in {0} iterations".format(it_count))
```

Cuando se resuelve el sistema de matrices, se obtiene '2' soluciones. Como vemos en la ecuación (7.16) obtenemos  $\varphi_1$  y  $\frac{\partial \varphi_1}{\partial n}$ , lo que visto en el código serían la *solution\_dirichl* y *solution\_neumann* respectivamente. Como nos interesa ver la forma de la onda disipada, graficamos esta onda, es decir, *solution\_dirichl* en torno a nuestra malla de estudio.

```
#Divide the solution
solution_dirichl, solution_neumann = sol

#Plot the solution
solution_dirichl.plot()
```

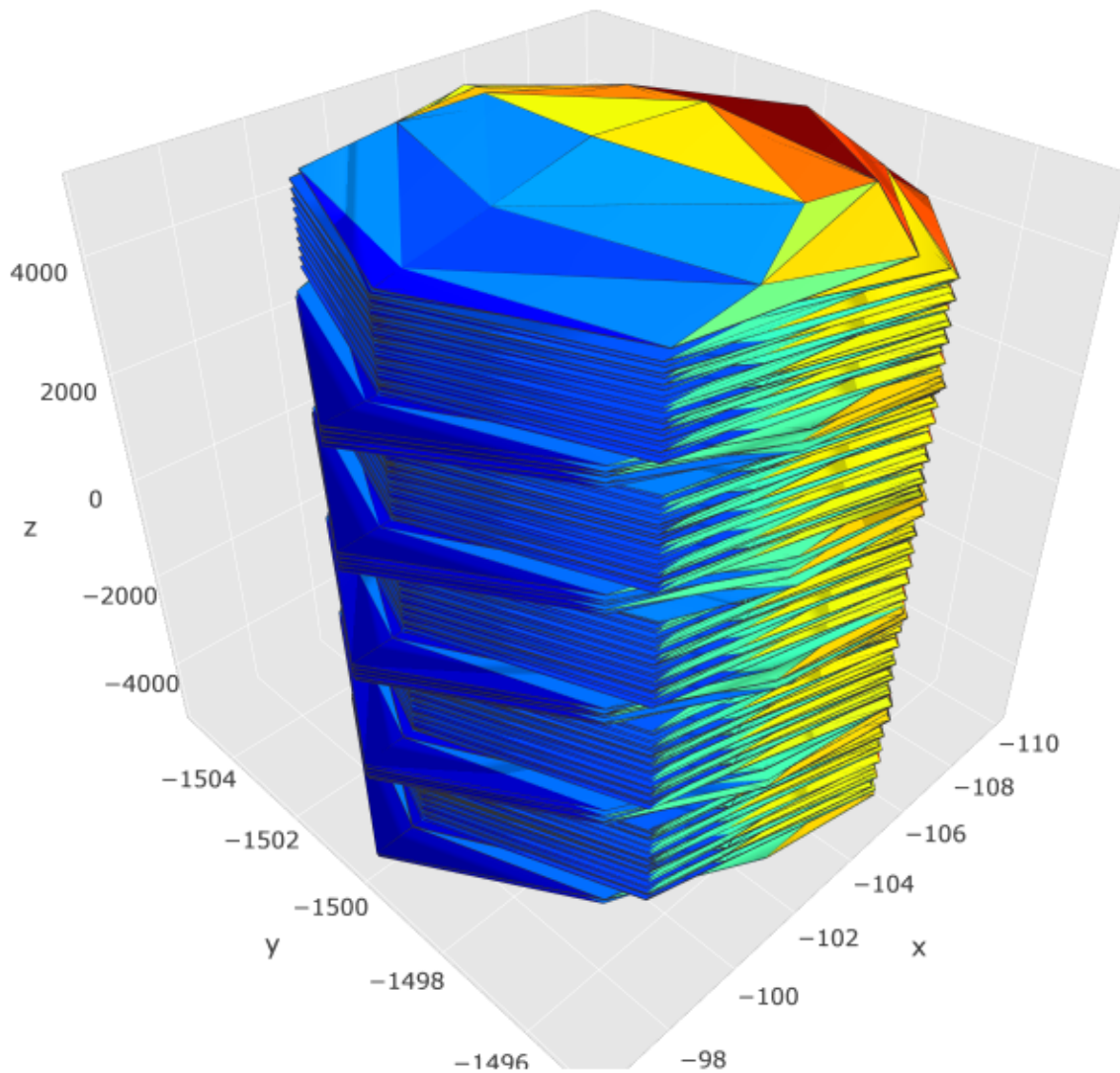


Figura 7.3: Solución situación 1

Además de ver la solución en 3D, se hace necesario ver en detalle que sucede el centro de nuestra malla, por lo que generar una malla 2D sobre un plano  $z$  determinado nos ayudaría a ver el comportamiento de la onda dentro de, en este caso, el cable.

```
#Generate the grid
n_grid_points = 200
xmin, xmax, ymin, ymax=[-3,3,-3,3]
plot_grid = np.mgrid[xmin:xmax:n_grid_points*1j,ymin:ymax:n_grid_points*1j]
points = np.vstack((plot_grid[0].ravel(),
                    plot_grid[1].ravel(),
                    np.zeros(plot_grid[0].size)))
```

Se crean las funciones de espacio utilizando la malla en 3D.

```
#Generate the spaces
dp0_space = bempp.api.function_space(grid, "DP", 0)
p1_space = bempp.api.function_space(grid, "DP", 0)
```

Se generan los operadores bajo los puntos especificados en la malla 2D, es decir bajo un plano  $z$  especificado.

```
#Generate the operators
slp_pot = bempp.api.operators.potential.laplace.single_layer(dp0_space, points)
dlp_pot = bempp.api.operators.potential.laplace.double_layer(p1_space, points)
```

Se evalúa la solución en todos los puntos del plano, que vienen siendo los puntos internos de la malla en un plano específico, es decir, se utiliza la ecuación (4.14).

```
#Evaluation in internal points
u_evaluated = slp_pot * solution_neumann - dlp_pot * solution_dirichl
```

Finalmente se grafica la solución obtenida, en este caso obtendremos una solución con una parte imaginaria, a nosotros para graficar solo nos interesa la parte real de la solución por lo que filtraremos bajo este criterio. Otro criterio de filtro es que los resultados fuera de la malla, son irrelevantes, por lo que se eliminan. —

```
# The next command ensures that plots are shown within the IPython notebook
%matplotlib inline
from matplotlib import pylab as plt

#Plot the results
fig,ax = plt.subplots()
ax.scatter(u_evaluated.real,u_evaluated.imag)
# Filter out solution values that are associated
# with points outside the unit circle.
u_evaluated = (u_evaluated).reshape((n_grid_points,n_grid_points))
radius = np.sqrt(plot_grid[0]**2 + plot_grid[1]**2)
u_evaluated[radius>2] = np.nan
fig = plt.figure(figsize=(10, 8))
plt.imshow((u_evaluated.real)), extent=(-3,3,-3,3))
plt.title('Computed solution')
plt.colorbar()
```

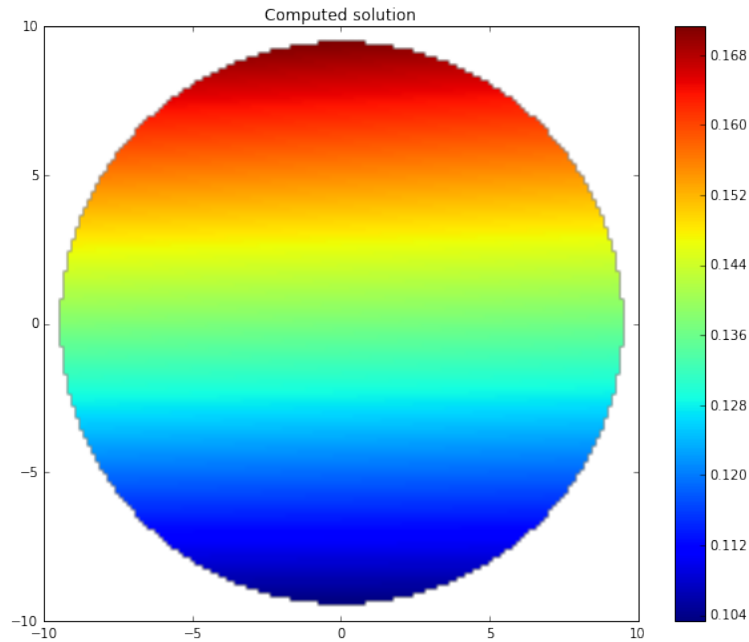


Figura 7.4: Solución situación 1

```
#Import libraries
import bempp.api
import numpy as np
import timeit
bempp.api.set_ipython_notebook_viewer()

#Set quadrature options
bempp.api.global_parameters.quadrature.near.double_order = 2
bempp.api.global_parameters.quadrature.medium.double_order = 2
bempp.api.global_parameters.quadrature.far.double_order = 2

#Define grid
grid_0 = bempp.api.import_grid("BH21_a5_l10_E5550.msh")

# Print out the number of elements
number_of_elements = grid_0.leaf_view.entity_count(0)
print("The grid has {0} elements.".format(number_of_elements))

#Plot the grid
grid_0.plot()
```



```

#Problem data
omega = 2.*np.pi*10.e9
e0 = 8.854*1e-12*1e-18
mu0 = 4.*np.pi*1e-7*1e6
mue = (1.)*mu0
ee = (16.)*e0
mui = (-2.9214+0.5895j)*mu0
ei = (82629.2677-200138.2211j)*e0
k = omega*np.sqrt(e0*mu0)
lam = 2*np.pi/k
nm = np.sqrt((ee*mue)/(e0*mu0))
nc = np.sqrt((ei*mui)/(e0*mu0))
alfa_m = mue/mu0
alfa_c = mui/mue
antena = np.array([[1e4],[0.],[0.]])
Amp = 1
print("Numero de onda exterior:", k)
print("Indice de refraccion matriz:", nm)
print("Indice de refraccion conductor:", nc)
print("Numero de onda interior matriz:", nm*k)
print("Numero de onda interior conductor:", nm*nc*k)
print("Indice de transmision matriz:", alfa_m)
print("Indice de transmision conductor:", alfa_c)
print("Longitud de onda:", lam, "micras")

```

```

#Dirichlet and Neumann functions
def dirichlet_fun(x, n, domain_index, result):
    result[0] = Amp * np.exp(1j * k * x[1])
def neumann_fun(x, n, domain_index, result):
    result[0] = Amp * 1j * k * n[1] * np.exp(1j * k * x[1])

```

```

#Multitrace Operators
Ai_0 = bempp.api.operators.boundary.helmholtz.multitrace_operator(grid_0,
    nm * nc * k)
Ae_0 = bempp.api.operators.boundary.helmholtz.multitrace_operator(grid_0,
    nm * k)

#Multitrace Transmission
Ai_0[0,1] = Ai_0[0,1]*alfa_c
Ai_0[1,1] = Ai_0[1,1]*alfa_c

#Interior and exterior link up
op_0 = (Ai_0 + Ae_0)

```

```
#Spaces
dirichlet_space_0 = Ai_0[0,0].domain
neumann_space_0 = Ai_0[0,1].domain
```

```
#Identity
ident_0 = bempp.api.operators.boundary.sparse.identity(neumann_space_0,
                                                         neumann_space_0, neumann_space_0)
```

```
#Left hand side
blocked = bempp.api.BlockedOperator(2,2)
blocked[0,0] = op_0[0,0]
blocked[0,1] = op_0[0,1]
blocked[1,0] = op_0[1,0]
blocked[1,1] = op_0[1,1]
blocked[1,1] = blocked[1,1] + 0.5 * ident_0 * (alfa_c - 1)

#Boundary conditions
dirichlet_grid_fun_0 = bempp.api.GridFunction(dirichlet_space_0,
                                                fun=dirichlet_fun)
neumann_grid_fun_0 = bempp.api.GridFunction(neumann_space_0, fun=neumann_fun)
```

```
start = timeit.default_timer()
#Solving the matrix system
sol, info, it_count = bempp.api.linalg.gmres(blocked,
                                              [dirichlet_grid_fun_0, neumann_grid_fun_0],
                                              use_strong_form=True, return_iteration_count=True,
                                              tol=1e-5, restart=20, maxiter=500000)
stop = timeit.default_timer()

#Print iterations
print("The linear system was solved in {0} iterations".format(it_count))
print('Time: ', stop - start)
```

```
#Divide the solution
solution_dirichl, solution_neumann = sol
print(sol)

#Plot the solution
solution_neumann.plot()
```

```

Nx = 150
Ny = 150
di=10
xmin, xmax, ymin, ymax = [-di,di,-di,di]
plot_grid = np.mgrid[xmin:xmax:Nx * 1j, ymin:ymax:Ny * 1j]
points = np.vstack((plot_grid[0].ravel(),
                    plot_grid[1].ravel(),
                    0*np.ones(plot_grid[0].size)))
u_evaluated = np.zeros(points.shape[1], dtype=np.complex128)

x, y = points[:2]
idx_ext = np.sqrt(x**2 + y**2) > di
idx_int = np.sqrt(x**2 + y**2) <= di

points_exterior = points[:, idx_ext]
points_interior = points[:, idx_int]

```

```

slp_pot_int = bempp.api.operators.potential.helmholtz.single_layer(
    dirichlet_space_0, points_interior, nm * k)
slp_pot_ext = bempp.api.operators.potential.helmholtz.single_layer(
    dirichlet_space_0, points_exterior, k)
dlp_pot_int = bempp.api.operators.potential.helmholtz.double_layer(
    dirichlet_space_0, points_interior, nm * k)
dlp_pot_ext = bempp.api.operators.potential.helmholtz.double_layer(
    dirichlet_space_0, points_exterior, k)

total_field_int = (slp_pot_int * solution_neumann
                  - dlp_pot_int * solution_dirichl).ravel()
total_field_ext = (dlp_pot_ext * solution_dirichl
                  - slp_pot_ext * solution_neumann).ravel() \
                  + np.exp(1j * k * points_exterior[0])

total_field = np.zeros(points.shape[1], dtype='complex128')
total_field[idx_ext] = total_field_ext
total_field[idx_int] = total_field_int
total_field = total_field.reshape([Nx, Ny])

```

```

%matplotlib inline
from matplotlib import pylab as plt
radius = np.sqrt(plot_grid[0]**2 + plot_grid[1]**2)
total_field[radius>di] = np.nan
fig = plt.figure(figsize=(10, 10))
plt.imshow(np.real(total_field.T), extent=[-di,di,-di,di])
plt.xlabel('x')
plt.ylabel('y')
plt.colorbar()

```

# Capítulo 8

## Conclusiones.

## Capítulo 9

# Anexos

### 9.1 Anexo I: Código 1.

# C1\_E2\_CIL Cilindro (1)

November 8, 2018

We are going to emulate the behavior of a magnetic field in microwires. Using the boundary element method and using Laplace as Green function, we're going to solve:

$$\begin{bmatrix} -K_{ext} - K_{int} & \alpha V_{int} + V_{ext} \\ -K'_{ext} - K'_{int} & \frac{\alpha-1}{2} + \alpha V'_{int} + V'_{ext} \end{bmatrix} \begin{bmatrix} \varphi_{ext} \\ \frac{\partial}{\partial n} \varphi_{ext} \end{bmatrix} = \begin{bmatrix} \varphi_{inc} \\ \frac{\partial \varphi_{inc}}{\partial n} \end{bmatrix} \quad (1)$$

Being  $K$  the double layer operator,  $V$  the single layer operator and  $I$  the identity. First of all, we call all the libraries that we'll use.

```
In [2]: #Import libraries
import bempp.api
import numpy as np
import timeit
bempp.api.set_ipython_notebook_viewer()

#Set quadrature options
bempp.api.global_parameters.quadrature.near.double_order = 2
bempp.api.global_parameters.quadrature.medium.double_order = 2
bempp.api.global_parameters.quadrature.far.double_order = 2
```

Now we import the mesh that we'll use, plotting them and printing the number of elements presents in the mesh.

```
In [3]: #Define grid
grid_0 = bempp.api.import_grid("BH21_a5_110_E5550.msh")

# Print out the number of elements
number_of_elements = grid_0.leaf_view.entity_count(0)
print("The grid has {0} elements.".format(number_of_elements))

#Plot the grid
#grid_0.plot()
```

The grid has 5550 elements.

First of all we have to define some constants:

```

In [4]: #Problem data
omega = 2.*np.pi*10.e9
e0 = 8.854*1e-12*1e-18
mu0 = 4.*np.pi*1e-7*1e6
mue = (1.)*mu0
ee = (16.)*e0
mui = (-2.9214+0.5895j)*mu0
ei = (82629.2677-200138.2211j)*e0
k = omega*np.sqrt(e0*mu0)
lam = 2*np.pi/k
nm = np.sqrt((ee*mue)/(e0*mu0))
nc = np.sqrt((ei*mui)/(e0*mu0))
alfa_m = mue/mu0
alfa_c = mui/mue
antena = np.array([[1e4],[0.],[0.]])
Amp = 1
print("Numero de onda exterior:", k)
print("Indice de refraccion matriz:", nm)
print("Indice de refraccion conductor:", nc)
print("Numero de onda interior matriz:", nm*k)
print("Numero de onda interior conductor:", nm*nc*k)
print("Indice de transmision matriz:", alfa_m)
print("Indice de transmision conductor:", alfa_c)
print("Longitud de onda:", lam, "micras")

Numero de onda exterior: 0.0002095822793
Indice de refraccion matriz: 4.0
Indice de refraccion conductor: (510.829219424+619.966251289j)
Numero de onda interior matriz: 0.000838329117198
Numero de onda interior conductor: (0.428243008559+0.519735760136j)
Indice de transmision matriz: 1.0
Indice de transmision conductor: (-2.9214+0.5895j)
Longitud de onda: 29979.5637693 micras

```

Also, we have to define the boundary functions that we will use to apply the boundary conditions. In this case an armonic wave for Dirichlet and his derivate for Neumann

```

In [5]: #Dirichlet and Neumann functions
def dirichlet_fun(x, n, domain_index, result):
    result[0] = Amp * np.exp(1j * k * x[1])
def neumann_fun(x, n, domain_index, result):
    result[0] = Amp * 1j * k * n[1] * np.exp(1j * k * x[1])

```

Now it's time to define the multitrace operators that represent the diagonal of the matrix. This operators have the information of the transmission between the geometries. The definition of the multitrace (A) is posible to see below:

$$A = \begin{bmatrix} -K & S \\ D & K' \end{bmatrix}$$



where  $K$  represent the double layer boundary operator,  $S$  the single layer,  $D$  the hypersingular and  $K'$  the adjoint double layer boundary operator

In [6]: *#Multitrace Operators*

```
Ai_0 = bempp.api.operators.boundary.helmholtz.multitrace_operator(grid_0, nm * nc * k)
Ae_0 = bempp.api.operators.boundary.helmholtz.multitrace_operator(grid_0, nm * k)

#Multitrace Transmission
Ai_0[0,1] = Ai_0[0,1]*alfa_c
Ai_0[1,1] = Ai_0[1,1]*alfa_c

#Interior and exterior link up
op_0 = (Ai_0 + Ae_0)
```

In order to obtain the spaces created with the multitrace operator it's possible to do the following:

In [7]: *#Spaces*

```
dirichlet_space_0 = Ai_0[0,0].domain
neumann_space_0 = Ai_0[0,1].domain
```

To make the complete diagonal of the main matrix showed at beginning is necessary to define the identity operators:

In [8]: *#Identity*

```
ident_0 = bempp.api.operators.boundary.sparse.identity(neumann_space_0, neumann_space_0)
```

The first subindex corresponds to the domain space, the second one to the range space. Now is time to create the big block that will have all the operators together, in this case the size is  $2 \times 2$ . And we create the right hand side and the left hand side of the equation

In [9]: *#Left hand side*

```
blocked = bempp.api.BlockedOperator(2,2)
blocked[0,0] = op_0[0,0]
blocked[0,1] = op_0[0,1]
blocked[1,0] = op_0[1,0]
blocked[1,1] = op_0[1,1]
blocked[1,1] = blocked[1,1] + 0.5 * ident_0 * (alfa_c - 1)

#Boundary conditions
dirichlet_grid_fun_0 = bempp.api.GridFunction(dirichlet_space_0, fun=dirichlet_fun)
neumann_grid_fun_0 = bempp.api.GridFunction(neumann_space_0, fun=neumann_fun)

#Discretization lado derecho
#rhs = np.concatenate([dirichlet_grid_fun_0.coefficients, neumann_grid_fun_0.coefficients])
#Discretization lado izquierdo
#blocked_discretizado = blocked.strong_form()
```

Now we solve the equation system, we use *gmres*. We print the number of iterations. Notice that the argument *use\_strong\_form = True* is the responsible for the discretization of both sides of the equation.

```
In [10]: start = timeit.default_timer()
         #Solving the matrix system
         sol, info, it_count = bempp.api.linalg.gmres(blocked, [dirichlet_grid_fun_0, neumann_
                                                         use_strong_form=True, return_iteration_co

         stop = timeit.default_timer()
         #Print iterations
         print("The linear system was solved in {0} iterations".format(it_count))
         print('Time: ', stop - start)

/usr/lib/python3/dist-packages/scipy/sparse/linalg/dsolve/linsolve.py:243: SparseEfficiencyWarning
warn('splu requires CSC matrix format', SparseEfficiencyWarning)
```

The linear system was solved in 199 iterations  
Time: 352.68147672800114

```
In [11]: #Divide the solution
         solution_dirichl, solution_neumann = sol
         print(sol)
         #Plot the solution
         solution_neumann.plot()

[<bempp.api.assembly.grid_function.GridFunction object at 0x7ff04af96b00>, <bempp.api.assembly
```

```
In [26]: Nx = 150
         Ny = 150
         di=10
         xmin, xmax, ymin, ymax = [-di,di,-di,di]
         plot_grid = np.mgrid[xmin:xmax:Nx * 1j, ymin:ymax:Ny * 1j]
         points = np.vstack((plot_grid[0].ravel(),
                               plot_grid[1].ravel(),
                               0*np.ones(plot_grid[0].size)))
         u_evaluated = np.zeros(points.shape[1], dtype=np.complex128)

         x, y = points[:2]
         idx_ext = np.sqrt(x**2 + y**2) > di
         idx_int = np.sqrt(x**2 + y**2) <= di

         points_exterior = points[:, idx_ext]
         points_interior = points[:, idx_int]

         print(points_interior)

[[-9.86577181 -9.86577181 -9.86577181 ...,  9.86577181  9.86577181
  9.86577181]
 [-1.54362416 -1.40939597 -1.27516779 ...,  1.27516779  1.40939597
  1.54362416]
```

```
[ 0.          0.          0.          ...,  0.          0.          0.          ]]
```

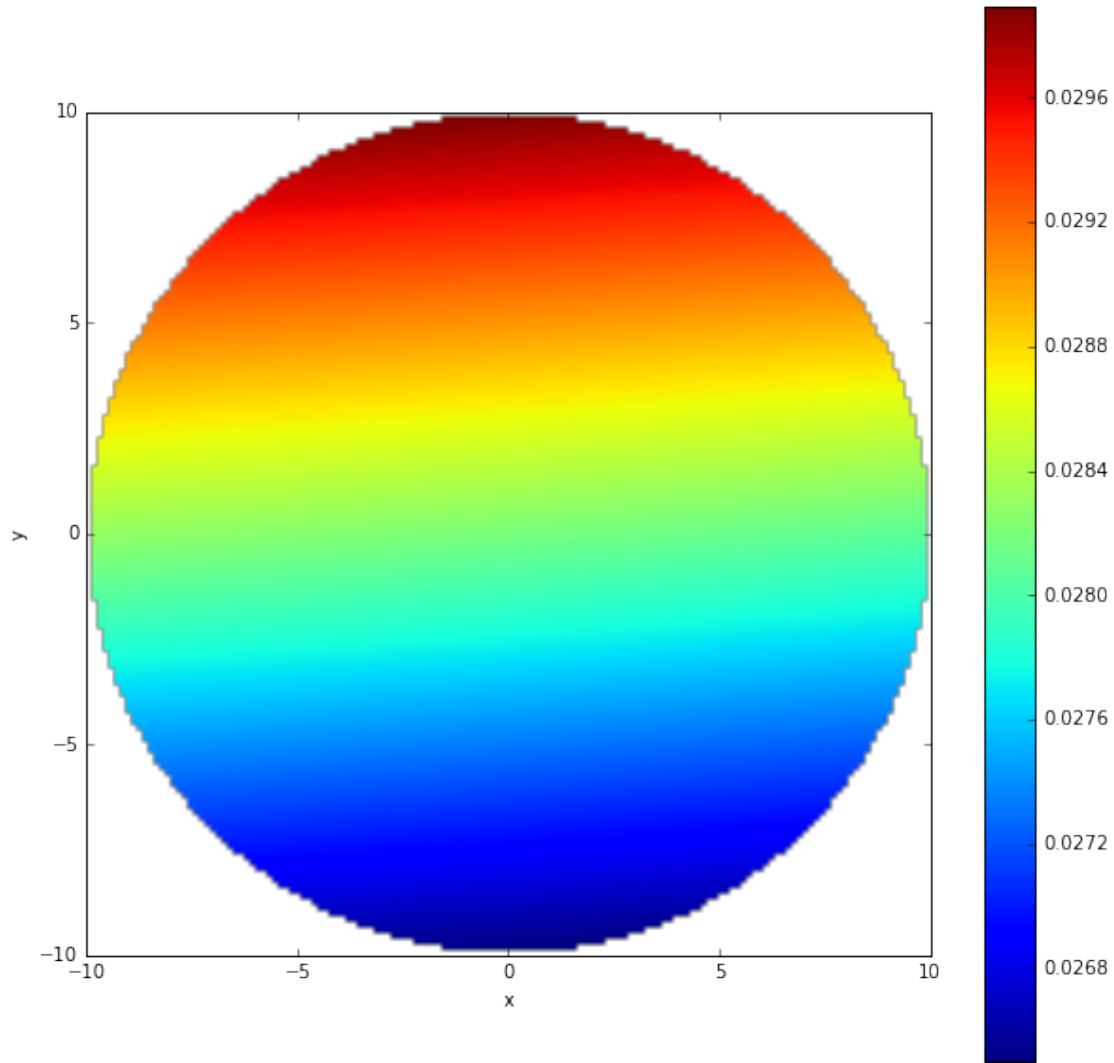
```
In [17]: slp_pot_int = bempp.api.operators.potential.helmholtz.single_layer(
        dirichlet_space_0, points_interior, nm * k)
        slp_pot_ext = bempp.api.operators.potential.helmholtz.single_layer(
        dirichlet_space_0, points_exterior, k)
        dlp_pot_int = bempp.api.operators.potential.helmholtz.double_layer(
        dirichlet_space_0, points_interior, nm * k)
        dlp_pot_ext = bempp.api.operators.potential.helmholtz.double_layer(
        dirichlet_space_0, points_exterior, k)

        total_field_int = (slp_pot_int * solution_neumann
                           - dlp_pot_int * solution_dirichl).ravel()
        total_field_ext = (dlp_pot_ext * solution_dirichl
                           - slp_pot_ext * solution_neumann).ravel() \
        + np.exp(1j * k * points_exterior[0])

        total_field = np.zeros(points.shape[1], dtype='complex128')
        total_field[idx_ext] = total_field_ext
        total_field[idx_int] = total_field_int
        total_field = total_field.reshape([Nx, Ny])

In [14]: %matplotlib inline
        from matplotlib import pylab as plt
        radius = np.sqrt(plot_grid[0]**2 + plot_grid[1]**2)
        total_field[radius>di] = np.nan
        fig = plt.figure(figsize=(10, 10))
        plt.imshow(np.real(total_field.T), extent=[-di,di,-di,di])
        plt.xlabel('x')
        plt.ylabel('y')
        plt.colorbar()
```

```
Out[14]: <matplotlib.colorbar.Colorbar at 0x7ff0151bce10>
```



```
In [20]: Nx = 300
        Ny = 300
        xmin, xmax, ymin, ymax = [-13, 13, -13, 13]
        plot_grid = np.mgrid[xmin:xmax:Nx * 1j, ymin:ymax:Ny * 1j]
        points = np.vstack((plot_grid[0].ravel(),
                             plot_grid[1].ravel(),
                             np.zeros(plot_grid[0].size)))
        u_evaluated = np.zeros(points.shape[1], dtype=np.complex128)

        x, y = points[:2]
        idx_ext = np.sqrt(x**2 + y**2) > 10
        idx_int = np.sqrt(x**2 + y**2) <= 10

        points_exterior = points[:, idx_ext]
        points_interior = points[:, idx_int]
```

```

In [23]: slp_pot_int = bempp.api.operators.potential.helmholtz.single_layer(
        dirichlet_space_0, points_interior, nm * k)
        slp_pot_ext = bempp.api.operators.potential.helmholtz.single_layer(
        dirichlet_space_0, points_exterior, k)
        dlp_pot_int = bempp.api.operators.potential.helmholtz.double_layer(
        dirichlet_space_0, points_interior, nm * k)
        dlp_pot_ext = bempp.api.operators.potential.helmholtz.double_layer(
        dirichlet_space_0, points_exterior, k)

        total_field_int = (slp_pot_int * solution_neumann
                           - dlp_pot_int * solution_dirichl).ravel()
        total_field_ext = (dlp_pot_ext * solution_dirichl
                           - slp_pot_ext * solution_neumann).ravel() \
                           + np.exp(1j * k * points_exterior[0])

        total_field = np.zeros(points.shape[1], dtype='complex128')
        total_field[idx_ext] = total_field_ext
        total_field[idx_int] = total_field_int
        total_field = total_field.reshape([Nx, Ny])

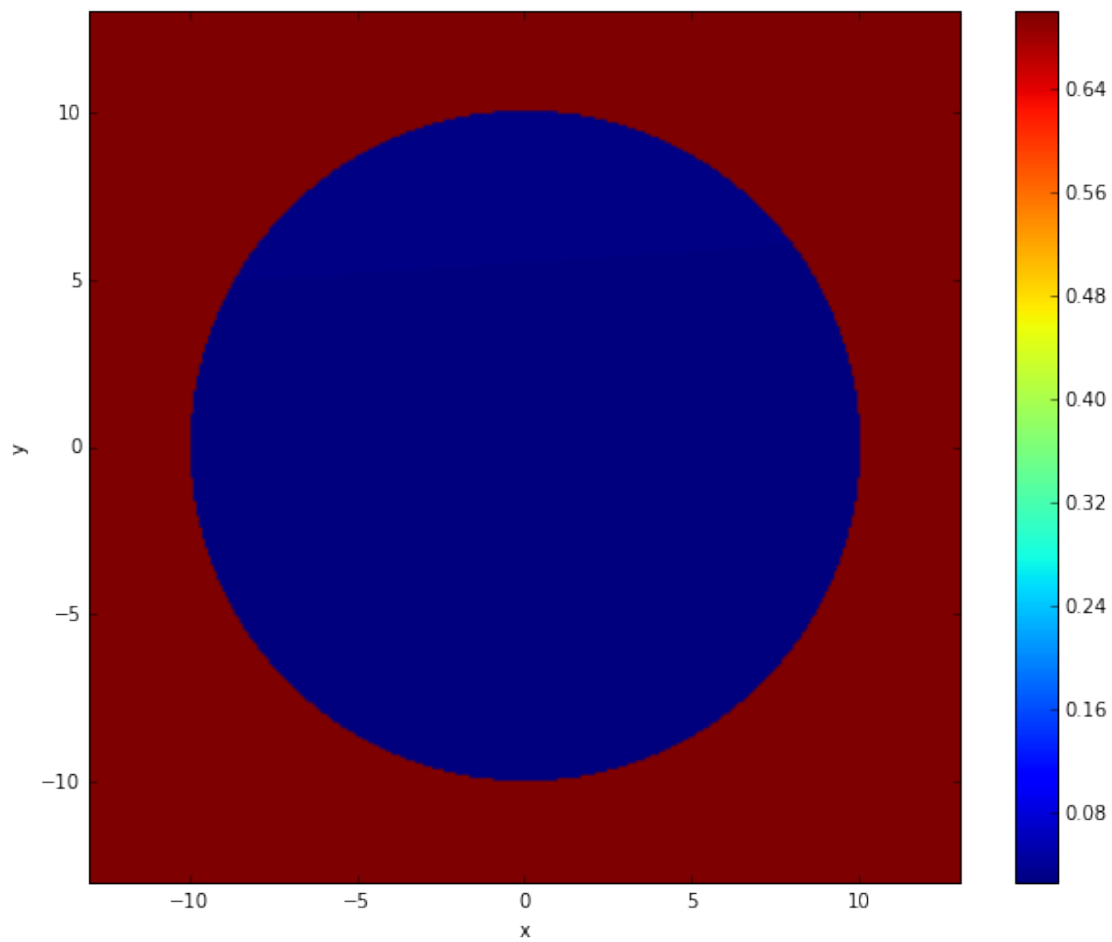
In [24]: %matplotlib inline
        from matplotlib import pylab as plt
        fig = plt.figure(figsize=(10, 8))
        plt.imshow(np.real(total_field.T), extent=[-13, 13, -13, 13])
        plt.xlabel('x')
        plt.ylabel('y')
        plt.colorbar()

```

```

Out[24]: <matplotlib.colorbar.Colorbar at 0x7ff068ab49e8>

```



# Referencias

- [1] *Gmsh*. URL <http://gmsh.info/>.
- [2] *Articulo wikipedia. Campo Vectorial*. URL [https://es.wikipedia.org/wiki/Campo\\_vectorial](https://es.wikipedia.org/wiki/Campo_vectorial).
- [3] *bempp services*. URL <https://bempp.com/>.
- [4] *Campos vectoriales*. Universidad Nacional de la Plata, 2014. URL [http://www.mate.unlp.edu.ar/practicas/114\\_8\\_16062015111652.pdf](http://www.mate.unlp.edu.ar/practicas/114_8_16062015111652.pdf).
- [5] C.A. Brebbia and J. Dominguez. *Boundary Elements: An Introductory Course*. Sydney Grammar School Press, 1994. ISBN 9781853123498. URL <https://books.google.cl/books?id=YBE1JxdldHYC>.
- [6] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. *MeshLab: an Open-Source Mesh Processing Tool*. The Eurographics Association, 2008. ISBN 978-3-905673-68-5. doi: 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136.
- [7] R.A. Serway, J.W. Jewett, and S.R.C. González. *Física para ciencias e ingeniería. Vol. 2 (9a. ed.)*. CENGAGE Learning, 2015. ISBN 9786075192017. URL <https://books.google.cl/books?id=zVuXAQAACAAJ>.
- [8] W. Śmigaj, T. Betcke, S. Arridge, J. Phillips, and M. Schweiger. Solving boundary integral problems with BEM++. *ACM Transactions on mathematical software*, 41(2):6:1–6:40, 2015. doi: {10.1145/2590830}.