



Industriales

Escuela Técnica Superior
de Ingeniería Industrial

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

Cálculo de los parámetros eléctricos de un transformador por medio del método de los elementos finitos

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

Autor: Rubén Sánchez Martos.
Director: Juan Álvaro Fuentes Moreno.
Codirector: Francisco Periago Esparza.

Cartagena, Curso 2014/2015



Universidad
técnica
Cartagena

Índice

1. Introducción.....	4
1.1 Objetivos.....	4
1.1 Método de elementos finitos.....	4
2. Programas	5
2.1 FreeFem++.....	5
2.2 ANSYS Maxwell	5
3. El método de los elementos finitos en electromagnetismo.....	5
4. Análisis del problema en FreeFem++.....	9
4.1 Transformador en 2D.....	10
4.1.1 Geometría y mallado del problema	10
4.1.2 Definición de parámetros	14
4.1.3 Resolución del problema y resultados.....	17
4.1.4 Código del programa.....	23
4.2 Transformador en 3D.....	24
4.2.1 Geometría y límites del problema	24
4.2.2 Definición de parámetros	28
4.2.3 Resolución del problema y resultados.....	29
4.2.4 Código del problema	35
5. Análisis del problema en ANSYS Maxwell.....	35
5.1 Diseño de la geometría y límites del problema.....	35
5.2 Definición de los parámetros del problema	37
5.3 Resolución del problema y resultados	39
6. Comparación entre los resultados 2D y 3D de FreeFem++	41
7. Comparativa entre los resultados de ANSYS y FreeFem++	42
8. Conclusiones.....	43
9. Referencias bibliográficas	45
10. Anexo	46
10.1 Anexo I: Planos del núcleo del transformador ETD 49/25/16.....	46
10.2 Anexo II: Definición de las características del hilo de litz	48
10.3 Anexo III: Código en FreeFem++ del problema 2D.....	49
10.4 Anexo IV: Permeabilidad del acero del núcleo del transformador.....	64

10.5 Anexo V: Código en FreeFem++ del mallado por partes en 3D	65
10.6 Anexo VI: Código en FreeFem++ del mallado en 3D extruido a partir del 2D .	86
10.7 Anexo VII: Código en FreeFem++ del problema en 3D	94

1. Introducción.

En este apartado haremos una breve introducción de los objetivos de nuestro proyecto así como una pequeña explicación del método de elementos finitos y su utilización en problemas de ingeniería.

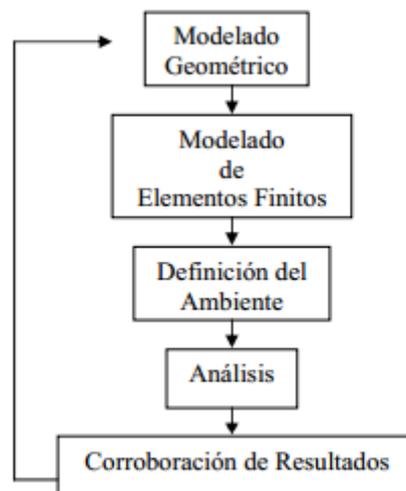
1.1 Objetivos.

El objetivo principal de este proyecto es realizar el diseño de un transformador con dos programas informáticos y comprobar los parámetros de diseño que éstos nos proporcionan con los reales, para poder observar cuál de los dos programas es más preciso a la hora de calcular estos parámetros, así como explicar los pros y los contras de cada uno de ellos.

1.1 Método de elementos finitos.

El método de los elementos finitos (MEF) es un procedimiento que permite resolver ecuaciones diferenciales de forma numérica y aproximada sustituyendo este problema diferencial por uno algebraico aproximadamente semejante, para el cual conocemos la solución. Para esto, se hace uso de la discretización de una región (sobre la cual están descritas las ecuaciones de forma geométrica) denominada elementos finitos.

En el proceso de análisis de los elementos finitos se distinguen las siguientes partes:



Modelado geométrico: Crear el modelo matemático del objeto o del conjunto. Reproducir tanto el sólido como la geometría de la superficie de forma precisa.

Modelado de elementos finitos: Subdividir la geometría del modelo en elementos discretos y asignar las propiedades del material y del elemento.

Definición del ambiente: Aplicar las cargas y las condiciones de contorno para simular las condiciones del entorno.

Análisis: calcular los resultados a partir de análisis estáticos, dinámicos o de transferencia de calor.

Corroboración de resultados: Comparar los resultados con los criterios de diseño. Rediseñar la estructura y repetir el proceso si fuese necesario.

El método de los elementos finitos supuso un punto de inflexión en ingeniería, ya que permitió el cálculo aproximado de problemas complejos sin la necesidad de la fabricación de tantos prototipos, esto hizo posible disminuir los tiempos y costes de fabricación al permitir realizar los primeros pasos del diseño mediante simulaciones numéricas.

2. Programas.

En este apartado vamos a presentar los dos software que vamos a utilizar para la resolución de nuestro problema de electromagnetismo.

2.1 FreeFem++.

FreeFem++ es un software libre creado para la resolución de ecuaciones diferenciales parciales de forma numérica aproximada mediante el método de los elementos finitos, que utiliza su propio lenguaje de programación de nivel alto. Este software se utiliza para resolver numerosos problemas de campos como la física e ingeniería.

Desarrollado en la Universidad "*Pierre y Marie Curie*" de París, su primera versión fue creada en 1987 por Olivier Pironneau, y fue llamada MacFem. En 1992, fue reescrito en C++ y fue llamado FreeFem. Versiones posteriores, FreeFem+ y FreeFem++ usaron el mismo lenguaje.

2.2 ANSYS Maxwell.

ANSYS es un software comercial de simulación de campo electromagnético para los ingenieros encargados de diseñar y analizar los dispositivos electromagnéticos y electromecánicos en 2-D y 3-D, estos incluyen motores, actuadores, sensores, bobinas y transformadores. ANSYS Maxwell utiliza el método de elementos finitos para resolver problemas estáticos, dominio de la frecuencia y los campos electromagnéticos y eléctricos variables en el tiempo. Un beneficio clave de este software es su proceso de solución automatizada, por la cual el usuario solo tendrá que especificar la geometría, las propiedades del material y la salida deseada. En función de estos parámetros el propio programa se encarga de generar la malla más adecuada para la solución de nuestro problema. Este proceso de mallado adaptativo automático elimina la complejidad del proceso de análisis y permite beneficiarse de un flujo altamente eficiente de diseño fácil de usar.

3. El método de los elementos finitos en electromagnetismo.

En este apartado definiremos las ecuaciones y magnitudes que vamos a necesitar para la resolución de nuestro problema.

Así pues, en primer lugar, explicaremos algunas magnitudes clave de nuestro problema:

Densidad de corriente (\vec{j}): Es una magnitud vectorial que se define como la cantidad de carga que circula por unidad de tiempo y superficie. A partir de esta definición su expresión quedará dada por:

$$\vec{j} = \frac{dI}{ds} \vec{u} \quad (3.1)$$

Siendo \vec{u} el vector perpendicular a la superficie s .

Permeabilidad magnética (μ): Es la capacidad de una sustancia o medio para atraer y hacer pasar a través de ella campos magnéticos. Esta magnitud está dada por la relación entre la inducción magnética y la intensidad de campo magnético que aparece en el material:

$$\mu = \frac{B}{H} \quad (3.2)$$

Siendo la permeabilidad referencia la del vacío, $\mu_0 = 4\pi * 10^{-7} NA^{-2}$, la permeabilidad relativa de cualquier material vendrá dada por:

$$\mu_r = \frac{\mu}{\mu_0} \quad (3.3)$$

Potencial vector magnético o potencial vectorial (\vec{A}): Es campo vectorial tridimensional, aquel cuyo conocimiento nos permite hallar el campo magnético. Debido a la inexistencia de cargas magnéticas en nuestro problema el campo magnético es solenoidal, que significa que éste puede escribirse como el rotacional del potencial vector magnético:

$$\nabla \cdot \vec{A} = 0 \rightarrow \vec{B} = \nabla \times \vec{A} \quad (3.4)$$

A su vez este puede relacionarse con el vector densidad de corriente:

$$\Delta \cdot \vec{A} = \mu_0 \cdot \vec{j} \quad (3.5)$$

Campo densidad de flujo magnético (\vec{B}): Es un campo vectorial que representa la influencia magnética de las corrientes eléctricas y los materiales magnéticos. De las ecuaciones de Maxwell en el vacío deducimos que:

$$\nabla \cdot \vec{B} = 0$$

Debido a que el campo magnético es solenoidal.

$$\nabla \times \vec{B} = \mu_0 \vec{j} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t} \quad (3.6)$$

Como el campo eléctrico de nuestro problema es estacionario, las derivadas respecto al tiempo serán nulas, quedando la expresión anterior definida como:

$$\nabla \times \vec{B} = \mu_0 \vec{J} \quad (3.7)$$

Intensidad de campo magnético (\vec{H}): Es una magnitud vectorial que designa de forma inequívoca la influencia que ejerce la corriente externa en la creación de campo magnético del material, independientemente de la respuesta magnética del material.

Siendo la expresión de éste:

$$\vec{H} = \frac{\vec{B}}{\mu} - \vec{M} \quad (3.8)$$

En nuestro caso al no tener un material magnético, la componente debida al material magnético es nula y la expresión queda dada por:

$$\vec{H} = \frac{\vec{B}}{\mu} \quad (3.9)$$

Antes de empezar a definir las inductancias, hay que tener en cuenta que un sistema lineal formado por dos bobinas, tal como el transformador que estamos considerando, los enlaces de flujo concatenados por estas en función de sus intensidades vienen dados por:

$$\begin{aligned} \lambda_1 &= L_{11} \cdot i_1 + L_{12} \cdot i_2 \\ \lambda_2 &= L_{21} \cdot i_1 + L_{22} \cdot i_2 \end{aligned}$$

Inductancia propia (L_{11} y L_{22}): Se define como el coeficiente entre el producto escalar del campo magnético y la intensidad de campo magnético a través del primario y la intensidad del primario al cuadrado por diferencial de volumen. De igual manera se define la del secundario pero referida a este.

$$L_{11} = \int_V \frac{\vec{B}_1 \cdot \vec{H}_1}{i_1^2} dV ; L_{22} = \int_V \frac{\vec{B}_2 \cdot \vec{H}_2}{i_2^2} dV \quad (3.10)$$

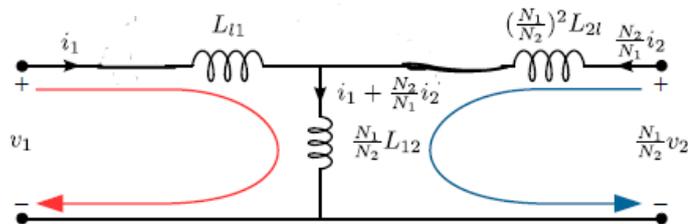
Inductancia mutua (M_{12} y M_{21}): Se define como la integral de volumen de la densidad de flujo magnético creada por la intensidad del primario por el producto escalar de la intensidad de campo magnético creada por la intensidad del secundario dividido por el producto de intensidades. De igual manera se define la del secundario pero referida a este.

$$M_{12} = \int_V \frac{\vec{B}_1 \cdot \vec{H}_2}{i_2 i_1} dV = M_{21} = \int_V \frac{\vec{B}_2 \cdot \vec{H}_1}{i_2 i_1} dV \quad (3.11)$$

Inductancia de dispersión (L_{1l} , L_{2l} y L_{cc}): La inductancia de dispersión se define como la autoinductancia debida al flujo de dispersión en un transformador. Esta se calcula aplicando la siguiente ecuación:

$$L_{1l} = L_{11} - \frac{N_1}{N_2} M_{21} ; L_{2l} = L_{22} - \frac{N_2}{N_1} M_{12} \quad (3.12)$$

Según estas definiciones y despreciando la resistividad de los conductores y la del hierro, el circuito quedaría así:



Para simplificar el problema, habitualmente la inductancia de dispersión se pone como la suma de la de dispersión debida al primario más la de dispersión debida al secundario referida al primario o viceversa:

$$L_{cc} = L_{1l} + rt^2 L_{2l} \quad (3.13)$$

Siendo rt la relación de transformación de nuestro transformador.

Una vez definidas todas las magnitudes y ecuaciones de nuestro problema, nos disponemos a plantear nuestra ecuación diferencial principal. En este caso, nuestra incógnita principal va a ser el potencial vector magnético (\vec{A}) y, a partir de éste, calcularemos las inductancias de nuestro transformador.

Sustituyendo (3.4) en (3.7) obtenemos que:

$$\nabla \times \nabla \times \vec{A} = \mu_0 \vec{j} \quad (3.14)$$

Que en coordenadas cartesianas y teniendo en cuenta la ecuación (3.4) se puede expresar como:

$$\nabla^2 \vec{A} = -\mu_0 \vec{j} \quad (3.15)$$

Esta es la ecuación diferencial del potencial magnetostático. Si descomponemos las magnitudes vectoriales en sus componentes cartesianas obtenemos que:

$$\begin{aligned} \nabla^2 A_x &= -\mu_0 j_x \\ \nabla^2 A_y &= -\mu_0 j_y \\ \nabla^2 A_z &= -\mu_0 j_z \end{aligned} \quad (3.15)$$

4. Análisis del problema en FreeFem++.

Para que FreeFem++ pueda resolver la ecuación diferencial debemos poner ésta en su forma variacional, por lo que necesitaremos poner nuestra ecuación diferencial en forma bilineal $b(u, v)$ mas una lineal $l(j, v)$ (donde u, v y j son funciones $\mathbb{R}^n \rightarrow \mathbb{R}$). Además, para resolver el problema necesitaremos una condición de contorno.

Debido a la geometría de nuestro problema, éste nos permitirá resolver la ecuación diferencial en x, z e y de forma separada. Siendo este último, en nuestro caso particular, igual a cero. Es decir:

$$\begin{aligned}\nabla^2 A_x &= -\mu_m j_x \\ A_y &= 0 ; j_y = 0 \\ \nabla^2 A_z &= -\mu_m j_z\end{aligned}\quad (3.16)$$

Donde μ_m hace referencia a la permeabilidad magnética de nuestro volumen control.

En referencia a las condiciones de contorno de nuestro problema. El potencial vectorial disminuye conforme nos alejamos y en los límites de nuestro volumen de control éste será igual a cero. Esto se debe a que el sumatorio total de las intensidades de nuestro circuito es igual a cero, lo que provoca que las líneas de potencial sigan trayectorias cerradas. Por lo que en los límites del volumen control se cumple que:

$$\vec{A} = 0 \text{ En los límites del volumen control.}$$

Una vez establecida la condición de contorno, transformamos las ecuaciones (3.16) a forma variacional:

$$\begin{aligned}\nabla \cdot (\nabla \vec{A}_x) &= -\mu_m \vec{j}_x \\ \nabla \cdot (\nabla \vec{A}_z) &= -\mu_m \vec{j}_z\end{aligned}\quad (3.17)$$

Multiplicamos por la función arbitraria v , que cumple las condiciones de contorno, y realizamos la integral en todo el volumen control Ω .

$$\int_{\Omega} \nabla \cdot \left(\frac{1}{-\mu_m} \nabla u \right) v \, d\Omega = \int_{\Omega} \vec{j}_z v \, d\Omega \quad (3.18)$$

Integramos la primera componente por partes.

$$\left[-\frac{1}{\mu_m} v \nabla u \right]_{\Omega} + \int_{\Omega} \frac{1}{\mu_m} \nabla u \nabla v \, d\Omega = \int_{\Omega} \vec{j}_z v \, d\Omega \quad (3.19)$$

Y como v cumple las condiciones de contorno en los límites del volumen control, éste será igual a cero, $[v]_{\Omega} = 0$.

$$\int_{\Omega} \frac{1}{\mu_m} \nabla u \nabla v \, d\Omega = \int_{\Omega} \vec{j}_z v \, d\Omega \quad (3.20)$$

La ecuación (3.20) es la forma variacional de nuestra ecuación diferencial principal.

Aunque la permeabilidad magnética no es una constante, en nuestro problema ésta permanece prácticamente uniforme, por lo que tomaremos un valor constante.

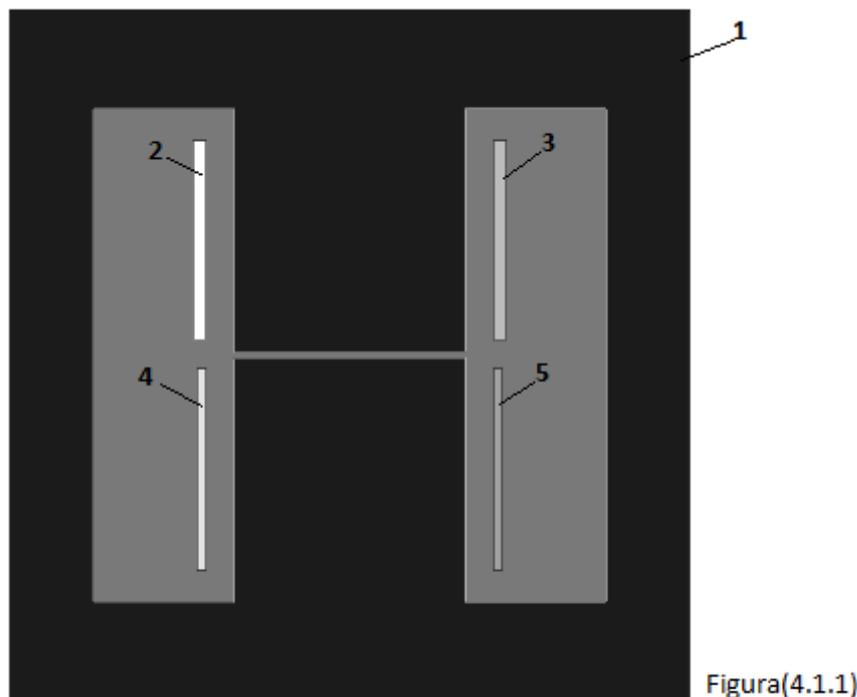
Una vez hecho esto, ya podemos proceder a resolver nuestro problema en FreeFem++.

4.1 Transformador en 2D.

Una vez explicado qué es FreeFem++ y las ecuaciones que vamos a utilizar, nos disponemos a plantear nuestro problema principal en 2D paso a paso.

4.1.1 Geometría y mallado del problema.

Para la resolución del problema cogemos el plano más significativo, que es el siguiente:



El dibujo de la Figura (4.1.1) es una representación de la geometría de nuestro transformador, donde la marca 1 hace referencia al núcleo y las marcas 2, 3, 4 y 5 a las bobinas. Como el problema es en 2D, esta representación es una sección transversal del transformador, donde 2 y 3 son una sección de la bobina del primario y donde 4 y 5 lo son de la bobina del secundario.

Las dimensiones del núcleo (1) se describen en el *Anexo I*. A diferencia del núcleo descrito en dicho anexo, la sección central de nuestro transformador es rectangular y no circular (tal y como lo especificaron los datos del transformador real), siendo el diámetro del descrito en el *Anexo I* similar al lado de nuestro transformador.

La bobina del primario (2 y 3) está formada por 3 capas de 18 vueltas cada una de ellas, lo que hace un total de 54 vueltas. Las dimensiones del hilo están indicadas en el *Anexo II*. Además, hay que tener en cuenta que la bobina del primario tiene una separación de 2mm con la parte superior del núcleo y otra separación de 2mm con la parte central del núcleo.

La bobina del secundario (4 y 5) está formada por 2 capas de 18 vueltas cada una de ellas, lo que hace un total de 36 vueltas. Las dimensiones del hilo son las mismas que las de la bobina del primario. Además, la bobina del secundario tiene una separación de 2mm con la parte inferior del núcleo y otra separación de 2mm con la parte central del núcleo.

Una vez definida la geometría del problema tendremos que realizar el mallado.

Para delimitar el contorno del problema definiremos un cuadrado de 60mm x 60mm, en el cuál se encontrará nuestro transformador. Quedando el mallado final de la siguiente manera:

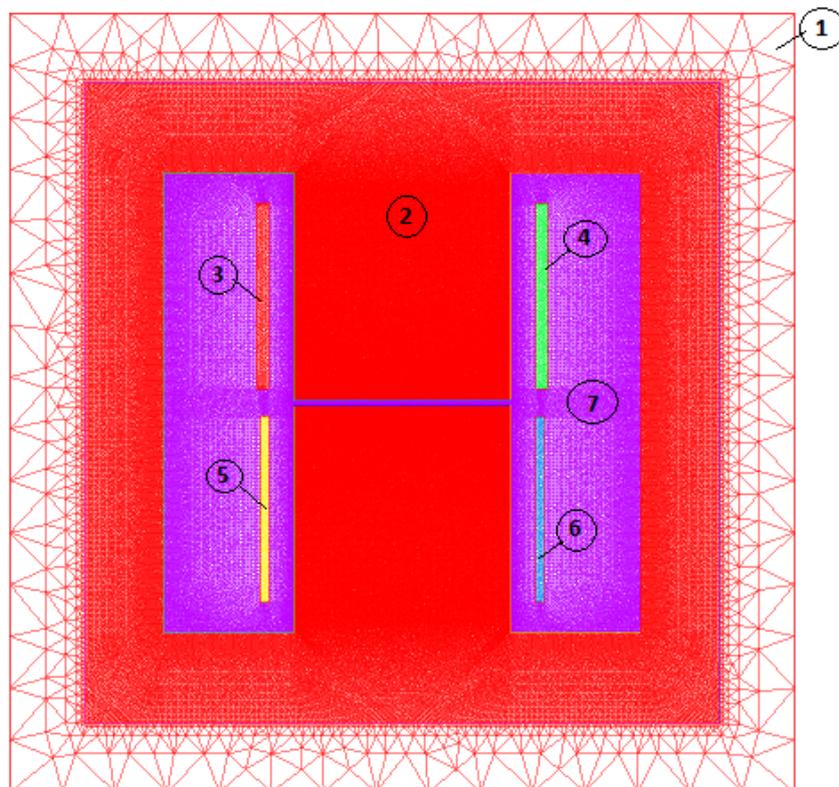
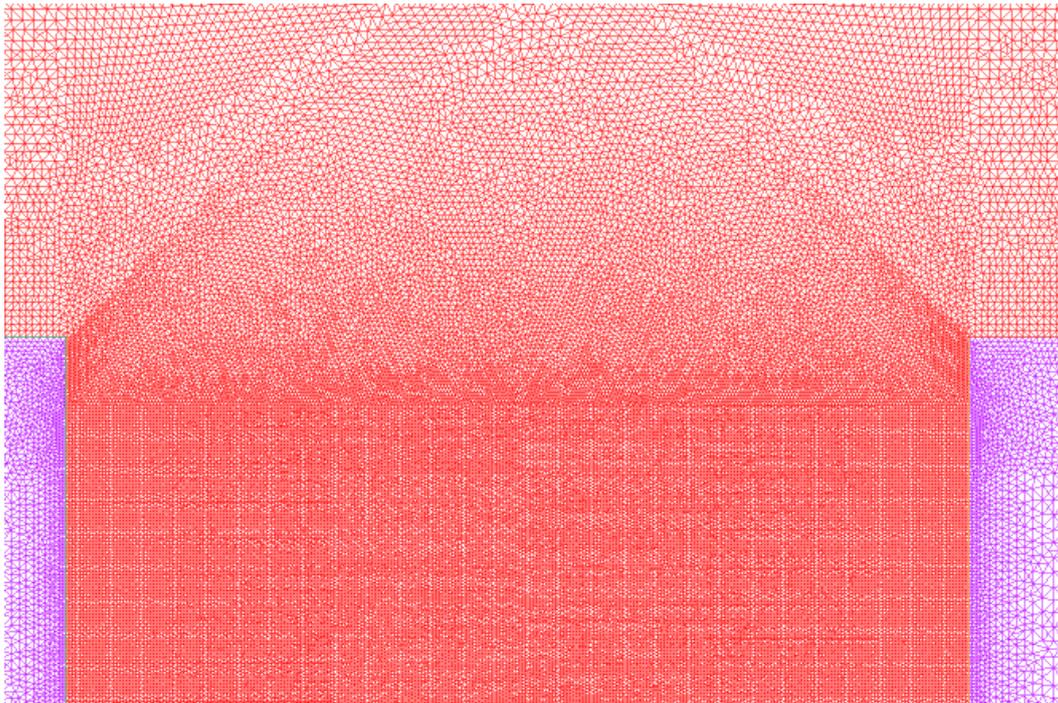


Figura (4.1.2)

La Figura (4.1.2) muestra la geometría, las regiones y el mallado final del problema. En esta figura se pueden distinguir 7 regiones.

La región 1 representa el aire exterior, esta región es la que tiene el mallado más grueso ya que es la que tiene menos peso en la solución final.

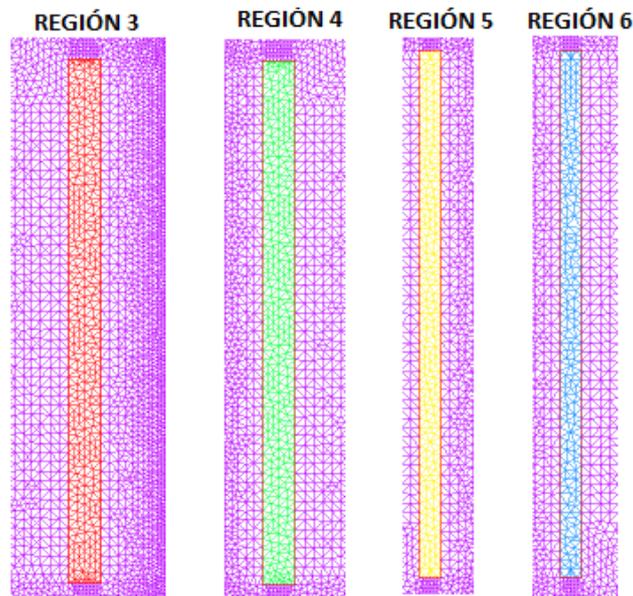
La región 2 representa el núcleo del transformador, siendo esta región la segunda más mallada debido a que la mayor parte de los parámetros que intervienen en la solución final se concentran en esta zona, dentro de esta región la parte central del núcleo tiene un mallado más fino debido a la importancia de la región central. Para mejor visualización de esta región mostrare una ampliación de la misma:



Figura(4.1.3)

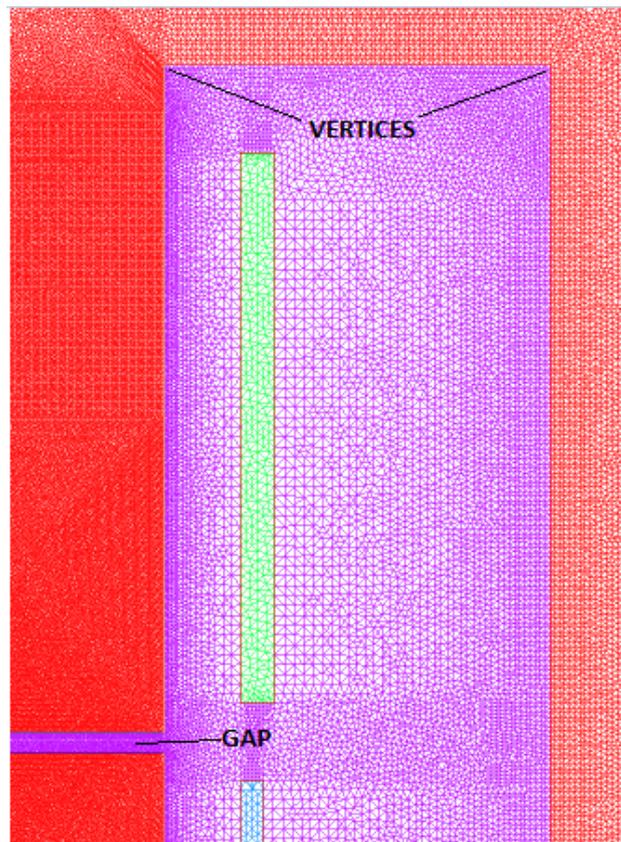
En la Figura (4.1.3) se puede observar como la parte central del núcleo esta mucho más mallada que el resto de éste (la parte central del núcleo es la figura roja inferior, y la parte exterior del núcleo es la figura roja superior).

Las regiones 3, 4, 5 y 6 de la Figura (4.1.2) representan las bobinas. Éstas tienen un mallado normal, ya que su importancia recae en la definición de la densidad de corriente y no en la solución de la ecuación diferencial (el potencial vector).



Figura(4.1.4)

Por último, tenemos la región 7 de la Figura (4.1.2), que representa el aire interior del núcleo. En esta región, las zonas más importantes son los vértices que forman con el núcleo y el entrehierro, siendo el entrehierro la zona más importante del problema y la más mallada. En la siguiente imagen se mostrará el entrehierro y vértices del problema ampliados.



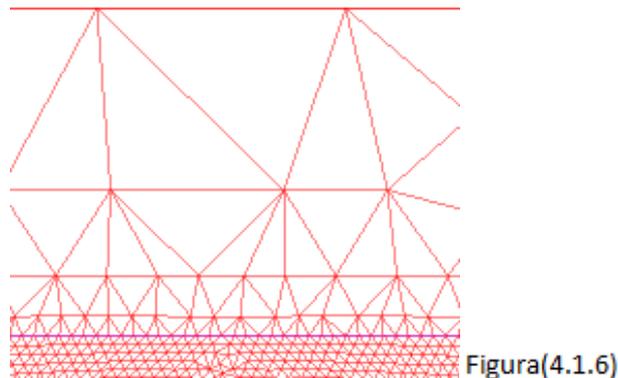
Figura(4.1.5)

Para el refinamiento del mallado se ha multiplicado una constante que hace referencia a cada región por el tamaño, en milímetros, de cada arista. Por lo que el refinamiento del mallado va en proporción con el tamaño. En la siguiente tabla se muestran las constantes que refinan cada región.

Tabla 1. Constantes de refinamiento en el caso 2D en FreeFem++.

Nombre del parámetro	Magnitud del parámetro	Regiones afectadas
nAire	0.25	Aire exterior del problema
nHierro	8	Núcleo de hierro excepto parte central
nAireIn	5	Aire interior del núcleo y bobinas
nAirefino	10	Gap y parte central del núcleo

Por ejemplo, el lado del núcleo del transformador mide 49,4mm, por lo que el número de nodos es: $aristaLN \times nHierro = 49,4 \times 8 = 395$ (tomaremos el resultado sin decimales porque el número de nodos es un entero), que es el número de nodos que tendrá ese lado del transformador. Una vez definidas dos aristas correlativas, el número de nodos intermedios se irá adaptando para casar ambos lados. El ejemplo más claro de esto es el del núcleo con la parte exterior de aire:



En la Figura(4.1.6) observamos perfectamente como el mallado se va adecuando en función del refinamiento que le hemos asignado a cada zona.

Para profundizar en la definición del mallado del problema revisar *anexo III*.

4.1.2 Definición de parámetros.

Una vez definida la geometría y el mallado del problema, el siguiente paso es definir los parámetros que intervienen en nuestro problema:

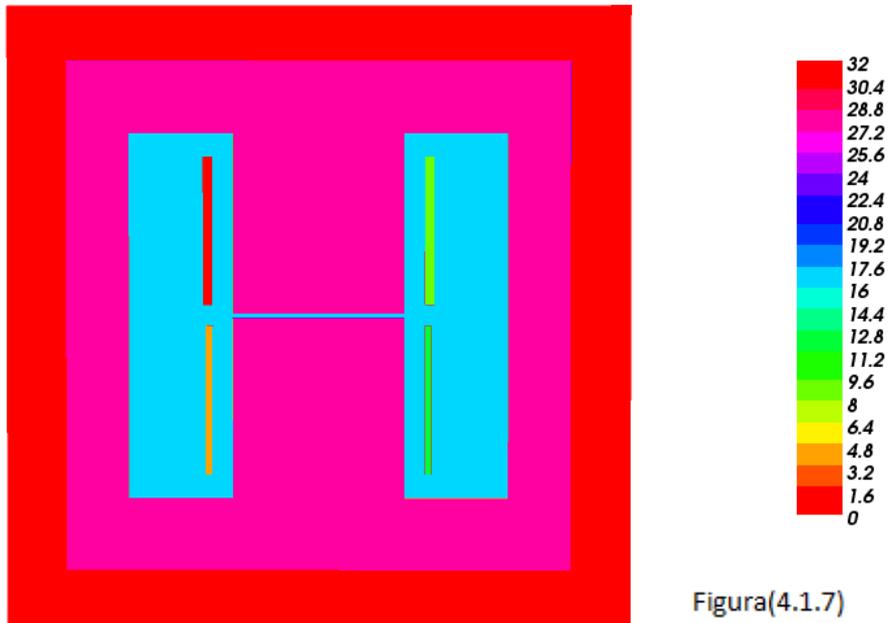
Permeabilidad magnética (μ) y densidad de corriente (\vec{j}).

Debido a que ambas magnitudes se definen en FreeFem++ de forma similar explicaremos el proceso de forma simultánea.

Para consultar los detalles de cómo se ha implementado, consultar el código en el *anexo III*.

Para definir estos dos parámetros tenemos dos formas de hacerlo.

1) Definir las regiones del problema asignándoles una marca.



Figura(4.1.7)

Como podemos observar en la Figura(4.1.7) cada región tiene un valor entero asignado.

Una vez hecho esto, le asignamos a cada región su valor correspondiente de permeabilidad magnética y densidad de corriente, teniendo en cuenta, en el caso de la densidad de corriente, el sentido de circulación de la intensidad.

2) Definir estos parámetros directamente por zonas. Es decir, asignamos el valor de permeabilidad magnética y densidad de corriente directamente a la geometría.

Como esta ha sido la forma en la que la hemos definido, para definirlo de la misma manera que en el caso en 3D.

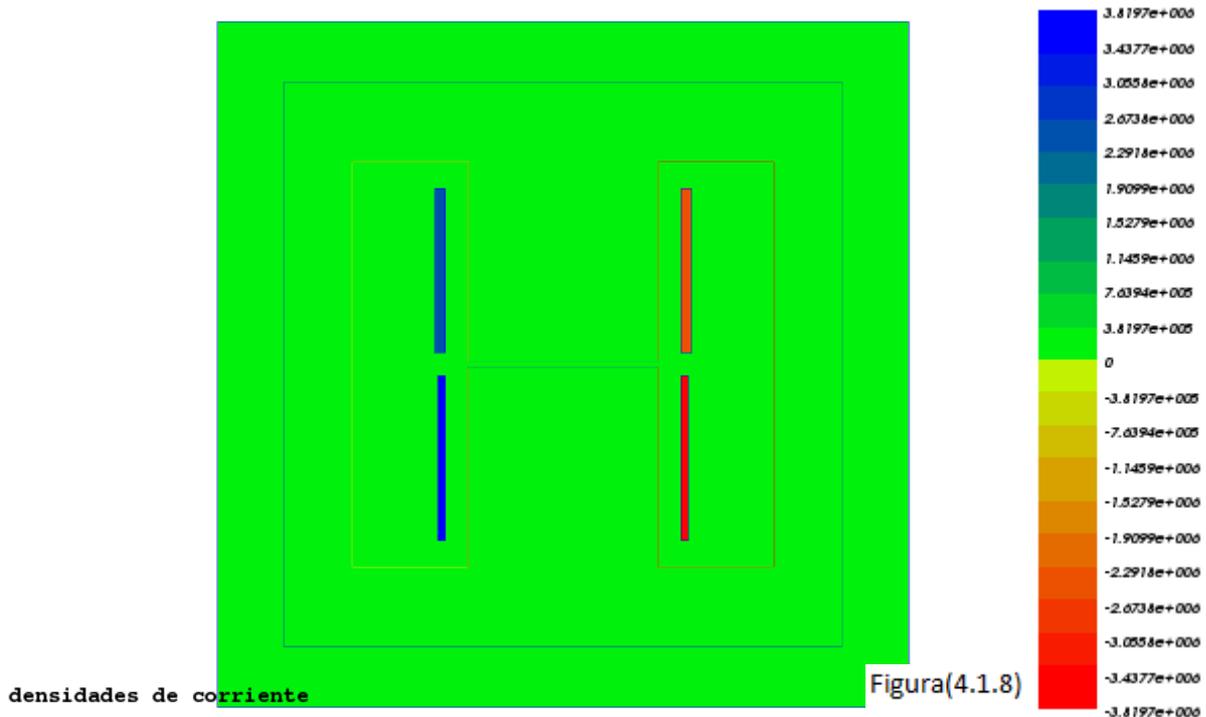
Como nuestro caso es en 2D la densidad de corriente solo tendrá componente z siendo el cálculo de esta:

$$DensidadCorriente\ Bobina = \frac{Intensidad}{Seccion\ del\ conductor}$$

Por lo que esta quedara definida de la siguiente forma:

$$\vec{j}_z \begin{cases} 0 & \text{en el hierro y el aire} \\ \frac{I_1}{Sec1} = 2546480 \frac{A}{m^2} & \text{en la bobina del primario} \\ \frac{I_2}{Sec2} = 3819720 \frac{A}{m^2} & \text{en la bobina del secundario} \end{cases}$$

Siendo los sentidos de las densidades de corrientes positivos para las dos bobinas de la izquierda y negativos para las dos de la derecha. Esto se debe a que hemos dado un sentido a la intensidad en contra de las agujas del reloj. Quedando definida la densidad de corriente de la siguiente forma:



La Figura (4.1.8) muestra la representación de las densidades de corriente tal y como hemos explicado anteriormente.

En el caso de la permeabilidad magnética, como es un escalar, solo habrá que asignar a cada material su valor de permeabilidad.

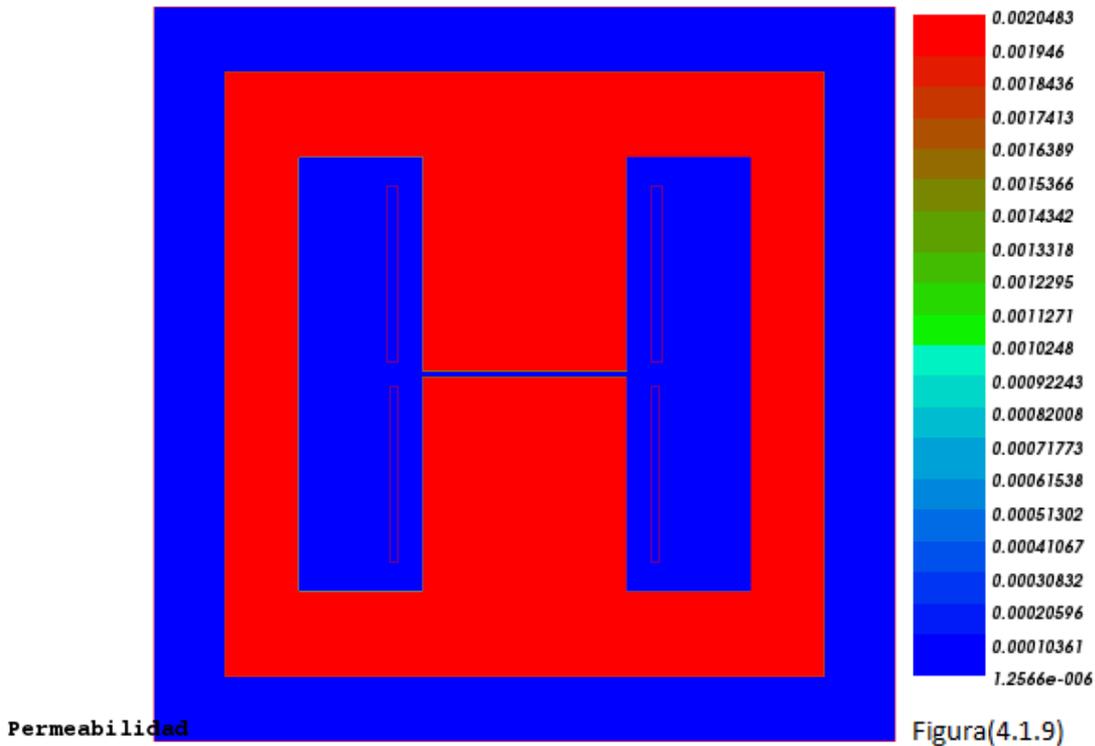
Los valores característicos de permeabilidad magnética de cada material son:

$$\mu_m \begin{cases} \mu_0 = 4\pi \cdot 10^{-7} \text{ para el aire y las bobinas de litz} \\ \mu_{Fe} = 1630 * \mu_0 \text{ para el acero del nucleo N87} \end{cases}$$

El aire y el hilo de litz son materiales paramagnéticos, es decir no son magnéticos por lo que la su permeabilidad magnética es aproximadamente la del vacío.

Para el acero N87 se han utilizado los datos del fabricante (ver *anexo IV*).

A continuación se incluye una figura donde se pueden ver las permeabilidades magnéticas asignadas:



La Figura(4.1.9) muestra la permeabilidad magnética de las distintas regiones de nuestro problema.

Una vez definida la geometría, el mallado y los parámetros de nuestro problema ya podemos disponernos a resolver la ecuación diferencial.

4.1.3 Resolución del problema y resultados.

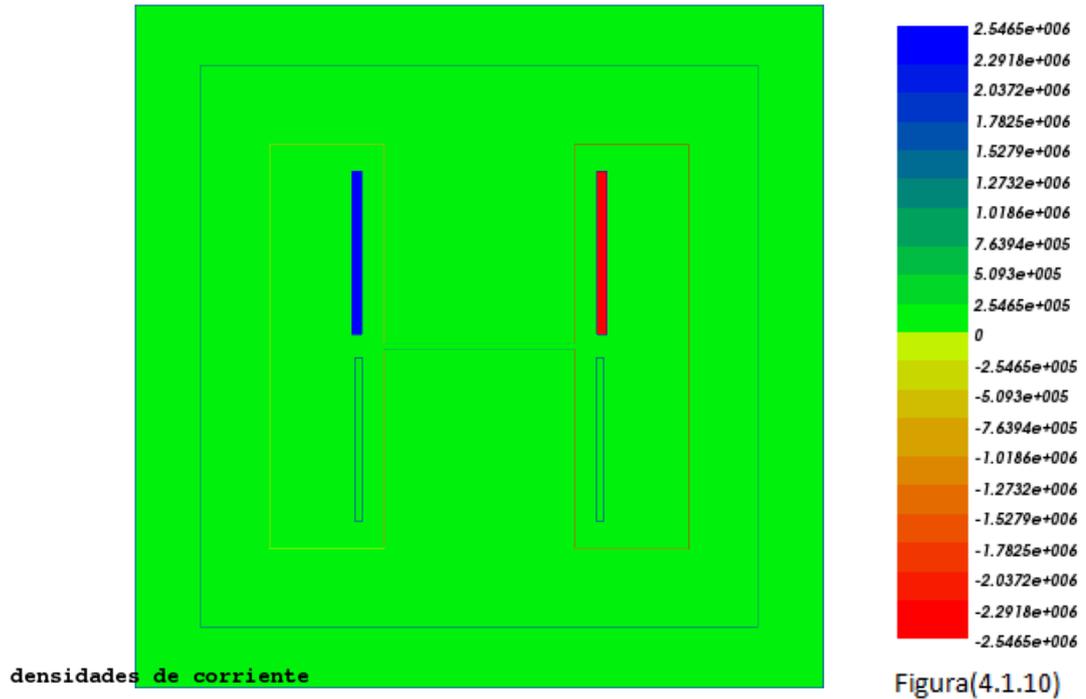
Para obtener el potencial magnetostático, en este caso tenemos que plantear nuestra ecuación diferencial en dos dimensiones, ya que la densidad de corriente más relevante está aplicada en el plano z y despreciamos la aplicada en el plano x. Por lo que nuestra ecuación diferencial será:

$$\begin{cases} \int_S \frac{1}{\mu_m} \nabla_{2D}(A_z) \nabla_{2D}(v) dS - \int_S j_z v dS = 0 \\ \text{Con la condicion de contorno } A_z|_S = 0 \end{cases} \quad (4.1)$$

Siendo $\nabla_{2D} = [dx, dy]$ y dS el diferencial de superficie.

Una vez llegado a este punto aclarar que lo que queremos calcular es la inductancia de nuestro transformador. Para calcular estas, primero hay que calcular las inductancias propias del primario y secundario (L_{11} y L_{22}).

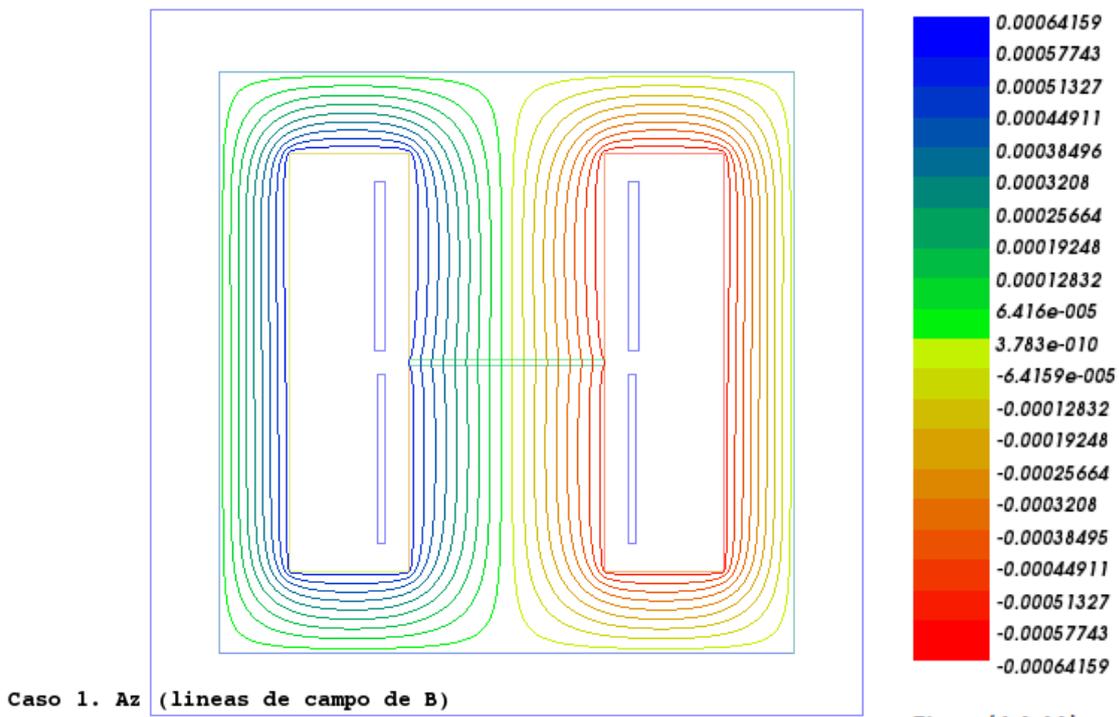
Caso 1. Calculo de L_{11} : Para el cálculo de L_{11} tendremos que dejar $I_2 = 0$. Esto significa que la densidad de corriente en el secundario es cero.



La Figura (4.1.10) muestra la distribución de la densidad de corriente del caso 1.

Una vez definida la nueva densidad de corriente ya podemos calcular el potencial magnetostático aplicando la ecuación (4.1) con la geometría y mallado mencionados anteriormente.

La representación del potencial magnetostático para el caso 1 puede observarse en la Figura (4.1.11).



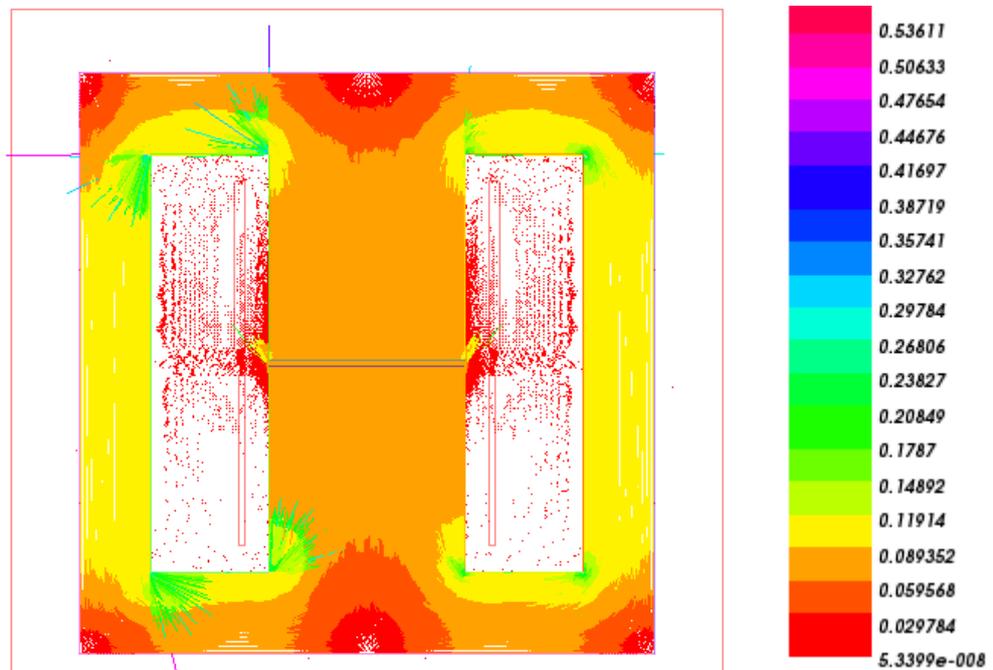
Una vez calculado el potencial magnetostático podemos calcular el campo magnético aplicando la siguiente ecuación:

$$\vec{B} = \vec{\nabla} \times \vec{A} \begin{cases} B_x = \frac{\partial A_z}{\partial y} - \frac{\partial A_y}{\partial z} \\ B_y = \frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x} \\ B_z = \frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y} \end{cases} \quad (4.2)$$

Como nuestro caso es bidimensional, teniendo solo componente en z, la ecuación quedará:

$$\begin{aligned} B_x &= \frac{\partial A_z}{\partial y} \\ B_y &= -\frac{\partial A_z}{\partial x} \\ B_z &= 0 \end{aligned} \quad (4.3)$$

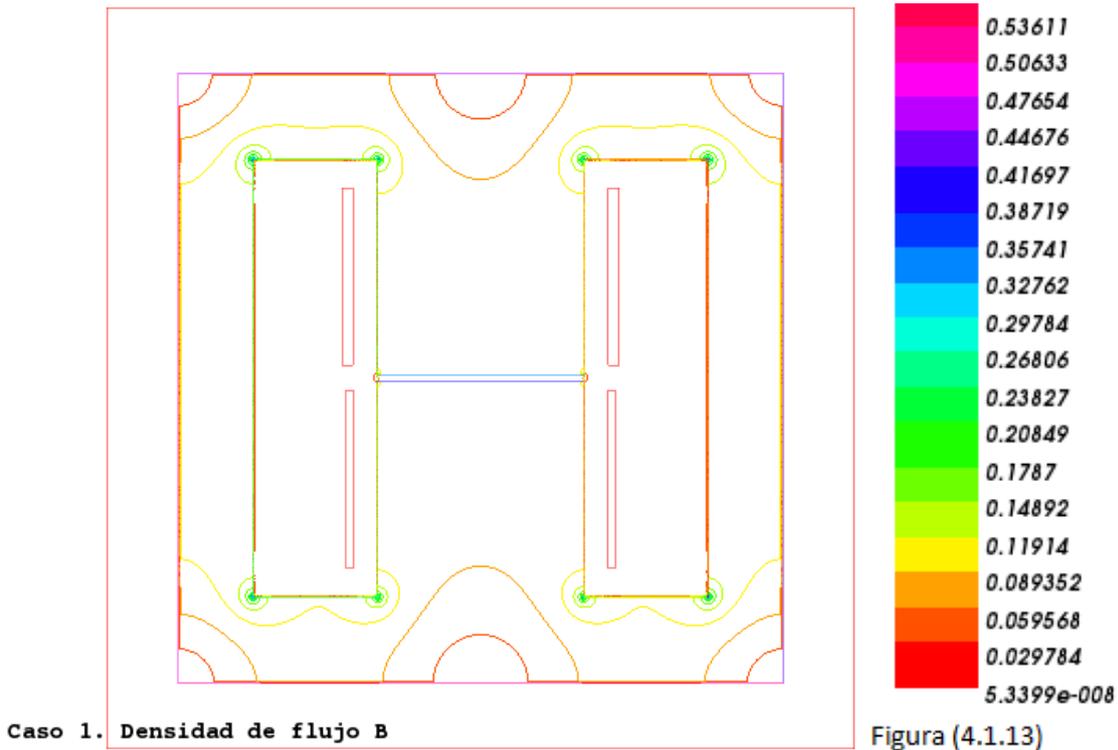
Siendo la representación del campo magnético para el caso 1 el mostrado en la Figura (4.1.12).



Caso 1. Vector de densidad de flujo B

Figura(4.1.12)

La Figura (4.1.13) representa el modulo del campo magnético para el caso 1.



Una vez calculado el campo magnético podemos calcular la intensidad de campo magnético aplicando la ecuación (3.9).

$$\begin{aligned} H_x &= \frac{B_x}{\mu_m} \\ H_y &= \frac{B_y}{\mu_m} \\ H_z &= 0 \end{aligned} \quad (4.4)$$

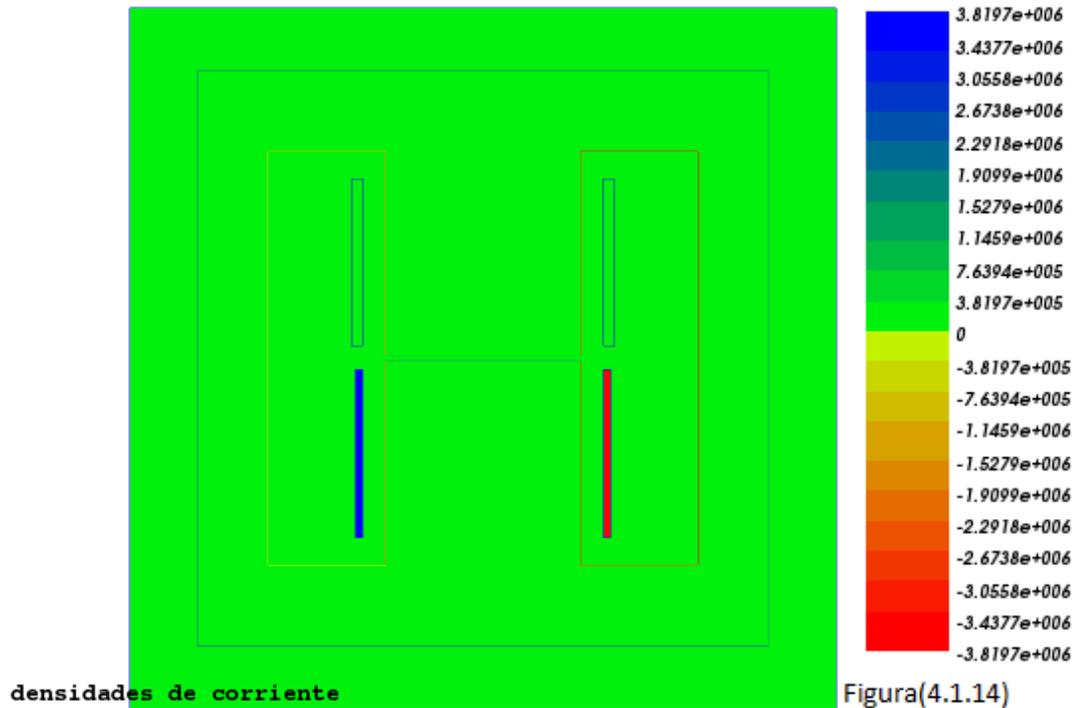
Una vez calculados estos parámetros ya podemos calcular la inductancia propia del primario aplicando la ecuación (3.10):

$$L_{11} = e * \int_S \frac{B_1 \cdot H_1}{I_1^2} dS = e * \int_S \frac{B_{1x} \cdot H_{1x} + B_{1y} \cdot H_{1y}}{I_1^2} dS$$

Siendo e el espesor del transformador e I_1 la intensidad del primario (sus valor se especifica en el *anexo III*).

El valor obtenido para la inductancia propia del primario es $L_{11} = 1,89235 \text{ mH}$.

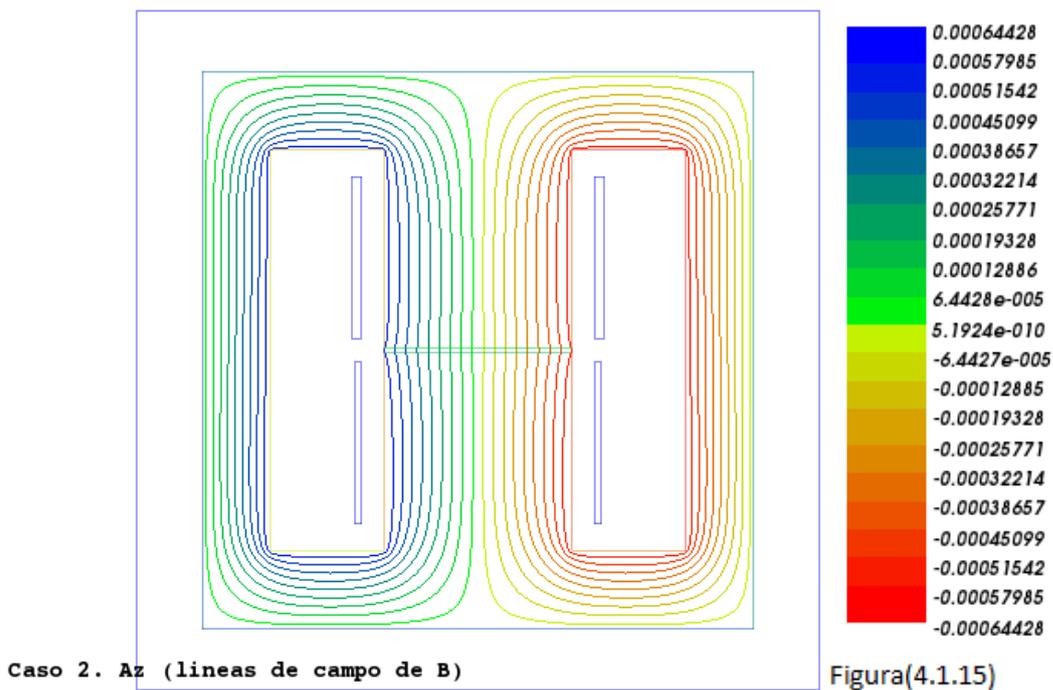
Caso 2. Calculo de L_{22} : Para el cálculo de L_{22} tendremos que dejar $I_1 = 0$. Esto significa que la densidad de corriente en el primario es cero.



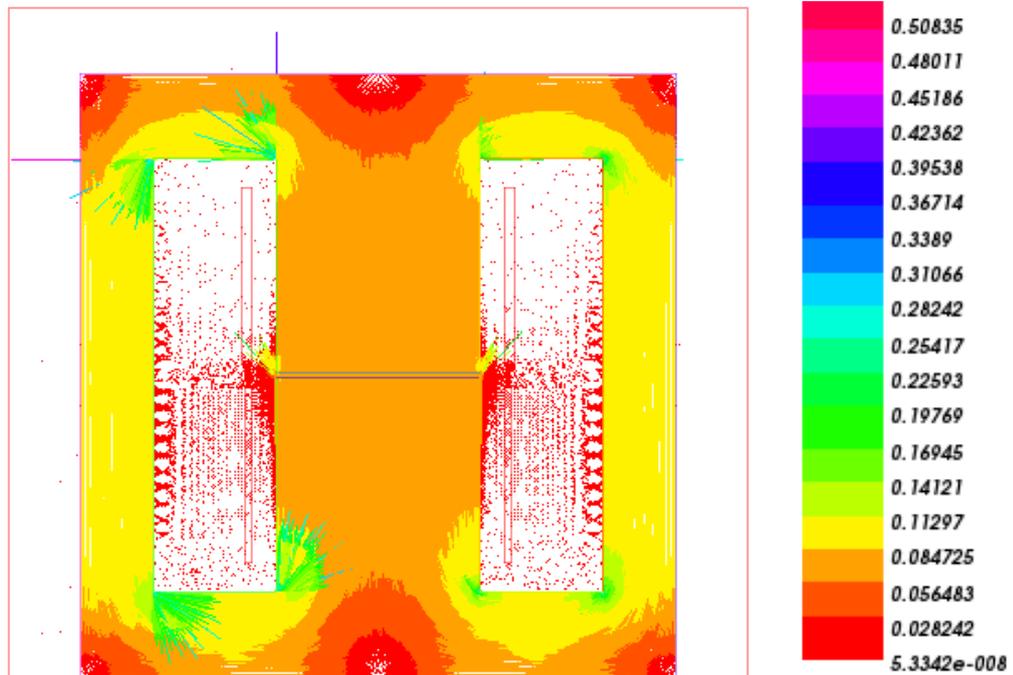
La Figura (4.1.14) muestra la distribución de la densidad de corriente del caso 2.

Una vez definida la nueva densidad de corriente ya podemos calcular el potencial magnetostático aplicando la ecuación (4.1) como en el caso 1.

La representación del potencial magnetostático para el caso 2 puede observarse en la Figura (4.1.15).



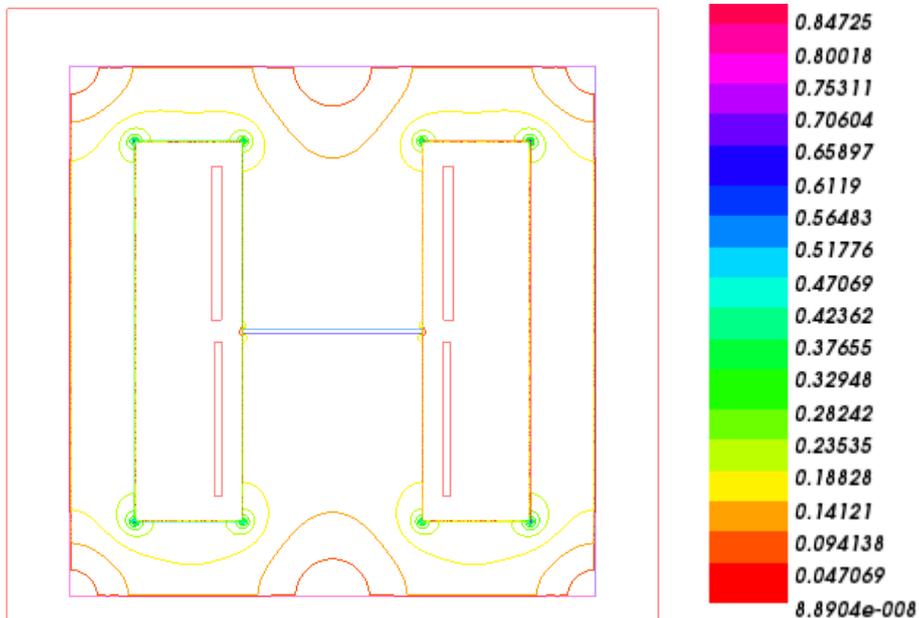
Ahora aplicamos la ecuación (4.3) y calculamos el campo magnético para el caso 2. En la Figura(4.1.16) se muestra su representación grafica .



Caso 2. Vector de densidad de flujo B

Figura(4.1.16):

La Figura (4.1.17) representa el modulo del campo magnético para el caso 2.



Caso 2. Densidad de flujo B

Figura (4.1.17)

Una vez calculado el campo magnético podemos calcular la intensidad de campo magnético aplicando la ecuación (4.4)

Y ahora ya podemos calcular la inductancia propia del secundario aplicando la ecuación (3.10).

$$L_{22} = e * \int_S \frac{\vec{B}_2 \cdot \vec{H}_2}{I_2^2} dS = e * \int_S \frac{B_{2x} \cdot H_{2x} + B_{2y} \cdot H_{2y}}{I_2^2} dS$$

Siendo e el espesor del transformador e I_2 la intensidad del secundario (sus valor se especifica en el *anexo III*).

El valor el valor calculado para la inductancia propia del secundario es $L_{22} = 0,847872 \text{ mH}$.

Como el problema es lineal la inductancia mutua se podrá calcular como combinación lineal del caso 1 y 2. Aplicando la ecuación (3.11) tenemos que:

$$M_{12} = M_{21} = e * \int_S \frac{\vec{B}_2 \cdot \vec{H}_1}{I_2 I_1} dS = e * \int_S \frac{B_{2x} \cdot H_{1x} + B_{2y} \cdot H_{1y}}{I_2 I_1} dS$$

Aplicando esta ecuación el valor obtenido de la inductancia mutua es $M_{12} = 1,21574 \text{ mH}$

Y por ultimo calculamos la impedancia de dispersión aplicando la ecuación (3.12).

$$L_{1l} = L_{11} - \frac{N_1}{N_2} M_{21} ; L_{2l} = L_{22} - \frac{N_2}{N_1} M_{12}$$

Siendo N_1 y N_2 el numero de vueltas del primario y del secundario respectivamente (para ver los valore de N_1 y N_2 consultar *anexo III*). Las impedancias de dispersión serán:

$$L_{1l} = 68,7434 \mu\text{H} ; L_{2l} = 37,3791 \mu\text{H}$$

Ahora sumamos ambas impedancias referidas al primario aplicando la ecuación (3.12).

$$L_{cc} = L_{1l} + rt^2 L_{2l}$$

Siendo el valor de la impedancia de dispersión total $L_{cc} = 152,846 \mu\text{H}$

El análisis de los resultados se realizara posteriormente en el apartado de comparación de resultados tomando como referencia los resultados del transformador real.

4.1.4 Código del programa.

En él *anexo III* se encuentra todo el código para la resolución de este problema.

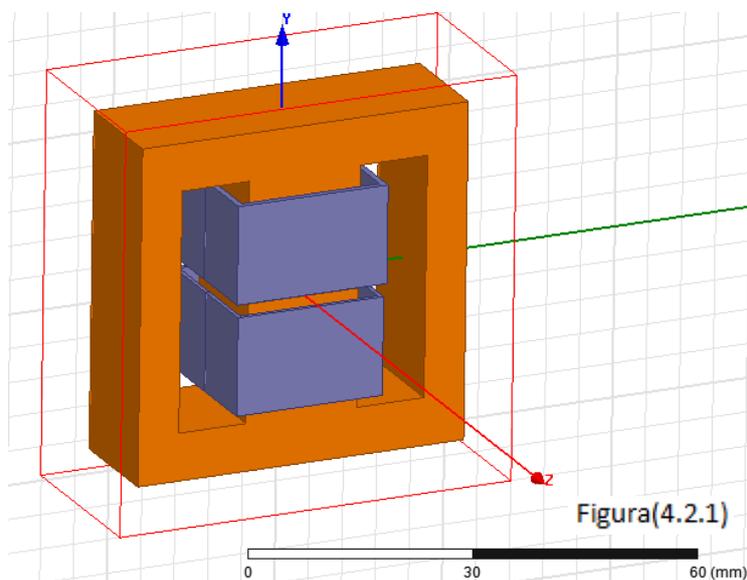
4.2 Transformador en 3D.

En este apartado nos disponemos a realizar el problema anterior pero en 3D. Este problema tiene unas peculiaridades que el caso en 2D no se encontraban.

4.2.1 Geometría y límites del problema.

La geometría del transformador será la geometría descrita en el *anexo I*, y al igual que en el caso 2D la parte central del transformador será rectangular en vez de circular, tomando como longitud del lado del rectángulo, el diámetro descrito en el *anexo I*.

Como hay que tener en cuenta el aire del transformador, este estará situado en el interior de un volumen de aire de 60mmx60mmx36mm como muestra la siguiente figura.

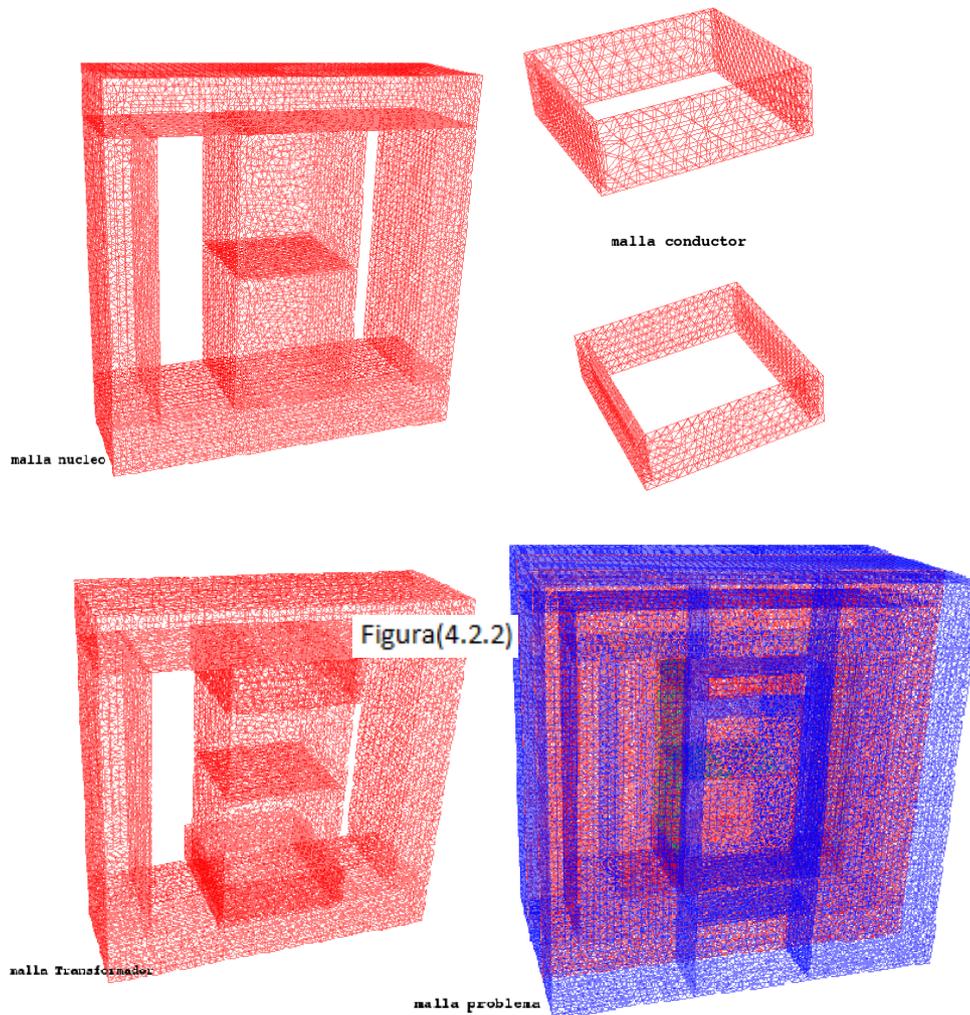


En el caso 3D nos encontramos ciertos problemas al describir la geometría y el mallado en FreeFem++. Estos problemas son los siguientes:

Opción 1. Describir la geometría por partes y luego unir todas las partes para formar la malla final.

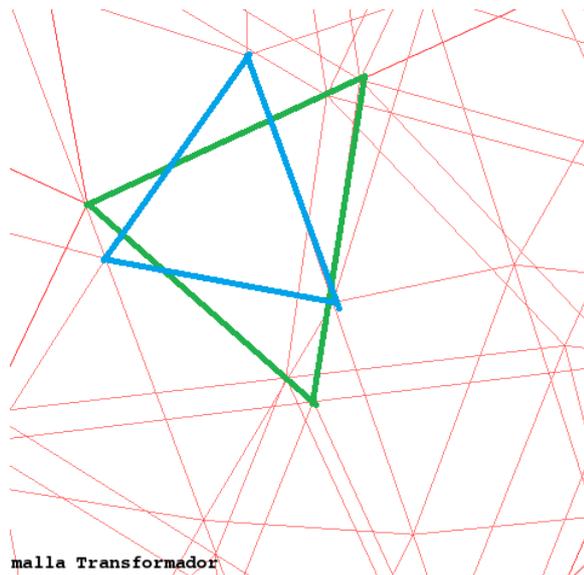
Esta opción sería la más óptima para realizar el mallado de la pieza ya que podemos darle el refinamiento que queramos a cada parte optimizando el tiempo de ejecución y la precisión de los resultados. Esto se debe a que podemos refinar mucho las zonas más cargadas y hacer un mallado más grueso para las zonas de menos interés para nuestro problema.

En la siguiente figura se muestra como quedaría el mallado.



La Figura (4.2.2) muestra el mallado realizado por partes y la unión de las mismas.

Pero esta forma de describir la geometría presenta un problema que nos impide utilizarla para la resolución del mismo. Esto se debe a que las mallas no casan entre ellas, es decir, los nodos de unas no coinciden con las que están en contacto con ellas.



Figura(4.2.3)

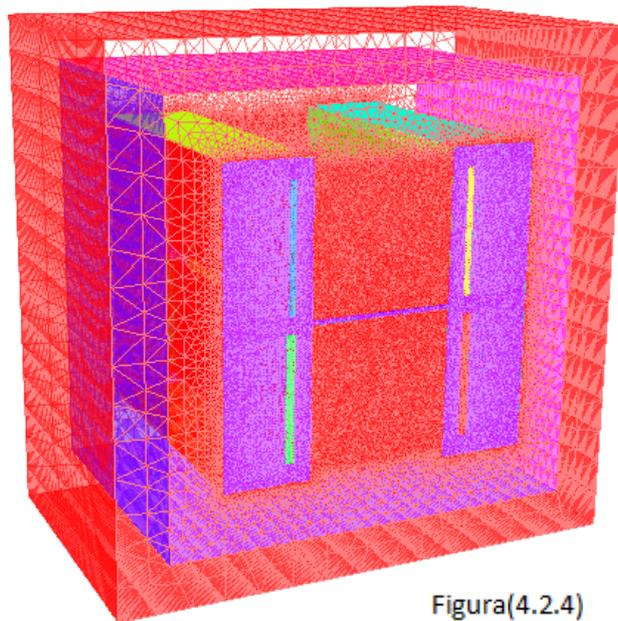
La Figura (4.2.3) muestra las mallas que están en contacto. Para que el mallado fuese uniforme este mallado debería coincidir. Para mejor visualización se han marcado de azul y verde los dos triángulos que deberían superponerse. Para mayor información el código de este mallado se encuentra en el *anexo V*.

Opción 2. Describir la geometría en 2D y se extruye para formar una figura en 3D.

Realizar la geometría de esta manera también nos permitiría optimizar el mallado de la pieza. No tanto como en el caso anterior, pero en cambio nos quitamos el problema de que las mallas casen.

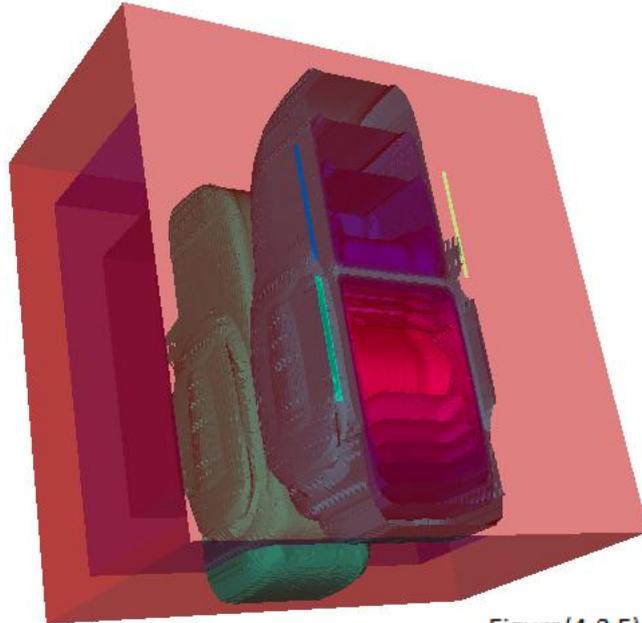
Esta opción no nos permite definir los parámetros por regiones ya que éstas no se corresponden exactamente con la geometría del mallado. Dichos parámetros tendrían que definirse por zonas como en el caso que hemos realizado en 2D.

El mallado quedaría como muestra la Figura (4.2.4):



Sin embargo, en esta opción, se nos presenta un problema que nos impide utilizarla para la resolución de nuestro problema. Esta opción nos impide ponerle una etiqueta a las superficies posterior y superior lo que desemboca en la imposibilidad de darle condiciones de contorno. Esto desemboca en que el potencial vectorial debido a la densidad de corriente en x no de cero en las superficies exteriores.

La Figura (4.2.5) muestra como daría el potencial vectorial en este caso:



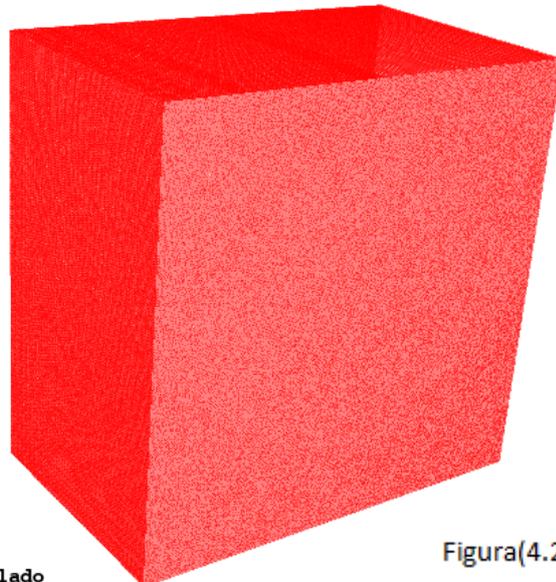
Caso 1. Ax (líneas de campo de B)

Figura(4.2.5)

Para mayor información el código de este mallado se encuentra en el *anexo VI*.

Opción 3. Describir un prisma rectangular con las dimensiones dichas al inicio de este apartado. Debido a los problemas de las otras opciones se ha optado por utilizar esta para la resolución de nuestro problema. Esta opción no nos permite diferenciar zonas de mallado ni definir los parámetros por regiones pero en cambio no presenta problemas de compatibilidad entre mallas ni problemas de etiquetación de superficies.

Según esta opción el mallado quedaría como se muestra en la Figura (4.2.5).



Mallado

Figura(4.2.5)

Para más información del mallado revisar *anexo VII*.

4.2.2 Definición de parámetros.

Debido a como se ha descrito la geometría y el mallado en el problema en 3D, no se nos permite diferenciar regiones directamente en la geometría para asignarles los valores de permeabilidad magnética o densidad de corriente.

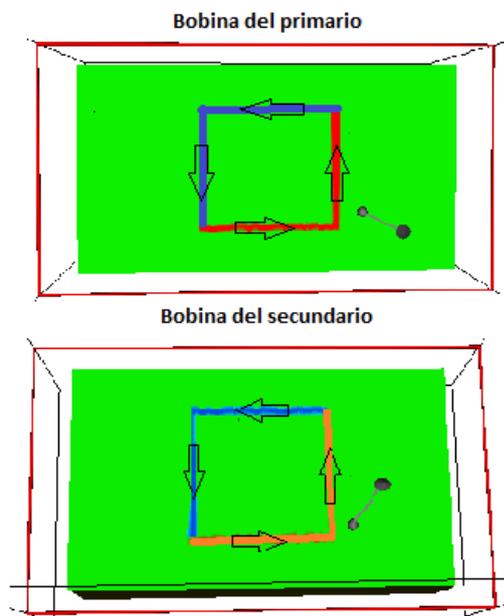
Por lo que tendremos que definir estos parámetros directamente por zonas como se hizo en el caso de 2D. Por problemas del programa al representar geometrías muy malladas las figuras que se mostraran en este apartado se han realizado con un mallado más grueso que el utilizado para resolver el problema.

Como en este caso nuestro problema es tridimensional, debido a que no vamos a despreciar ninguna componente de la densidad de corriente, tendremos dos componentes de la misma.

Siendo el módulo el mismo que el especificado en el caso 2D y teniendo en cuenta que el sentido de circulación de la corriente es en contra de las agujas del reloj, la densidad de corriente quedará definida de la siguiente forma:

$$j_z, j_x \begin{cases} 0 & \text{en el hierro y el aire} \\ \frac{I_1}{Sec1} = 2546480 \frac{A}{m^2} & \text{en la bobina del primario} \\ \frac{I_2}{Sec2} = 3819720 \frac{A}{m^2} & \text{en la bobina del secundario} \end{cases}$$

Quedando la representación de la densidad de corriente de la siguiente forma:



densidades de corriente

Figura(4.2.6)

La Figura (4.2.6) muestra la representación de las densidades de corriente tal y como hemos explicado anteriormente. Para más información sobre el código consultar *anexo VII*.

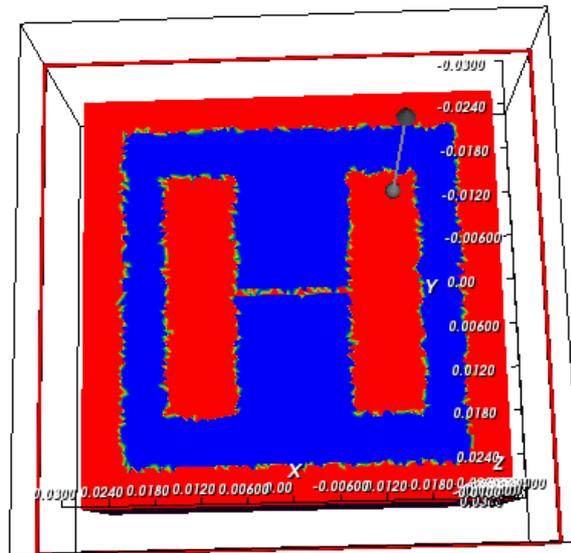
En el caso de la permeabilidad magnética tenemos que realizar lo mismo que en el caso en 2D pero teniendo en cuenta también la profundidad.

Los valores característicos de permeabilidad magnética de cada material son:

$$\mu_m \begin{cases} \mu_0 = 4\pi \cdot 10^{-7} \text{ para el aire y las bobinas de litz} \\ \mu_{Fe} = 1630 * \mu_0 \text{ para el acero del nucleo N87} \end{cases}$$

Para ver la permeabilidad magnética del acero N87 revisar *anexo IV*.

Quedando la permeabilidad magnética de nuestra geometría definida de la siguiente forma:



Permeabilidad

Figura(4.2.7)

La Figura (4.2.7) muestra la permeabilidad magnética de las distintas regiones de nuestro problema. Para más información sobre el código consultar *anexo VII*.

Una vez definida la geometría, el mallado y los parámetros de nuestro problema, ya podemos disponernos a resolver la ecuación diferencial.

4.2.3 Resolución del problema y resultados.

Para obtener el potencial magnetostático, en este caso hemos optado por plantear dos ecuaciones diferenciales, una para la componente z y otra para la componente x, por lo que nuestras ecuaciones diferenciales serán:

$$\begin{cases} \int_{\Omega} \frac{1}{\mu_m} \nabla_z(A_z) \nabla_z(v) d\Omega - \int_{\Omega} j_z v d\Omega = 0 \\ \int_{\Omega} \frac{1}{\mu_m} \nabla_x(A_x) \nabla_x(v) d\Omega - \int_{\Omega} j_x v d\Omega = 0 \end{cases} \quad (4.5)$$

Con la condicion de contorno $A_z, A_x|_{\Omega} = 0$

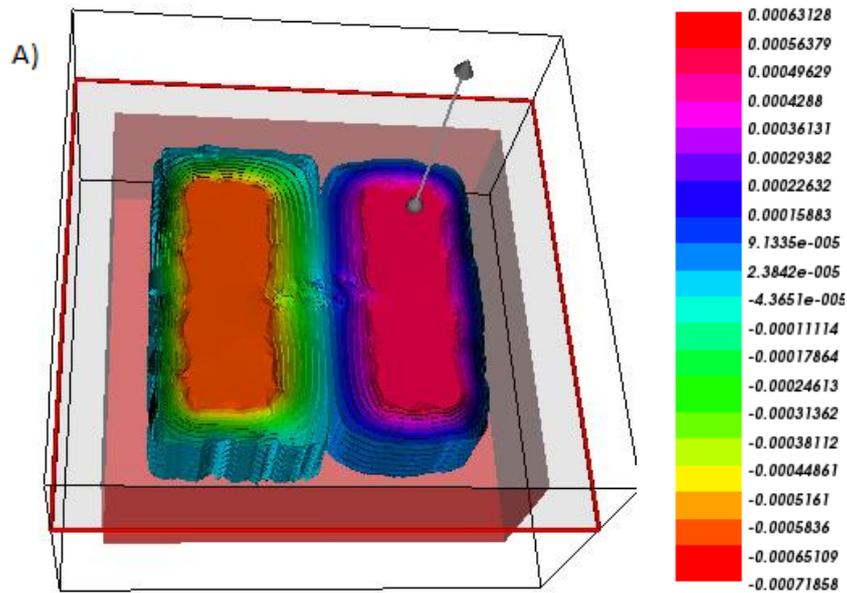
Siendo $\nabla_z = [dx, dy]$, $\nabla_x = [dy, dz]$ y $d\Omega$ el diferencial de volumen.

Y como lo que queremos calcular son las inductancias de nuestro transformador. Primero calculamos las inductancias propias del primario y secundario (L_{11} y L_{22}).

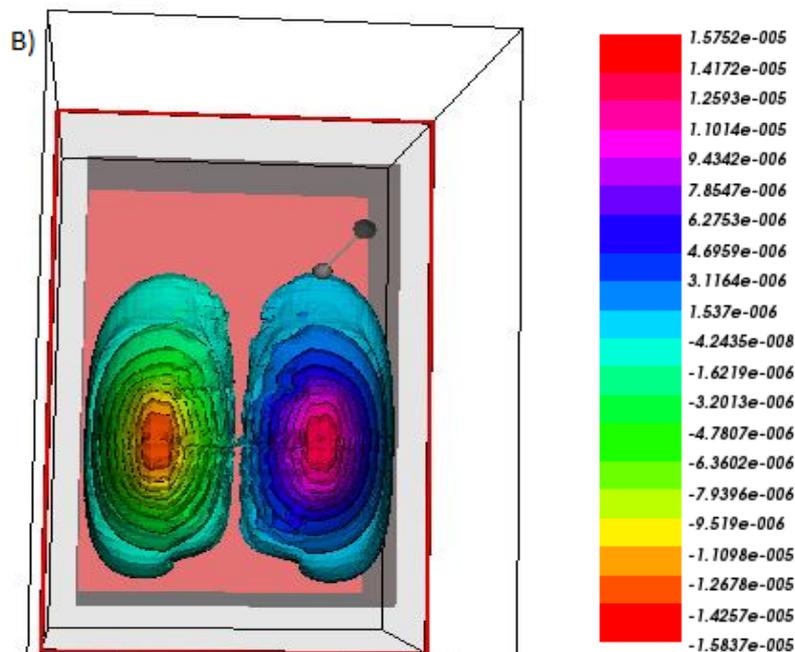
Caso 1. Calculo de L_{11} : Para el cálculo de L_{11} tendremos que dejar $I_2 = 0$. Esto significa que la densidad de corriente en el secundario es cero.

Una vez definida la nueva densidad de corriente ya podemos calcular el potencial magnetostático aplicando la ecuación (4.5) con la geometría y mallado mencionados anteriormente.

La representación del potencial magnetostático para el caso 1 puede observarse en la Figura (4.2.8), la figura A muestra A_z y la figura B muestra A_x .



Caso 1. A_z (lineas de campo de B)



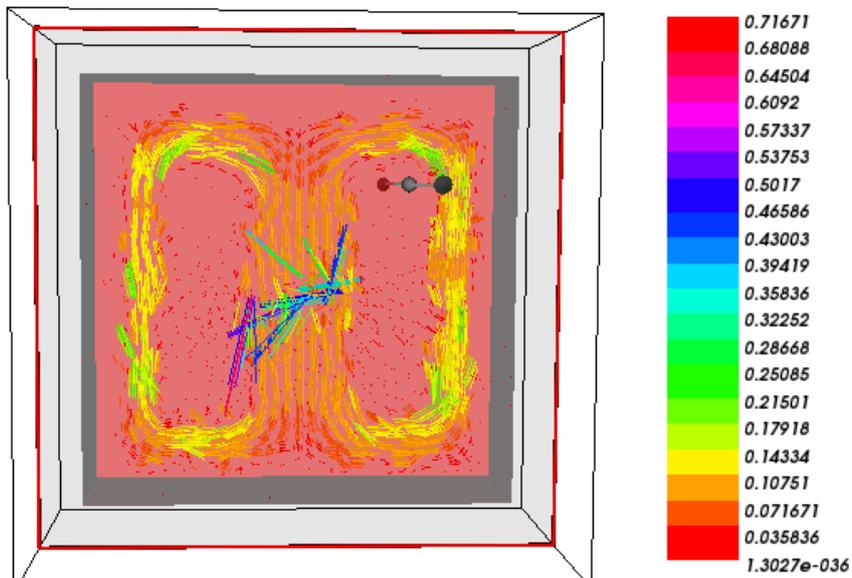
Caso 1. A_x (lineas de campo de B)

Figura(4.2.8)

Calculamos el potencial magnético aplicando la ecuación (4.2). Teniendo en cuenta que nosotros solo tenemos potencial magnetostático en x y z, la ecuación se nos queda:

$$\begin{aligned} B_x &= \frac{\partial A_z}{\partial y} \\ B_y &= \frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x} \\ B_z &= -\frac{\partial A_x}{\partial y} \end{aligned} \quad (4.6)$$

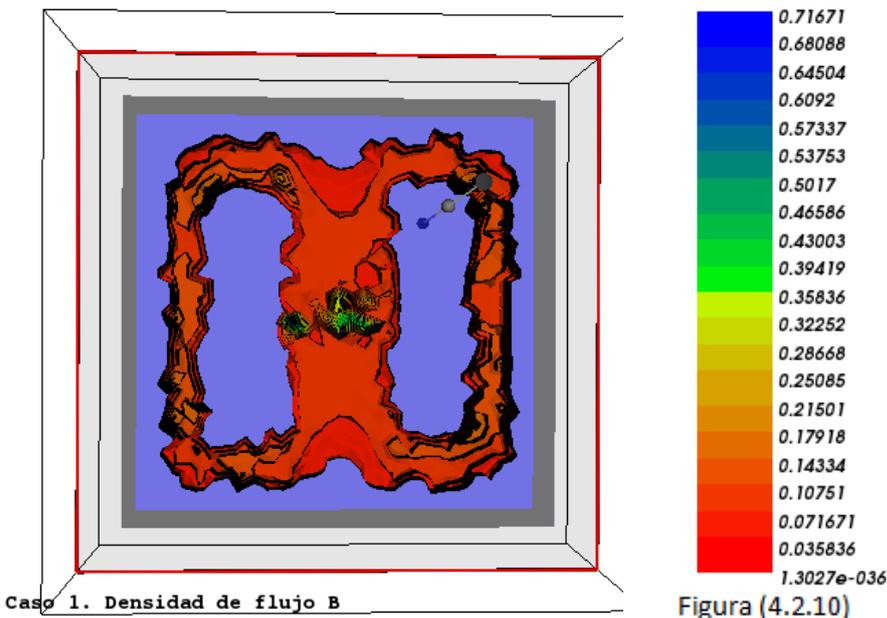
Siendo la representación del campo magnético para el caso 1 el mostrado en la Figura (4.2.9):



Caso 1. Vector de densidad de flujo B

Figura(4.2.9)

La Figura (4.2.10) representa el modulo del campo magnético para el caso 1.



Caso 1. Densidad de flujo B

Figura (4.2.10)

Una vez calculado el campo magnético podemos calcular la intensidad de campo magnético aplicando la ecuación (3.9):

$$\begin{aligned} H_x &= \frac{B_x}{\mu_m} \\ H_y &= \frac{B_y}{\mu_m} \\ H_z &= \frac{B_z}{\mu_m} \end{aligned} \quad (4.7)$$

Una vez calculados estos parámetros, ya podemos obtener la inductancia propia del primario aplicando la ecuación (3.10):

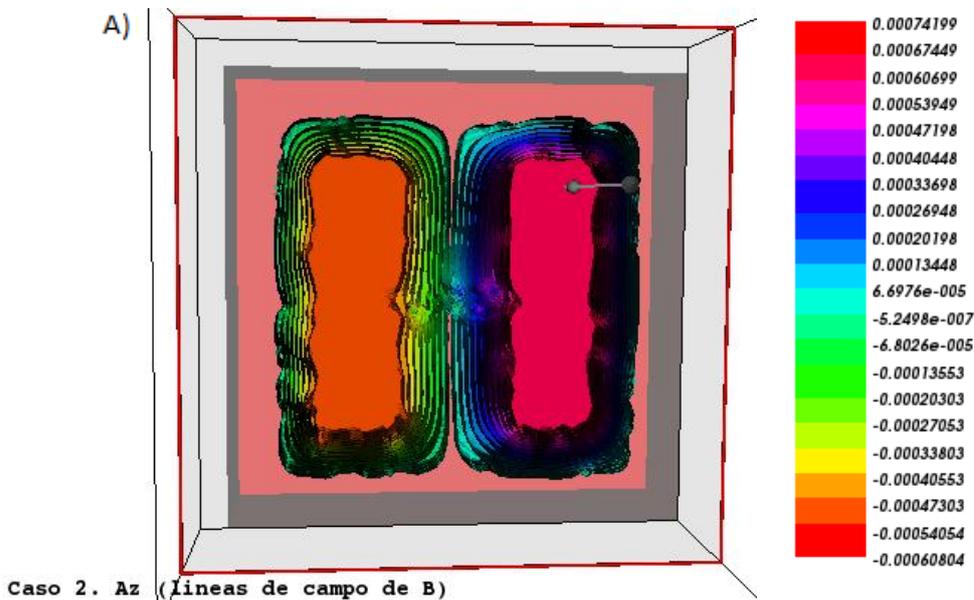
$$L_{11} = \int_{\Omega} \frac{\vec{B}_1 \cdot \vec{H}_1}{I_1^2} d\Omega = \int_{\Omega} \frac{B_{1x} \cdot H_{1x} + B_{1y} \cdot H_{1y} + B_{1z} \cdot H_{1z}}{I_1^2} d\Omega$$

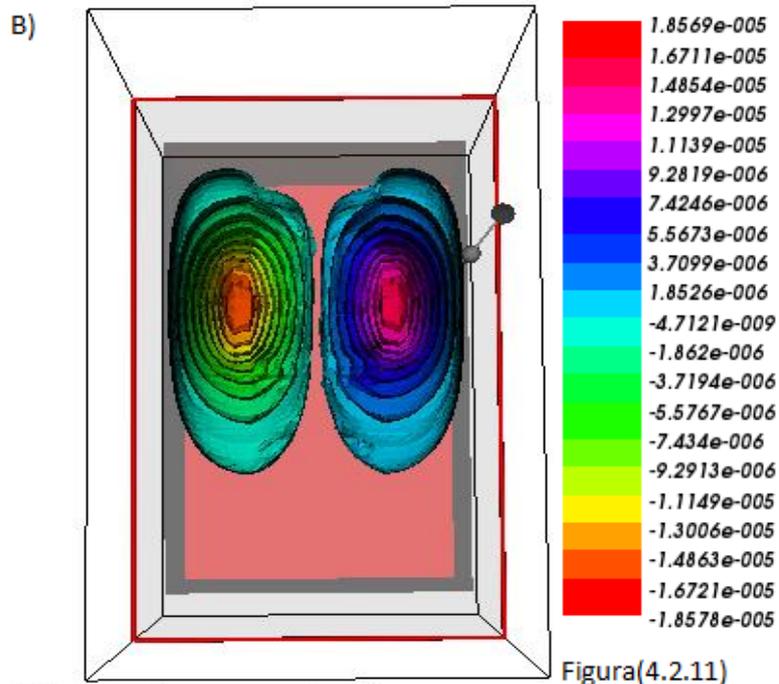
El valor obtenido para la inductancia propia del primario es $L_{11} = 1,98789 \text{ mH}$.

Caso 2. Calculo de L_{22} : Para el cálculo de L_{22} tendremos que dejar $I_1 = 0$. Esto significa que la densidad de corriente en el primario es cero.

Una vez definida la nueva densidad de corriente ya podemos calcular el potencial magnetostático aplicando la ecuación (4.5) como en el caso 1.

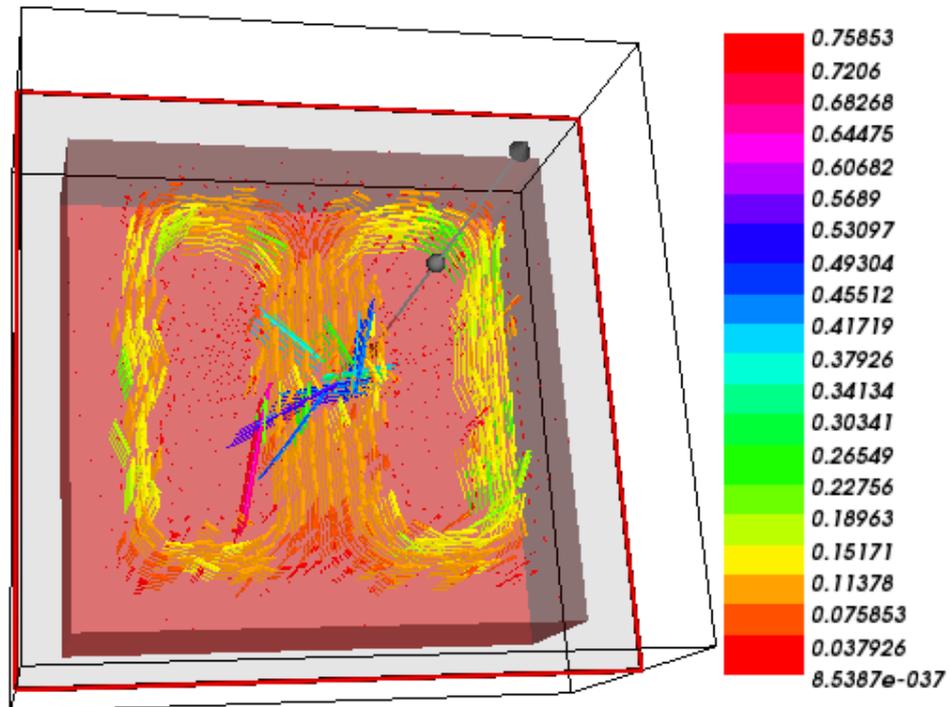
La representación del potencial magnetostático para el caso 2 puede observarse en la Figura (4.2.11), la figura A muestra A_z y la figura B muestra A_x :





Caso 2. Ax (líneas de campo de B)

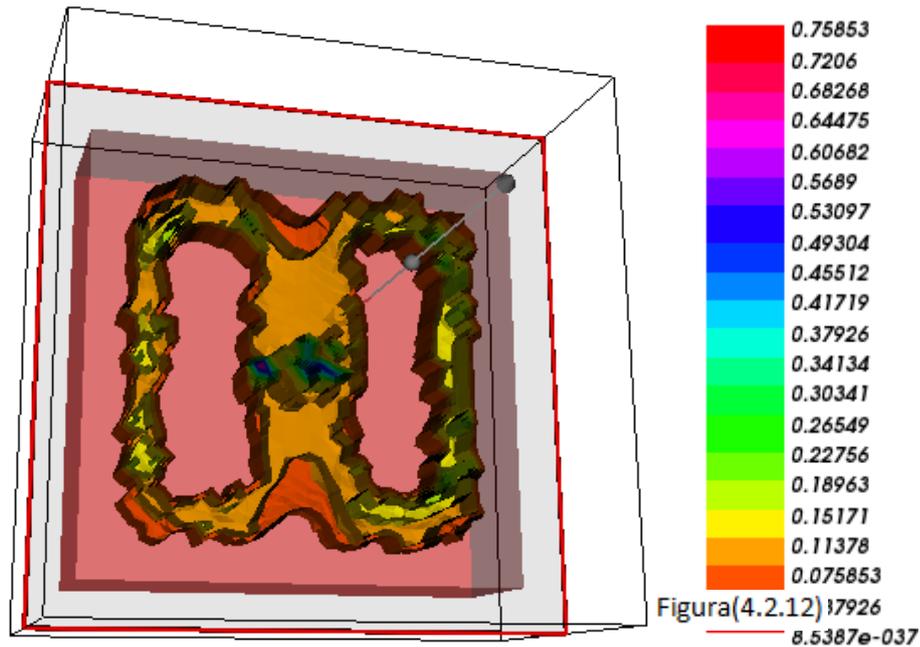
Ahora aplicamos la ecuación (4.6) y calculamos el campo magnético para el caso 2. En la Figura (4.2.12) se muestra su representación gráfica:



Caso 2. Vector de densidad de flujo B

Figura(4.2.12)

La Figura (4.1.13) representa el módulo del campo magnético para el caso 2:



Caso 2. Densidad de flujo B

Figura (4.1.13)

Una vez calculado el campo magnético podemos obtener la intensidad de campo magnético aplicando la ecuación (4.7).

Y ahora ya podemos calcular la inductancia propia del secundario aplicando la ecuación (3.10):

$$L_{22} = \int_{\Omega} \frac{\vec{B}_2 \cdot \vec{H}_2}{I_2^2} d\Omega = \int_{\Omega} \frac{B_{2x} \cdot H_{2x} + B_{2y} \cdot H_{2y} + B_{2z} \cdot H_{2z}}{I_2^2} d\Omega$$

El valor obtenido para la inductancia propia del secundario es $L_{22} = 0,898908 \text{ mH}$.

Como el problema es lineal la inductancia mutua se puede calcular como combinación lineal del caso 1 y 2. Aplicando la ecuación (3.11) tenemos que:

$$M_{12} = M_{21} = \int_{\Omega} \frac{\vec{B}_2 \cdot \vec{H}_1}{I_2 I_1} d\Omega = \int_{\Omega} \frac{B_{2x} \cdot H_{1x} + B_{2y} \cdot H_{1y} + B_{2z} \cdot H_{1z}}{I_2 I_1} d\Omega$$

Aplicando esta ecuación el valor obtenido de la inductancia mutua es $M_{12} = 1,25611 \text{ mH}$

Y por último, calculamos la inductancia de dispersión aplicando la ecuación (3.12).

$$L_{1l} = L_{11} - \frac{N_1}{N_2} M_{21} ; L_{2l} = L_{22} - \frac{N_2}{N_1} M_{12}$$

Siendo N_1 y N_2 el número de vueltas del primario y del secundario respectivamente (para ver los valores de N_1 y N_2 consultar *anexo VII*). Las impedancias de dispersión serán:

$$L_{1l} = 103,719 \mu\text{H}; L_{2l} = 61,4983 \mu\text{H}$$

Ahora sumamos ambas impedancias referidas al primario aplicando la ecuación (3.12).

$$L_{cc} = L_{1l} + r t^2 L_{2l}$$

Siendo el valor obtenido de la impedancia de dispersión total $L_{cc} = 242,09 \mu\text{H}$

El análisis de los resultados se realizará posteriormente en el apartado de comparación de resultados, tomando como referencia los resultados del transformador real.

4.2.4 Código del problema.

En el *anexo VII* se encuentra todo el código para la resolución de este problema.

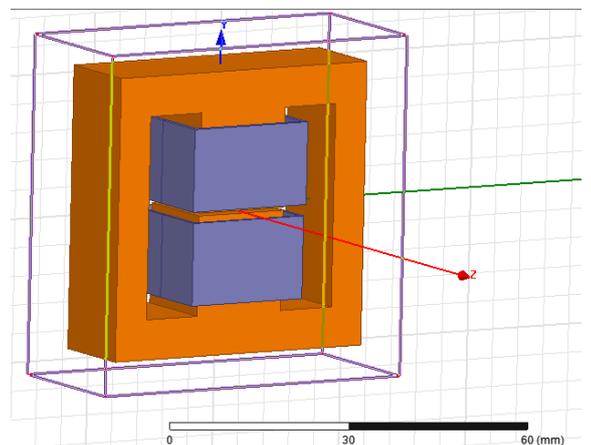
5. Análisis del problema en ANSYS Maxwell.

Para el análisis de nuestro problema en ANSYS Maxwell no hay que introducir ninguna ecuación, el software ya las lleva internamente. Pero ANSYS Maxwell solo nos calcula las inductancias propias y la inductancia mutua por lo que a partir de estos valores hay que calcular la inductancia de dispersión.

5.1 Diseño de la geometría y límites del problema.

El software de ANSYS Maxwell nos permite introducir la geometría como si fuese un software de diseño. Lo que facilita mucho el problema de describir la geometría.

Primero diseñamos el núcleo y las bobinas de manera normal, utilizando las medidas que nos proporcionan el *anexo I* y el *anexo II* sobre el núcleo y el hilo de litz, teniendo en cuenta que las bobinas están separadas 2mm del núcleo. Una vez hecho esto elegimos una pestaña que se llama "*create region*" e introducimos los límites de nuestro problema, que son los mismos que en el caso en 3D de FreeFem++, un prisma de 60mm x 60mm x 36mm. Esta pestaña le dice al programa donde tienen que aplicarse las condiciones de contorno.

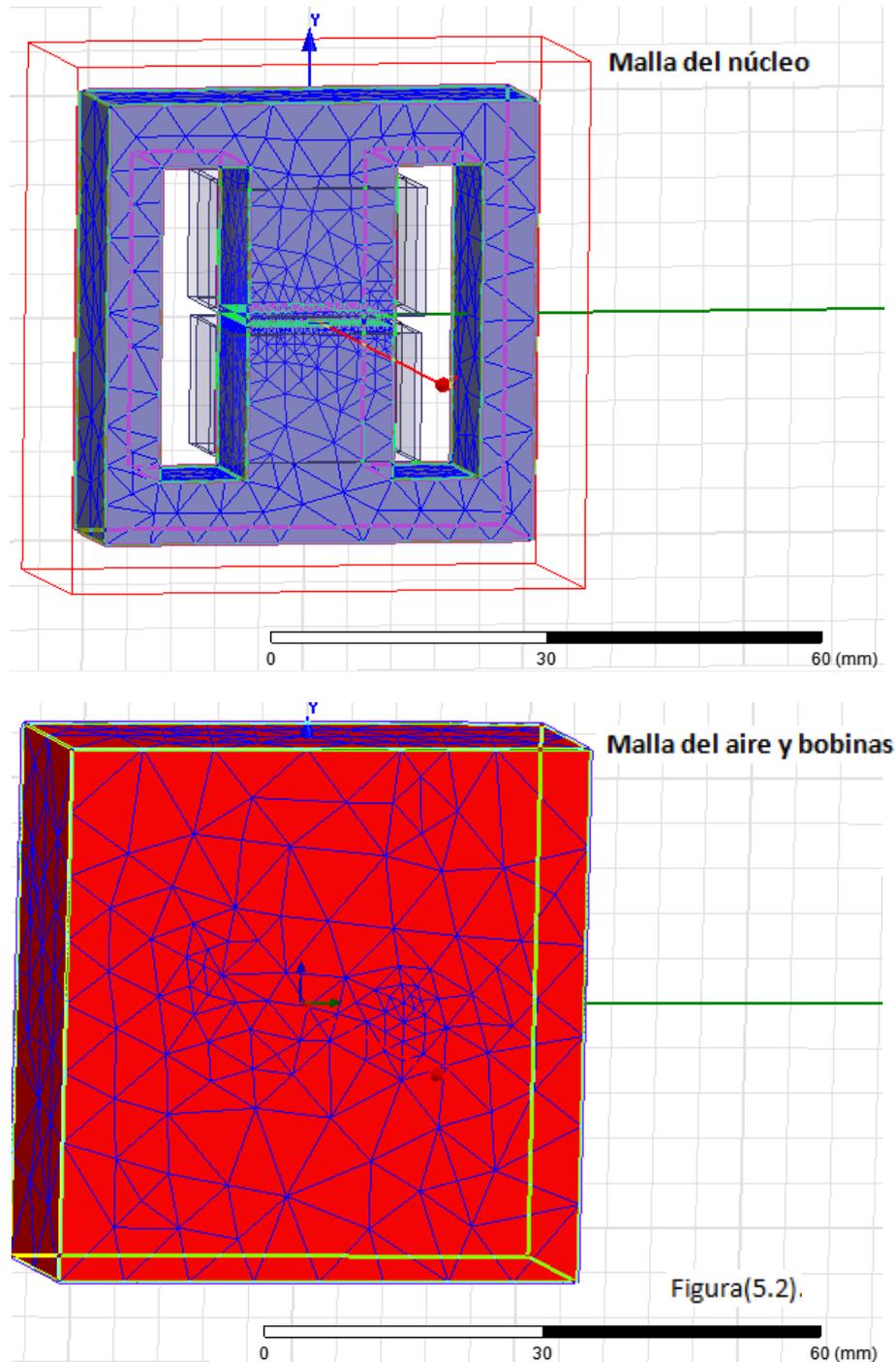


Figura(5.1)

La Figura (5.1) representa la geometría del problema.

A diferencia de FreeFem++, ANSYS realiza un auto mallado en función de los parámetros que queremos calcular y el tanto por ciento de error que podemos tener.

Según lo elegido nuestra malla quedaría como muestra la Figura (5.2):

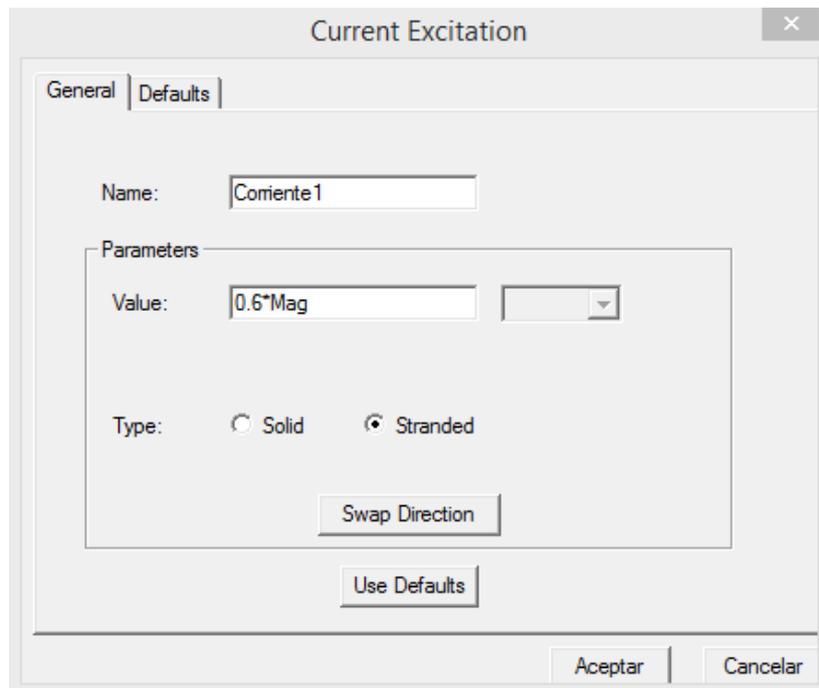


5.2 Definición de los parámetros del problema.

Como explicaré a continuación, la definición de los parámetros característicos para la resolución de nuestro problema es mucho más cómoda e intuitiva en ANSYS que en FreeFem++ .

Primero describiremos cómo se define la densidad de corriente. Una vez que ya hemos diseñado nuestras bobinas, lo que necesitaremos para definir nuestro parámetro será concretar una superficie que corte transversalmente a cada una de nuestras bobinas, de manera que la superficie describa uno de los planos de propagación de la corriente en la bobina.

Ahora, lo que tenemos que hacer para definir la densidad de corriente es elegir las superficies que hemos creado en las bobinas y asignarle a cada una de estas superficies un valor de corriente. Esto se hace de la siguiente manera. Pulsamos "*Maxwell 3D > Excitations > Assign > Current*" y nos saldrá la ventana que muestra la Figura (5.3):

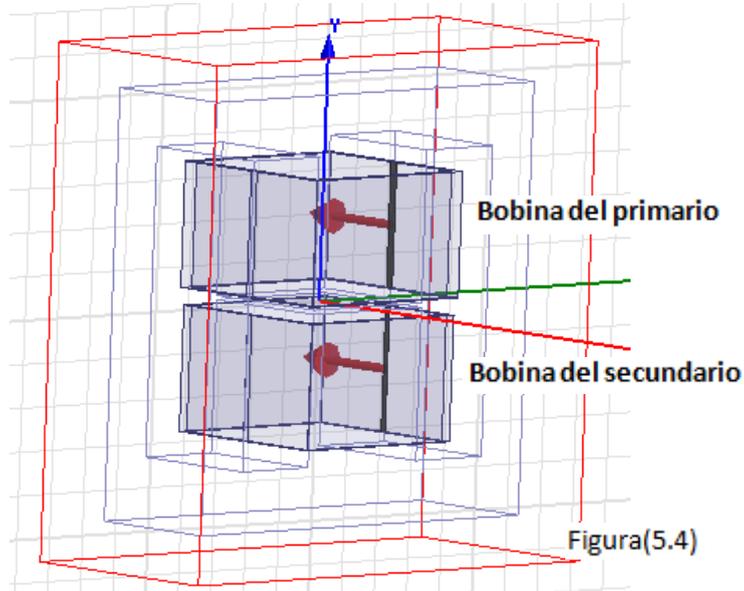


Figura(5.3)

En este cuadro de diálogo asignamos un nombre a nuestra corriente. Para ello, introducimos el valor de la intensidad que pasa por nuestra superficie (que será el producto de la intensidad que pasa por un conductor por el número de vuelta de nuestra bobina, $Mag = N1 = 54$). En segundo lugar, elegimos si nuestro hilo es un sólido compacto o está formado por varios hilos más pequeños (en nuestro caso el hilo de litz está formado por 30 hilos más pequeños) y, por último, elegimos la dirección de la intensidad (ANSYS asigna una dirección automáticamente, si no es la que tenemos en

nuestro problema pulsamos el botón "*Swap Direction*"). Para finalizar, pulsamos "*Aceptar*" y repetimos el proceso para la otra bobina.

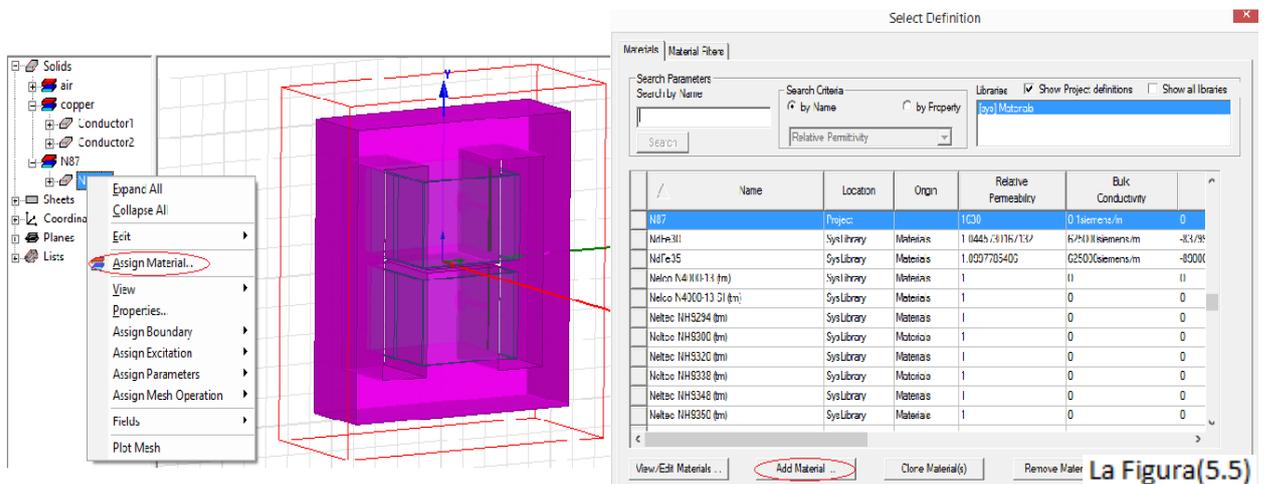
Una vez hecho esto la corriente debe quedar definida como muestra la Figura (5.4):



Para la definición de la permeabilidad lo único que tendremos que hacer será elegir el material de cada volumen y ANSYS le asignará la permeabilidad que le corresponda. Para ello, pulsamos el segundo botón del ratón en el nombre del volumen que queramos asignar un material y pulsamos "*Assign Material*". Si realizamos esto se nos abrirá un cuadro de diálogo donde podemos elegir el material pulsando encima de él y dándole a "*Aceptar*".

En el caso de que no se encuentre el material de nuestro problema, tendremos que añadirlo manualmente. Para hacer esto, pulsamos "*Add Material*" e introducimos el nombre del material, su permeabilidad relativa y su conductancia por metro (la conductancia es la inversa de la resistencia eléctrica).

La Figura (5.5) muestra como asignar el material y como añadir uno nuevo:



Para la asignación del material de las bobinas hemos elegido el cobre ya que el hilo de litz está hecho de cobre.

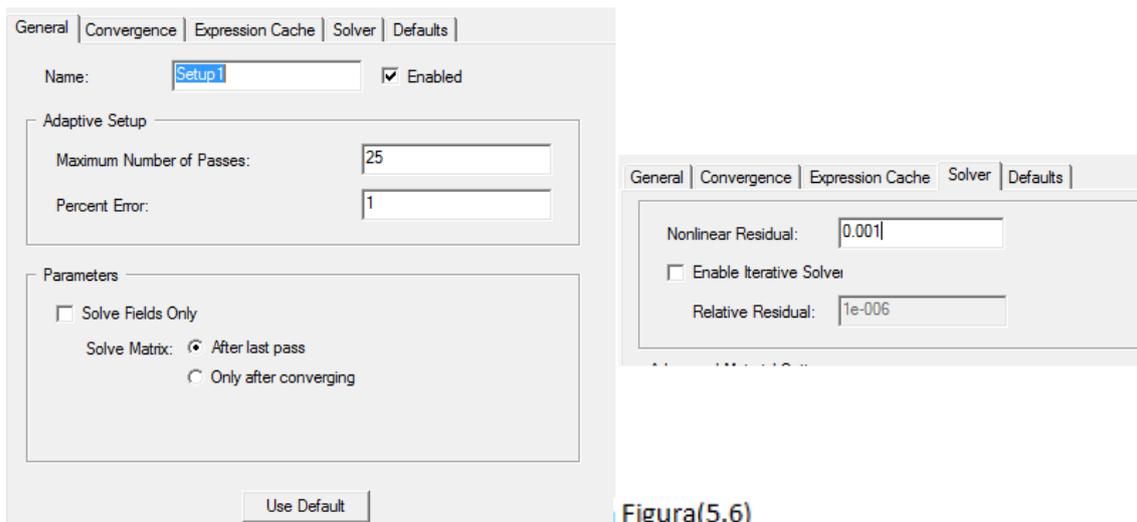
Una vez realizado esto, ya tendremos introducidos todos los parámetros de nuestro problema.

5.3 Resolución del problema y resultados.

Para la resolución de nuestro problema en ANSYS Maxwell, una vez que hemos creado un nuevo proyecto, debemos especificar el tipo de problema que es. Para ello, tenemos que darle a "*Maxwell 3D > Solution Type*" y elegir, en nuestro caso, magnetostático "*Magnetostatic*".

Una vez hecho todo lo anterior descrito, ya podemos disponernos a solucionar nuestro problema. Primero creamos un nuevo sistema de análisis (aunque no es necesario porque ANSYS te proporciona uno por defecto), para ello, pulsamos "*Maxwell 3D > Analysis Setup > Add Solution Setup*". Cuando realicemos esto se nos abrirá una pantalla donde tenemos que elegir el tanto por ciento de error de nuestro problema, en nuestro caso hemos elegido 1%, y el residuo no lineal, que hace referencia al error al realizar un problema de manera lineal con materiales con un comportamiento no lineal, en nuestro caso es un 0,001. Debido a que el número de paso predeterminado por ANSYS no es suficiente para asegurar un 1% de error, hemos tenido que aumentarlo de 10 a 25. Los pasos hacen referencia al mallado, ANSYS resuelve el problema con un mallado predeterminado y si el error que le proporciona el problema es mayor que el elegido, ANSYS refina la malla y vuelve a resolver el problema (este refinamiento por paso también puede elegirse en ANSYS, nosotros lo tenemos en un 30%), así hasta que el error que nos da sea igual o menor que el elegido.

La Figura (5.6) muestra las ventanas con los parámetros que hemos introducido:

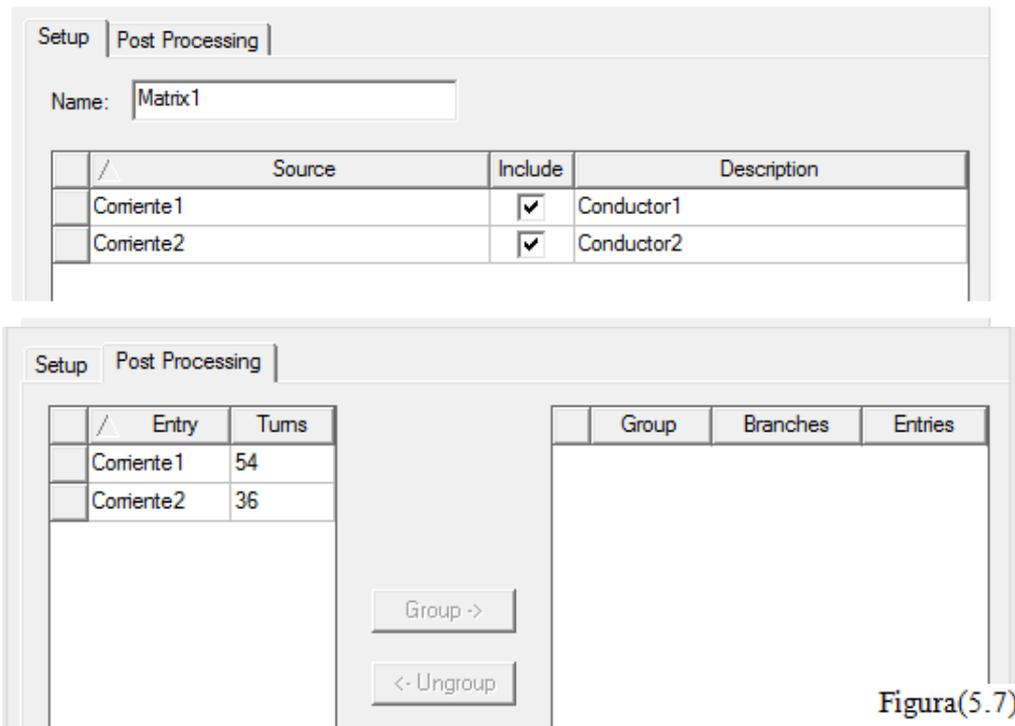


Figura(5.6)

Por último tenemos que decirle a ANSYS qué queremos calcular. Para ello, pulsamos "*Maxwell 3D > Parameters > Assign > Matrix*" y nos aparecerá una ventana con las

dos corrientes que hemos definido anteriormente. Marcamos las dos corrientes y le damos a la pestaña de post proceso, en esta pestaña nos saldrán las corrientes que hemos elegido en la ventana anterior. Al lado de cada corriente podremos poner el número de vueltas de cada bobina (en nuestro caso 54 para el primario y 36 para el secundario). Introducimos el número de vueltas y seleccionamos ambas corrientes. Una vez hecho esto, le damos al botón de **"Aceptar"**.

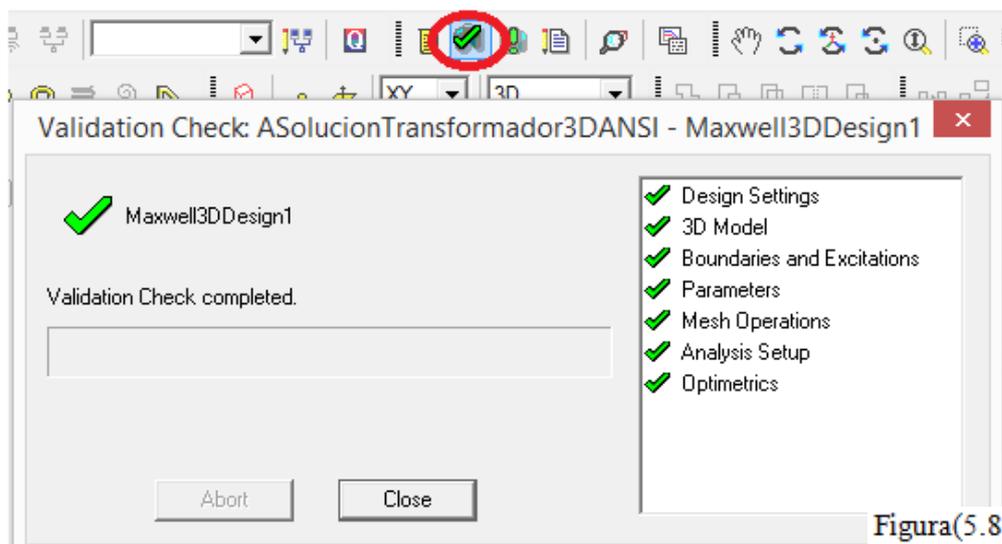
La Figura (5.7) muestra lo que debemos elegir en esta ventana:



Figura(5.7)

Para comprobar si hemos realizado todos estos pasos bien pulsamos el botón **"Validate"**.

La Figura (5.8) muestra qué tiene que salir si hemos hecho todos los pasos bien:



Figura(5.8)

Ahora pulsamos "**Analyze All**" y se resuelve nuestro problema. Para ver los resultados pulsamos "**Maxwell 3D > Results > Solution Data**" y nos saldrá una ventana con los valores de las inductancias propias y la mutua tomando la bobina como si tuviera una vuelta. Para ver el resultado con nuestro número de vueltas, marcamos una ventana que pone "**PostProcessed**" y ya nos muestra los valores de nuestras inductancias.

Los valores que ANSYS nos da son $L_{11} = 1,9859 \text{ mH}$, $L_{22} = 0,88301 \text{ mH}$ y $M_{12} = 1,2381 \text{ mH}$ para la inductancia propia del primario, secundario y la inductancia mutuas, respectivamente.

Ahora para calcular la inductancia de dispersión hay que hacerlo a parte, ya que ANSYS no nos proporciona estos valores.

Calculamos la inductancia de dispersión aplicando la ecuación (3.12):

$$L_{1l} = L_{11} - \frac{N_1}{N_2} M_{21} ; L_{2l} = L_{22} - \frac{N_2}{N_1} M_{12}$$

Siendo N_1 y N_2 el numero de vueltas del primario y del secundario, respectivamente. Las inductancias de dispersión serán:

$$L_{1l} = 128,75 \mu\text{H} ; L_{2l} = 57,61 \mu\text{H}$$

Ahora sumamos ambas inductancias referidas al primario aplicando la ecuación (3.12):

$$L_{cc} = L_{1l} + r t^2 L_{2l}$$

Y obtenemos el valor de la inductancia de dispersión total $L_{cc} = 258,373 \mu\text{H}$.

6. Comparación entre los resultados 2D y 3D de FreeFem++.

En este apartado compararemos los resultados del problema bidimensional con los tridimensionales y explicaremos cuáles de estos se acercan más a los valores reales.

Los valores reales han sido obtenidos del trabajo de Villarejo et al.¹

Tabla 2. Comparativa de datos 2D y 3D en FreeFem++

Inductancia	Simulación 2D	Simulación 3D	Datos reales
Propia del primario (mH)	1,89235	1,98789	-
Propia del secundario (mH)	0,847872	0,898908	-
Mutua (mH)	1,21574	1,25611	1,3
Dispersión del primario (μH)	68,7434	103,719	-
Dispersión del secundario (μH)	37,3791	61,4983	-
Dispersión total (μH)	152,846	242,09	270

Llegados a este punto, antes de comparar los resultados, aclarar que los valores reales no son 100% precisos, ya que al fabricar un transformador no se hace de manera perfecta. Pero como en este proyecto una de las cosas que queremos averiguar es si nuestros programas nos pueden servir de referencia a la hora de diseñar un transformador, los valores experimentales son los que vamos a tomar como referencia, por lo que las simulaciones que más se acerquen a estos valores serán mejores aproximaciones.

Como podemos observar en la Tabla 2 el valor de la inductancia mutua, en ambos casos, se aproxima mucho al valor real, siendo el caso en 3D mejor aproximación que el caso en 2D, en cambio, la inductancia de dispersión varía mucho en el caso 2D, siendo el valor en 3D muy aproximado al valor real. En mi opinión, esto se debe a que el valor de la inductancia de dispersión se calcula en un post proceso a partir de los valores de inductancia propia y mutua. Si tuviésemos que juzgar teniendo en cuenta sólo los valores de las inductancias propias y las inductancias mutuas, el problema en 2D sería una buena simplificación del problema del transformador, ya que estos valores difieren poco entre ambos resultados. Pero como en nuestro caso, debemos ser más precisos y tener menor porcentaje de error (ya que los valores de inductancias propias y mutua se utilizan en un post proceso y los errores se van arrastrando), resolver el problema como si fuese 2D no es un buen método para dar solución a nuestro problema. En cambio, la simulación en 3D es muy buen método para calcular las inductancias de nuestro transformador y darnos una aproximación fiable de los valores que nos daría si fabricásemos el transformador.

7. Comparativa entre los resultados de ANSYS y FreeFem++.

En este apartado compararemos los resultados del problema tridimensional resueltos con ANSYS, con los tridimensionales resueltos con FreeFem++, y explicaremos cuáles de éstos se acercan más a los valores reales.

Tabla 3. Comparativa de datos 3D en ANSYS y datos 3D en FreeFem++

Inductancia	ANSYS	FreeFem++	Datos reales
Propia del primario (mH)	1,9859	1,98789	-
Propia del secundario (mH)	0,88301	0,898908	-
Mutua (mH)	1,2381	1,25611	1,3
Dispersión del primario (μ H)	128,75	103,719	-
Dispersión del secundario (μ H)	57,61	61,4983	-
Dispersión total (μ H)	258,373	242,09	270

Como podemos comprobar en la Tabla 3, ambos programas nos aportan prácticamente los mismos resultados, siendo en este caso la mejor aproximación de la inductancia mutua la que nos proporciona FreeFem++ y en el caso de la inductancia de dispersión, observamos que es mejor la que nos proporciona ANSYS.

En este caso, aunque ANSYS se acerca más al valor de la inductancia de dispersión y por su parte, FreeFem++ se aproxima más al valor de la inductancia mutua, creo que ANSYS es algo más preciso que FreeFem++ a la hora de calcular las inductancias propias y mutua, ya que a la inductancia de dispersión le afectan mucho los errores que acumulemos. Viendo la tendencia que seguía la inductancia mutua y de dispersión de la simulación en FreeFem++ conforme refinábamos el mallado, el valor de la inductancia mutua bajaba y el de la inductancia de dispersión aumentaba, si hubiésemos seguido refinando el mallado, los valores se hubiesen acercado más a los de ANSYS.

En esta ocasión ambos programas han sido muy buenos calculando las inductancias de nuestro transformador por lo que ambos programas serían igual de válidos para hacer una aproximación de los valores de nuestro transformador real antes de fabricarlo.

8. Conclusiones.

En este apartado, explicaremos los pros y los contras de los dos programas utilizados para resolver nuestro problema y aclararemos en qué casos utilizaríamos cada uno de ellos.

Así pues, las desventajas de FreeFem++ son:

- Necesidad de conocer un alto lenguaje de programación. Ya que todo el problema se resuelve a base de programar código.
- Requerimiento de altos conocimientos matemáticos y físicos. Ya que tenemos que trabajar con ecuaciones diferenciales.
- La realización del problema precisa una elevada inversión de tiempo.
- Existen problemas en algunas sentencias del código que nos obligan a plantear alternativas para calcular la solución que no serían las más óptimas para nuestro problema.
- Nuestro problema requiere un tiempo excesivo de ejecución. Una vez terminado el código, el programa tardó 3 días aproximadamente en ejecutar el problema.

En cuanto a las ventajas, podemos destacar:

- Numerosa variedad de alternativas que nos ofrece a la hora de introducir los parámetros de nuestro problema.
- Capacidad de cambiar parámetros de forma instantánea.
- Gran flexibilidad a la hora de utilizar ecuaciones en el programa.
- Capacidad de controlar todos los pasos de la resolución de nuestro problema.
- Posibilidad de mostrarnos representaciones de nuestro problema.
- El coste del software, ya que este es gratuito.

Debido a sus ventajas y desventajas, FreeFem++ es un programa que utilizaría más como software de enseñanza que de diseño, ya que necesitamos tener conocimientos físicos y matemáticos para su utilización y podemos tener un seguimiento de cada proceso que nos lleva a la solución final.

Por otro lado, las desventajas de ANSYS son:

- Escasa flexibilidad a la hora de introducir los parámetros de nuestro problema.
- Poco seguimiento del proceso que realiza para obtener la solución del problema.
- Elevado precio del software.

Así pues, las ventajas son:

- Fácil utilización del software, ya que es muy intuitivo. Debido a esto, aprender a utilizar el programa requiere poco tiempo.
- No necesitas un alto nivel matemático, ya que no hay que utilizar ninguna ecuación directamente.
- Te permite realizar tu simulación con un error predeterminado.
- Gran velocidad de ejecución. El programa se ejecuta en pocos minutos y no en días como FreeFem++.
- Posibilidad de introducir geometrías muy complejas que darían problemas en FreeFem++.

Debido a esto, ANSYS es un buen software de diseño, sin embargo, no sería muy buen software de enseñanza porque no trabaja con las ecuaciones que definen el problema.

En conclusión, si lo que deseamos es diseñar un transformador, el programa que utilizaría sería ANSYS, aunque con ambos llegaríamos a los mismos resultados. En cambio, si lo que quiero realizar es el análisis de un problema para enseñanza, sin duda utilizaría antes FreeFem++.

9. Referencias bibliográficas.

¹Villarejo J, Chaouki S, de Jodar E, García-Valverde R, Krebs F. Convertidor CC/CC de media tensión de entrada para paneles solares orgánicos. UPCT.

❖ Elementos finitos:

- Periago Esparza F. Apuntes de clase.
- El método de los elementos finitos en la ingeniería.
(<http://finitosxtecnologia.blogspot.com.es/>)
- Pezzotti S, Antico F. Introducción a la Teoría de Elementos Finitos. Facultad de Ingeniería Universidad Nacional de La Plata.
(<http://www.monografias.com/trabajos-pdf4/elementos-finitos-elementos-resorte-barra-y-viga/elementos-finitos-elementos-resorte-barra-y-viga.pdf>)

❖ FreeFem++:

- Periago Esparza F. Apuntes de clase.
- Chacón Vera, Eliseo. Notas sobre FreeFem++ 2D y 3D. 2010
- Hecht, Frédéric. FreeFem++ manual, Third Edition. 2013.
(<http://www.freefem.org/ff++/ftp/freefem++doc.pdf>)

❖ ANSYS Maxwell:

- Web oficial de de ANSYS.
(<http://www.ansys.com/Products/Simulation+Technology/Electronics/Electromechanical/ANSYS+Maxwell>)

❖ Magnitudes:

- Fuentes Moreno J. A. Apuntes sobre inducción electromagnética y energía magnetica.
- Taylor & Francis Group. Inductors and Transformers for Power Electronics. 2005.
- Hurley W. G, Wölfle W.H. Transformers and Inductor for Power Electronics. National University of Ireland, Galway, Ireland, 2013.
- Fuentes Moreno, Juan Álvaro. Apuntes sobre circuitos magnéticos y transformadores.
- Web de conceptos fisicos.
(<http://hyperphysics.phy-astr.gsu.edu/hbase/hph.html>)
- Web del Departamento de Física Aplicada de la Universidad de Sevilla.
(http://laplace.us.es/wiki/index.php/P%C3%A1gina_Principal)

10. Anexo.

10.1 Anexo I: Planos del núcleo del transformador ETD 49/25/16.

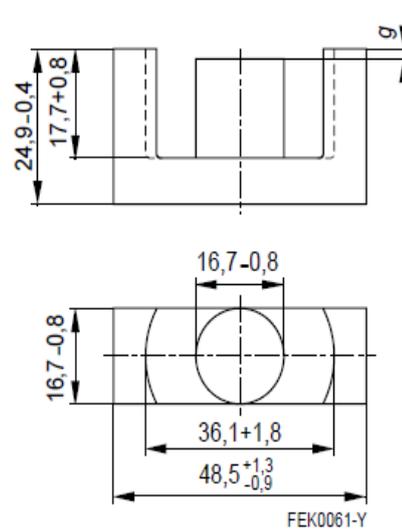
ETD 49/25/16	
Core	B66367

- To IEC 61185
- For SMPS transformers with optimum weight/performance ratio at small volume
- Delivery mode: single units

Magnetic characteristics (per set)

$$\begin{aligned} \Sigma l/A &= 0.54 \text{ mm}^{-1} \\ l_e &= 114 \text{ mm} \\ A_e &= 211 \text{ mm}^2 \\ A_{\min} &= 209 \text{ mm}^2 \\ V_e &= 24100 \text{ mm}^3 \end{aligned}$$

Approx. weight 124 g/set



Ungapped

Material	A_L value nH	μ_e	P_V W/set	Ordering code
N27	3700 +30/-20%	1590	< 4.59 (200 mT, 25 kHz, 100 °C)	B66367G0000X127
N87	3800 +30/-20%	1630	< 12.40 (200 mT, 100 kHz, 100 °C)	B66367G0000X187
N97	3900 +30/-20%	1680	< 10.60 (200 mT, 100 kHz, 100 °C)	B66367G0000X197

Gapped

Material	g mm	A_L value approx. nH	μ_e	Ordering code ** = 27 (N27) = 87 (N87)
N27, N87	0.20 ± 0.02	1035	444	B66367G0200X1**
	0.50 ± 0.05	525	225	B66367G0500X1**
	1.00 ± 0.05	314	135	B66367G1000X1**
	2.00 ± 0.05	188	81	B66367G2000X1**

The A_L value in the table applies to a core set comprising one ungapped core (dimension $g = 0$) and one gapped core (dimension $g > 0$).

ETD 49/25/16

Accessories

B66368

Coil former

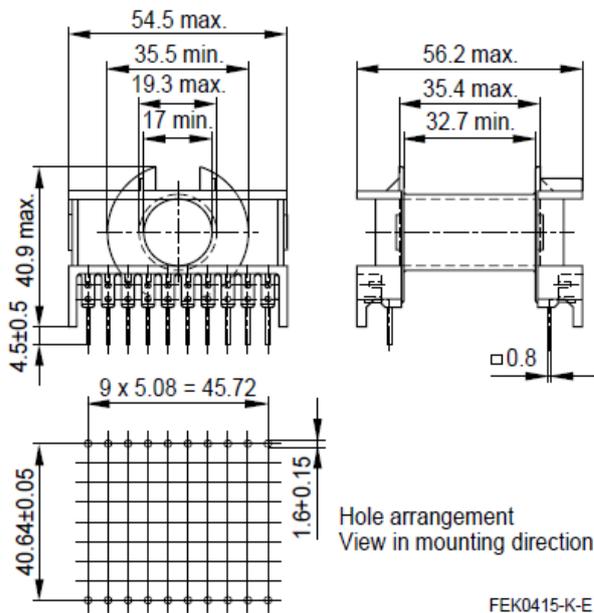
Material: GFR polyterephthalate, UL 94 V-0, insulation class to IEC 60085:
 B66368B: F \triangleq max. operating temperature 155 °C, color code black
 Valox 420-SE0® [E45329 (M)], GE PLASTICS B V
 B66368W: H \triangleq max. operating temperature 180 °C, color code black
 Rynite FR 530® [E41938 (M)], E I DUPONT DE NEMOURS & CO INC
 Solderability: to IEC 60068-2-20, test Ta, method 1 (aging 3): 235 °C, 2 s
 Resistance to soldering heat: to IEC 60068-2-20, test Tb, method 1B: 350 °C, 3.5 s
 Winding: see Data Book 2007, chapter "Processing notes, 2.1"

Yoke

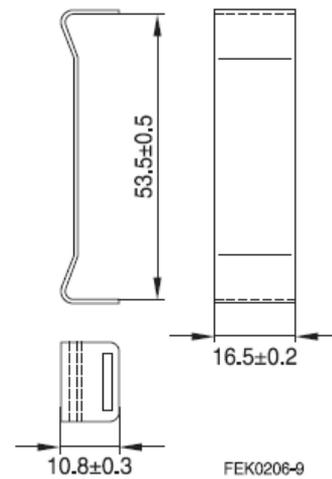
Material: Stainless spring steel (0.4 mm)

Coil former					Ordering code
Sections	A_N mm ²	l_N mm	A_R value $\mu\Omega$	Pins	
1	269.4	86	11	20	B66368B1020T001 B66368W1020T001
Yoke (ordering code per piece, 2 are required)					B66368A2000X000

Coil former

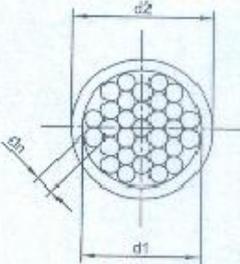


Yoke



10.2 Anexo II: Definición de las características del hilo de litz.

Litz wire/ 多股绞合漆包铜线
Cu1L 30x0.1



d1 ... external diameter without braiding/wrapping [mm]
多股线外直径 (无丝包层) [毫米]
d2 ... external diameter with braiding/wrapping [mm]
多股线外直径 (含丝包层) [毫米]
dn ... nominal wire diameter of single wire [mm]
单股裸铜线直径 [毫米]
R₂₀... DC-Resistance per kilometer at 20°C [Ohm/km]
直流电阻/每千米 @ 20° C [欧姆/千米]

Litz wire construction / 多股绞合漆包铜线规格:

number of strands 股数	dn [mm] 单股线径	bunching 束数	length of twist [mm] 绞合节距	d1 [mm]		d2 [mm]		cross-section [mm ²] 横截面积
				min. 最小值	max. 最大值	min. 最小值	max. 最大值	
30	0.1	1 x 30	23±3	-	-	-	0.8	0.23550
braiding/wrapping layers * material 丝包层数/丝包材料			twisting turns [Ts/m] 绞合圈数	Standard/标准		R ₂₀ [Ohm/km]		
1 x 尼龙 / nylon			-	IEC 60950		nom. 中间值	min. 最小值	max. 最大值
						74.02	67.80	81.70

Single wire construction / 单股漆包铜线规格:

Grade/等级: Grade 1	single wire diameter (with varnish) 漆包铜线外直径 (涂漆膜后) [mm] / [毫米]
Standard/标准: JIS_C_3202	
UL: UL-listed NEMA: MW79	min. / 最小值
Temperature class/ 温度等级: 155 °C	max. / 最大值
	- 0.118

approved single wire suppliers and models / 已批准的单线供应商与型号:

DES 文件控制

Dielectric strength/ 绝缘强度: U_{eff} ≥ 700 V

test method: twisted pair test
测试方法: 绞线电击测试

AQL: 0.15	检验标准: I	日期: ---
日期: ---	标准: ---	---

材料: Cu + mod. Polyurethane

Dimensions in mm/尺寸单位 mm	DATE	BY	CHK
00	HZ	DAT	03.06.10
01	HZ	DAT	31.08.10
02	Helen ZHANG		10.10.10

30x0.1mm G1, 23Ts/m, 155Deg

DANTRAFO F104 08.12.24 19110030 1/1

10.3 Anexo III: Código en FreeFem++ del problema 2D.

```
//*****
PREPROCESO*****

//Cargamos las librerías necesarias para nuestra simulación

load "medit";

load "msh3";

load "tetgen";

//Constantes que definen el refinamiento del mallado para cada región

real nHierro = 5, nAirefino = 10, nAireIn = 8, nAire = 0.25;

//Constante que utilizaremos para darle un valor a los límites del volumen control

real CondContorno = 50;

real Eaisx = 2e-3; //Distancia entre el conductor y el núcleo central

real Eaisy = 2.3e-3; //Distancia entre el conductor y el núcleo superior e inferior

real longitudMaquina = 16e-3; //Espesor del transformador

//Parámetros del problema

real Po = 300; //Potencia total del problema

//Conductor 1

//Definición de parámetros de la bobina 1

real N1 = 54; //Número de vueltas bobina 1

real d1 = 0.1e-3;

real Sec01 = pi*(d1^2)/4; //Sección de uno de los hilos que forma el conductor

real Nh1 = 30; //Número de hilos que forman el conductor

real Sec1 = Nh1*Sec01; //Sección del conductor 1 en milímetros S = 0.2356
```

```

real I1 = 0.6;//Intensidad de la bobina 1

real V1 = 500;

//Conductor 2

//Definición de parámetros de la bobina 2

real N2 = 36;//Numero de vueltas bobina 2

real d2 = 0.1e-3;

real Sec02 = pi*(d2^2)/4;

real Nh2 = 30;//Numero de hilos que forman el conductor

real Sec2 = Nh2*Sec02;//Sección del conductor 1 en milímetros

real rt = N1/N2;

real I2 = I1*rt;//Intensidad de la bobina 2

real V2 = V1/rt;

//*****Parámetros*****//

bool esperar = true;

real x0 = 30e-3;

real x1 = -30e-3;

real x2 = 24.25e-3;

real x3 = -24.25e-3;

real x4 = 18.25e-3;

real x5 = 8.25e-3;

real x6 = -8.25e-3;

real x7 = -18.25e-3;

real x8 = -9.1335e-3;

real x9 = 9.1335e-3;

real x10 = -8.839e-3;

real x11 = 8.839e-3;

```

real y0 = 30e-3;

real y1 = 24.7e-3;

real y2 = -24.7e-3;

real y3 = -30e-3;

real y4 = 17.7e-3;

real y5 = 0.25e-3;

real y6 = -0.25e-3;

real y7 = -17.7e-3;

real y8 = 3.36e-3;

real y9 = -3.3e-3;

////////////////////////////////////

//////////////////////////////////// VENTANA DEL CIRCUITO MAGNÉTICO

////////////////////////////////////

// Defino el hueco del interior por tres rectángulos => el izquierdo (Izq), el centro (Cen) y el

// derecho (Der)

// Rectángulo de la derecha

real xIzquierdaDer = x5;

real xDerechaDer = x4;

real ySuperiorDers = y4;

real ySuperiorDeri = y5;

real yInferiorDeri = y7;

real yInferiorDers = y6;

```

// Rectángulo del centro
real xlzquierdaCen = x6;
real xDerechaCen = x5;
real ySuperiorCen = y5;
real yInferiorCen = y6;

// Rectángulo de la izquierda
real xlzquierdalzq = x7;
real xDerechalzq = x6;
real ySuperiorlzqs = y4;
real ySuperiorlzqi = y5;
real yInferiorlzqi = y7;
real yInferiorlzqs = y6;

//////////Aire dentro del transformador//////////

//Rectángulo de la Derecha
border CDerRD (t = yInferiorDeri, ySuperiorDers) {x = xDerechaDer; y = t;};
border CSupRD (t = xDerechaDer, xlzquierdaDer) {x = t; y = ySuperiorDers;};
border ClzqRDi (t = yInferiorDers, yInferiorDeri) {x = xlzquierdaDer; y = t;};
border ClzqRDs (t = ySuperiorDers, ySuperiorDeri) {x = xlzquierdaDer; y = t;};
border CInfRD (t = xlzquierdaDer, xDerechaDer) {x = t; y = yInferiorDeri;};

//Rectángulo de la Izquierda
border CDerRli (t = yInferiorlzqi, yInferiorlzqs) {x = xDerechalzq; y = t;};
border CDerRls (t = ySuperiorlzqi, ySuperiorlzqs) {x = xDerechalzq; y = t;};
border CSupRI (t = xDerechalzq, xlzquierdalzq) {x = t; y = ySuperiorlzqs;};
border ClzqRI (t = ySuperiorlzqs, yInferiorlzqi) {x = xlzquierdalzq; y = t;};
border CInfRI (t = xlzquierdalzq, xDerechalzq) {x = t; y = yInferiorlzqi;};

```

```
//Rectángulo Central

border CDerRC (t = yInferiorCen, ySuperiorCen) {x = xDerechaCen; y = t;};

border CSupRC (t = xDerechaCen, xlzquierdaCen) {x = t; y = ySuperiorCen;};

border ClzqRC (t = ySuperiorCen, yInferiorCen) {x = xlzquierdaCen; y = t;};

border CInfRC (t = xlzquierdaCen, xDerechaCen) {x = t; y = yInferiorCen;};

//////////Núcleo de hierro//////////

border CDerN (t = y2, y1) {x = x2; y = t;};

border CSupN (t = x2, x3) {x = t; y = y1;};

border ClzqN (t = y1, y2) {x = x3; y = t;};

border CInfN (t = x3, x2) {x = t; y = y2;};

//////////Contorno del problema//////////

border CDerP (t = y3, y0) {x = x0; y = t; label = CondContorno;};

border CSupP (t = x0, x1) {x = t; y = y0; label = CondContorno;};

border ClzqP (t = y0, y3) {x = x1; y = t; label = CondContorno;};

border CInfP (t = x1, x0) {x = t; y = y3; label = CondContorno;};

//////////Bobinas//////////

// Contorno de la Bobina Inferior Derecha (BID)

// l -> límite, izq y der, b -> bobina, i -> inferior, d -> derecha

real lizqbid = x5+Eaisx;

real lderbid = x11+Eaisx;

real lsupbid = y9+Eaisy;

real linfbid = y7+Eaisy;

border CDerBID (t = linfbid, lsupbid) {x = lderbid; y = t;};

border CSupBID (t = lderbid, lizqbid) {x = t; y = lsupbid;};
```

```
border ClzqBID (t = lsupbid, linfbid) {x = lizqbid; y = t;};
```

```
border ClnfBID (t = lizqbid, lderbid) {x = t; y = linfbid;};
```

```
// Contorno de la Bobina Superior Derecha (BSD)
```

```
real lizqbsd = x5+Eaisx;
```

```
real lderbsd = x9+Eaisx;
```

```
real lsupbsd = y4-Eaisy;
```

```
real linfbsd = y8-Eaisy;
```

```
border CDerBSD (t = linfbsd, lsupbsd) {x = lderbsd; y = t;};
```

```
border CSupBSD (t = lderbsd, lizqbsd) {x = t; y = lsupbsd;};
```

```
border ClzqBSD (t = lsupbsd, linfbsd) {x = lizqbsd; y = t;};
```

```
border ClnfBSD (t = lizqbsd, lderbsd) {x = t; y = linfbsd;};
```

```
// Contorno de la Bobina Superior Izquierda (BSI)
```

```
real lizqbsi = x8-Eaisx;
```

```
real lderbsi = x6-Eaisx;
```

```
real lsupbsi = y4-Eaisy;
```

```
real linfbsi = y8-Eaisy;
```

```
border CDerBSI (t = linfbsi, lsupbsi) {x = lderbsi; y = t;};
```

```
border CSupBSI (t = lderbsi, lizqbsi) {x = t; y = lsupbsi;};
```

```
border ClzqBSI (t = lsupbsi, linfbsi) {x = lizqbsi; y = t;};
```

```
border ClnfBSI (t = lizqbsi, lderbsi) {x = t; y = linfbsi;};
```

```
// Contorno de la Bobina Inferior Izquierda (BII)
```

```
real lizqbii = x10-Eaisx;
```

```

real lderbii = x6-Eaisx;

real lsupbii = y9+Eaisy;

real linfbii = y7+Eaisy;

border CDerBII (t = linfbii, lsupbii) {x = lderbii; y = t;};

border CSupBII (t = lderbii, lizqbii) {x = t;    y = lsupbii;};

border ClzqBII (t = lsupbii, linfbii) {x = lizqbii; y = t;};

border CInfBII (t = lizqbii, lderbii) {x = t;    y = linfbii;};

//////////

////Mallado del problema 2D

//////////

mesh Th = buildmesh(CDerP(60*nAire) +CSupP(60*nAire) +ClzqP(60*nAire) +CInfP(60*nAire)
+CDerN(49.4*nHierro) +CSupN(48.5*nHierro) +ClzqN(49.4*nHierro)+CInfN(48.5*nHierro)
+CDerRD(36.2*nAireIn)+CSupRD(10.0*nAireIn)+ClzqRDs(17.85*2*nAireIn)
+CSupRC(16.5*2*nAirefino) +CDerRIs(17.85*2*nAireIn) +CSupRI(10.0*nAireIn)
+ClzqRI(36.2*nAireIn) + CInfRI(10.0*nAireIn)+CDerRli(17.85*2*nAireIn)
+CInfRC(16.5*2*nAirefino)+ClzqRDi(17.85*2*nAireIn) + CInfRD(10.0*nAireIn)
+CDerBID(5.6916*nAirefino)+CSupBID(0.6324*nAirefino)+ClzqBID(5.6916*nAirefino)
+CInfBID(0.6324*nAirefino)+CDerBSD(5.6916*nAirefino)+CSupBSD(0.9486*nAirefino)
+ClzqBSD(5.6916*nAirefino)+CInfBSD(0.9486*nAirefino)+CDerBII(5.6916*nAirefino)
+CSupBII(0.6324*nAirefino)+ClzqBII(5.6916*nAirefino)+CInfBII(0.6324*nAirefino)
+CDerBSI(5.6916*nAirefino)+CSupBSI(0.9486*nAirefino)+ClzqBSI(5.6916*nAirefino)
+CInfBSI(0.9486*nAirefino) );

plot(Th, fill = 1, wait = esperar, value = 1, cmm = "Malla 2D");

```

```

////////////////////////////////////
// Espacios de elementos finitos
////////////////////////////////////

fespace Vh(Th,P2); // Espacio para calcular el potencial vectorial magnético A
fespace Bh(Th,P1); // Espacio para calcular B
fespace Ch(Th,P0); // Espacio para definir ctes

// funciones de Vh para cálculo del potencial vectorial A
Vh u1, v1, u2, v2, u3, v3;

// funciones de Bh para el cálculo de B
Bh Bx1, By1, mu, Bmodulo1, Bx2, By2, muRelativa, Bmodulo2, Bx3, By3, Bmodulo3;

////////////////////////////////////
// Obtención de las regiones para definir las densidades de corriente y la permeabilidad
magnética
////////////////////////////////////
Ch reg = region;
plot(reg, fill = 1, wait = esperar, value = 1, cmm = "regiones");

int bobinaSuperiorDerecha = reg(lizqbsd+1e-3, linfbsd+1e-3);
int bobinaSuperiorIzquierda = reg(-lderbsd+1e-3, linfbsd+1e-3);
int bobinaInferiorDerecha = reg(lizqbsd+1e-3, -lsupbsd+1e-3);
int bobinaInferiorIzquierda = reg(-lderbsd+1e-3, -lsupbsd+1e-3);
int aireinterior = reg(0, 0);
int aireexterior = reg(x0-1e-3, y0-1e-3);
int hierro = reg(-0.8,-0.8);

```

```

//*****PERMEABILIDADES*****//

//Permeabilidad del aire

real mu0 = 4*pi*1e-7;

//Permeabilidad del hierro

real muFe0 = 1630;

//Definición de la región con permeabilidad del aire

func muAire = mu0*(x <= x0)*(x >= x1)*(y <= y0)*(y > y1)+
    mu0*(x <= x0)*(x >= x1)*(y < y2)*(y >= y3)+
    mu0*(x <= x0)*(x > x2)*(y <= y1)*(y >= y2)+
    mu0*(x < x3)*(x >= x1)*(y <= y1)*(y >= y2)+
    mu0*(x < x4)*(x > x5)*(y < y4)*(y > y7)+
    mu0*(x < x6)*(x > x7)*(y < y4)*(y > y7)+
    mu0*(x <= x5)*(x >= x6)*(y < y5)*(y > y6);

//Definición de la región con permeabilidad del hierro

func muFe = muFe0*mu0*(x <= x2)*(x >= x3)*(y <= y1)*(y >= y4)+
    muFe0*mu0*(x <= x2)*(x >= x3)*(y <= y7)*(y >= y2)+
    muFe0*mu0*(x <= x2)*(x >= x4)*(y < y4)*(y > y7)+
    muFe0*mu0*(x <= x7)*(x >= x3)*(y < y4)*(y > y7)+
    muFe0*mu0*(x <= x5)*(x >= x6)*(y < y4)*(y >= y5)+
    muFe0*mu0*(x <= x5)*(x >= x6)*(y <= y6)*(y > y7);

//Función que suma ambas regiones con sus permeabilidades correspondientes

func muP = muAire + muFe;

mu = muP;

```

```
//Permeabilidad relativa
muRelativa = muFe0/mu0;

Ch muT = muP;

//*****//DENSIDAD DE CORRIENTE*****//

//Valor densidad de corriente
real DensidadCorrienteCu1 = I1/(Sec1); //Módulo de la densidad de corriente conductor 1
real DensidadCorrienteCu2 = I2/(Sec2); //Módulo de la densidad de corriente conductor 2

cout<<"Valor de la densidad corriente->"<<DensidadCorrienteCu1<<endl;
cout<<"Valor de la densidad corriente->"<<DensidadCorrienteCu2<<endl;

//Caso 1 I2=0.
real I11 = 0.6;
func j1 = ((I11/(Sec1))*(x <= x6-Eaisx)*(x >= x8-Eaisx)*(y <= y4-Eaisy)*(y >= y8-Eaisy)
+(-1)*(I11/(Sec1))*(x <= x9+Eaisx)*(x >= x5+Eaisx)*(y <= y4-Eaisy)*(y >= y8-Eaisy));

//Caso 2 I1=0.
real I22 = 0.9;
func j2 = ((I22/(Sec2))*(x <= x6-Eaisx)*(x >= x10-Eaisx)*(y <= y9+Eaisy)*(y >= y7+Eaisy)
+(-1)*(I22/(Sec2))*(x <= x11+Eaisx)*(x >= x5+Eaisx)*(y <= y9+Eaisy)*(y >= y7+Eaisy));

//Caso 3 I1 e I2 distintos de 0.
func j3 = (DensidadCorrienteCu1*(x <= x6-Eaisx)*(x >= x8-Eaisx)*(y <= y4-Eaisy)*(y >= y8-Eaisy)
+(-1)*DensidadCorrienteCu1*(x <= x9+Eaisx)*(x >= x5+Eaisx)*(y <= y4-Eaisy)*(y >= y8-
Eaisy)
+DensidadCorrienteCu2*(x <= x6-Eaisx)*(x >= x10-Eaisx)*(y <= y9+Eaisy)*(y >= y7+Eaisy)
+(-1)*DensidadCorrienteCu2*(x <= x11+Eaisx)*(x >= x5+Eaisx)*(y <= y9+Eaisy)*(y >=
y7+Eaisy));
```

```

Ch jh = j3;

//Representación de la permeabilidad magnética
plot(muT, fill = 1, wait = esperar, value = 1, cmm = "Permeabilidad");

//Representación de la densidad de corriente
plot(jh, fill = 1, wait = esperar, value = 1, cmm = "densidades de corriente");

////////////////////////////////////
////////////////////////////////////
// Resolvemos el problema Caso 1
////////////////////////////////////
////////////////////////////////////

macro Grad2(u) [dx(u),dy(u)] // EOM, gradiente

problem magnetostatica1(u1,v1)=int2d(Th)((1/muP)*(Grad2(v1)'*Grad2(u1)))
-int2d(Th)(j1*v1)
+on(CondContorno, u1 = 0);

magnetostatica1;

// Calculo de las componentes de B
Bx1 = dy(u1);
By1 = -dx(u1);
Bmodulo1 = sqrt(Bx1^2+By1^2);

```

```

////////////////////////////////////
// Representación gráfica de la solución
////////////////////////////////////
// Potencial vector (componente z)
plot(u1, cmm ="Caso 1. Az (lineas de campo de B)");

// Densidad de flujo
plot([Bx1,By1], value = true, coef = -1.6, wait = esperar, cmm = "Caso 1. Vector de densidad de
flujo B");
plot(Bmodulo1, value = true , wait = esperar, cmm ="Caso 1. Densidad de flujo B");

// Intensidad de campo magnético
Bh Hx1, Hy1;
Hx1 = Bx1/mu;
Hy1 = By1/mu;

//Calculo de la inductancia L1
real L1 = longitudMaquina*int2d(Th) ((Bx1*Hx1+By1*Hy1)/(l11^2));
cout << "Inductancia L1: " << L1 << endl;

////////////////////////////////////
////////////////////////////////////
// Resolvemos el problema Caso 2
////////////////////////////////////
////////////////////////////////////
problem magnetostatica2(u2,v2)=int2d(Th)((1/muP)*(Grad2(v2)*Grad2(u2)))
-int2d(Th)(j2*v2)
+on(CondContorno, u2 = 0);
magnetostatica2;

```

```
// Calculo de las componentes de B

Bx2 = dy(u2);

By2 = -dx(u2);

Bmodulo2 = sqrt((Bx2^2+By2^2)/(l11^2));

////////////////////////////////////

// Representación gráfica de la solución

////////////////////////////////////

// Potencial vector (componente z)

plot(u2, cmm ="Caso 2. Az (lineas de campo de B)");

// Densidad de flujo

plot([Bx2,By2], value = true, coef = -1.6, wait = esperar, cmm = "Caso 2. Vector de densidad de
flujo B");

plot(Bmodulo2, value = true , wait = esperar, cmm ="Caso 2. Densidad de flujo B");

// Intensidad de campo magnético

Bh Hx2, Hy2;

Hx2 = Bx2/mu;

Hy2 = By2/mu;

//Calculo de la inductancia L2

real L2 = longitudMaquina*int2d(Th) ((Bx2*Hx2+By2*Hy2)/(l22^2));

cout << "Inductancia L2: " << L2 << endl;
```

```
//Calculo de la inductancia M12
```

```
//Como nuestro problema es lineal la inductancia mutua se puede calcular como combinación de los dos casos anteriores
```

```
real M12 = longitudMaquina*int2d(Th) ((Bx1*Hx2+By1*Hy2)/(I11*I22));
```

```
cout << "Inductancia M12: " << M12 << endl;
```

```
real M21 = longitudMaquina*int2d(Th) ((Bx2*Hx1+By2*Hy1)/(I11*I22));
```

```
cout << "Inductancia M21: " << M21 << endl;
```

```
////////////////////////////////////
```

```
//Por ultimo calculamos la inductancia observado desde el primario
```

```
////////////////////////////////////
```

```
//Primero calculamos L1l
```

```
real L1l = longitudMaquina*int2d(Th) (((Bx1*Hx1+By1*Hy1)/(I11^2))
```

```
- rt*((Bx1*Hx2+By1*Hy2)/(I11*I22)));
```

```
cout << "Inductancia L1l: " << L1l << endl;
```

```
//Segundo calculamos L2l
```

```
real L2l = longitudMaquina*int2d(Th) (((Bx2*Hx2+By2*Hy2)/(I22^2))
```

```
- ((Bx1*Hx2+By1*Hy2)/(I11*I22*rt)));
```

```
cout << "Inductancia L2l: " << L2l << endl;
```

//Para obtener la inductancia de dispersión total sumamos las inductancias observadas desde el primario

```
real Lcc1 = L1l + (L2l*((rt)^2));
```

```
cout << "Inductancia Lcc1: " << Lcc1 << endl;
```

10.4 Anexo IV: Permeabilidad del acero del núcleo del transformador.

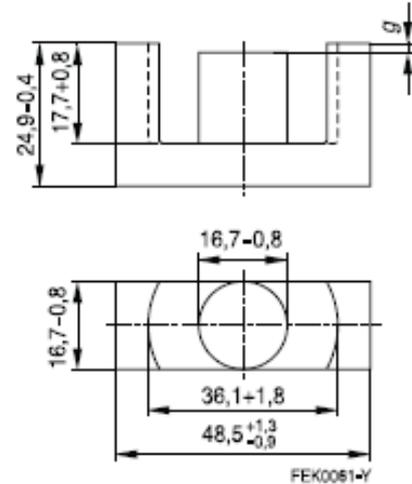
ETD 49/25/16	
Core	B66367

- To IEC 61185
- For SMPS transformers with optimum weight/performance ratio at small volume
- Delivery mode: single units

Magnetic characteristics (per set)

$$\begin{aligned} \Sigma l/A &= 0.54 \text{ mm}^{-1} \\ l_e &= 114 \text{ mm} \\ A_e &= 211 \text{ mm}^2 \\ A_{\min} &= 209 \text{ mm}^2 \\ V_e &= 24100 \text{ mm}^3 \end{aligned}$$

Approx. weight 124 g/set



Ungapped

Material	A_L value nH	μ_e	B_S^* mT	P_V W/set	Ordering code
N27	3700 +30/-20%	1590	320	< 4.59 (200 mT, 25 kHz, 100 °C)	B66367G0000X127
N87	3800 +30/-20%	1630	320	< 12.40 (200 mT, 100 kHz, 100 °C)	B66367G0000X187
N97	3900 +30/-20%	1680	320	< 10.60 (200 mT, 100 kHz, 100 °C)	B66367G0000X197

* $H = 250 \text{ A/m}$; $f = 10 \text{ kHz}$; $T = 100 \text{ °C}$

Gapped

Material	g mm	A_L value approx. nH	μ_e	Ordering code ** = 27 (N27) = 87 (N87)
N27,	0.20 ± 0.02	1035	444	B66367G0200X1**
N87	0.50 ± 0.05	525	225	B66367G0500X1**
	1.00 ± 0.05	314	135	B66367G1000X1**
	2.00 ± 0.05	188	81	B66367G2000X1**

The A_L value in the table applies to a core set comprising one ungapped core (dimension $g = 0$) and one gapped core (dimension $g > 0$).

10.5 Anexo V: Código en FreeFem++ del mallado por partes en 3D.

```

load "medit";

load "msh3";

load "tetgen";

//*****Parámetros*****//

//////////Todas las medidas en mm //////////

bool esperar = true;

int CondContorno = 50;

//Parámetros del problema

//Conductor 1

real I1 = 0.6;//Intensidad de la bobina 1

int N1 = 54;//Numero de vueltas bobina 1

real Sec1 = 0.1e-6;//Sección del conductor 1 en milímetros

//Conductor 2

real I2 = 1.5/sqrt(2);//Intensidad de la bobina 2

int N2 = 36;//Numero de vueltas bobina 2

real Sec2 = 0.1e-6;//Sección del conductor 2 en milímetros

//DATOS PARA LAS REGIONES

int nRegFe = 0; //Hierro

int nRegAire = 4; //Aire

int nRegCu = 8; //Cobre

//Parámetros del mallado

int nbobina = 1; // Número de divisiones de los contornos de las bobinas

int naire = 1; // Número de divisiones de los contornos del hueco interior

int ncontorno = 1; // Número de divisiones de los contornos exteriores

```

```
int nnucleo = 1;
int ncontornol = 1;
int ncapas = 1;
//Dimensiones en metros
real x0 = 30e-3;
real x1 = -30e-3;
real x2 = 24.25e-3;
real x3 = -24.25e-3;
real x4 = 18.25e-3;
real x5 = 8.25e-3;
real x6 = -8.25e-3;
real x7 = -18.25e-3;
real x8 = -9.1986e-3;
real x9 = 9.1986e-3;
real x10 = -8.8824e-3;
real x11 = 8.8824e-3;

real y0 = 30e-3;
real y1 = 24.7e-3;
real y2 = -24.7e-3;
real y3 = -30e-3;
real y4 = 18.1e-3;
real y5 = 0.25e-3;
real y6 = -0.25e-3;
real y7 = -18.1e-3;
real y8 = 12.4084e-3;
real y9 = -12.4084e-3;
```

real z0 = 26e-3;

real z1 = -10e-3;

real z2 = 16e-3;

real z3 = 0e-3;

real z4 = -0.9486e-3;

real z5 = 16.9486e-3;

real z6 = -0.6324e-3;

real z7 = 16.6324e-3;

////////////////////////////////////

//////////////////////////////////// VENTANA DEL CIRCUITO MAGNÉTICO

////////////////////////////////////

// Defino el hueco del interior por tres rectángulos => el izquierdo (Izq), el centro (Cen) y el

// derecho (Der)

// Rectangulo de la derecha

real xIzquierdaDer = x9;

real xDerechaDer = x4;

real ySuperiorDer = y4;

real yInferiorDer = y7;

// Rectangulo del centro

real xIzquierdaCen = x6;

real xDerechaCen = x5;

real ySuperiorCen = y5;

real yInferiorCen = y6;

// Rectangulo de la izquierda

```
real xlzquierdalzq = x7;
real xDerechalzq = x8;
real ySuperiorlzq = y4;
real yInferiorlzq = y7;
// Rectángulo de la Derecha Central Exterior
real xlzquierdaDerCenEx = x11;
real xDerechaDerCenEx = x9;
real ySuperiorDerCenEx = y8;
real yInferiorDerCenEx = y7;

// Rectángulo de la Derecha Central Interior
real xlzquierdaDerCenIn = x5;
real xDerechaDerCenIn = x11;
real ySuperiorDerCenIn = y8;
real yInferiorDerCenIn = y9;

// Rectángulo de la Izquierda Central Exterior
real xlzquierdalzqCenEx = x8;
real xDerechalzqCenEx = x10;
real ySuperiorlzqCenEx = y8;
real yInferiorlzqCenEx = y7;

// Rectángulo de la Izquierda Central Interior
real xlzquierdalzqCenIn = x10;
real xDerechalzqCenIn = x6;
real ySuperiorlzqCenIn = y8;
real yInferiorlzqCenIn = y9;
```

// Espesor transformador

real zBT = z3;

real zAT = z2;

real zCu1TSup = z5;

real zCu1TInf = z4;

real zCu2TSup = z7;

real zCu2TInf = z6;

real zSup = z0;

real zInf = z1;

////////Aire dentro del transformador

//Rectángulo de la Derecha

border CDerRD (t = yInferiorDer, ySuperiorDer) {x = xDerechaDer; y = t;};

border CSupRD (t = xDerechaDer, xlzquierdaDer) {x = t; y = ySuperiorDer;};

border ClzqRD (t = ySuperiorDer, yInferiorDer) {x = xlzquierdaDer; y = t;};

border CInfRD (t = xlzquierdaDer, xDerechaDer) {x = t; y = yInferiorDer;};

//Rectángulo de la Izquierda

border CDerRI (t = yInferiorIzq, ySuperiorIzq) {x = xDerechalzq; y = t;};

border CSupRI (t = xDerechalzq, xlzquierdalzq) {x = t; y = ySuperiorIzq;};

border ClzqRI (t = ySuperiorIzq, yInferiorIzq) {x = xlzquierdalzq; y = t;};

border CInfRI (t = xlzquierdalzq, xDerechalzq) {x = t; y = yInferiorIzq;};

//Rectángulo Central

border CDerRC (t = yInferiorCen, ySuperiorCen) {x = xDerechaCen; y = t};

border CSupRC (t = xDerechaCen, xlzquierdaCen) {x = t; y = ySuperiorCen};

border ClzqRC (t = ySuperiorCen, yInferiorCen) {x = xlzquierdaCen; y = t};

border CInfRC (t = xlzquierdaCen, xDerechaCen) {x = t; y = yInferiorCen};

//Rectángulo Central de la Derecha Exterior

border CDerRDCE (t = yInferiorDerCenEx, ySuperiorDerCenEx) {x = xDerechaDerCenEx; y = t};

border CSupRDCE (t = xDerechaDerCenEx, xlzquierdaDerCenEx) {x = t; y = ySuperiorDerCenEx};

border ClzqRDCE (t = ySuperiorDerCenEx, yInferiorDerCenEx) {x = xlzquierdaDerCenEx; y = t};

border CInfRDCE (t = xlzquierdaDerCenEx, xDerechaDerCenEx) {x = t; y = yInferiorDerCenEx};

//Rectángulo Central de la Derecha Interior

border CDerRDCl (t = yInferiorDerCenIn, ySuperiorDerCenIn) {x = xDerechaDerCenIn; y = t};

border CSupRDCl (t = xDerechaDerCenIn, xlzquierdaDerCenIn) {x = t; y = ySuperiorDerCenIn};

border ClzqRDCl (t = ySuperiorDerCenIn, yInferiorDerCenIn) {x = xlzquierdaDerCenIn; y = t};

border CInfRDCl (t = xlzquierdaDerCenIn, xDerechaDerCenIn) {x = t; y = yInferiorDerCenIn};

//Rectángulo Central de la Izquierda Exterior

border CDerRICE (t = yInferiorIzqCenEx, ySuperiorIzqCenEx) {x = xDerechalzqCenEx; y = t};

border CSupRICE (t = xDerechalzqCenEx, xlzquierdalzqCenEx) {x = t; y = ySuperiorIzqCenEx};

border ClzqRICE (t = ySuperiorIzqCenEx, yInferiorIzqCenEx) {x = xlzquierdalzqCenEx; y = t};

```
border CInFRICE (t = xlzquierdalzqCenEx, xDerechalzqCenEx) {x = t; y =
yInferiorlzqCenEx;};
```

```
//Rectángulo Central de la Izquierda Interior
```

```
border CDerRICE (t = yInferiorlzqCenIn, ySuperiorlzqCenIn) {x = xDerechalzqCenIn; y = t;};
```

```
border CSupRICE (t = xDerechalzqCenIn, xlzquierdalzqCenIn) {x = t; y =
ySuperiorlzqCenIn;};
```

```
border ClzqRICE (t = ySuperiorlzqCenIn, yInferiorlzqCenIn) {x = xlzquierdalzqCenIn; y = t;};
```

```
border CInFRICE (t = xlzquierdalzqCenIn, xDerechalzqCenIn) {x = t; y =
yInferiorlzqCenIn;};
```

```
////////////////////////////////////
```

```
//////////////////////////////////Definimos el núcleo de Fe////////////////////////////////////
```

```
////////////////////////////////////
```

```
real xDeN = x2;
```

```
real yInfiN = y2;
```

```
real ySupsN = y1;
```

```
real xleN = x3;
```

```
real xDiN = x4;
```

```
real yInfsN = y7;
```

```
real ySupiN = y4;
```

```
real xliN = x7;
```

```
real xCDN = x5;
```

```
real yCInfN = y6;
```

```
real yCSupN = y5;
```

```
real xCIN = x6;
```

```
//Rectángulo Superior Fe
```

```
border CDerNSup(t = ySupiN, ySupsN) {x = xDeN; y = t;};
```

```
border CSupNSup(t = xDeN, xleN) {x = t; y = ySupsN;};
```

```
border ClzqNSup(t = ySupsN, ySupiN) {x = xleN; y = t;};
```

```
border CInfNSup(t = xleN, xDeN) {x = t; y = ySupiN;};
```

```
//Rectángulo Inferior Fe
```

```
border CDerNInf(t = yInfiN, yInfsN) {x = xDeN; y = t;};
```

```
border CSupNInf(t = xDeN, xleN) {x = t; y = yInfsN;};
```

```
border ClzqNInf(t = yInfsN, yInfiN) {x = xleN; y = t;};
```

```
border CInfNInf(t = xleN, xDeN) {x = t; y = yInfiN;};
```

```
//Rectángulo Derecha Fe
```

```
border CDerND(t = yInfsN, ySupiN) {x = xDeN; y = t;};
```

```
border CSupND(t = xDeN, xDiN) {x = t; y = ySupiN;};
```

```
border ClzqND(t = ySupiN, yInfsN) {x = xDiN; y = t;};
```

```
border CInfND(t = xDiN, xDeN) {x = t; y = yInfsN;};
```

```
//Rectángulo Izquierda Fe
```

```
border CDerNI(t = yInfsN, ySupiN) {x = xliN; y = t;};
```

```
border CSupNI(t = xliN, xleN) {x = t; y = ySupiN;};
```

```
border ClzqNI(t = ySupiN, yInfsN) {x = xleN; y = t;};
```

```
border CInfNI(t = xleN, xliN) {x = t; y = yInfsN;};
```

```

//Rectángulo Central Superior Fe
border CDerNCSup(t = yCSupN, ySupiN) {x = xCDN; y = t;};
border CSupNCSup(t = xCDN, xCIN) {x = t; y = ySupiN;};
border ClzqNCSup(t = ySupiN, yCSupN) {x = xCIN; y = t;};
border CInfNCSup(t = xCIN, xCDN) {x = t; y = yCSupN;};

//Rectángulo Central Inferior Fe
border CDerNCInf(t = yInfsN, yCInfN) {x = xCDN; y = t;};
border CSupNCInf(t = xCDN, xCIN) {x = t; y = yCInfN;};
border ClzqNCInf(t = yCInfN, yInfsN) {x = xCIN; y = t;};
border CInfNCInf(t = xCIN, xCDN) {x = t; y = yInfsN;};

////////////////////////////////////
//////// Rectángulo exterior de Aire P = Dominio del Problema
////////////////////////////////////

real xDeP = x0;
real yInfiP = y3;
real ySupSP = y0;
real xleP = x1;
real xDiP = x2;
real yInfsP = y2;
real ySupiP = y1;
real xliP = x3;

//Rectángulo Superior Aire
border CDerPSup(t = ySupiP, ySupSP) {x = xDeP; y = t;};
border CSupPSup(t = xDeP, xleP) {x = t; y = ySupSP; label = CondContorno;};

```

```

border ClzqPSup(t = ySupSP, ySupiP) {x = xleP; y = t;};

border ClnfPSup(t = xleP, xDeP) {x = t; y = ySupiP;};

//Rectángulo Inferior Aire

border CDerPInf(t = yInfiP, yInfsP) {x = xDeP; y = t;};

border CSupPInf(t = xDeP, xleP) {x = t; y = yInfsP;};

border ClzqPInf(t = yInfsP, yInfiP) {x = xleP; y = t;};

border ClnfPInf(t = xleP, xDeP) {x = t; y = yInfiP; label = CondContorno;};

//Rectángulo Derecha Aire

border CDerPD(t = yInfsP, ySupiP) {x = xDeP; y = t; label = CondContorno;};

border CSupPD(t = xDeP, xDiP) {x = t; y = ySupiP;};

border ClzqPD(t = ySupiP, yInfsP) {x = xDiP; y = t;};

border ClnfPD(t = xDiP, xDeP) {x = t; y = yInfsP;};

//Rectángulo Izquierda Aire

border CDerPI(t = yInfsP, ySupiP) {x = xliP; y = t;};

border CSupPI(t = xliP, xleP) {x = t; y = ySupiP;};

border ClzqPI(t = ySupiP, yInfsP) {x = xleP; y = t; label = CondContorno;};

border ClnfPI(t = xleP, xliP) {x = t; y = yInfsP;};

////////////////////////////////////

////////////////////////////////////Tapa del transformador T = Tapadera

////////////////////////////////////

real xDePT = x0;

real yInfiPT = y3;

real ySupSP = y0;

real xlePT = x1;

real xDiPT = x9;

```

```

real yInfsPT = y7;

real ySupiPT = y4;

real xliPT = x8;

real xInDPT = x11;

real yInSPT = y8;

real yInIPT = y9;

real xInIPT = x10;

//Parte derecha

border CDerPTD(t = yInfiPT, ySupsPT) {x = xDePT; y = t; label = CondContorno;};

border CSupPTD(t = xDePT, xDiPT) {x = t; y = ySupsPT;};

border ClzqPTD(t = ySupsPT, yInfiPT) {x = xDiPT; y = t;};

border CInfPTD(t = xDiPT, xDePT) {x = t; y = yInfiPT;};

//Parte izquierda

border CDerPTI(t = yInfiPT, ySupsPT) {x = xliPT; y = t;};

border CSupPTI(t = xliPT, xlePT) {x = t; y = ySupsPT;};

border ClzqPTI(t = ySupsPT, yInfiPT) {x = xlePT; y = t; label = CondContorno;};

border CInfPTI(t = xlePT, xliPT) {x = t; y = yInfiPT;};

//Parte derecha interna

border CDerPTDi(t = yInfsPT, yInSPT) {x = xDiPT; y = t;};

border CSupPTDi(t = xDiPT, xInDPT) {x = t; y = yInSPT;};

border ClzqPTDi(t = yInSPT, yInfsPT) {x = xInDPT; y = t;};

border CInfPTDi(t = xInDPT, xDiPT) {x = t; y = yInfsPT;};

```

//Parte izquierda interna

border CDerPTIi(t = yInfsPT, yInSPT) {x = xInIPT; y = t;};

border CSupPTIi(t = xInIPT, xliPT) {x = t; y = yInSPT;};

border ClzqPTIi(t = yInSPT, yInfsPT) {x = xliPT; y = t;};

border CInfPTIi(t = xliPT, xInIPT) {x = t; y = yInfsPT;};

//Parte Superior

border CDerPTSUp(t = ySupiPT, ySupsPT) {x = xDiPT; y = t;};

border CSupPTSUp(t = xDiPT, xliPT) {x = t; y = ySupsPT; label = CondContorno;};

border ClzqPTSUp(t = ySupsPT, ySupiPT) {x = xliPT; y = t;};

border CInfPTSUp(t = xliPT, xDiPT) {x = t; y = ySupiPT;};

//Parte Inferior

border CDerPTInf(t = yInfiPT, yInfsPT) {x = xDiPT; y = t;};

border CSupPTInf(t = xDiPT, xliPT) {x = t; y = yInfsPT;};

border ClzqPTInf(t = yInfsPT, yInfiPT) {x = xliPT; y = t;};

border CInfPTInf(t = xliPT, xDiPT) {x = t; y = yInfiPT; label = CondContorno;};

//Parte central

border CDerPTCen(t = yInIPT, yInSPT) {x = xInDPT; y = t;};

border CSupPTCen(t = xInDPT, xInIPT) {x = t; y = yInSPT;};

border ClzqPTCen(t = yInSPT, yInIPT) {x = xInIPT; y = t;};

border CInfPTCen(t = xInIPT, xInDPT) {x = t; y = yInIPT;};

//Parte central superior

border CDerPTCens(t = yInSPT, ySupiPT) {x = xDiPT; y = t;};

border CSupPTCens(t = xDiPT, xliPT) {x = t; y = ySupiPT;};

border ClzqPTCens(t = ySupiPT, yInSPT) {x = xliPT; y = t;};

```

border ClnfPTCens(t = xliPT, xDiPT) {x = t;    y = ylnSPT;};

//Parte central inferior

border CDerPTCeni(t = ylnfsPT, ylnIPT) {x = xlnDPT;  y = t;};

border CSupPTCeni(t = xlnDPT, xlnIPT) {x = t;    y = ylnIPT;};

border ClzqPTCeni(t = ylnIPT, ylnfsPT) {x = xlnIPT;  y = t;};

border ClnfPTCeni(t = xlnIPT, xlnDPT) {x = t;    y = ylnfsPT;};

////////////////////////////////////

////////////////////////////////////Conductor de Cobre

////////////////////////////////////

real xDeCu1 = x9;

real ylnfCu1 = y8;

real ySupCu1 = y4;

real xleCu1 = x8;

real xDiCu1 = x5;

real xliCu1 = x6;

real xDeCu2 = x11;

real ylnfCu2 = y7;

real ySupCu2 = y9;

real xleCu2 = x10;

real xDiCu2 = x5;

real xliCu2 = x6;

//Parte derecha Cu1

border CDerCu1D(t = ylnfCu1, ySupCu1) {x = xDeCu1;  y = t;};

```

```
border CSupCu1D(t = xDeCu1, xDiCu1) {x = t; y = ySupCu1};
```

```
border ClzqCu1D(t = ySupCu1, yInfCu1) {x = xDiCu1; y = t};
```

```
border CInfCu1D(t = xDiCu1, xDeCu1) {x = t; y = yInfCu1};
```

```
//Parte Izquierda Cu1
```

```
border CDerCu1I(t = yInfCu1, ySupCu1) {x = xliCu1; y = t};
```

```
border CSupCu1I(t = xliCu1, xleCu1) {x = t; y = ySupCu1};
```

```
border ClzqCu1I(t = ySupCu1, yInfCu1) {x = xleCu1; y = t};
```

```
border CInfCu1I(t = xleCu1, xliCu1) {x = t; y = yInfCu1};
```

```
//Parte Tapadera Cu1
```

```
border CDerCu1T(t = yInfCu1, ySupCu1) {x = xDeCu1; y = t};
```

```
border CSupCu1T(t = xDeCu1, xleCu1) {x = t; y = ySupCu1};
```

```
border ClzqCu1T(t = ySupCu1, yInfCu1) {x = xleCu1; y = t};
```

```
border CInfCu1T(t = xleCu1, xDeCu1) {x = t; y = yInfCu1};
```

```
//Parte derecha Cu2
```

```
border CDerCu2D(t = yInfCu2, ySupCu2) {x = xDeCu2; y = t};
```

```
border CSupCu2D(t = xDeCu2, xDiCu2) {x = t; y = ySupCu2};
```

```
border ClzqCu2D(t = ySupCu2, yInfCu2) {x = xDiCu2; y = t};
```

```
border CInfCu2D(t = xDiCu2, xDeCu2) {x = t; y = yInfCu2};
```

```
//Parte Izquierda Cu2
```

```
border CDerCu2I(t = yInfCu2, ySupCu2) {x = xliCu2; y = t};
```

```
border CSupCu2I(t = xliCu2, xleCu2) {x = t; y = ySupCu2};
```

```
border ClzqCu2I(t = ySupCu2, yInfCu2) {x = xleCu2; y = t};
```

```
border CInfCu2I(t = xleCu2, xliCu2) {x = t; y = yInfCu2};
```

```
//Parte Tapadera Cu2

border CDerCu2T(t = yInfCu2, ySupCu2) {x = xDeCu2; y = t;};

border CSupCu2T(t = xDeCu2, xleCu2) {x = t; y = ySupCu2;};

border ClzqCu2T(t = ySupCu2, yInfCu2) {x = xleCu2; y = t;};

border CInfCu2T(t = xleCu2, xDeCu2) {x = t; y = yInfCu2;};

////////////////////////////////////

////////////////////////////////////

// Malla del problema

////////////////////////////////////

////////////////////////////////////

////////////////////////////////////

////////Malla del Núcleo de hierro en 2D y 3D

////////////////////////////////////

mesh RSupN = buildmesh(CDerNSup(6*nnucleo) + CSupNSup(48*nnucleo) +
ClzqNSup(6*nnucleo) + CInfNSup(48*nnucleo));

mesh RInfN = buildmesh(CDerNInf(6*nnucleo) + CSupNInf(48*nnucleo) + ClzqNInf(6*nnucleo)
+ CInfNInf(48*nnucleo));

mesh RDN = buildmesh(CDerND(36*nnucleo) + CSupND(6*nnucleo) + ClzqND(36*nnucleo) +
CInfND(6*nnucleo));

mesh RIN = buildmesh(CDerNI(36*nnucleo) + CSupNI(6*nnucleo) + ClzqNI(36*nnucleo) +
CInfNI(6*nnucleo));

mesh RCSupNCen = buildmesh(CDerNCSup(17*nnucleo) + CSupNCSup(16*nnucleo) +
ClzqNCSup(17*nnucleo) + CInfNCSup(16*nnucleo));

mesh RCInfNCen = buildmesh(CDerNCInf(17*nnucleo) + CSupNCInf(16*nnucleo) +
ClzqNCInf(17*nnucleo) + CInfNCInf(16*nnucleo));

mesh Nucleo2D = RSupN +RInfN + RDN + RIN + RCSupNCen + RCInfNCen;

//Malla total 3D

int [int] regHierro = [0,nRegFe]; //Unifico el numero de región
```

```

mesh3 Nucleo3D = buildlayers(Nucleo2D,16*ncapas,zbound=[zBT,zAT],region = regHierro);

////////////////////////////////////

/////Malla del Conductor de cobre en 2D y 3D

////////////////////////////////////

mesh   RCu1D   =   buildmesh(CDerCu1D(5*nbobina)   +   CSupCu1D(1*nbobina)   +
ClzqCu1D(5*nbobina) + CInfCu1D(1*nbobina));

mesh RCu1I = buildmesh(CDerCu1I(5*nbobina) + CSupCu1I(1*nbobina) + ClzqCu1I(5*nbobina)
+ CInfCu1I(1*nbobina));

mesh   RCu1T   =   buildmesh(CDerCu1T(5*nbobina)   +   CSupCu1T(18*nbobina)   +
ClzqCu1T(5*nbobina) + CInfCu1T(18*nbobina));

mesh   RCu2D   =   buildmesh(CDerCu2D(5*nbobina)   +   CSupCu2D(1*nbobina)   +
ClzqCu2D(5*nbobina) + CInfCu2D(1*nbobina));

mesh RCu2I = buildmesh(CDerCu2I(5*nbobina) + CSupCu2I(1*nbobina) + ClzqCu2I(5*nbobina)
+ CInfCu2I(1*nbobina));

mesh   RCu2T   =   buildmesh(CDerCu2T(5*nbobina)   +   CSupCu2T(18*nbobina)   +
ClzqCu2T(5*nbobina) + CInfCu2T(18*nbobina));

//Malla total 3D

mesh3 RCu1D3D = buildlayers(RCu1D,18*ncapas,zbound=[zBT,zAT]);
mesh3 RCu1I3D = buildlayers(RCu1I,18*ncapas,zbound=[zBT,zAT]);
mesh3 RCu1TS3D = buildlayers(RCu1T,ncapas,zbound=[zAT,zCu1TSup]);
mesh3 RCu1TI3D = buildlayers(RCu1T,ncapas,zbound=[zCu1TInf,zBT]);

mesh3 RCu2D3D = buildlayers(RCu2D,18*ncapas,zbound=[zBT,zAT]);
mesh3 RCu2I3D = buildlayers(RCu2I,18*ncapas,zbound=[zBT,zAT]);
mesh3 RCu2TS3D = buildlayers(RCu2T,ncapas,zbound=[zAT,zCu2TSup]);
mesh3 RCu2TI3D = buildlayers(RCu2T,ncapas,zbound=[zCu2TInf,zBT]);

mesh3 CCu13D = RCu1D3D + RCu1I3D + RCu1TS3D + RCu1TI3D;
mesh3 CCu23D = RCu2D3D + RCu2I3D + RCu2TS3D + RCu2TI3D;

```

```
//zona de Cobre suma Cu1 y Cu2

int [int] regCobre = [0,nRegCu,4,nRegCu];//Le asigno su número de región

mesh3 Cobre3D = CCu13D + CCu23D;

Cobre3D = change(Cobre3D,region = regCobre);//Unifico el numero de región

////////////////////////////////////

////////Malla de Aire dentro del transformador en 2D y 3D

////////////////////////////////////

mesh RAID2D = buildmesh(CDerRD(36*naire) + CInfRD(9*naire) + ClzqRD(36*naire) +
CSupRD(9*naire));

mesh RAII2D = buildmesh(CDerRI(36*naire) + CSupRI(9*naire) + ClzqRI(36*naire) +
CInfRI(9*naire));

mesh RAIC2D = buildmesh(CDerRC(naire) + CSupRC(16*naire) + ClzqRC(naire) +
CInfRC(16*naire));

mesh RAICDE2D = buildmesh(CDerRDCE(30*naire) + CSupRDCE(naire) + ClzqRDCE(30*naire) +
CInfRDCE(naire));

mesh RAICIE2D = buildmesh(CDerRICE(30*naire) + CSupRICE(naire) + ClzqRICE(30*naire) +
CInfRICE(naire));

mesh RAICDI2D = buildmesh(CDerRDCl(30*naire) + CSupRDCl(naire) + ClzqRDCl(30*naire) +
CInfRDCl(naire));

mesh RAICII2D = buildmesh(CDerRICI(30*naire) + CSupRICI(naire) + ClzqRICI(30*naire) +
CInfRICI(naire));

mesh AireInterior2D = RAID2D +RAII2D + RAIC2D +RAICDE2D + RAICIE2D + RAICDI2D +
RAICII2D;

////////////////////////////////////

//////// Rectángulo exterior de Aire

////////////////////////////////////

mesh RAEPSup2D = buildmesh(CDerPSup(5*naire) + CSupPSup(60*naire) + ClzqPSup(5*naire)
+ CInfPSup(60*naire));

mesh RAEPInf2D = buildmesh(CDerPInf(5*naire) + CSupPInf(60*naire) + ClzqPInf(5*naire) +
CInfPInf(60*naire));
```

```
mesh RAEPD2D = buildmesh(CDerPD(60*naire) + CSupPD(5*naire) + ClzqPD(60*naire) +
ClnfPD(5*naire));
```

```
mesh RAEPI2D = buildmesh(CDerPI(60*naire) + CSupPI(5*naire) + ClzqPI(60*naire) +
ClnfPI(5*naire));
```

```
mesh AireExterior2D = RAEPSup2D + RAEPInf2D + RAEPD2D + RAEPI2D;
```

```
mesh AireTrafo2D = AireExterior2D + AireInterior2D;
```

```
//Malla total 3D
```

```
mesh3 AireTrafo3D = buildlayers(AireTrafo2D,16*ncapas,zbound=[zBT,zAT]);
```

```
////////////////////////////////////
```

```
////////Malla de la Tapadera de aire en 2D y 3D
```

```
////////////////////////////////////
```

```
mesh RTD2D = buildmesh(CDerPTD(60*naire) + CSupPTD(20*naire) + ClzqPTD(60*naire) +
ClnfPTD(20*naire));
```

```
mesh RTI2D = buildmesh(CDerPTI(60*naire) + CSupPTI(20*naire) + ClzqPTI(60*naire) +
ClnfPTI(20*naire));
```

```
mesh RTSup2D = buildmesh(CDerPTSup(18*naire) + CSupPTSup(12*naire) +
ClzqPTSup(18*naire) + ClnfPTSup(12*naire));
```

```
mesh RTInf2D = buildmesh(CDerPTInf(18*naire) + CSupPTInf(12*naire) + ClzqPTInf(18*naire) +
ClnfPTInf(12*naire));
```

```
mesh RTCeD2D = buildmesh(CDerPTDi(24*naire) + CSupPTDi(naire) + ClzqPTDi(24*naire) +
ClnfPTDi(naire));
```

```
mesh RTCeI2D = buildmesh(CDerPTIi(24*naire) + CSupPTIi(naire) + ClzqPTIi(24*naire) +
ClnfPTIi(naire));
```

```
mesh RTCeSup2D = buildmesh(CDerPTCens(5*naire) + CSupPTCens(18*naire) +
ClzqPTCens(5*naire) + ClnfPTCens(18*naire));
```

```
mesh RTCeInf2D = buildmesh(CDerPTCeni(5*naire) + CSupPTCeni(17*naire) +
ClzqPTCeni(5*naire) + ClnfPTCeni(17*naire));
```

```
mesh RTCe2D = buildmesh(CDerPTCen(24*naire) + CSupPTCen(17*naire) +
ClzqPTCen(24*naire) + ClnfPTCen(17*naire));
```

```

//Asigno el valor de las condiciones de contorno a la tapadera exterior

//superior y inferior

int[int] IExtAireUp = [0,50], IExtAireDown = [0,50];

//Tapadera Aire Superior

mesh3 RTDS3D = buildlayers(RTD2D,10*ncapas,zbound=[zAT,zSup],labelup = IExtAireUp);
mesh3 RTIS3D = buildlayers(RTI2D,10*ncapas,zbound=[zAT,zSup],labelup = IExtAireUp);
mesh3 RTSupS3D = buildlayers(RTSup2D,10*ncapas,zbound=[zAT,zSup],labelup = IExtAireUp);
mesh3 RTInfS3D = buildlayers(RTInf2D,10*ncapas,zbound=[zAT,zSup],labelup = IExtAireUp);
mesh3 RTCeDS3D = buildlayers(RTCeD2D,10*ncapas,zbound=[zAT,zSup],labelup = IExtAireUp);
mesh3 RTCeIS3D = buildlayers(RTCeI2D,10*ncapas,zbound=[zAT,zSup],labelup = IExtAireUp);
mesh3 RTCeSupS3D = buildlayers(RTCeSup2D,10*ncapas,zbound=[zCu1TSup,zSup],labelup =
IExtAireUp);
mesh3 RTCeInfS3D = buildlayers(RTCeInf2D,10*ncapas,zbound=[zCu2TSup,zSup],labelup =
IExtAireUp);
mesh3 RTCeS3D = buildlayers(RTCe2D,10*ncapas,zbound=[zAT,zSup],labelup = IExtAireUp);

//Tapadera Aire Inferior

mesh3 RTDI3D = buildlayers(RTD2D,10*ncapas,zbound=[zInf,zBT],labeldown = IExtAireDown);
mesh3 RTII3D = buildlayers(RTI2D,10*ncapas,zbound=[zInf,zBT],labeldown = IExtAireDown);
mesh3 RTSupI3D = buildlayers(RTSup2D,10*ncapas,zbound=[zInf,zBT],labeldown =
IExtAireDown);
mesh3 RTInfI3D = buildlayers(RTInf2D,10*ncapas,zbound=[zInf,zBT],labeldown =
IExtAireDown);
mesh3 RTCeDI3D = buildlayers(RTCeD2D,10*ncapas,zbound=[zInf,zBT],labeldown =
IExtAireDown);
mesh3 RTCeII3D = buildlayers(RTCeI2D,10*ncapas,zbound=[zInf,zBT],labeldown =
IExtAireDown);

```

```

mesh3 RTCeSupI3D = buildlayers(RTCeSup2D,10*ncapas,zbound=[zInf,zCu1TInf],labeldown =
IExtAireDown);

mesh3 RTCeInfI3D = buildlayers(RTCeInf2D,10*ncapas,zbound=[zInf,zCu2TInf],labeldown =
IExtAireDown);

mesh3 RTCeI3D = buildlayers(RTCe2D,10*ncapas,zbound=[zInf,zBT],labeldown =
IExtAireDown);

//Malla total 3D

mesh3 TapaderaSup3D = RTDS3D + RTIS3D + RTSupS3D + RTInfs3D + RTCeDS3D + RTCeIS3D +
RTCeSupS3D + RTCeInfs3D + RTCeS3D;

mesh3 TapaderaInf3D = RTDI3D + RTII3D + RTSupI3D + RTInfi3D + RTCeDI3D + RTCeII3D +
RTCeSupI3D + RTCeInfi3D + RTCeI3D;

plot (TapaderaSup3D, wait = esperar, cmm = "malla tapadera");

////////////////////

////////Mallas finales

////////////////////

//Malla núcleo

plot (Nucleo3D, wait = esperar, cmm = "malla núcleo");

//Malla Cobre

plot (Cobre3D, wait = esperar, cmm = "malla conductor");

//Malla del transformador núcleo

mesh3 NucleoCobre3D = Nucleo3D + Cobre3D;

plot (NucleoCobre3D, wait = esperar, cmm = "malla Transformador");

//Malla aire

int [int] regAire = [0,nRegAire,4,nRegAire,8,nRegAire]; //Unifico el numero de región

mesh3 Aire3D = AireTrafo3D + TapaderaSup3D + TapaderaInf3D;

Aire3D = change(Aire3D,region = regAire);

//plot (Aire3D, wait = esperar, cmm = "malla Aire");

```

```
//Malla del problema

mesh3 Th3D = NucleoCobre3D + Aire3D;

plot (Th3D, wait = esperar, cmm = "malla problema");

/////////////////////////////////DEFINICION DEL ESPACIO DE ELEMENTOS FINITOS/////////////////////////////////

fespace Ch(Th3D,P03d);//Elementos constantes

fespace Bh(Th3D,P13d);//Grado 1

fespace Vh(Th3D,P23d);//Grado 2

Vh ux,vx,uz,vz ;//Vh para el potencial

Bh Bx,By,Bz ;//Bh para el campo magnético

Ch regiones ;//Ch para las regiones, que se identifican por un número entero

/////////////////////////////////DEFINICION DE LAS REGIONES/////////////////////////////////

regiones = region; //Recupero el valor de las regiones en mi variable

//Creo funciones para las distintas regiones

func regionHierro = (regiones == nRegFe);

func regionBobina = (regiones == nRegCu);

func regionAire = (regiones == nRegAire);

plot(regiones, fill = 1, wait = esperar, value = 1, cmm = "regiones");
```

10.6 Anexo VI: Código en FreeFem++ del mallado en 3D extruido a partir del 2D.

```
load "medit";

load "msh3";

load "tetgen";

//*****Parámetros*****//

//////////Todas las medidas en mm //////////

bool esperar = true;

int CondContorno = 50;

//Parámetros del problema

//Conductor 1

real N1 = 54;//Numero de vueltas bobina 1

real d1 = 0.1e-3;

real Sec01 = pi*(d1^2)/4;

real Sec1 = 30*Sec01;//Sección del conductor 1 en milímetros S = 0.2356

real I1 = 0.6;//Intensidad de la bobina 1

real V1 = 500;

//Conductor 2

real N2 = 36;//Numero de vueltas bobina 2

real d2 = 0.1e-3;

real Sec02 = pi*(d2^2)/4;

real Sec2 = 30*Sec02;//Sección del conductor 1 en milímetros

real rt = N1/N2;

real I2 = I1*rt;//Intensidad de la bobina 2

real V2 = V1/rt;
```

```

real Eaisx = 2e-3;//Distancia entre el conductor y el núcleo central

real Eaisy = 2.3e-3;//Distancia entre el conductor y el núcleo superior e inferior

//DATOS PARA LAS REGIONES

int nRegFe=0; //Hierro

int nRegAire=4; //Aire

int nRegCu=8; //Cobre

//Parámetros del mallado

real nHierro = 0.5, nAirefino = 8,nAireIn = 6,nAire = 0.25;

int ncapas = 4;

//*****Parámetros*****//

real x0 = 30e-3;

real x1 = -30e-3;

real x2 = 24.25e-3;

real x3 = -24.25e-3;

real x4 = 18.25e-3;

real x5 = 8.25e-3;

real x6 = -8.25e-3;

real x7 = -18.25e-3;

real x8 = -9.1335e-3;

real x9 = 9.1335e-3;

real x10 = -8.839e-3;

real x11 = 8.839e-3;

real y0 = 30e-3;

real y1 = 24.7e-3;

real y2 = -24.7e-3;

```

```

real y3 = -30e-3;

real y4 = 17.7e-3;

real y5 = 0.25e-3;

real y6 = -0.25e-3;

real y7 = -17.7e-3;

real y8 = 3.36e-3;

real y9 = -3.3e-3;

real z0 = 26e-3;

real z1 = -10e-3;

real z2 = 16e-3;

real z3 = 0e-3;

real z4 = -0.9486e-3;

real z5 = 16.9486e-3;

real z6 = -0.6324e-3;

real z7 = 16.6324e-3;

////////////////////////////////////

//////////////////////////////////// VENTANA DEL CIRCUITO MAGNÉTICO

////////////////////////////////////

// Defino el hueco del interior por tres rectángulos => el izquierdo (Izq), el centro (Cen) y el
// derecho (Der)

// Rectángulo de la derecha

real xIzquierdaDer = x5;

real xDerechaDer = x4;

real ySuperiorDers = y4;

real ySuperiorDeri = y5;

real yInferiorDeri = y7;

```

```

real yInferiorDers = y6;

// Rectangulo del centro
real xlzquierdaCen = x6;
real xDerechaCen = x5;
real ySuperiorCen = y5;
real yInferiorCen = y6;

// Rectángulo de la izquierda
real xlzquierdalzq = x7;
real xDerechalzq = x6;
real ySuperiorlzqs = y4;
real ySuperiorlzqi = y5;
real yInferiorlzqi = y7;
real yInferiorlzqs = y6;

//////////Aire dentro del transformador
//Rectángulo de la Derecha

border CDerRD (t = yInferiorDeri, ySuperiorDers) {x = xDerechaDer; y = t;};
border CSupRD (t = xDerechaDer, xlzquierdaDer) {x = t; y = ySuperiorDers;};
border ClzqRDi (t = yInferiorDers, yInferiorDeri) {x = xlzquierdaDer; y = t;};
border ClzqRDs (t = ySuperiorDers, ySuperiorDeri) {x = xlzquierdaDer; y = t;};
border CInFRD (t = xlzquierdaDer, xDerechaDer) {x = t; y = yInferiorDeri;};

```

//Rectángulo de la Izquierda

border CDerRli (t = yInferiorIzqi, yInferiorIzqs) {x = xDerechalzq; y = t};

border CDerRis (t = ySuperiorIzqi, ySuperiorIzqs) {x = xDerechalzq; y = t};

border CSupRI (t = xDerechalzq, xlzquierdalzq) {x = t; y = ySuperiorIzqs};

border ClzqRI (t = ySuperiorIzqs, yInferiorIzqi) {x = xlzquierdalzq; y = t};

border CInfRI (t = xlzquierdalzq, xDerechalzq) {x = t; y = yInferiorIzqi};

//Rectángulo Central

border CDerRC (t = yInferiorCen, ySuperiorCen) {x = xDerechaCen; y = t};

border CSupRC (t = xDerechaCen, xlzquierdaCen) {x = t; y = ySuperiorCen};

border ClzqRC (t = ySuperiorCen, yInferiorCen) {x = xlzquierdaCen; y = t};

border CInfRC (t = xlzquierdaCen, xDerechaCen) {x = t; y = yInferiorCen};

//Núcleo de hierro

border CDerN (t = y2, y1) {x = x2; y = t};

border CSupN (t = x2, x3) {x = t; y = y1};

border ClzqN (t = y1, y2) {x = x3; y = t};

border CInfN (t = x3, x2) {x = t; y = y2};

//Contorno del problema

border CDerP (t = y3, y0) {x = x0; y = t; label = CondContorno};

border CSupP (t = x0, x1) {x = t; y = y0; label = CondContorno};

border ClzqP (t = y0, y3) {x = x1; y = t; label = CondContorno};

border CInfP (t = x1, x0) {x = t; y = y3; label = CondContorno};

//////////////////// BOBINAS

// Contorno de la Bobina Inferior Derecha (BID)

// l -> límite, izq y der, b -> bobina, i -> inferior, d-> derecha

real lizqbid = x5+Eaisx;

real lderbid = x11+Eaisx;

real lsupbid = y9+Eaisy;

real linfbid = y7+Eaisy;

border CDerBID (t = linfbid, lsupbid) {x = lderbid; y = t};

border CSupBID (t = lderbid, lizqbid) {x = t; y = lsupbid};

border ClzqBID (t = lsupbid, linfbid) {x = lizqbid; y = t};

border CInfBID (t = lizqbid, lderbid) {x = t; y = linfbid};

// Contorno de la Bobina Superior Derecha (BSD)

real lizqbsd = x5+Eaisx;

real lderbsd = x9+Eaisx;

real lsupbsd = y4-Eaisy;

real linfbsd = y8-Eaisy;

border CDerBSD (t = linfbsd, lsupbsd) {x = lderbsd; y = t};

border CSupBSD (t = lderbsd, lizqbsd) {x = t; y = lsupbsd};

border ClzqBSD (t = lsupbsd, linfbsd) {x = lizqbsd; y = t};

border CInfBSD (t = lizqbsd, lderbsd) {x = t; y = linfbsd};

// Contorno de la Bobina Superior Izquierda (BSI)

real lizqbsi = x8-Eaisx;

real lderbsi = x6-Eaisx;

real lsupbsi = y4-Eaisy;

real linfbsi = y8-Eaisy;

```

border CDerBSI (t = linfbsi, lsupbsi) {x = lderbsi; y = t;};

border CSupBSI (t = lderbsi, lizqbsi) {x = t;    y = lsupbsi;};

border ClzqBSI (t = lsupbsi, linfbsi) {x = lizqbsi; y = t;};

border CInfBSI (t = lizqbsi, lderbsi) {x = t;    y = linfbsi;};

// Contorno de la Bobina Inferior Izquierda (BII)

real lizqbii = x10-Eaisx;

real lderbii = x6-Eaisx;

real lsupbii = y9+Eaisy;

real linfbii = y7+Eaisy;

border CDerBII (t = linfbii, lsupbii) {x = lderbii; y = t;};

border CSupBII (t = lderbii, lizqbii) {x = t;    y = lsupbii;};

border ClzqBII (t = lsupbii, linfbii) {x = lizqbii; y = t;};

border CInfBII (t = lizqbii, lderbii) {x = t;    y = linfbii;};

//Mallado del problema 2D

mesh Th2D = buildmesh(CDerP(60*nAire) +CSupP(60*nAire) +ClzqP(60*nAire)
+ClnP(60*nAire)+CDerN(49.4*nHierro) +CSupN(48.5*nHierro) +ClzqN(49.4*nHierro)
+ClnFN(48.5*nHierro)+CDerRD(36.2*nAireIn)+CSupRD(10.0*nAireIn)+ClzqRDs(17.85*nAireIn)
+CSupRC(16.5*nAirefino) +CDerRIs(17.85*nAireIn)+ CSupRI(10.0*nAireIn)
+ ClzqRI(36.2*nAireIn) + ClnfRI(10.0*nAireIn)+CDerRli(17.85*nAireIn)
+ ClnfRC(16.5*nAirefino)+ClzqRDi(17.85*nAireIn) + ClnfRD(10.0*nAireIn)
+CDerBID(5.6916*nAirefino)+CSupBID(0.6324*nAirefino)+ClzqBID(5.6916*nAirefino)
+ClnfBID(0.6324*nAirefino)+CDerBSD(5.6916*nAirefino)+CSupBSD(0.9486*nAirefino)
+ClzqBSD(5.6916*nAirefino)+ClnfBSD(0.9486*nAirefino)+CDerBII(5.6916*nAirefino)
+ CSupBII(0.6324*nAirefino)+ClzqBII(5.6916*nAirefino)+ClnfBII(0.6324*nAirefino)
+CDerBSI(5.6916*nAirefino)+CSupBSI(0.9486*nAirefino)+ClzqBSI(5.6916*nAirefino)
+ClnfBSI(0.9486*nAirefino) );

```

```
plot(Th2D);  
  
////////////////////////////////////  
// Definimos regiones  
////////////////////////////////////  
  
fespace Rh(Th2D,P0); // Espacio para definir ctes  
  
int[int] lExtAireUp=[0,60], lExtAireDown=[0,60];  
  
  
mesh3 Th3D=buildlayers(Th2D,36*ncapas,zbound=[z1,z0],labelup = lExtAireUp,labeldown =  
lExtAireDown); //malla 3D  
  
plot(Th3D, cmm = "Mallado");
```

10.7 Anexo VII: Código en FreeFem++ del problema en 3D.

```

load "medit";

load "msh3";

load "tetgen";

//*****Parámetros*****//

//////////Todas las medidas en mm //////////

bool esperar = true;

int CondContorno = 50;

//Parámetros del problema

//Conductor 1

real N1 = 54;//Numero de vueltas bobina 1

real d1 = 0.1e-3;

real Sec01 = pi*(d1^2)/4;

real Sec1 = 30*Sec01;//Sección del conductor 1 en milímetros S = 0.2356

real I1 = 0.6;//Intensidad de la bobina 1

real V1 = 500;

//Conductor 2

real N2 = 36;//Numero de vueltas bobina 2

real d2 = 0.1e-3;

real Sec02 = pi*(d2^2)/4;

real Sec2 = 30*Sec02;//Sección del conductor 1 en milímetros

real rt = N1/N2;

real I2 = I1*rt;//Intensidad de la bobina 2

real V2 = V1/rt;

real Eaisx = 2e-3;//Distancia entre el conductor y el núcleo central

real Eaisy = 2.3e-3;//Distancia entre el conductor y el núcleo superior e inferior

```

//Parámetros del mallado

real nAire = 2.5;

real ncapas = 4;

//*****Parámetros*****//

real x0 = 30e-3;

real x1 = -30e-3;

real x2 = 24.25e-3;

real x3 = -24.25e-3;

real x4 = 18.25e-3;

real x5 = 8.25e-3;

real x6 = -8.25e-3;

real x7 = -18.25e-3;

real x8 = -9.1335e-3;

real x9 = 9.1335e-3;

real x10 = -8.839e-3;

real x11 = 8.839e-3;

real y0 = 30e-3;

real y1 = 24.7e-3;

real y2 = -24.7e-3;

real y3 = -30e-3;

real y4 = 17.7e-3;

real y5 = 0.25e-3;

real y6 = -0.25e-3;

real y7 = -17.7e-3;

real y8 = 3.36e-3;

real y9 = -3.3e-3;

real z0 = 26e-3;

real z1 = -10e-3;

real z2 = 16e-3;

real z3 = 0e-3;

real z4 = -0.9486e-3;

real z5 = 16.9486e-3;

real z6 = -0.6324e-3;

real z7 = 16.6324e-3;

////////////////////////////////////

//////////////////////////////// VENTANA DEL CIRCUITO MAGNÉTICO

////////////////////////////////////

//Contorno del problema

border CDerP (t = y3, y0) {x = x0; y = t; label = CondContorno;};

border CSupP (t = x0, x1) {x = t; y = y0; label = CondContorno;};

border ClzqP (t = y0, y3) {x = x1; y = t; label = CondContorno;};

border CInfP (t = x1, x0) {x = t; y = y3; label = CondContorno;};

//Mallado del problema 2D

mesh Th2D = buildmesh(CDerP(60*nAire) +CSupP(60*nAire) +ClzqP(60*nAire)
+CInfP(60*nAire));

plot(Th2D);

```
//Mallado del problema 3D

int[int] lExtAireUp=[0,60], lExtAireDown=[0,60]; //Asignamos marcas a la cara superior e
inferior

mesh3 Th3D=buildlayers(Th2D,36*ncapas,zbound=[z1,z0],labelup = lExtAireUp,labeldown =
lExtAireDown); //malla 3D

plot(Th3D, cmm = "Mallado");

////////////////////////////////////

// Espacios de elementos finitos

////////////////////////////////////

fespace Vh(Th3D,P2); // Espacio para calcular el potencial vectorial magnético A

fespace Bh(Th3D,P1); // Espacio para calcular B

fespace Ch(Th3D,P0); // Espacio para definir ctes

// funciones de Vh para cálculo del potencial vectorial A
Vh uz1, vz1, uz2, vz2, uz3, vz3, ux1, vx1, ux2, vx2, ux3, vx3;

// funciones de Bh para el cálculo de B
Bh Bx1, By1, Bz1, Bmodulo1, Bx2, By2, Bz2, muRelativa, Bmodulo2, Bx3, By3, Bz3, Bmodulo3;

//*****PERMEABILIDADES*****//

//Permeabilidad del aire
real mu0 = 4*pi*1e-7;

//Permeabilidad del hierro
real muFe0 = 1630;

func muAire = mu0*(x <= x0)*(x >= x1)*(y <= y0)*(y > y1)*(z <= z0)*(z >= z1)+
```

```

mu0*(x <= x0)*(x >= x1)*(y < y2)*(y >= y3)*(z <= z0)*(z >= z1)+
mu0*(x <= x0)*(x > x2)*(y <= y1)*(y >= y2)*(z <= z0)*(z >= z1)+
mu0*(x < x3)*(x >= x1)*(y <= y1)*(y >= y2)*(z <= z0)*(z >= z1)+
mu0*(x < x4)*(x > x5)*(y < y4)*(y > y7)*(z <= z2)*(z >= z3)+
mu0*(x < x6)*(x > x7)*(y < y4)*(y > y7)*(z <= z2)*(z >= z3)+
mu0*(x <= x5)*(x >= x6)*(y < y5)*(y > y6)*(z <= z2)*(z >= z3)+
mu0*(x <= x2)*(x >= x3)*(y <= y1)*(y >= y2)*(z <= z0)*(z > z2)+
mu0*(x <= x2)*(x >= x3)*(y <= y1)*(y >= y2)*(z < z3)*(z >= z1);

```

```

func muFe = muFe0*mu0*(x <= x2)*(x >= x3)*(y <= y1)*(y >= y4)*(z <= z2)*(z >= z3)+
muFe0*mu0*(x <= x2)*(x >= x3)*(y <= y7)*(y >= y2)*(z <= z2)*(z >= z3)+
muFe0*mu0*(x <= x2)*(x >= x4)*(y < y4)*(y > y7)*(z <= z2)*(z >= z3)+
muFe0*mu0*(x <= x7)*(x >= x3)*(y < y4)*(y > y7)*(z <= z2)*(z >= z3)+
muFe0*mu0*(x <= x5)*(x >= x6)*(y < y4)*(y >= y5)*(z <= z2)*(z >= z3)+
muFe0*mu0*(x <= x5)*(x >= x6)*(y <= y6)*(y > y7)*(z <= z2)*(z >= z3);

```

```

func muT = muAire + muFe;

```

```

Ch mu = muT;

```

```

//////////DEFINICION DE LA DENSIDAD DE CORRIENTE//////////

```

```

//////////Trabajamos con valores medios

```

```

real AmpVueltaCu1 = I1*N1; //Valor de amperios-vuelta conductor 1

```

```

real AmpVueltaCu2 = I2*N2; //Valor de amperios-vuelta conductor 2

```

```

real DensidadCorrienteCu1 = I1/(Sec1); //Módulo de la densidad de corriente conductor 1

```

```

real DensidadCorrienteCu2 = I2/(Sec2); //Módulo de la densidad de corriente conductor 2

```

```
cout<<"Valor de la densidad corriente->"<<DensidadCorrienteCu1<<endl;
```

```
cout<<"Valor de la densidad corriente->"<<DensidadCorrienteCu2<<endl;
```

```
//Caso 1 I2=0.
```

```
real I11 = 0.6;
```

```
func jz1 = ((I11/(Sec1))*(x <= x6-Eaisx)*(x >= x8-Eaisx)*(y <= y4-Eaisy)*(y >= y8-Eaisy)*(z <= z2+Eaisx)*(z >= z3-Eaisx)
```

```
+(-1)*(I11/(Sec1))*(x <= x9+Eaisx)*(x >= x5+Eaisx)*(y <= y4-Eaisy)*(y >= y8-Eaisy)*(z <= z2+Eaisx)*(z >= z3-Eaisx));
```

```
func jx1 = ((I11/(Sec1))*(x <= x9+Eaisx)*(x >= x8-Eaisx)*(y <= y4-Eaisy)*(y >= y8-Eaisy)*(z <= z5+Eaisx)*(z >= z2+Eaisx)
```

```
+(-1)*(I11/(Sec1))*(x <= x9+Eaisx)*(x >= x8-Eaisx)*(y <= y4-Eaisy)*(y >= y8-Eaisy)*(z <= z3-Eaisx)*(z >= z4-Eaisx));
```

```
//Caso 2 I1=0.
```

```
real I22 = 0.9;
```

```
func jz2 = ((I22/(Sec2))*(x <= x6-Eaisx)*(x >= x10-Eaisx)*(y <= y9+Eaisy)*(y >= y7+Eaisy)*(z <= z2+Eaisx)*(z >= z3-Eaisx)
```

```
+(-1)*(I22/(Sec2))*(x <= x11+Eaisx)*(x >= x5+Eaisx)*(y <= y9+Eaisy)*(y >= y7+Eaisy)*(z <= z2+Eaisx)*(z >= z3-Eaisx));
```

```
func jx2 = ((I22/(Sec2))*(x <= x11+Eaisx)*(x >= x10-Eaisx)*(y <= y9+Eaisy)*(y >= y7+Eaisy)*(z <= z7+Eaisx)*(z >= z2+Eaisx)
```

```
+(-1)*(I22/(Sec2))*(x <= x11+Eaisx)*(x >= x10-Eaisx)*(y <= y9+Eaisy)*(y >= y7+Eaisy)*(z <= z3-Eaisx)*(z >= z6-Eaisx));
```

```
//Caso 3 I1 e I2 distintos de 0.
```

```
func jz3 =jz1 + jz2;
```

```
func jx3 = jx1 + jx2;
```

Ch jh = jx3 + jz3;

plot(mu, fill = 1, wait = esperar, value = 1, cmm = "Permeabilidad");

plot(jh, fill = 1, wait = esperar, value = 1, cmm = "densidades de corriente");

////////////////////////////////////

////////////////////////////////////

// Resolvemos el problema Caso 1

////////////////////////////////////

////////////////////////////////////

macro Grad3z(u) [dx(u),dy(u)] // EOM, gradiente

problem magnetostatica1z(uz1,vz1,solver=CG) = int3d(Th3D)((1/(muT))*(Grad3z(vz1)'
*Grad3z(uz1))) -int3d(Th3D)(jz1*vz1) + on(CondContorno, uz1 = 0);

magnetostatica1z;

macro Grad3x(u) [dy(u),dz(u)] // EOM

problem magnetostatica1x(ux1,vx1,solver=CG) = int3d(Th3D)((1/(muT))*(Grad3x(vx1)'
*Grad3x(ux1))) - int3d(Th3D)(jx1*vx1) + on(CondContorno,60,ux1 = 0);

magnetostatica1x;

// Calculo de las componentes de B

Bx1 = dy(uz1);

By1 = -dx(uz1)+dz(ux1);

Bz1 = -dy(ux1);

Bmodulo1 = sqrt(Bx1^2 + By1^2 + Bz1^2);

```

////////////////////////////////////
// Representación gráfica de la solución
////////////////////////////////////
// Potencial vector (componente z)
plot(uz1, cmm ="Caso 1. Az (lineas de campo de B)");
plot(ux1, cmm ="Caso 1. Ax (lineas de campo de B)");

// Densidad de flujo

plot([Bx1,By1,Bz1], value = true, coef = -1.6, wait = esperar, cmm = "Caso 1. Vector de
densidad de flujo B");

plot(Bmodulo1, value = true , wait = esperar, cmm ="Caso 1. Densidad de flujo B");

// Intensidad de campo magnético

Bh Hx1, Hy1, Hz1;

Hx1 = Bx1/(mu);

Hy1 = By1/(mu);

Hz1 = Bz1/(mu);

//Calculo de la inductancia L1

real L1 = int3d(Th3D) ((Bx1*Hx1+By1*Hy1+Bz1*Hz1)/(l11^2));

cout << "Inductancia L1: " << L1 << endl;

////////////////////////////////////
////////////////////////////////////
// Resolvemos el problema Caso 2
////////////////////////////////////
////////////////////////////////////

```

```
problem magnetostaticaz2(uz2,vz2,solver=CG) = int3d(Th3D)((1/(muT))*(Grad3z(vz2)'
*Grad3z(uz2))) -int3d(Th3D)(jz2*vz2) +on(CondContorno, uz2 = 0);
```

```
magnetostaticaz2;
```

```
problem magnetostatica2x(ux2,vx2,solver=CG) = int3d(Th3D)((1/(muT))*(Grad3x(vx2)'
*Grad3x(ux2))) -int3d(Th3D)(jx2*vx2) +on(CondContorno,60,ux2 = 0);
```

```
magnetostatica2x;
```

```
// Calculo de las componentes de B
```

```
Bx2 = dy(uz2);
```

```
By2 = -dx(uz2)+dz(ux2);
```

```
Bz2 = -dy(ux2);
```

```
Bmodulo2 = sqrt(Bx2^2 + By2^2 + Bz2^2);
```

```
////////////////////////////////////
```

```
// Representación gráfica de la solución
```

```
////////////////////////////////////
```

```
// Potencial vector (componente z)
```

```
plot(uz2, cmm ="Caso 2. Az (lineas de campo de B)");
```

```
plot(ux2, cmm ="Caso 2. Ax (lineas de campo de B)");
```

```
// Densidad de flujo
```

```
plot([Bx2,By2,Bz2], value = true, coef = -1.6, wait = esperar, cmm = "Caso 2. Vector de
densidad de flujo B");
```

```
plot(Bmodulo2, value = true , wait = esperar, cmm ="Caso 2. Densidad de flujo B");
```

```
// Intensidad de campo magnético
```

```
Bh Hx2, Hy2, Hz2;
```

Hx2 = Bx2/(mu);

Hy2 = By2/(mu);

Hx2 = Bz2/(mu);

//Calculo de la inductancia L2

real L2 = int3d(Th3D) ((Bx2*Hx2+By2*Hy2+Bz2*Hz2)/(l2^2));

cout << "Inductancia L2: " << L2 << endl;

//Calculo de la inductancia L2

real M12 = int3d(Th3D) ((Bx1*Hx2+By1*Hy2+Bz1*Hz2)/(l1*l2));

cout << "Inductancia M12: " << M12 << endl;

////////////////////

//Por ultimo calculamos la inductancia observado desde el primario

////////////////////

//Primero calculamos L1l

real L1l = L1 - M12*rt;

cout << "Inductancia L1l: " << L1l << endl;

//Segundo calculamos L2l

real L2l = L2 - M12/rt;

cout << "Inductancia L2l: " << L2l << endl;

//Por últimos calculamos la inductancia final

real Lcc = L1l + (L2l*((rt)^2));

cout << "Inductancia Lcc: " << Lcc << endl;