

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA

**PROGRAMA DE MAESTRÍA Y DOCTORADO EN CIENCIAS E
INGENIERÍA**

ÁREA ELÉCTRICA



**DESARROLLO DE UN SOFTWARE DE ELEMENTOS FINITOS EN 2D
APLICADO AL ELECTROMAGNETISMO USANDO GPU Y CUDA C**

TESIS

**que para cubrir parcialmente los requisitos para obtener el grado de MAESTRO EN
CIENCIAS presenta:**

José Ángel Beltrán Pérez

**Director de Tesis:
Dr. Juan Mauricio Díaz Chacón**

Mexicali, Baja California, México, agosto 2021

RESUMEN de la tesis de José Ángel Beltrán Pérez, presentada como requisito parcial para la obtención del grado de MAESTRO EN CIENCIAS. Mexicali, Baja California, México. Agosto de 2021.

DESARROLLO DE UN SOFTWARE DE ELEMENTO FINITO EN 2D APLICADO AL ELECTROMAGNETISMO USANDO GPU Y CUDA C

Resumen aprobado por:

Dr. Juan Mauricio Díaz Chacón
Director de tesis

En esta tesis se desarrolla un Software de Elementos Finitos en 2D (EF-2D) mediante el cual se pueden realizar simulaciones de diversos modelos electromagnéticos usando GPU y CUDA C. Para lograr dicho objetivo, se generan mallas de estos modelos, a través del software libre Gmsh, cuyos parámetros son descritos en archivos de texto para su uso posterior. El software desarrollado es capaz de incorporar las características físicas y las condiciones de frontera respectivas de cada modelo. A su vez, en este software se implementan diversas formulaciones electromagnéticas en 2D. Mediante el uso del lenguaje de programación CUDA C y de hardware GPU, se realiza la resolución del sistema de ecuaciones matricial de los modelos. A su vez, se realiza el desarrollo de una interfaz gráfica, la cual tiene integrado las formulaciones implementadas para interactuar con el usuario final. Este software es capaz de generar archivos de texto derivados de la etapa de post-procesamiento para su visualización en el software libre Gmsh. Después de esto, se puede realizar el despliegue gráfico de los resultados obtenidos de este software usando el software libre Octave. Finalmente, se lleva a cabo una etapa de validación y desempeño del software desarrollado empleando diversas soluciones analíticas, tomadas de la literatura eléctrica.

Palabras clave: Método de los Elementos Finitos, modelos electromagnéticos, software libre, GPU's.

ABSTRACT of the thesis, presented by José Ángel Beltrán Pérez, in order to obtain the degree of MASTER IN SCIENCES. Mexicali, Baja California, Mexico. August, 2021.

DEVELOPMENT OF A 2D FINITE ELEMENT SOFTWARE APPLIED TO ELECTROMAGNETISM USING GPU AND CUDA C

Approved by:

Dr. Juan Mauricio Díaz Chacón

Thesis advisor

In this thesis, a 2D Finite Element software (FE-2D) is developed through which simulations of various electromagnetic models can be carried out using GPU and CUDA C. To achieve this objective, meshes of these models are generated through the free Gmsh software, whose parameters are described in text files for later use. The software developed is capable of incorporating the physical characteristics and the respective boundary conditions of each model. At the same time, various 2D electromagnetic formulations are implemented in this software. Through the use of the CUDA C programming language and GPU hardware, the resolution of the matrix system of equations of the models is performed. At the same time, the development of a graphical interface is carried out, which has integrated the formulations implemented to interact with the end user. This software is capable of generating text files derived from the post-processing stage for viewing in free Gmsh software. After this, the graphical display of the results obtained from this software can be made using the free Octave software. Finally, a stage of validation and performance of developed software is carried out using various analytical solutions, taken from the electrical literature.

Keywords: Finite Element Method, electromagnetic models, free software, GPU's.

Dedicatoria

A mis padres Amilcar Beltran Villa y Bertha Lidia Perez Figuerca por su apoyo incondicional en cada etapa de mi vida.

A mis abuelos Juan José Perez Inda y María Lidia Figuerca Salazar por su apoyo y consejos a lo largo de cada etapa de mi vida.

Agradecimientos

Al Dr. Juan Mauricio Díaz Chacón por su dirección en esta tesis y por brindar su conocimiento, disponibilidad y amistad.

A los miembros del comité de tesis Dr. César Amaro Hernández y Dr. Pedro Francisco Rosales Escobedo por el apoyo y consejos en el trayecto de mis estudios de maestría.

Al Consejo Nacional de la Ciencia y Tecnología (CONACYT) por el apoyo otorgado a través de la beca de estudios de posgrado.

Publicaciones realizadas

Como resultado de esta tesis se publicaron los siguientes artículos:

1. J. A. Beltrán Pérez, J. M. Díaz Chacón, C. Amaro Hernández, A. Navarro Valle, P. F. Rosales Escobedo, “Simulación Electroestática de un Tanque con Líquido Cargado mediante un Software de Elementos Finitos en 2D”, *Revista Aristas: Investigación Básica y Aplicada*, vol. 8, no. 16, pp. 318-324, 2021, ISSN: 2007-9478.
2. J. A. Beltrán Pérez, J. M. Díaz Chacón, C. Amaro Hernández, A. Navarro Valle, P. F. Rosales Escobedo, “Aceleración en la Resolución de Modelos Transitorios de Transferencia de Calor mediante un Software de Elementos Finitos en 2D y CUDA”, *Revista Ciencia, Ingeniería y Desarrollo Tec Lerdo*, vol. 1, no. 7, pp. 87-92, 2021, ISSN: 2448-623X.

Además, se realizó una colaboración en el siguiente artículo:

1. J. M. Díaz Chacón, J. A. Beltrán Pérez, C. Amaro Hernández, A. Navarro Valle, P. F. Rosales Escobedo, “Simulación Electromagnética del Modelo de Validación TEAM 7 usando la formulación T, phi-phi”, *Revista Ciencia, Ingeniería y Desarrollo Tec Lerdo*, vol. 1, no. 7, pp. 93-98, 2021, ISSN: 2448-623X.

Índice

1. Introducción	1
1.1 <i>Antecedentes</i>	1
1.2 <i>Estado del arte</i>	2
1.3 <i>Objetivos de la investigación</i>	3
1.3.1 <i>Objetivo general</i>	3
1.3.2 <i>Objetivos específicos</i>	3
1.4 <i>Justificación</i>	3
1.5 <i>Hipótesis</i>	4
1.6 <i>Organización de la tesis</i>	5
2. Implementación del Método de los Elementos Finitos para Determinar una Solución Aproximada de Modelos Electromagnéticos	6
2.1 <i>Método de los Elementos Finitos</i>	6
2.2 <i>Ecuaciones de Maxwell</i>	8
2.3 <i>Formulación magnetostática</i>	9
2.4 <i>Formulación Electrostática</i>	11
2.4.1 <i>Formulación Laplace</i>	11
2.5 <i>Formulación magnetodinámica</i>	12
2.5.1 <i>Transitoria</i>	12
2.5.2 <i>Compleja</i>	13
2.6 <i>Formulación de transferencia de calor</i>	13
2.7 <i>Almacenamiento de matriz en formato COO (Coordinate)</i>	14
2.8 <i>Almacenamiento de matriz en formato CSR (Compressed Storage Row)</i>	15
2.9 <i>Solver INTEL MKL PARDISO</i>	15
3. Aceleración en la resolución de modelos electromagnéticos mediante CUDA	18
3.1 <i>CUDA: Software especializado en paralelismo</i>	18
3.2 <i>Librerías de resolución de ecuaciones CUDA</i>	20
3.2.1 <i>CUSPARSE</i>	20
3.2.2 <i>CUBLAS</i>	21
3.3 <i>Solver ICCG utilizando librerías de CUDA</i>	22
3.4 <i>wxWidgets: Software utilizado para la creación de una interfaz grafica</i>	23
4. Simulación de modelos electromagnéticos mediante el software desarrollado de Elementos Finitos en 2D	25

4.1 Simulación de modelos magnetostáticos.....	25
4.1.1 Caso 1	25
4.1.2 Caso 2	33
4.2 Simulación de modelos electrostáticos	38
4.2.1 Caso 1	38
4.2.2 Caso 2	41
4.2.3 Caso 3	45
4.3 Simulación de modelos magnetodinámicos.....	53
4.3.1 Caso 1	53
4.4 Simulación de modelos electromagnéticos transitorios.....	56
4.4.1 Caso 1	56
4.5 Simulación de modelos de conducción de calor	60
4.5.1 Caso 1	60
4.5.2 Caso 2	65
4.5.3 Caso 3	71
5. Conclusiones y trabajo futuro	76
5.1 Conclusiones	76
5.2 Trabajo futuro	77
Referencias.....	78

Índice de figuras

2.1.	Malla de elementos finitos de una figura geometría cuadrada en 2D.....	7
2.2.	Elemento cuadrilátero y triangulo de segundo orden utilizados en el mallado del MEF.....	7
3.1.	Estructura de código con sección secuencial y paralela.....	19
3.2.	Jerarquía en una aplicación CUDA.....	19
3.3.	Interfaz gráfica de software de elementos finitos desarrollada para la interacción con el usuario final.....	24
4.1.	Geometría de bloque y barra conductora, dada en mm.....	27
4.2.	Malla de modelo barra conductora conformada por elementos triangulares.....	27
4.3.	Acercamiento de malla del modelo barra conductora para mejorar visualización de elementos triangulares.....	28
4.4.	Malla de barra conductora conformada por elementos cuadriláteros.....	28
4.5.	Acercamiento de malla de modelo barra conductora para mejorar visualización de elementos cuadriláteros.....	29
4.6.	Distribución de los potenciales eléctricos (V) en bloque y barra conductora, obtenidos con elementos triangulares y solver INTEL MKL PARDISO.....	29
4.7.	Distribución de la magnitud de campo eléctrico (V/m) en bloque y barra conductora, obtenidos con elementos triangulares y solver INTEL MKL PARDISO	30
4.8.	Distribución de los potenciales eléctricos (V) en bloque y barra conductora, obtenidos con elementos triangulares y solver ICCG CUDA.....	30
4.9.	Distribución de la magnitud del campo eléctrico (V/m) en bloque y barra conductora, obtenidos con elementos triangulares y solver ICCG CUDA.....	31
4.10.	Distribución de los potenciales eléctricos (V) en bloque y barra conductora, obtenido con elementos cuadriláteros y solver INTEL MKL PARDISO.....	31
4.11.	Distribución de la magnitud del campo eléctrico (V/m) en bloque y barra conductora, obtenido con elementos cuadriláteros y solver INTEL MKL PARDISO	32
4.12.	Distribución de los potenciales eléctricos (V) en bloque y barra conductora, obtenido con elementos cuadriláteros y solver ICCG CUDA	32
4.13.	Distribución de la magnitud del campo eléctrico (V/m) en bloque y barra conductora, obtenido con elementos cuadriláteros y solver ICCG CUDA.....	33
4.14.	Geometría del conductor coaxial, dada en m	35
4.15.	Malla del conductor coaxial conformada por elementos triangulares.....	35
4.16.	Distribución de los potenciales eléctricos (V) en conductor coaxial.....	36
4.17.	Distribución del campo magnético (A/m) en conductor coaxial.....	36

4.18.	Comparación entre los valores de la solución analítica y el software de elementos finitos, desde la línea recta del conductor	37
4.19.	Porcentaje de error entre los valores de la solución analítica y el software de elementos finitos	37
4.20.	Dimensiones geométricas de modelo electrostático, dado en mm.....	39
4.21.	Malla de modelo electrostático conformada por elementos triangulares.....	39
4.22.	Acercamiento de malla para visualización de los elementos triangulares que lo conforman	40
4.23.	Distribución de los potenciales eléctricos (V) en modelo electrostático.....	40
4.24.	Distribución de la magnitud del campo eléctrico (V/m) en modelo electrostático....	41
4.25.	Dimensiones geométricas de conductor, dada en m	42
4.26.	Malla de modelo conductor conformada por elementos triangulares.....	43
4.27.	Distribución de los potenciales eléctricos (V) en conductor, obtenidas por software de elementos finitos	43
4.28.	Distribución de los potenciales eléctricos (V) en conductor, obtenida por solución analítica	44
4.29.	Comparación entre los potenciales obtenidos en la solución analítica y el software de elementos finitos	44
4.30.	Porcentaje de error entre los potenciales de la solución analítica y el software de elementos finitos.....	45
4.31.	Dimensiones geométricas de tanque aterrizado que contiene un líquido con cargas electrostáticas, dada en m	48
4.32.	Malla de tanque aterrizado conformado por elementos triangulares.....	48
4.33.	Distribución de los potenciales eléctricos (V) en tanque aterrizado (simulación)....	49
4.34.	Distribución de la magnitud del campo eléctrico (V/m) en tanque aterrizado (simulación).....	49
4.35.	Distribución de los potenciales eléctricos (V) en tanque aterrizado (Solución analítica).....	50
4.36.	Distribución de la magnitud del campo eléctrico (V/m) en tanque aterrizado (Solución analítica).....	50
4.37.	Comparación de la distribución de los potenciales eléctricos escalares a lo largo de $x = 0.4m, 0 \leq y \leq 2m$	51
4.38.	Comparación de la distribución de potenciales eléctricos escalares en la interface gas-líquido.....	51
4.39.	Porcentaje de error entre los potenciales eléctricos obtenidos en la simulación y la solución analítica, a lo largo de $x = 0.4m, 0 \leq y \leq 2m$	52
4.40.	Porcentaje de error entre los potenciales eléctricos obtenidos en la simulación y la solución analítica en la interface gas- líquido.....	52
4.41.	Geometría del bloque de cobre en el entrehierro del circuito magnético, dada en cm.....	54
4.42.	Malla del circuito magnético con elementos triangulares.....	54
4.43.	Visualización de elementos triangulares que conforman la malla del modelo circuito magnético.....	55

4.44.	Distribución de los potenciales eléctricos (V) en el circuito magnético.....	55
4.45.	Distribución de la magnitud del campo eléctrico (V/m) en el circuito magnético....	56
4.46.	Dimensiones geométricas de modelo transitorio, dado en mm.....	57
4.47.	Malla de modelo transitorio con elementos triangulares en 2D.....	58
4.48.	Distribución del potencial magnético vectorial en $1\mu s$	58
4.49.	Distribución del potencial magnético vectorial en $20\mu s$	59
4.50.	Distribución del potencial magnético vectorial.....	59
4.51.	Dimensiones geométricas de conductor (caso 1), dada en m.....	61
4.52.	Malla del conductor (caso 1) conformada por elementos triangulares.....	62
4.53.	Distribución de temperatura ($^{\circ}F$) en conductor (caso 1), en 5 ms.....	62
4.54.	Distribución de temperatura ($^{\circ}F$) en conductor (caso 1), en 0.1 s.....	63
4.55.	Distribución de temperatura ($^{\circ}F$) en conductor (caso 1).....	63
4.56.	Distribución de temperatura ($^{\circ}F$) en conductor (caso 1), obtenido en solución analítica.....	64
4.57.	Comparación entre temperaturas obtenidas en solución analítica y simulación (caso 1).....	64
4.58.	Porcentaje de error entre resultados de solución analítica y simulación (caso 1)....	65
4.59.	Dimensiones geométricas de conductor (caso 2), dada en m.....	67
4.60.	Malla del conductor (caso 2) conformada por elementos triangulares.....	67
4.61.	Distribución de temperatura ($^{\circ}F$) en conductor (caso 2), en 5 ms.....	68
4.62.	Distribución de temperatura ($^{\circ}F$) en conductor (caso 2), en 0.1 s.....	68
4.63.	Distribución de temperatura ($^{\circ}F$) en conductor (caso 2).....	69
4.64.	Distribución de temperatura ($^{\circ}F$) en conductor (caso 2), obtenido en solución analítica.....	69
4.65.	Comparación entre temperaturas obtenidas en solución analítica y simulación (caso 2).....	70
4.66.	Porcentaje de error entre resultados de solución analítica y simulación (caso 2)....	70
4.67.	Dimensiones geométricas de conductor (caso 3), dada en m.....	72
4.68.	Malla del conductor (caso 3) conformada por elementos triangulares.....	73
4.69.	Distribución de temperatura ($^{\circ}F$) en conductor (caso 3), en 3.125 ms.....	73
4.70.	Distribución de temperatura ($^{\circ}F$) en conductor (caso 3), en 0.0625 s.....	74
4.71.	Distribución de temperatura ($^{\circ}F$) en conductor (caso 3).....	74
4.72.	Distribución de temperatura ($^{\circ}F$) en conductor (caso 3), obtenido en solución analítica.....	75
4.73.	Comparación entre temperaturas obtenidas en solución analítica y simulación (caso 3).....	75
4.74.	Porcentaje de error entre resultados de solución analítica y simulación (caso 3)....	76

Índice de tablas

4.1.	Tiempo requerido de simulación de modelo barra conductora, utilizando elementos triangulares.....	26
4.2.	Tiempo requerido de simulación de modelo barra conductora, utilizando elementos cuadriláteros.....	26
4.3.	Tiempo requerido de simulación de modelo cable coaxial, utilizando elementos triangulares	34
4.4.	Tiempo requerido de simulación de modelo cable coaxial, utilizando elementos cuadriláteros	34
4.5.	Tiempo requerido de simulación de modelo electrostático, utilizando elementos triangulares	38
4.6.	Tiempo de solución requerido de modelo electrostático, utilizando elementos cuadriláteros	38
4.7.	Tiempo requerido de simulación de modelo conductor, utilizando elementos triangulares	42
4.8.	Tiempo requerido de simulación de modelo conductor, utilizando elementos triangulares	42
4.9.	Tiempo requerido de simulación de modelo tanque, utilizando elementos triangulares.....	47
4.10.	Tiempo requerido de simulación de modelo tanque, utilizando elementos cuadriláteros	47
4.11.	Tiempo requerido de simulación de modelo circuito magnético, utilizando elementos triangulares.....	53
4.12.	Tiempo requerido de simulación de modelo circuito magnético, utilizando elementos cuadriláteros.....	53
4.13.	Tiempo requerido de simulación de modelo transitorio, utilizando elementos triangulares.....	57
4.14.	Tiempo requerido de simulación de modelo transitorio, utilizando elementos cuadriláteros.....	57
4.15.	Tiempo requerido de simulación de conductor (caso 1), utilizando elementos triangulares.....	61
4.16.	Tiempo requerido de simulación de conductor (caso 1), utilizando elementos cuadriláteros.....	61
4.17.	Tiempo requerido de simulación de conductor (caso 2), utilizando elementos triangulares.....	66
4.18.	Tiempo requerido de simulación de conductor (caso 2), utilizando elementos cuadriláteros.....	66
4.19.	Tiempo requerido de simulación de conductor (caso 3), utilizando elementos triangulares.....	71
4.20.	Tiempo requerido de simulación de conductor (caso 3), utilizando elementos cuadriláteros.....	72

Capítulo 1

Introducción

1.1 Antecedentes

En 1943, R. Courant dio origen al Método de los Elementos Finitos (MEF) en una publicación enfocada en el área de la mecánica estructural [1]. En la publicación, se realiza la solución de problemas de torsiones aplicando interpolación polinomial mediante el uso de elementos triangulares.

En 1956, L. C. Turner realizó el primer artículo científico en el área de la aeronáutica [2], en el cual emplea el MEF para el análisis estructural en dos dimensiones (2D).

En 1960, R. W. Clough fue el primero en utilizar el término Elemento Finito [3]. Además, el MEF era aplicado en deformaciones y problemas no lineales.

En 1965, O. C. Zienkiewicz y Y. K. Cheung utilizaron por primera vez el MEF para aplicaciones no estructurales como flujo de fluidos, transferencia de calor, entre otros [4].

En esa misma década, la mayoría de los autores empleaban métodos analíticos para aplicarlos en diseños reales. Sin embargo, eran sumamente complejos aplicarlos tanto en las características de los materiales de fabricación que lo componen como en sus dimensiones geométricas. Con la llegada de equipos computacionales, la NASA desarrolló el software Nastran donde fueron implementadas diversas técnicas computacionales [5]. En 1967, se realizó la publicación del primer libro del MEF aplicado a ingeniería civil [6].

En 1968, S. Ahmed realizó la publicación del primer artículo del MEF aplicado en electromagnetismo [7]. Este artículo sirvió como referencia para comprobar que este método sería una herramienta fundamental para el análisis de fenómenos electromagnéticos [8].

En 1969, P. Silvester realizó la publicación de diversos trabajos del MEF en electromagnetismo para bajas y altas frecuencias [9]-[10].

En 1970, surgió el software comercial de Elementos Finitos llamado ANSYS para aplicaciones en diversas áreas de la ingeniería [11].

En 1992, la compañía Silicon Graphics desarrolló la librería OpenGL utilizado en el área de diseño asistido por computadora [12].

En 1998, fue desarrollado el software COMSOL para el modelado numérico de diversos problemas físicos y de ingeniería [13].

En 2001, la compañía NVIDIA realizó el lanzamiento de la tarjeta gráfica GeForce 3 siendo un avance importante en la tecnología de GPU para el cómputo paralelo [14]. Además, en 2006 introdujo la plataforma de computación paralela CUDA con la finalidad de programar las GPU para la solución de problemas computacionales complejos de forma más eficiente en las CPU [15].

1.2 Estado del arte

En el sector industrial ha evolucionado constantemente la aplicación de métodos numéricos para el diseño eficiente de máquinas eléctricas. Con la finalidad de reducir la cantidad de prototipos fallidos y/o la realización de un gran número de pruebas empíricas, las empresas fabricantes de equipo eléctrico, desarrollan simulaciones para modelar y analizar diversos equipos eléctricos tales como motores, transformadores, generadores, entre otros, para lograr la mayor eficiencia y calidad posible. Y a su vez, reducir los costos de manufactura.

Una de las técnicas numéricas computacionales modernas empleadas es el Método de los Elementos Finitos (MEF) para desarrollar el análisis electromagnético de los mismos, antes de su construcción o prueba [16]-[18]. Para la implementación del MEF es requerida la aplicación de formulaciones o modelos matemáticos aplicados al modelado electromagnético en bajas o altas frecuencias. El uso de este método permite analizar dispositivos electromagnéticos, considerando sus dimensiones geométricas y las características de sus materiales de construcción.

En [19], J. Smajic utilizó elementos finitos para el análisis del acoplamiento electromagnético-mecánico en disyuntores de un generador, concluyendo con la validación del modelo al obtener resultados precisos en la simulación.

En [20], J. Li, P. Liu y V. Dinavahi utilizaron el MEF para implementar el método de transmisión lineal que es ampliamente aplicado en aparatos electromagnéticos, concluyendo con las ventajas obtenidas en la aplicación de este método.

Sin embargo, una de las principales desventajas del MEF es el elevado tiempo de solución requerido para su ejecución. Por tal motivo, es necesario contar con recursos de cómputo de alto desempeño. Esto impide su aplicación en procesos de optimización de dispositivos electromagnéticos, los cuales requieren la evaluación de un gran número de casos para obtener un modelo finalizado.

Para mitigar esta desventaja, se han empleado metodologías de cómputo paralelo en el MEF con el objetivo de reducir considerablemente el tiempo requerido de solución. Una de las metodologías, utilizadas en la industria, es el desarrollo de software basado en hardware GPU's (Graphics Processing Unit) mediante el lenguaje de programación CUDA (Compute Unified Device Architecture) [21]-[24].

En la actualidad, existen diversos software's comerciales para el análisis de fenómenos electromagnéticos utilizando el MEF [25]. Sin embargo, la adquisición de su licencia conlleva un alto costo monetario. Como alternativa, existen diversos software's de licencia libre para realizar el mismo tipo de análisis, entre los cuales son: ELMER [26], SALOME [27], Agros2D[28], ONELAB [29], etc.

Recientemente se ha presentado una tendencia por el desarrollo personalizado de software de Elementos Finitos [30]-[31], en el cual, se pueden emplear diversas metodologías o técnicas de vanguardia de cómputo acelerado para el análisis de modelos electromagnéticos.

1.3 Objetivos de la investigación

1.3.1 Objetivo general

El objetivo general de esta investigación consiste en desarrollar un software de elementos finitos en 2D mediante el uso de GPU y programación paralela en lenguaje CUDA C.

1.3.2 Objetivos específicos

1. Realizar una búsqueda exhaustiva del estado del arte del Método de los Elementos Finitos en 2D (MEF-2D) y del uso de GPU's para el desarrollo de aplicaciones de cómputo paralelo en electromagnetismo.
2. Implementar formulaciones electromagnéticas del MEF-2D usando lenguaje C++ (Microsoft Visual Studio 2017 e Intel Parallel 2018).
3. Implementar formulaciones electromagnéticas del MEF-2D usando el lenguaje CUDA C (librerías CuSPARSE y/o CuBLAS).
4. Desarrollar una interfaz gráfica para llevar a cabo la interacción con el usuario final.
5. Desarrollar un software, conformado por una interfaz gráfica y las formulaciones electromagnéticas del MEF-2D implementadas.
6. Evaluar y validar el desempeño (en términos de aceleración computacional) del software desarrollado mediante la simulación de modelos analíticos, tomados de la literatura eléctrica.

1.4 Justificación

En la actualidad, la generación de energía eléctrica es uno de los temas de mayor prioridad para el desarrollo económico del país. Debido, a que es un recurso fundamental que es utilizado en todos los sectores para el desarrollo de actividades y procesos.

Con la promulgación de la Reforma Energética en México, se busca la modernización del sector energético que lo conforman las empresas públicas y privadas dedicadas a la producción de energía eléctrica por medio de fuentes de generación como geotérmicas, hidráulicas, nucleares, entre otras. Además, del desarrollo de proyectos de energías renovables para la implementación de fuentes de generación como eólica, fotovoltaica, entre otros. Con la finalidad de disminuir el impacto ambiental producido por las fuentes de generación convencionales. Esto, favorece la generación de nuevos empleos y personal especializado, con lo cual, elevará la calidad de vida de las familias mexicanas.

Los generadores y transformadores son equipos eléctricos que primordialmente forman parte de los sistemas de potencia, los cuales, realizan la producción y/o transformación de electricidad a partir de fuentes de generación.

Comúnmente, estos equipos presentan diversas deficiencias, los cuales, son áreas de mejoramiento por parte de las empresas fabricantes de equipo eléctrico. El principal inconveniente que se presenta en el diseño de máquinas eléctricas es el elevado costo económico que conlleva realizar los diversos prototipos para lograr el producto final. Generalmente las modificaciones realizadas a los prototipos se basan en la experiencia del diseñador y/o en pruebas empíricas.

Para reducir la cantidad de prototipos fallidos y/o de pruebas exhaustivas, las empresas fabricantes de equipo eléctrico emplean técnicas numéricas computacionales modernas tales como el MEF para desarrollar el análisis electromagnético de los mismos, antes de su construcción. Este método permite analizar cualquier dispositivo electromagnético, sin importar sus dimensiones geométricas y/o características de los materiales de construcción.

Sin embargo, una de las principales desventajas del uso del MEF es el tiempo elevado de solución y el requerimiento de recursos de cómputo de alto desempeño para su ejecución. Esto impide su aplicación en procesos de optimización de los dispositivos electromagnéticos, los cuales, requieren la evaluación de un gran número de casos para un modelo final.

Por tal motivo, se han empleado metodologías de cómputo paralelo con el objetivo de mitigar las desventajas del MEF. Una de las metodologías, utilizadas a nivel industrial, es el desarrollo de software basado en hardware GPU's mediante el lenguaje de programación CUDA (Compute Unified Device Architecture).

Cabe mencionar que existe una gran variedad de software de elemento finito tanto de licencia comercial como libre mediante los cuales es posible realizar diversas simulaciones de fenómenos electromagnéticos. Sin embargo, una de las principales desventajas en la adquisición de software comercial es el alto costo económico de su licencia. Además, del desconocimiento de la metodología implementada para la aceleración y resolución de los diversos modelos, aunado a la imposibilidad de realizar modificaciones al código para la implementación de otras metodologías de investigación y/o desarrollo. Por otra parte, la adquisición de software libre conlleva restricciones en su uso. Por ejemplo, para llevar a cabo el registro de propiedad intelectual (derechos de autor).

El desarrollo de un software propio otorga la capacidad de ser empleado en el ámbito de la investigación y en la enseñanza de diversas metodologías de cómputo paralelo que pueden propiciar la incursión de estudiantes en el conocimiento de técnicas de vanguardia que se pueden emplear en el modelado, diseño y análisis acelerado de diversos dispositivos eléctricos. Al mismo tiempo, el aprendizaje de estas metodologías les permitirá a los estudiantes encontrar áreas de oportunidad de mejora en el diseño de máquinas eléctricas, generando con esto recurso humano especializado para el campo laboral.

Por tal motivo, en esta tesis se realizó el desarrollo de un software de elementos finitos en 2D para elementos de primer y segundo orden para la simulación de diversos fenómenos electromagnéticos.

Cabe mencionar que se espera que los resultados derivados de esta investigación sean utilizados en posteriores investigaciones para el modelado de fenómenos electromagnéticos más complejos y el análisis y/o diseño de otras máquinas eléctricas.

1.5 Hipótesis

El desarrollo de un software basado en hardware GPU puede acelerar la solución de un modelo electromagnético de elementos finitos en 2D.

1.6 Organización de la tesis

Esta tesis se organiza de la siguiente manera: en el capítulo 2, se presentan los conceptos generales del MEF. Posteriormente, se describen las ecuaciones de Maxwell que son utilizadas para el desarrollo de la formulación de los diversos modelos electromagnéticos, las cuales son descritas de manera detallada. Además, se presentan los formatos COO y CSR utilizados para la dispersión de matrices. Por último, se describe el solver INTEL MKL PARDISO utilizado para la resolución de la matriz.

En el capítulo 3, se presentan los conceptos generales de la interfaz de cómputo paralelo CUDA. Posteriormente, son descritas las librerías que contiene CUDA para la resolución de operaciones matemáticas. Además, se presenta el solver ICCG para la resolución de la matriz desarrollado mediante las librerías CUDA. Por último, son descritas las librerías wxWidgets que son utilizadas para la creación de la interfaz donde se realiza la selección del modelo, la inserción de parámetros y el solver.

En el capítulo 4, se presentan los resultados de las simulaciones de los diversos modelos electromagnéticos realizados en el software desarrollado de Elementos Finitos utilizando los solver's mencionados anteriormente. Además, se presentan los tiempos requeridos de ejecución de cada solver para un posterior análisis comparativo. En el caso de que el modelo contenga su respectiva solución analítica, se realiza una comparación entre los resultados de la solución analítica y los obtenidos de la simulación, para determinar el porcentaje de error entre ambos.

Finalmente, en el capítulo 5 se presentan las conclusiones obtenidas y el trabajo futuro a realizarse.

Capítulo 2

Implementación del Método de los Elementos Finitos para Determinar una Solución Aproximada de Modelos Electromagnéticos

En este capítulo se presentan las bases del Método de los Elementos Finitos aplicado para el desarrollo de modelos electromagnéticos utilizados en este trabajo de investigación. También se presentan, de manera detallada, las ecuaciones que gobiernan la aplicación del MEF así como el software INTEL MKL PARDISO utilizado para la solución de la matriz global del sistema y método comparativo.

2.1 Método de los Elementos Finitos

Para determinar la solución aproximada de un sistema de ecuaciones diferenciales parciales utilizadas para la descripción de fenómenos electromagnéticos utilizados en problemas de ingeniería y física, es utilizado el Método de Elementos Finitos (MEF). En la actualidad, el MEF es uno de los métodos más utilizados para el desarrollo de simulaciones de diversos fenómenos electromagnéticos debido a su mejora para la aplicación de recurso computacional (hardware de alto desempeño).

En el sector industrial el MEF es utilizado para el modelado de prototipos físicos, con la finalidad de reducir la cantidad de prototipos fallidos. Para desarrollar el MEF es requerido realizar un diseño asistido por computadora o también llamado CAD (por sus siglas en inglés, Computer-Aided Design), así como también es requerido el desarrollo de formulaciones matemáticas, el cual, depende del software utilizado para realizar el modelado. Algunos software's de elementos finitos cuentan con formulaciones de forma predeterminada. Se puede deducir que la implementación de este método requiere de diversas áreas tales como eléctrica, computación, matemáticas aplicadas, entre otras.

El MEF consiste en dividir (discretizar) la región o dominio de interés en pequeños elementos llamados elementos finitos (EF) donde el conjunto de elementos en la región es llamado malla. Al elaborar una malla bien discretizada de un modelo geométrico utilizando una gran cantidad de elementos, son generados resultados más confiables. Sin embargo, es requerido hardware computacional especializado para la simulación de una gran cantidad de elementos. En la Figura 2.1 se muestra el ejemplo de una malla simple para una geometría cuadrada en dos dimensiones.

De forma jerárquica las mallas están conformadas por elementos, los elementos se conforman por aristas y las aristas se crean por medio de dos nodos.

La representación de los elementos es mediante diferentes figuras geométricas tales como lo son: triángulos o cuadriláteros para el caso de simulaciones en dos dimensiones. En la Figura 2.2 se presentan los elementos que conforman el dominio de interés.

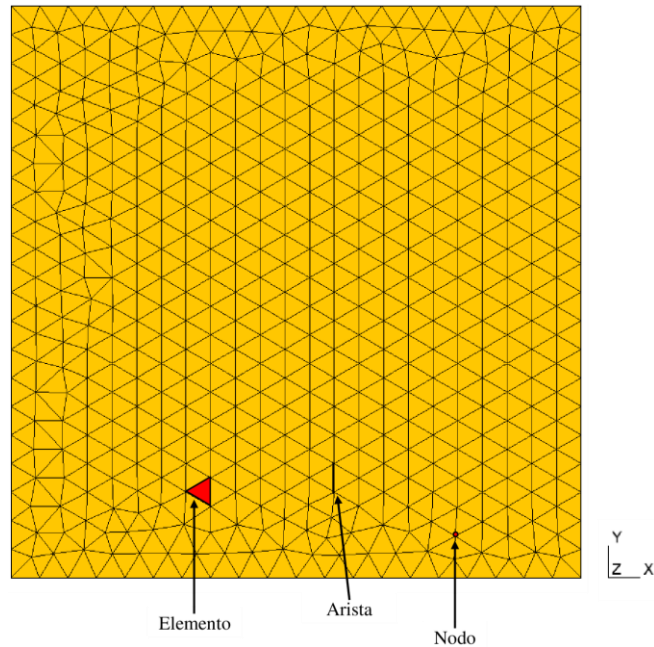


Figura 2.1. Malla de elementos finitos de una figura geométrica cuadrada en 2D.

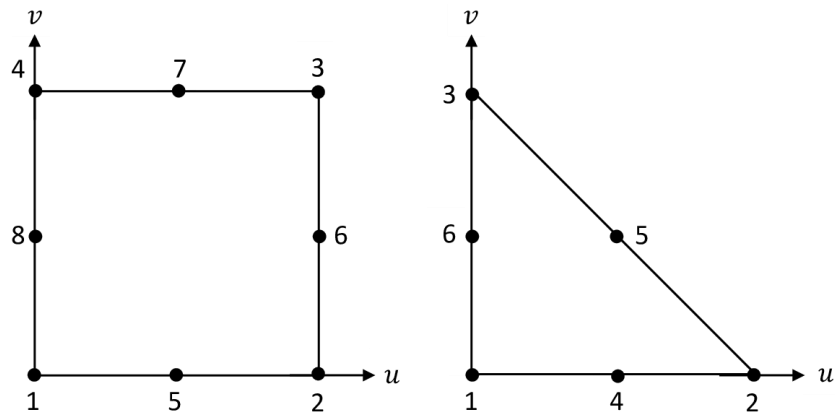


Figura 2.2. Elemento cuadrilátero y triángulo de segundo orden utilizados en el mallado del MEF.

Para mejorar la discretización de modelos que contienen una geometría curva, son utilizados los elementos isoperimétricos, los cuales son utilizados en el desarrollo de esta tesis para el proceso de formación de la matriz de rigidez.

Para determinar una solución aproximada mediante el MEF, es requerido emplear el método de Galerkin o método variacional de residuos ponderados, siendo el método de Galerkin el utilizado para el desarrollo de esta tesis.

Para la aplicación del MEF es requerido especificar cada región con sus respectivas condiciones físicas, por ejemplo, las propiedades del material, si es un material conductor o no, entre otras características.

En la aplicación del método de Galerkin es requerido emplear funciones de interpolación aproximadas en los nodos de los elementos que discretizan la región o dominio de interés. Estas

funciones de interpolación llamadas “funciones de forma” tienen su respectiva configuración para cada tipo de elemento [32]. Las funciones de forma correspondiente para elementos triangulares de segundo orden son:

$$N_1 = (1 - 2u - 2v)(1 - u - v) \quad (2.1)$$

$$N_2 = u(2u - 1) \quad (2.2)$$

$$N_3 = v(2v - 1) \quad (2.3)$$

$$N_4 = 4u(1 - u - v) \quad (2.4)$$

$$N_5 = 4uv \quad (2.5)$$

$$N_6 = 4v(1 - u - v) \quad (2.6)$$

Las funciones de forma correspondiente para elementos cuadriláteros de segundo orden son:

$$N_j = \frac{1}{4}(1 + u_j u)(1 + v_j v)(u_j u + v_j v - 1) \quad (j = 1, 2, 3, 4) \quad (2.7)$$

$$N_j = \frac{1}{4}(1 + u^2)(1 + v_j v) \quad (j = 5, 7) \quad (2.8)$$

$$N_j = \frac{1}{4}(1 + v^2)(1 + u_j u) \quad (j = 6, 8) \quad (2.9)$$

El MEF está conformado por 3 etapas fundamentales: pre-procesamiento, procesamiento y post-procesamiento.

La fase de pre-procesamiento consiste en la generación de la geometría y malla del modelo, donde a su vez, son incorporados las características físicas en cada región y las condiciones de frontera del modelo a resolver.

La fase del procesamiento consiste en la implementación de la formulación matemática de Galerkin y se realiza la solución del sistema de ecuaciones resultante.

Finalmente, la fase del post-procesamiento consiste en el despliegue gráfico y/o numérico de los resultados obtenidos del modelo para su posterior interpretación y análisis.

2.2 Ecuaciones de Maxwell

Para la descripción de los fenómenos electromagnéticos son utilizadas las ecuaciones de Maxwell. Particularmente en esta tesis, se utilizan para el desarrollo de un software de elementos finitos en 2D para el modelado de fenómenos electromagnéticos presentados en máquinas eléctricas.

Las ecuaciones de Maxwell presentadas en forma diferencial, son utilizadas en el desarrollo de la formulación matemática para el modelado de diversos fenómenos electromagnéticos presentados en esta tesis. La ecuación (2.10) representa la ley de Ampere, la ecuación (2.11) representa la ley de Gauss para materiales magnéticos, la ecuación (2.12)

representa la ley de Faraday y la ecuación (2.13) representa la ley de Gauss para materiales eléctricos.

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \quad (2.10)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (2.11)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (2.12)$$

$$\nabla \times \mathbf{D} = \rho \quad (2.13)$$

donde:

\mathbf{H} es la intensidad de campo magnético (A/m).

\mathbf{J} es la densidad de corriente eléctrica (A/m²).

\mathbf{D} es la densidad de campo eléctrico (V/m²).

\mathbf{B} es la densidad de flujo magnético (T).

\mathbf{E} es el campo eléctrico (V/m).

ρ es la densidad de carga superficial.

2.3 Formulación magnetostática

Para el desarrollo de la formulación magnetostática, es utilizada la ley de Ampere mostrada en la ecuación (2.10). Debido a que la formulación es desarrollada para el análisis en bajas frecuencias, la corriente de desplazamiento $\partial \mathbf{D} / \partial t$ es despreciable. Por tal motivo, la ley de Ampere es convertida en:

$$\nabla \times \mathbf{H} = \mathbf{J} \quad (2.14)$$

donde se requiere utilizar las ecuaciones constitutivas, dadas por:

$$\mathbf{B} = \mu \mathbf{H} \quad (2.15)$$

$$\mathbf{J} = \sigma \mathbf{E} \quad (2.16)$$

donde:

μ es la permeabilidad magnética (H/m).

σ es la conductividad eléctrica del material (S/m).

Al despejar la intensidad de campo magnético de la ecuación (2.15) es obtenida:

$$\mathbf{H} = \nu \mathbf{B} \quad (2.17)$$

donde ν es la reluctancia del material ($\nu = 1/\mu$).

Sustituyendo la ecuación (2.17) en la ecuación (2.14) es obtenido:

$$\nabla \times \nu \mathbf{B} = \mathbf{J} \quad (2.18)$$

Al aplicar una identidad vectorial en la ecuación (2.11) es obtenido el potencial magnético vectorial \mathbf{A} , determinado como:

$$\mathbf{B} = \nabla \times \mathbf{A} \quad (2.19)$$

Sustituyendo la ecuación (2.19) en la ecuación (2.18) es obtenido lo siguiente:

$$\nabla \times v(\nabla \times \mathbf{A}) = \mathbf{J} \quad (2.20)$$

Aplicando el vector laplaciano en la ecuación (2.20) es obtenido:

$$v\{\nabla \times (\nabla \times \mathbf{A})\} = \mathbf{J} \quad (2.21)$$

$$\nabla(\nabla \cdot \mathbf{A}) - \nabla^2 \mathbf{A} = \mathbf{J} \quad (2.22)$$

Debido a que el potencial vectorial tiene divergencia cero ($\nabla \cdot \mathbf{A} = 0$), la ecuación (2.22) es reducida a:

$$-\nabla^2 \mathbf{A} = \mathbf{J} \quad (2.23)$$

Dado a que el potencial magnético vectorial y la densidad de corriente eléctrica, en el plano bidimensional, solamente tiene componente en z :

$$-\nabla \cdot (\nabla A_z) = J_z \quad (2.24)$$

$$\nabla \cdot (\nabla A_z) + J_z = 0 \quad (2.25)$$

donde A_z es la magnitud de la componente en z del vector de potencial magnético.

Aplicando el método de residuos ponderados de Galerkin en la ecuación (2.25) para determinar una solución aproximada, es producido un residual, R , dado por:

$$\nabla \cdot (\nabla A_z) + J_z = R \quad (2.26)$$

Sustituyendo el residual R en el método de Galerkin, es obtenido:

$$\int_{\Omega} W[\nabla \cdot (\nabla A_z) + J_z] d\Omega = 0 \quad (2.27)$$

donde, W , es la función de peso.

Al aplicar el teorema de la divergencia en la ecuación (2.27), es obtenida la ecuación en “forma débil”, la cual debe su nombre por la reducción del orden de la ecuación diferencial y permite la implementación del MEF en la discretización de dicha ecuación sea de manera más sencilla, como se muestra en:

$$\sum_{n=1,N} \int_{\Omega_n} [\nabla A_z \cdot \nabla \phi_n + J_z \phi_n] d\Omega = 0 \quad (2.28)$$

donde ϕ_n representa la función de forma del elemento.

2.4 Formulación Electroestática

Para describir el campo electrostático es utilizada la Ley de Gauss para materiales eléctricos mostrada en la ecuación (2.13). Utilizando la relación constitutiva $\mathbf{D} = \epsilon\mathbf{E}$ y realizando la sustitución en la ecuación (2.13) obtenemos:

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon} \quad (2.29)$$

donde:

ϵ es la permitividad eléctrica del material.

Para definir el potencial eléctrico escalar, V , es aplicado la definición del campo eléctrico escalar dada por $\mathbf{E} = -\nabla V$. Al sustituir esta definición en la ecuación (2.29) es obtenida la ecuación de Poisson que es empleado para el modelado electrostático:

$$\nabla \cdot (\epsilon \nabla V) + \rho = 0 \quad (2.30)$$

Aplicando el método de residuos ponderados de Galerkin en la ecuación (2.30) se produce un residual, R , dado por:

$$\nabla \cdot (\epsilon \nabla V) + \rho = R \quad (2.31)$$

Al sustituir el residual, R , en el método de Galerkin es obtenido:

$$\int_{\Omega} W [\nabla \cdot (\epsilon \nabla V) + \rho] d\Omega = 0 \quad (2.32)$$

Al aplicar el teorema de la divergencia en la ecuación (2.32), es implementado el MEF, como se muestra en:

$$\sum_{n=1,N} \int_{\Omega_n} [\epsilon \nabla V \cdot \nabla \phi_n - \rho \phi_n] d\Omega = 0 \quad (2.33)$$

2.4.1 Formulación Laplace

Cuando no están presentes cargas estáticas ($\rho = 0$) y únicamente está presente un material dieléctrico en la región o dominio de interés, la ecuación (2.30) se reduce a:

$$\nabla \cdot (\epsilon \nabla V) = 0 \quad (2.34)$$

La ecuación (2.34) es llamada Laplace, en donde la fuente del campo eléctrico en el dominio de interés son las condiciones de frontera en el cual diferentes potenciales son impuestas. Al aplicar el método de residuos ponderados y el teorema de divergencia en la ecuación (2.34) es obtenido:

$$\sum_{n=1,N} \int_{\Omega_n} [\epsilon \nabla V \cdot \nabla \phi_n] d\Omega = 0 \quad (2.35)$$

2.5 Formulación magnetodinámica

2.5.1 Transitoria

Utilizando la ley de Ampere mostrada en la ecuación (2.10) para el análisis de fenómenos electromagnéticos de bajas frecuencias, es obtenido:

$$\nabla \times \mathbf{H} = \mathbf{J} \quad (2.36)$$

Utilizando la ley de Gauss para materiales magnéticos mostrada en la ecuación (2.11) y aplicando una identidad vectorial, es obtenido:

$$\nabla \times \mathbf{A} = \mathbf{B} \quad (2.37)$$

donde \mathbf{A} es el potencial magnético vectorial.

Utilizando la relación constitutiva mostrada en la ecuación (2.38), es despejada la intensidad de campo eléctrico \mathbf{H} , como se muestra en la ecuación (2.39):

$$\mathbf{B} = \mu \mathbf{H} \quad (2.38)$$

$$\mathbf{H} = \nu \mathbf{B} \quad (2.39)$$

donde ν es la reluctancia del material ($\nu = 1/\mu$).

Al sustituir las ecuación (2.37) en (2.39), es obtenido:

$$\mathbf{H} = \nu (\nabla \times \mathbf{A}) \quad (2.40)$$

Sustituyendo la ecuación (2.40) en (2.36), es obtenido:

$$\nu \nabla \times (\nabla \times \mathbf{A}) = \mathbf{J} \quad (2.41)$$

donde:

$$\mathbf{J} = \mathbf{J}_s + \mathbf{J}_e \quad (2.42)$$

\mathbf{J}_s es la densidad de corriente impuesta y \mathbf{J}_e representa la densidad de corriente inducida en los materiales conductores, que es definida por:

$$\mathbf{J}_e = -\sigma \left[\frac{\partial \mathbf{A}}{\partial t} + \nabla V \right] \quad (2.43)$$

donde σ es la conductividad eléctrica del material (S/m).

Debido a la variación de la densidad de flujo magnético en el tiempo, es generado un campo eléctrico en el conductor y es obtenido:

$$\mathbf{J}_e = -\sigma \frac{\partial \mathbf{A}}{\partial t} \quad (2.44)$$

Al sustituir las ecuaciones (2.44) y (2.42) en (2.41), se obtiene la ecuación de difusión electromagnética:

$$\nu \nabla \times (\nabla \times \mathbf{A}) + \sigma \frac{\partial \mathbf{A}}{\partial t} - \mathbf{J}_s = 0 \quad (2.45)$$

Considerando que el dominio de estudio es en el plano bidimensional (2D), solamente son considerados los componentes en el eje z del potencial magnético vectorial y la densidad de corriente aplicada e inducida. Al aplicar una identidad vectorial al primer término del lado izquierdo de la ecuación (2.45):

$$\nabla \times v(\nabla \times \mathbf{A}) = v\nabla \cdot \nabla A_z \quad (2.46)$$

Por tal motivo la ecuación de difusión en 2D es:

$$\nabla \cdot v\nabla A_z - \sigma \frac{\partial A_z}{\partial t} + J_z = 0 \quad (2.47)$$

Al aplicar el teorema de la divergencia en la ecuación (2.47), es implementado el MEF, como se muestra en:

$$\sum_{n=1,N} \int_{\Omega_n} [\nabla v A_z \cdot \nabla \phi_n + J_z \phi_n] d\Omega = 0 \quad (2.48)$$

2.5.2 Compleja

En el caso de que la densidad de corriente inducida sea una señal sinusoidal, su notación es compleja, por lo que se obtiene:

$$\nabla^2 A_z - j\omega \frac{\partial A_z}{\partial t} + J_z = 0 \quad (2.49)$$

$$\sum_{n=1,N} \int_{\Omega_n} [\nabla v A_z \cdot \nabla \phi_n + J_z \phi_n] d\Omega = 0 \quad (2.50)$$

2.6 Formulación de transferencia de calor

Para el desarrollo matemático de este modelo, es utilizando la terminología y definiciones tomadas de la referencia [33].

Para determinar la distribución de calor en un material en función del tiempo es utilizada la ley de Fourier, la cual establece el tiempo de transferencia de calor a través de las características térmicas del mismo y está dada por:

$$-k\nabla T = q \quad (2.51)$$

donde:

q es el flujo de calor (W/m^2).

k es la conductividad térmica del material (W/mK).

∇T es el gradiente de temperatura (K/m).

La proporcionalidad de la variación de energía interna por unidad de volumen en el material y el calor generado están dados por:

$$\nabla Q = \rho c \nabla T \quad (2.52)$$

donde:

Q es el flujo de calor externamente aplicado.

ρ es la densidad de masa.

c es el calor específico.

Para determinar la ecuación diferencial general de conducción de calor para un medio estacionario en dos dimensiones son utilizadas las ecuaciones (2.20) y (2.21), como se muestra en:

$$\frac{\partial}{\partial x} \left(k_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k_y \frac{\partial T}{\partial y} \right) + Q = \rho c \frac{\partial T}{\partial t} \quad (2.53)$$

Al aplicar el método de residuos ponderados y el teorema de divergencia en la ecuación (2.) es obtenido:

$$\int_{\Omega} W \left[\frac{\partial}{\partial x} \left(k_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k_y \frac{\partial T}{\partial y} \right) + Q - \rho c \frac{\partial T}{\partial t} \right] d\Omega = 0 \quad (2.54)$$

2.7 Almacenamiento de matriz en formato COO (Coordinate)

Al utilizar métodos tradicionales de almacenamiento para matrices de gran tamaño, es requerido una gran cantidad de memoria y tiempo de procesamiento para la resolución de sistemas de ecuaciones lineales. Por tal motivo, se han diseñado diversos formatos de almacenamiento de matrices dispersas [34].

Uno de los formatos utilizados en el desarrollo de esta tesis para el almacenamiento de la matriz global del sistema generada por el MEF, es el formato COO, el cual es una de los métodos de almacenamiento de matrices dispersas más sencillos de aplicar y cuya composición consta de tres vectores llamados VAL, JA e IA.

A continuación, se realiza la aplicación del formato COO a la matriz no simétrica de 5 x 5 llamada A :

$$A = \begin{bmatrix} 5 & 8 & 0 & -3 & 0 \\ 4 & -7 & 0 & 0 & 0 \\ 0 & 0 & -2 & 1 & 7 \\ 10 & 0 & 6 & -2 & 0 \\ 0 & 11 & 0 & 0 & 9 \end{bmatrix}$$

$$\begin{aligned} \text{VAL} &= [5 \ 8 \ -3 \ 4 \ -7 \ -2 \ 1 \ 7 \ 10 \ 6 \ -2 \ 11 \ 9] \\ \text{JA} &= [0 \ 1 \ 3 \ 0 \ 1 \ 2 \ 3 \ 4 \ 0 \ 2 \ 3 \ 1 \ 4] \\ \text{IA} &= [0 \ 0 \ 0 \ 1 \ 1 \ 2 \ 2 \ 2 \ 3 \ 3 \ 3 \ 4 \ 4] \end{aligned}$$

donde:

- VAL Contiene los elementos diferentes de cero de la matriz A , donde su longitud llamado “nnz” es igual al número de elementos diferentes de cero.
- JA Contiene los índices de la columna de los elementos de VAL.
- IA Contiene los índices del renglón de los elementos de VAL.

2.8 Almacenamiento de matriz en formato CSR (Compressed Storage Row)

Debido al requerimiento de ingresar la matriz en formato CSR para la ejecución de los solver’s utilizados en esta tesis, es utilizado el formato CSR para la resolución del sistema de ecuaciones, el cual es un método muy utilizado debido a su simplicidad y compactibilidad, que no genera una disminución en el rendimiento computacional.

A diferencia del formato COO, este formato implementa un esquema de compresión aplicado en los índices de IA, lo que permite una fácil conversión entre ambos formatos.

Utilizando la misma matriz A , se realiza la dispersión de la matriz no simétrica utilizando el formato CSR:

$$A = \begin{bmatrix} 5 & 8 & 0 & -3 & 0 \\ 4 & -7 & 0 & 0 & 0 \\ 0 & 0 & -2 & 1 & 7 \\ 10 & 0 & 6 & -2 & 0 \\ 0 & 11 & 0 & 0 & 9 \end{bmatrix}$$

$$\begin{aligned} \text{VAL} &= [5 \ 8 \ -3 \ 4 \ -7 \ -2 \ 1 \ 7 \ 10 \ 6 \ -2 \ 11 \ 9] \\ \text{JA} &= [0 \ 1 \ 3 \ 0 \ 1 \ 2 \ 3 \ 4 \ 0 \ 2 \ 3 \ 1 \ 4] \\ \text{IA} &= [0 \ 3 \ 5 \ 8 \ 11 \ 13] \end{aligned}$$

donde:

- VAL Contiene los elementos diferentes de cero de la matriz A introducidos por IA y su longitud es nnz.
- JA Contiene los índices de la columna de los elementos de VAL y su longitud es nnz.
- IA Contiene los índices del renglón de los elementos de VAL y su longitud es igual a $n + 1$.

2.9 Solver INTEL MKL PARDISO

Para realizar el análisis comparativo de aceleración en la solución de modelos electromagnéticos utilizando GPU’s de CUDA, es utilizada la librería INTEL MKL (Math Kernel Library, por sus siglas en inglés) [35], la cual es una librería optimizada para el desarrollo de aplicaciones en el ámbito científico, ingeniería, entre otros. Las librerías contienen funciones matemáticas básicas que incluyen BLAS, solucionadores dispersos, etc.

Para la resolución de la matriz global del sistema, es utilizado el solver INTEL MKL PARDISO, el cual es un paquete de alto desempeño y memoria eficiente para la solución de largos sistemas dispersos de ecuaciones utilizando multiprocesadores de memoria compartida. El solver utiliza una combinación de técnicas de supernodo BLAS de nivel 3 de izquierda a derecha para mejorar el rendimiento de la factorización numérica dispersa de manera secuencial y paralela. Algunas de las funciones que utiliza el solver PARDISO son:

- iparm [1] Realiza el ordenamiento reducido de la matriz de entrada.
- iparm [3] Realiza el preconditionamiento del CGS (Conjugate Gradients Squared) para matrices no simétricas o estructuralmente simétricas y el CG (Conjugate Gradient) para matrices simétricas.
- iparm [5] Escribe la solución en x.
- iparm [19] Reporta el diagnostico de CGS/CG.

Para la ejecución del solver, es requerido implementar la dispersión de la matriz global del sistema generado por el MEF en formato COO, para posteriormente realizar la conversión a formato CSR. La función utilizada es:

```
sparse_status_t mkl_sparse_?_create_coo (sparse_matrix *A, const sparse_index_base_t indexing, const MKL_INT rows, const MKL_INT cols, const MKL_INT nnz, MKL_INT *row_idx, MKL_INT *col_idx, double*values);
```

donde:

- ? Indica el tipo de variable.
- indexing Indica como la entrada de la matriz es indexada.
- rows Indica el número de renglones de la matriz A.
- cols Indica el número de columnas de la matriz A.
- nnz Especifica el número de elementos diferentes de cero de la matriz A.
- row_idx Es el vector de longitud nnz que contiene el índice del renglón de cada elemento diferente de cero de la matriz A.
- col_idx Es el vector de longitud nnz que contiene el índice de columna de cada elemento diferente de cero de la matriz A.
- values Es el vector de longitud nnz que contiene los elementos diferentes de cero de la matriz A, ordenados de forma arbitraria.

Para realizar la conversión de la matriz dispersa en formato COO a CSR, es utilizada la función:

```
sparse_status_t mkl_sparse_convert_csr (const sparse_matrix_t source, const sparse_operation_t operation, sparse_matrix_t *dest);
```

donde:

- source Es el manejador que contiene los datos internos.
- operation Especifica el tipo de operación (transpuesta, no transpuesta, etc.) en la entrada de la matriz.

Capítulo 3

Aceleración en la resolución de modelos electromagnéticos mediante CUDA

En este capítulo se presenta la metodología del software especializado en paralelismo CUDA, el cual es utilizado con el objetivo de reducir el tiempo requerido de simulación de modelos electromagnéticos utilizando el solver ICCG desarrollado por medio de las librerías cuBLAS y cuSPARSE.

Además, se presentan las librerías wxWidgets que son utilizados para la creación de una interfaz gráfica para llevar a cabo una interacción con el usuario final mediante la selección del modelo electromagnético que se desea simular, el establecimiento de parámetros y el solver para la resolución de la matriz global del sistema.

3.1 CUDA: Software especializado en paralelismo

En 2006 fue propuesto el esquema de programación CUDA (Compute Unified Device Architecture), para el uso óptimo de las capacidades computacionales de las tarjetas gráficas de NVIDIA. Lo que estableció un modelo de programación para las tarjetas gráficas más eficientes y rápidamente fue considerado por la comunidad científica un gran avance para la Unidad de Procesamiento de Graficos de Propósito General (GPGPU-General-Purpose Computing on Graphics Processing Units).

Sin embargo, el problema principal al utilizar tarjetas gráficas para el cálculo científico de GPU (abreviatura de GPGPU) era el uso específico de lenguajes de programación para gráficos como OpenGL para realizar la programación del mismo. Como consecuencia, su acceso era limitado en el campo científico de la eficiencia y rendimiento de las GPU's.

NVIDIA realizó modificaciones en la arquitecturas de los GPU's con la finalidad de realizar adaptaciones para aplicaciones científicas, además de añadir asesoría y soporte para lenguajes de programación como C y C++.

En 2009, se realizó el lanzamiento de la arquitectura CUDA (Compute Unified Device Architecture) para el desarrollo de aplicaciones de programación mediante el uso de funciones especializados en paralelismo.

Las tarjetas gráficas de NVIDIA están constituidas por cientos de núcleos que procesan, de forma concurrente, miles de hilos de ejecución. Cada núcleo comparte ciertos recursos, además de registros y memoria. La memoria compartida, se encuentra integrada en el procesador y permite la ejecución de tareas en los núcleos que comparten datos sin enviar información a través del bus de memoria, para eficientar el procesamiento de datos.

Una tarjeta gráfica tiene integrada una arquitectura que consiste en la presencia de las Unidades de Ejecución denominadas Streaming Multiprocessors (SM), interconectadas entre sí en la zona de memoria común. La composición de una SM está constituida por núcleos de cómputo denominados núcleos CUDA o Streaming Processors (SP), encargados de la ejecución de las instrucciones. Esta arquitectura, permite la programación de los núcleos que contienen los GPU's, utilizando el lenguaje CUDA C.

El desarrollo de un programa consiste en un código de forma secuencial ejecutado en el CPU. En la sección donde sea requerido realizar el paralelismo, se realiza la invocación de un kernel que es ejecutado en el GPU, como se muestra en la Figura 3.1.

El kernel invocado realiza su ejecución de forma paralela en el GPU. El conjunto de hilos son organizados en una jerarquía donde son agrupados en bloques, que a su vez son agrupados en mallas, como se muestra en la Figura 3.2.

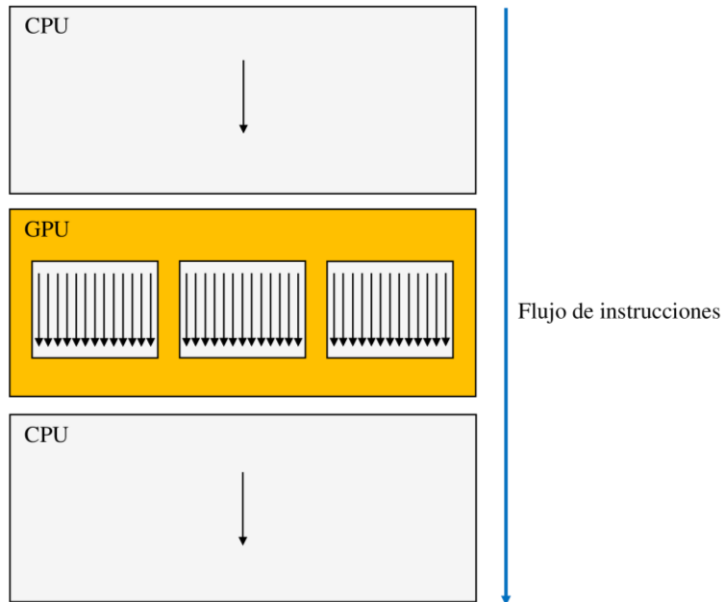


Figura 3.1. Estructura de código con sección secuencial y paralela.

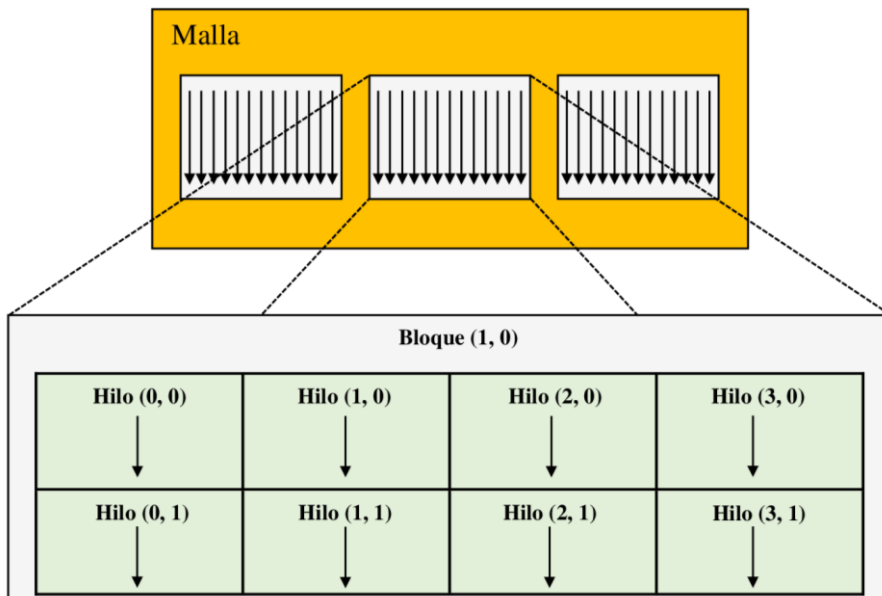


Figura 3.2. Jerarquía en una aplicación CUDA.

Los bloques y mallas creados pueden ser de una, dos o tres dimensiones. Además, existen diversas situaciones donde los datos tienen una estructura de malla y descomponer los datos en una jerarquía de hilos es compleja. Por tal motivo, un bloque es un conjunto de hilos concurrentes que cooperan entre sí, a través de mecanismos de sincronización y acceso a un espacio de memoria exclusivo del mismo. Una malla es un conjunto de bloques que son ejecutados de manera independientemente y de forma paralela en los Streaming Multiprocessors (SM).

Al realizar la invocación de un kernel, es especificado el número de hilos que contiene cada bloque y a su vez, es especificado el número de bloques que conforman a la malla. Cada hilo tiene asignado un número de identificación único dentro de cada bloque, y a su vez, cada bloque recibe un identificador único dentro de la malla. Esto permite que cada hilo realice la selección de los datos que ejecutara, simplificando de manera significativa el direccionamiento de memoria en la ejecución de datos multidimensionales, como en el procesamiento de imágenes o la resolución numérica de ecuaciones diferenciales en dos y tres dimensiones.

3.2 Librerías de resolución de ecuaciones CUDA

Para el desarrollo de rutinas de algebra lineal, CUDA ofrece diversas bibliotecas gratuitas que están optimizadas para la ejecución de rutinas de forma rápida y eficiente. Lo que permite optimizar de manera significativa el código desarrollado, estandarizar la sintaxis de las rutinas y una mayor portabilidad del código.

3.2.1 CUSPARSE

La librería cuSPARSE (CUDA Sparse Matrix Library) [36], es un conjunto de rutinas básicas de álgebra lineal utilizadas en la solución de matrices dispersas y está diseñado para ser llamado desde C y C ++. Las rutinas de la librería se clasifican en cuatro categorías:

Nivel 1	Operaciones entre un vector en formato disperso y un vector en formato denso.
Nivel 2	Operaciones entre una matriz en formato disperso y un vector en formato denso.
Nivel 3	Operaciones entre una matriz en formato disperso y un conjunto de vectores en formato denso formato (que también puede verse normalmente como una matriz alta densa).
Conversión	Operaciones que permiten la conversión entre diferentes formatos matriciales, y compresión de matrices CSR.

Utilizando la siguiente función de nivel 1, es determinado el producto punto de un vector x en formato disperso y un vector y en formato denso:

```
cusparse<t>doti(cusparseHandle_t handle, int nnz, const double* xVal, const int* xInd, const double* y, double* resultDevHostPtr, cusparseIndexBase_t idxBase);
```

donde:

<t>	Corresponde al tipo de dato.
handle	Manejo del contexto de la librería cuSPARSE.
nnz	Número de elementos en vector x.
xVal	Vector con nnz valores diferentes de cero del vector x.
xInd	Vector con nnz índices de valores diferentes de cero de vector x.
y	Vector en formato denso
resultDevHostPtr	Puntero con la locación del resultado en la memoria del device (GPU) o host (CPU).
idxBase	CUSPARSE_INDEX_BASE_ZERO o CUSPARSE_INDEX_BASE_ONE.
resultDevHostPtr	Resultado escalar en la memoria del device o host.

Para utilizar las funciones, es necesario implementar los siguientes ficheros en la cabecera del código:

```
#include <cuda_runtime.h>
#include <cusparse_V2.h>
```

Además, es requerido realizar la asignación de memoria y copiar los datos entre la memoria del GPU y CPU mediante las rutinas estándar de CUDA como `cudaMalloc()`, `cudaFree()`, `cudaMemcpy()` y `cudaMemcpyAsync()`.

Las funciones están disponibles para datos de tipo `float`, `double`, `cuComplex`, `cuDoubleComplex`.

3.2.2 CUBLAS

La librería cuBLAS (CUDA Basic Linear Algebra Subprograms) [37], es un conjunto de rutinas básicas de BLAS para realizar operaciones comunes de algebra lineal como producto escalar, multiplicación escalar, combinaciones lineales y multiplicación de matrices. Las rutinas se dividen en 3 categorías:

- Nivel 1 Operaciones vectoriales en matrices escalares.
- Nivel 2 Operaciones entre una matriz y un vector.
- Nivel 3 Operaciones entre matrices.

Utilizando la siguiente función de nivel e, es determinado la multiplicación entre matrices:

```
cublas<t>gemm(cublasHandle_t handle, cublasOperation_t transa, cublasOperation_t transb, int m, int n, int k, const float *alpha, const float *A, int lda, const float *B, int ldb, const float *beta, float *C, int ldc);
```

donde:

transa	Operación op(A) que es non- o (conj.) transpuesta.
transb	Operación op(B) que es non- o (conj.) transpuesta.
m	Número de renglones de matriz op(A) y C.
n	Número de columnas de matriz op(B) y C.
k	Número de columnas de op(A) y renglones de op(B).
alpha	Escalar utilizado para multiplicación.
A	Arreglo de dimensiones $lda \times k$ con $lda \geq \max(1, m)$ si $transa = = CUBLAS_OP_N$ y $lda \times m$ con $lda \geq \max(1, k)$ de lo contrario.
lda	Dimensión principal de arreglo bidimensional usado para almacenar matriz A.
B	Dimensiones de arreglo $ldb \times n$ con $ldb \geq \max(1, k)$ si $transb = = CUBLAS_OP_N$ y $lda \times k$ con $ldb \geq \max(1, n)$ de lo contrario.
ldb	Dimensión principal de arreglo bidimensional usado para almacenar matriz B.
beta	Escalar usado para multiplicación. Si $beta = = 0$, C no tiene que ser una entrada valida.
C	Arreglo de dimensiones $ldc \times n$ con $ldc \geq \max(1, m)$.
ldc	Dimensión principal de arreglo bidimensional usado para almacenar matriz C.

Para utilizar las funciones, es necesario implementar los siguientes ficheros en la cabecera del código:

```
#include <cuda_runtime.h>
#include <cublas_v2.h>
```

3.3 Solver ICCG utilizando librerías de CUDA

En 1954, Hestenes y Stiefel introdujeron el método CG (Conjugate Gradient) para la resolución de sistemas lineales de ecuaciones [38]. Considerado como una curiosidad matemática debido a su alto requerimiento de recurso computacional.

En 1974, Meijerink y Van der Vorst, realizaron la combinación del CG con la descomposición IC (Incomplete Cholesky). Implementando un estrategia de preconditionamiento para acelerar el tiempo de convergencia y disminuir el requerimiento de recurso computacional.

El solver ICCG CUDA, es desarrollado mediante las librerías cuBLAS y cusparse, las cuales, realizan el cálculo del factor Cholesky de una matriz denominada “A”. Con el enfoque de determinar una matriz triangular inferior denominada “H”, que tiene la propiedad de un estructura de dispersión manejable y muy cercano al factor Cholesky de la matriz A.

En [39] es presentado el código computacional de preconditionamiento Incompleto Cholesky:

```

for k = 1:n
    A(k,k) = sqrt(A(k,k))
    for i = k + 1:n
        if A(i,k) ≠ 0
            A(i,k) = A(i,k)/A(k,k)
        end
    end
    for j = k + 1:n
        for i = j:n
            if A(i,j) ≠ 0
                A(i,j) = A(i,j) - A(i,k)A(j,k)
            end
        end
    end
end
end

```

3.4 wxWidgets: Software utilizado para la creación de una interfaz grafica

En 1992, Julian Smart realizo el lanzamiento del software wxWidgets [40], el cual consiste en un conjunto de librerías que permiten el desarrollo de interfaces graficas en el lenguaje de programación C++. Las librerías de licencia libre, permiten la creación de una interfaz gráfica que se integra de forma óptima y es portable entre diversos sistemas operativos como Windows, MacOS, etc.

Para la creación de una tabla de eventos para la declaración de los eventos provenientes del usuario o de otras fuentes, son utilizadas las siguientes instrucciones:

```

BEGIN_EVENT_TABLE(MyFrame, wxFrame)
    EVT_MENU(wxID_ABOUT, MyFrame::OnAbout)
    EVT_MENU(wxID_EXIT, MyFrame::OnQuit)
END_EVENT_TABLE()

```

Con wxID_ABOUT y wxID_EXIT, son identificados los clics del mouse en los elementos del menú y se enruta a las funciones MyFrame::OnAbout y MyFrame::OnQuit, respectivamente.

EVT_MENU es una de las diversas tablas de eventos para declarar que tipo de evento debe enrutarse a una función.

La interfaz creada para el desarrollo de esta tesis se muestra en la Figura 3.2, en el cual, el usuario final puede realizar la selección del tipo de modelo que desea simular y posteriormente realizar la inserción de los parámetros que conforman al modelo.

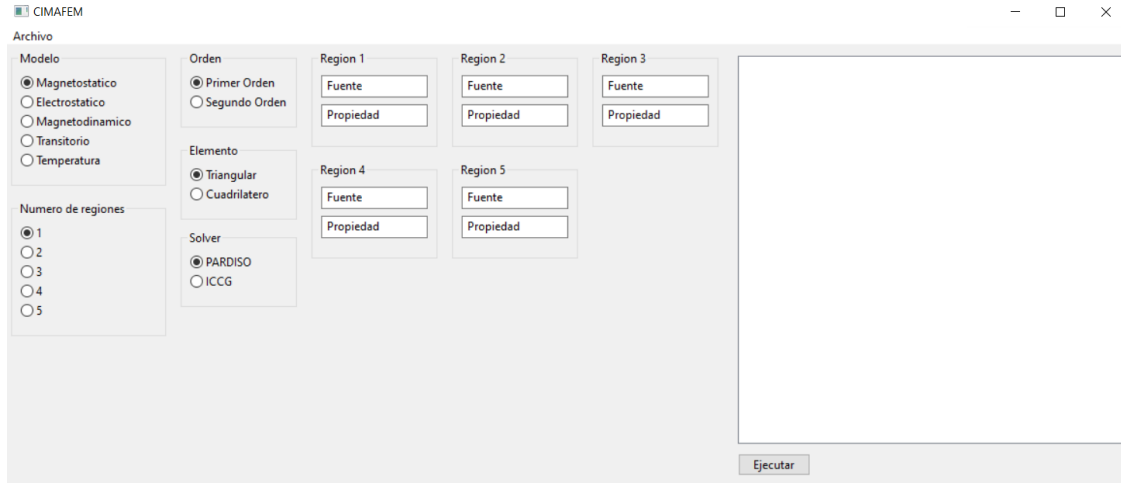


Figura 3.3. Interfaz gráfica de software de elementos finitos desarrollada para la interacción con el usuario final.

Capítulo 4

Simulación de modelos electromagnéticos mediante el software desarrollado de Elementos Finitos en 2D

En este capítulo se presenta la simulación de diversos modelos electromagnéticos tomados de diversas referencias para validar la metodología implementada en el software de elementos finitos en 2D. Debido a la similitud de los resultados obtenidos en la resolución de los modelos al utilizar los solver's INTEL MKL PARDISO e ICCG CUDA con elementos triangulares y cuadriláteros, solamente en el caso 1 de simulación de modelos magnetostáticos, son presentados los resultados utilizando ambos tipos de elementos y solver's.

Las simulaciones fueron ejecutadas en una laptop Alienware M15, procesador Intel Core i7 Octava Generación, 16 Gb de memoria RAM.

4.1 Simulación de modelos magnetostáticos

4.1.1 Caso 1

El modelo magnetostático, desarrollado para fines didácticos en la referencia [41], consiste en un bloque de material con una permeabilidad igual a $10\mu_0$, donde fluye un flujo magnético \mathbf{B} . En la parte interna del bloque, una barra conductora con permeabilidad μ_0 (cobre o aluminio), transporta una densidad de corriente, \mathbf{J} , como se muestra en la Figura 4.1.

El software Gmsh [42], es un generador de mallas donde es descrito la geometría del modelo y el tipo de elemento que conforma la malla. En la Figura 4.2 es mostrado el mallado del modelo elaborado en Gmsh, el cual contiene 354,025 nodos y 176,418 elementos triangulares de segundo orden. En la Figura 4.3 es mostrado con un mayor acercamiento el mallado del modelo para mejorar la visualización de los elementos triangulares. En la Figura 4.4 es mostrado el mallado del modelo con elementos cuadriláteros de segundo orden, el cual contiene 265,816 nodos y 88,209 elementos. En la Figura 4.5 es mostrado con un mayor acercamiento el mallado del modelo para mejorar la visualización de los elementos cuadriláteros.

Posteriormente, es generado un archivo de texto de la malla del modelo creado en Gmsh, el cual es utilizado por el software de elementos finitos para la lectura de datos geométricos e inserción de parámetros de los materiales que conforman cada región del modelo, durante la etapa de pre-procesamiento.

En la etapa de procesamiento, el software de elementos finitos utiliza la metodología del MEF para la generación de la matriz global del sistema y termino fuente, para su posterior solución mediante los solver's INTEL MKL PARDISO e ICCG CUDA.

En la etapa de post-procesamiento, son generados archivos de texto que contienen los resultados obtenidos en la simulación del modelo con un formato específico, para su posterior visualización en Gmsh.

En la Figura 4.6 y 4.7 es mostrada la distribución de los potenciales eléctricos y la magnitud del campo eléctrico desarrollado en el modelo, al utilizar la malla con elementos triangulares y solver INTEL MKL PARDISO.

En la Figura 4.8 y 4.9 es mostrada la distribución de los potenciales eléctricos y la magnitud del campo eléctrico desarrollado en el modelo, al utiliza la malla con elementos triangulares y solver ICCG CUDA.

En las Figuras 4.10 y 4.11 es mostrada la distribución de los potenciales eléctricos y la magnitud del campo eléctrico desarrollado en el modelo, al utilizar la malla con elementos cuadriláteros y solver INTEL MKL PARDISO.

En las Figuras 4.12 y 4.13 es mostrada la distribución de los potenciales eléctricos y la magnitud del campo eléctrico desarrollado en el modelo, al utilizar la malla con elementos cuadriláteros y solver ICCG CUDA.

Para el análisis de desempeño, es determinado el tiempo requerido de simulación al utilizar ambos del solver's. En la Tabla 4.1 se muestra el tiempo requerido de simulación por ambos solver's al utilizar elementos triangulares. En la Tabla 4.2, se muestra el tiempo requerido de simulación al utilizar elementos cuadriláteros.

Tabla 4.1. Tiempo requerido de simulación de modelo barra conductora, utilizando elementos triangulares.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	2,192,724	165
ICCG CUDA	4,031,423	164

Tabla 4.2. Tiempo requerido de simulación de modelo barra conductora, utilizando elementos cuadriláteros.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	2,189,208	177
ICCG CUDA	4,112,600	133

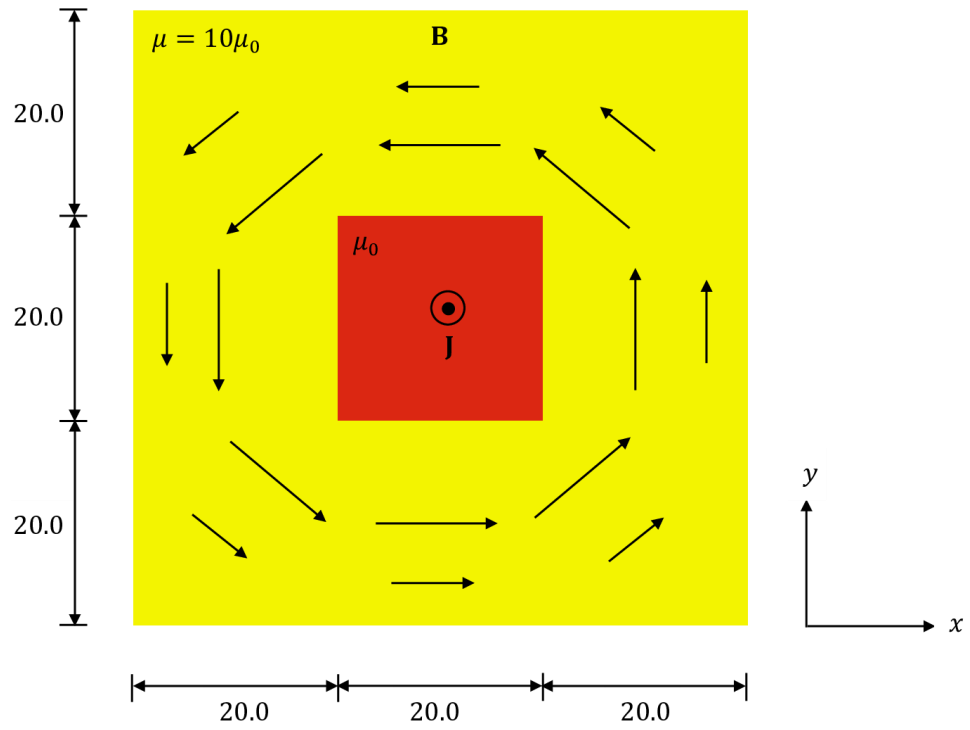


Figura 4.1. Geometría de bloque y barra conductora, dada en mm.

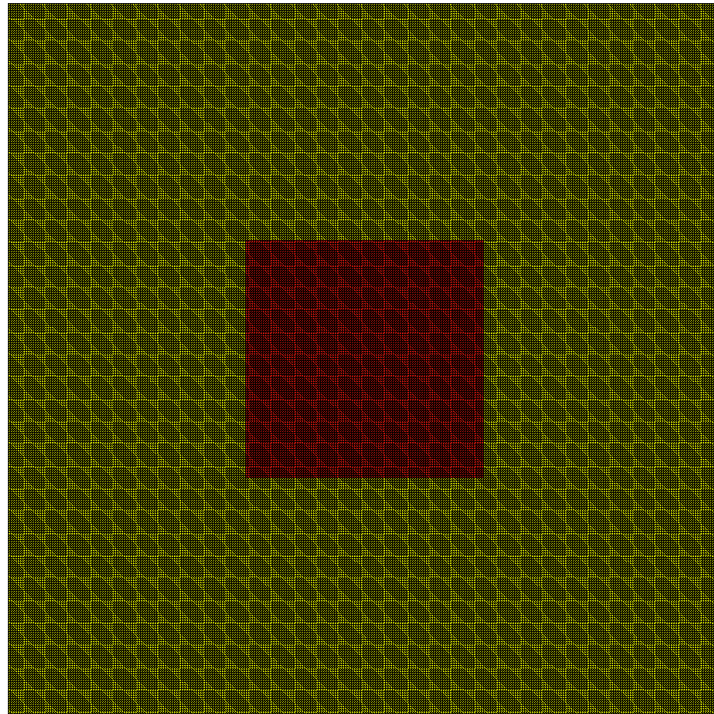


Figura 4.2. Malla de modelo barra conductora conformada por elementos triangulares.

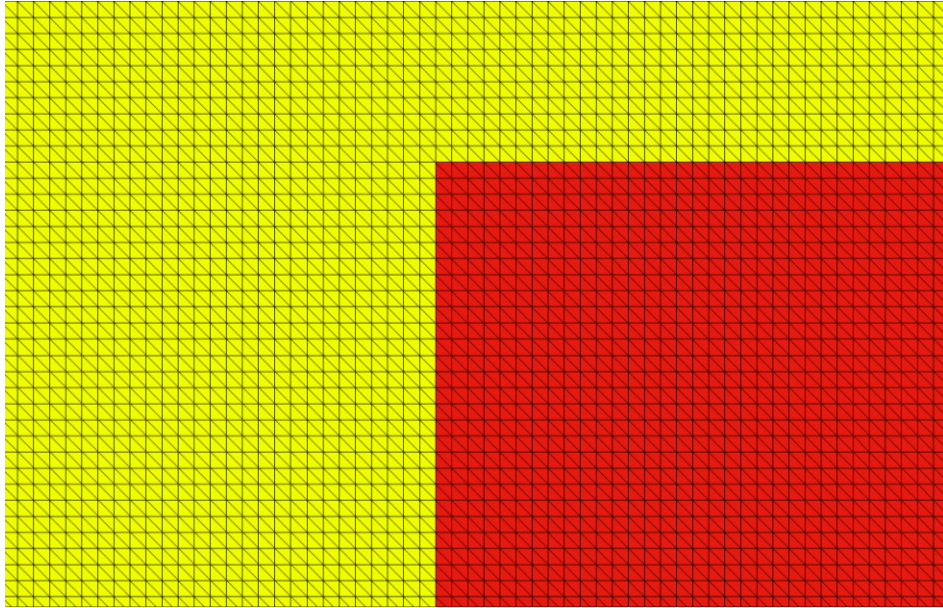


Figura 4.3. Acercamiento de malla del modelo barra conductora para mejorar visualización de elementos triangulares.

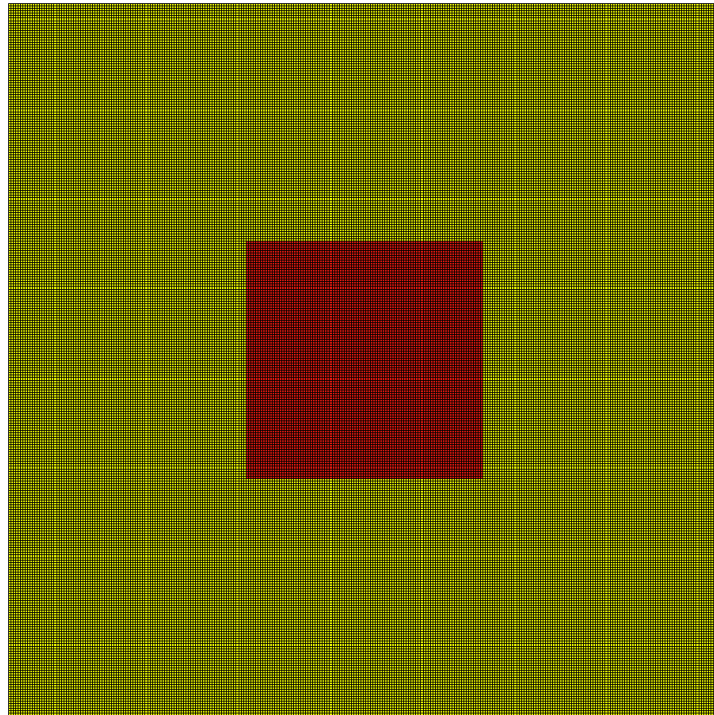


Figura 4.4. Malla de barra conductora conformada por elementos cuadriláteros.

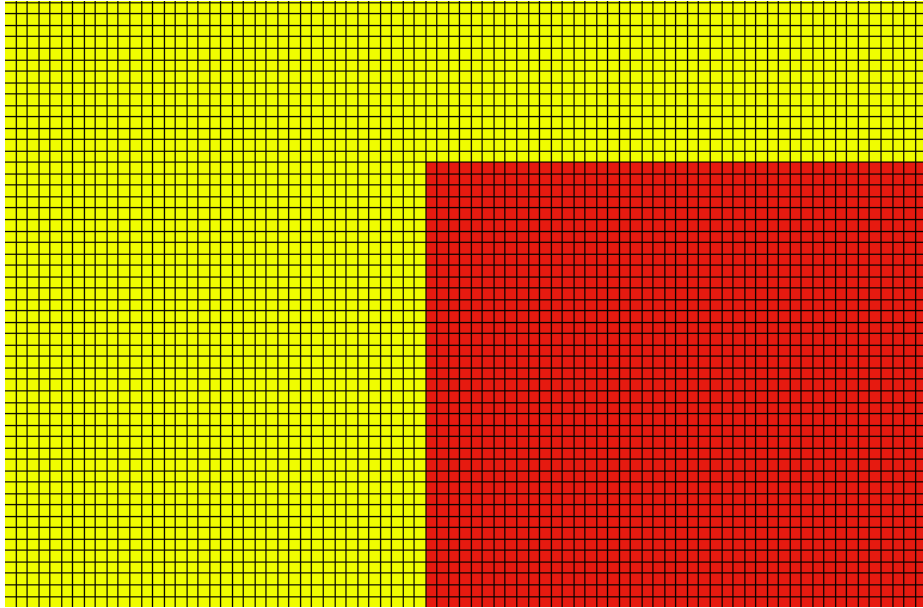


Figura 4.5. Acercamiento de malla de modelo barra conductora para mejorar visualización de elementos cuadriláteros.

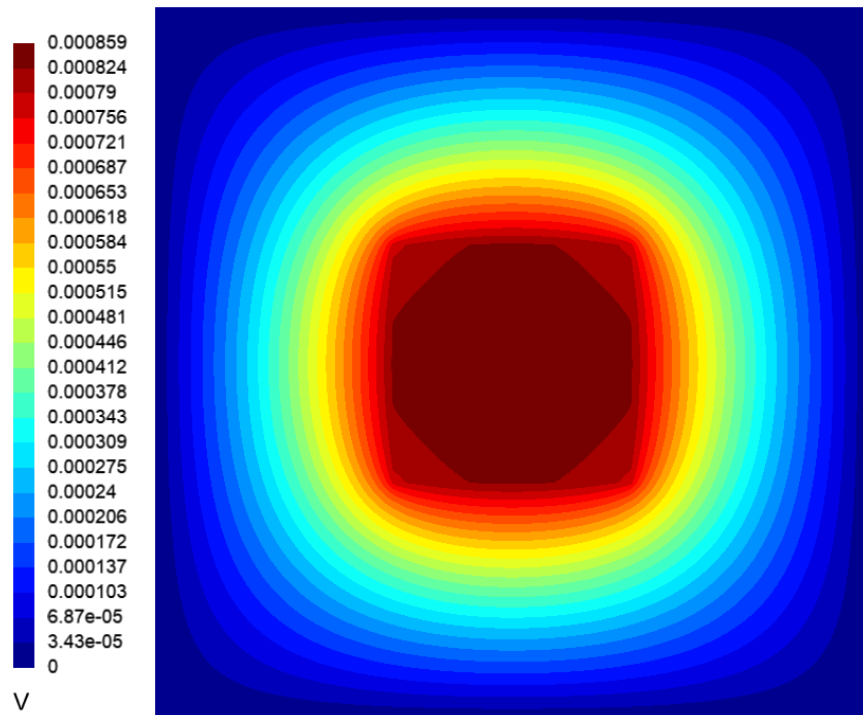


Figura 4.6. Distribución de los potenciales eléctricos (V) en bloque y barra conductora, obtenidos con elementos triangulares y solver INTEL MKL PARDISO.

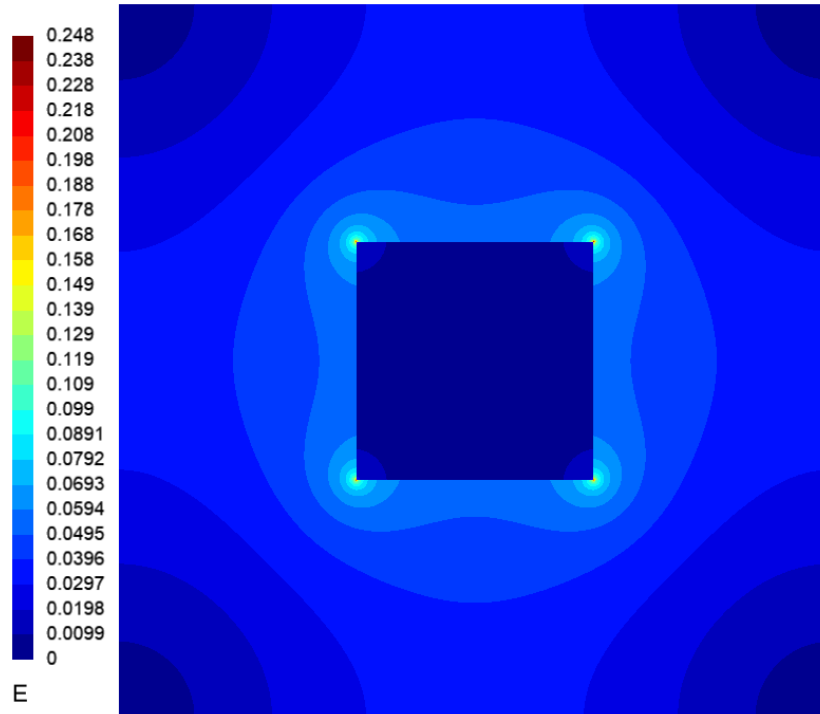


Figura 4.7. Distribución de la magnitud de campo eléctrico (V/m) en bloque y barra conductora, obtenidos con elementos triangulares y solver INTEL MKL PARDISO.

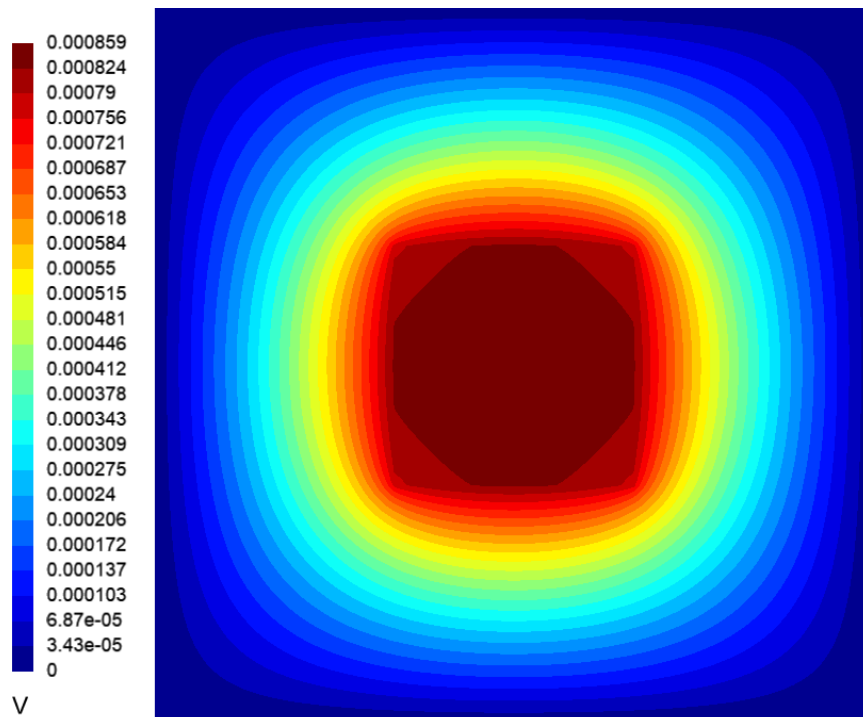


Figura 4.8. Distribución de los potenciales eléctricos (V) en bloque y barra conductora, obtenidos con elementos triangulares y solver ICCG CUDA.

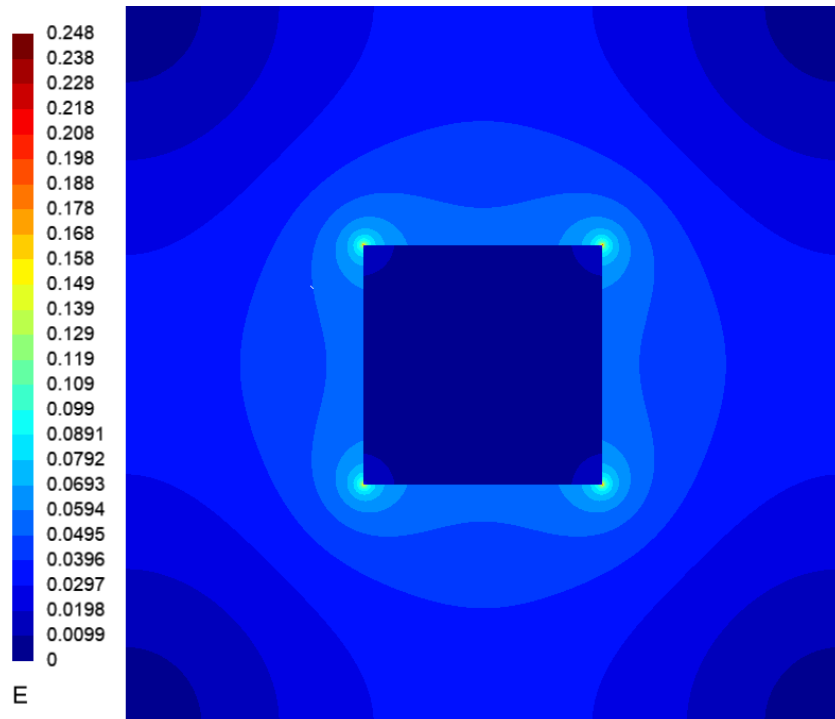


Figura 4.9. Distribución de la magnitud del campo eléctrico (V/m) en bloque y barra conductora, obtenidos con elementos triangulares y solver ICCG CUDA.

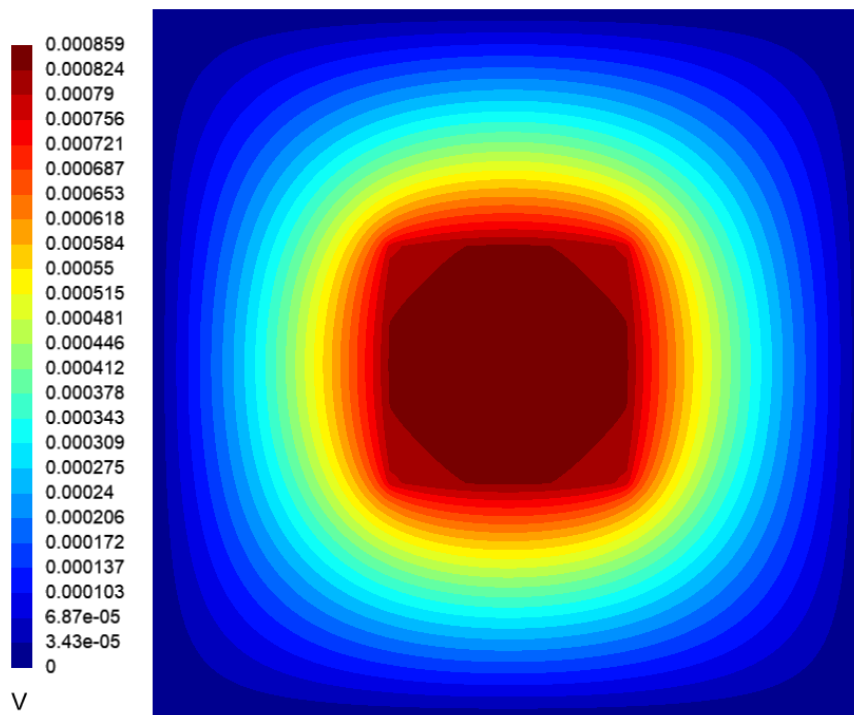


Figura 4.10. Distribución de los potenciales eléctricos (V) en bloque y barra conductora, obtenido con elementos cuadriláteros y solver INTEL MKL PARDISO.

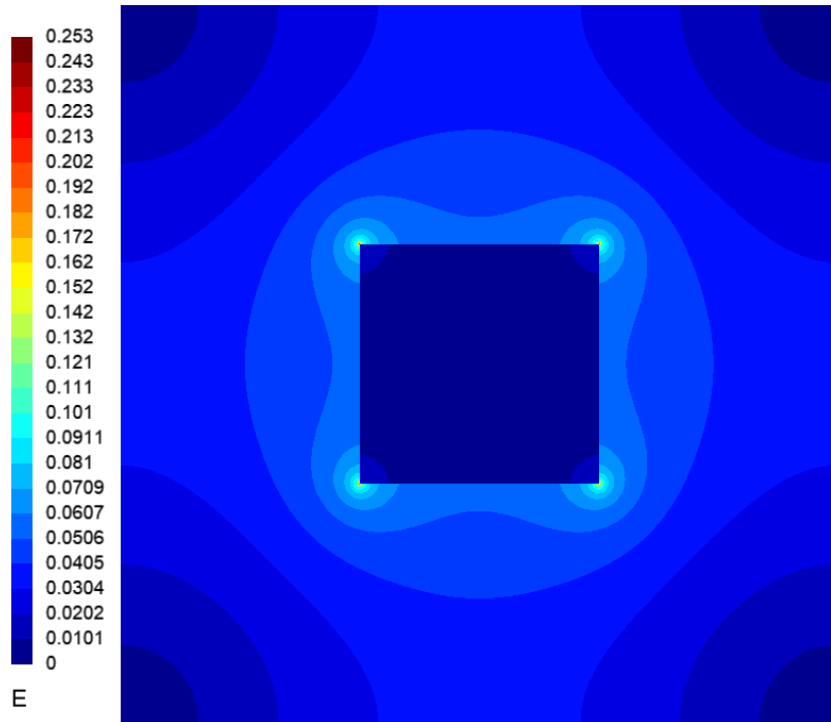


Figura 4.11. Distribución de la magnitud del campo eléctrico (V/m) en bloque y barra conductora, obtenido con elementos cuadriláteros y solver INTEL MKL PARDISO.

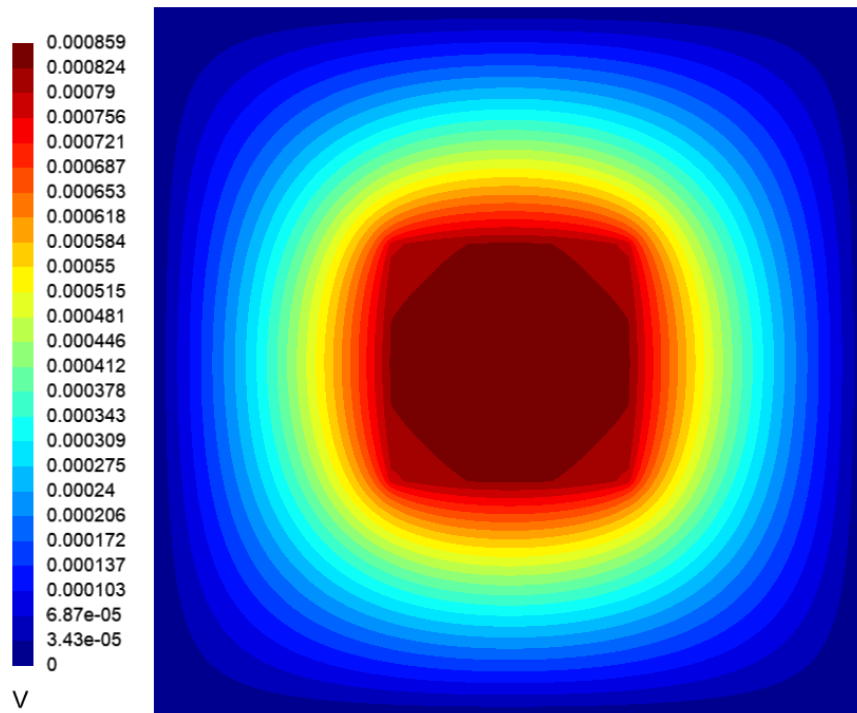


Figura 4.12. Distribución de los potenciales eléctricos (V) en bloque y barra conductora, obtenido con elementos cuadriláteros y solver ICCG CUDA.

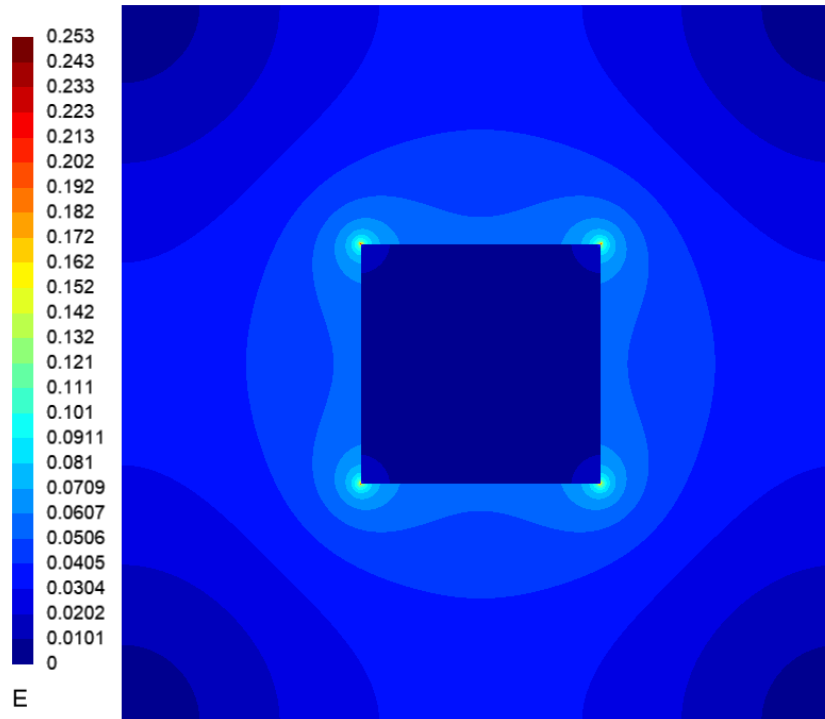


Figura 4.13. Distribución de la magnitud del campo eléctrico (V/m) en bloque y barra conductora, obtenido con elementos cuadriláteros y solver ICCG CUDA.

4.1.2 Caso 2

Utilizando el modelo magnetostático desarrollado en las referencias [43]-[44], el cual corresponde a un conductor coaxial que se encuentra compuesto por tres regiones: un núcleo magnético y 2 conductores eléctricos (interno y externo). La permeabilidad del espacio libre es asignada en ambos conductores mientras que la permeabilidad del núcleo es de $10000\mu_0$. En cada conductor es aplicada una fuerza magnetomotriz de 70000 AT en dirección opuesta del eje z.

En las Figuras 4.14 y 4.15 es mostrado la geometría y el mallado del conductor coaxial que contiene 19,279 nodos y 9,504 elementos de segundo orden. Se puede apreciar, que son utilizados elementos triangulares para la formación de la malla y solamente es requerido un cuarto del modelo para la obtención de resultados numéricos.

Las figuras 4.16 y 4.17 muestran la distribución de los potenciales eléctricos y del campo magnético desarrollado en el modelo del conductor coaxial, respectivamente. Obtenido con solver ICCG CUDA.

La solución analítica para determinar la distribución de la intensidad del campo magnético en el conductor [45], es mostrado a continuación:

$$\mathbf{H} = \begin{cases} \frac{rI}{2\pi a^2} \hat{a}_\theta, & r \leq a \\ \frac{I}{2\pi r} \hat{a}_\theta, & a < r \leq b \\ \frac{I(c^2 - r^2)}{2\pi r(c^2 - b^2)} \hat{a}_\theta, & b < r \leq c \end{cases} \quad (4.1)$$

donde I es la corriente aplicada en el conductor, b representa el radio del núcleo magnético, a y c representan el radio de los conductores interno y externo.

El software Octave [46], contiene una fuerte sintaxis orientado a las matemáticas para el cálculo numérico y herramientas de visualización. La Figura 4.18, muestra la comparación realizada en Octave, entre los valores de intensidad de campo magnético obtenidos en la solución analítica (resuelta en Octave), y los valores obtenidos por el software de elementos finitos. La Figura 4.19 muestra el porcentaje de error entre ambos.

El tiempo requerido de simulación al utilizar la malla conformada con elementos triangulares, es mostrado en la Tabla 4.3. En la Tabla 4.4, se muestra el tiempo requerido de simulación al utilizar una malla que contiene 14, 527 nodos y 4,752 elementos cuadriláteros de segundo orden.

Tabla 4.3. Tiempo requerido de simulación de modelo cable coaxial, utilizando elementos triangulares.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	118,841	14
ICCG CUDA	218,403	10

Tabla 4.4. Tiempo requerido de simulación de modelo cable coaxial, utilizando elementos cuadriláteros.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	118,706	16
ICCG CUDA	222,885	11

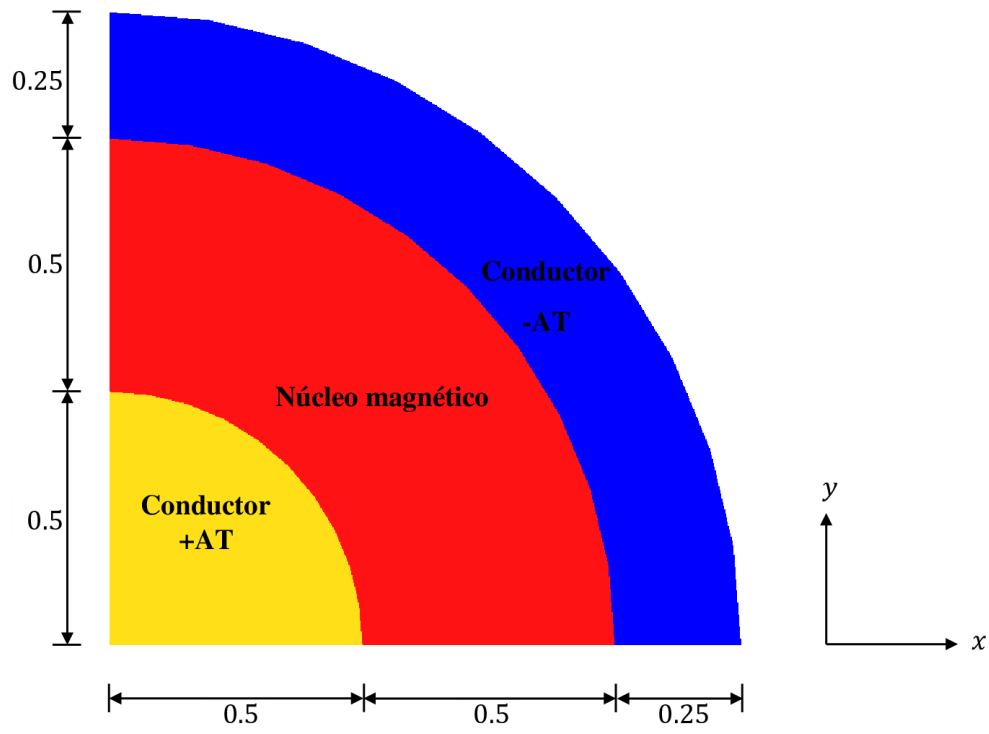


Figura 4.14. Geometría del conductor coaxial, dada en m.

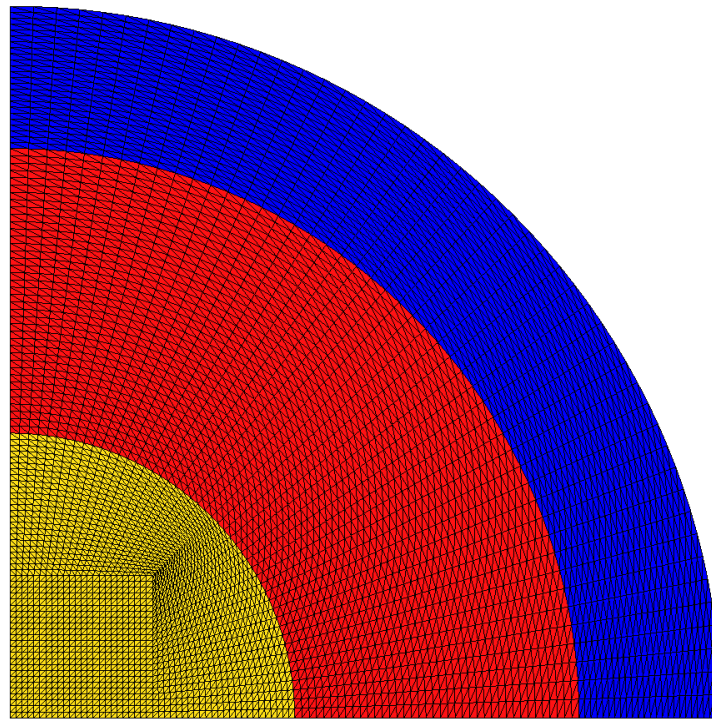


Figura 4.15. Malla del conductor coaxial conformada por elementos triangulares.

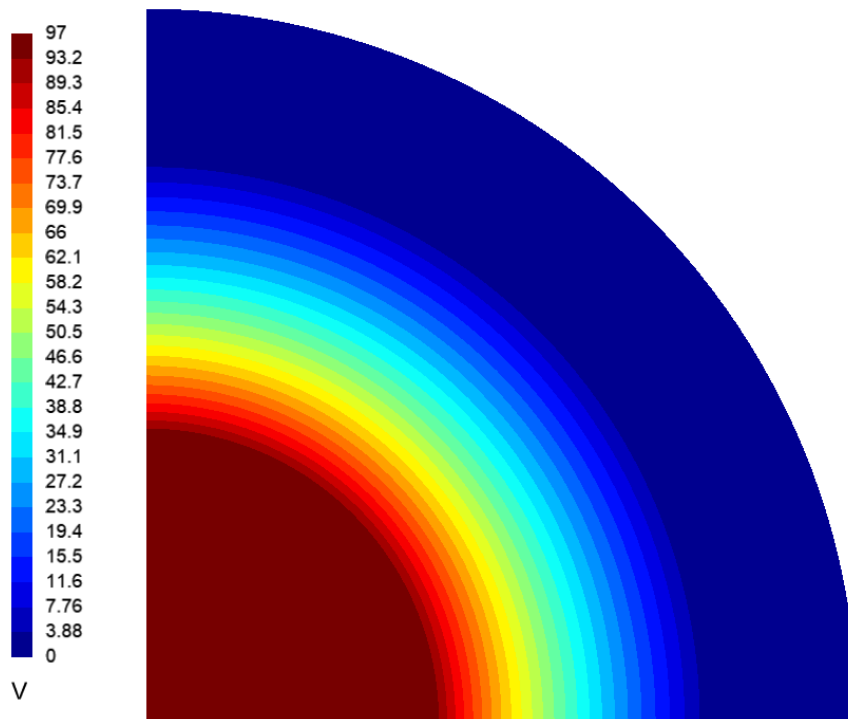


Figura 4.16. Distribución de los potenciales eléctricos (V) en conductor coaxial.

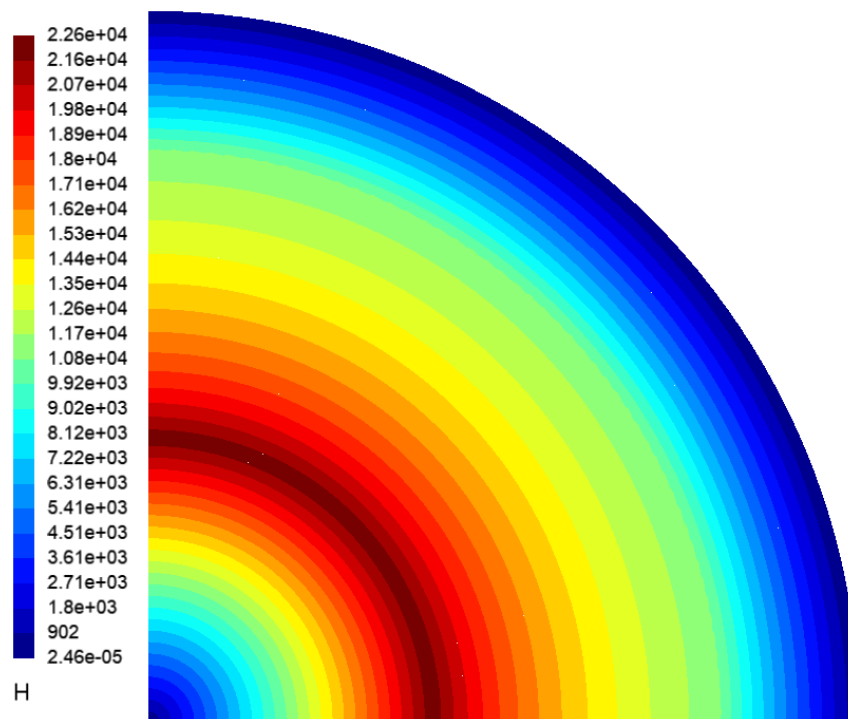


Figura 4.17. Distribución del campo magnético (A/m) en conductor coaxial.

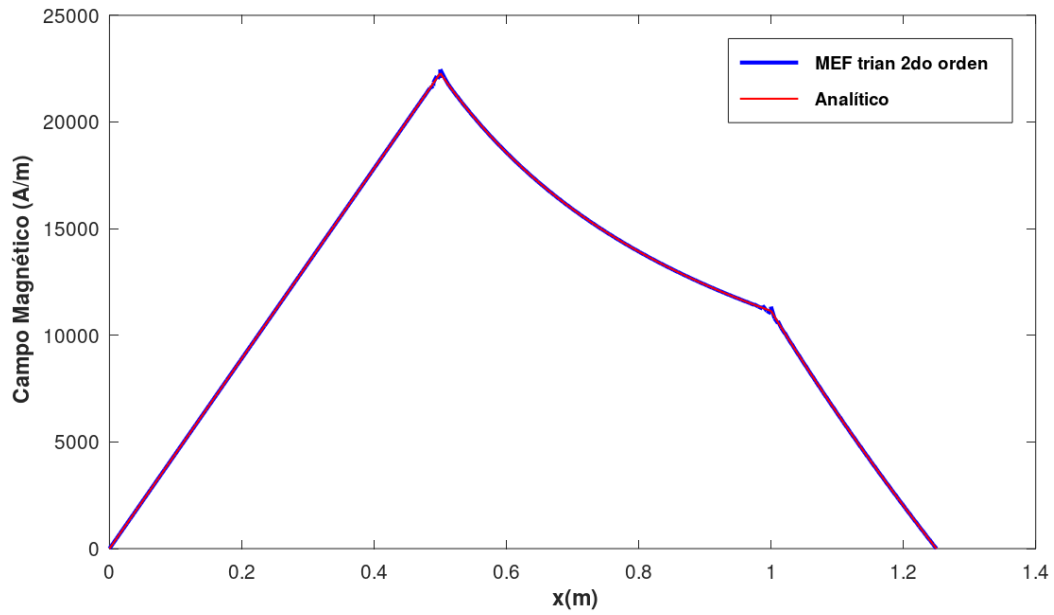


Figura 4.18. Comparación entre los valores de la solución analítica y el software de elementos finitos, desde la línea recta del conductor.

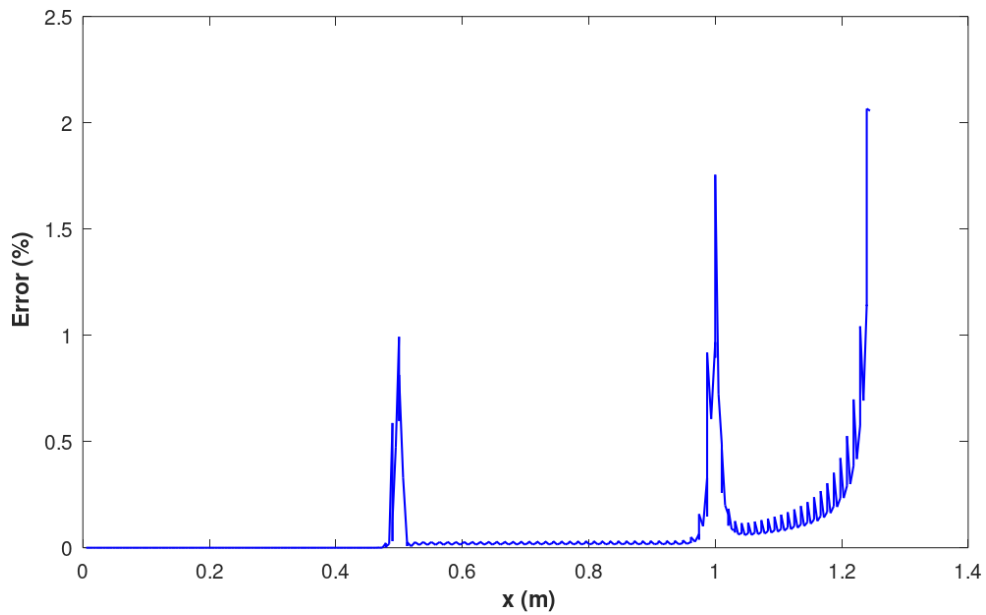


Figura 4.19. Porcentaje de error entre los valores de la solución analítica y el software de elementos finitos.

4.2 Simulación de modelos electrostáticos

4.2.1 Caso 1

Tomando el modelo electrostático desarrollado para fines demostrativos en la referencia [41], en el cual, es requerido determinar la distribución de los potenciales y la intensidad de campo eléctrico en una simple geometría, mostrada en la Figura 4.20. La parte superior se encuentra conectado a 100 V, mientras que la parte inferior se encuentra aterrizada. La región de color verde tiene la permitividad del espacio libre asignado y la región de color gris tiene una permitividad de $\epsilon_1 = 5\epsilon_0$.

La malla del modelo electrostático elaborada y mostrada en la Figura 4.21, contiene 442,225 nodos y 442,225 elementos triangulares de segundo orden. Debido a la gran cantidad de elementos que conforman la malla, en la Figura 4.22 es mostrada la malla del modelo electrostático con un mayor acercamiento, con la finalidad de una mejor visualización de los elementos triangulares.

La distribución de los potenciales eléctricos y el campo eléctrico obtenidos en la simulación del modelo, son mostrados en las Figuras 4.23 y 4.24 respectivamente.

El tiempo requerido en la simulación del modelo utilizando la malla conformada por elementos triangulares, es mostrado en la Tabla 4.5.

En la simulación del modelo con elementos cuadriláteros, fue utilizada una malla que contiene 332,001 nodos y 110,224 elementos, en el cual, el tiempo requerido de simulación es mostrado en la Tabla 4.6.

Tabla 4.5. Tiempo requerido de simulación de modelo electrostático, utilizando elementos triangulares.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	2,750,221	208
ICCG CUDA	5,058,217	237

Tabla 4.6. Tiempo de solución requerido de modelo electrostático, utilizando elementos cuadriláteros.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	2,748,269	217
ICCG CUDA	5,164,537	169

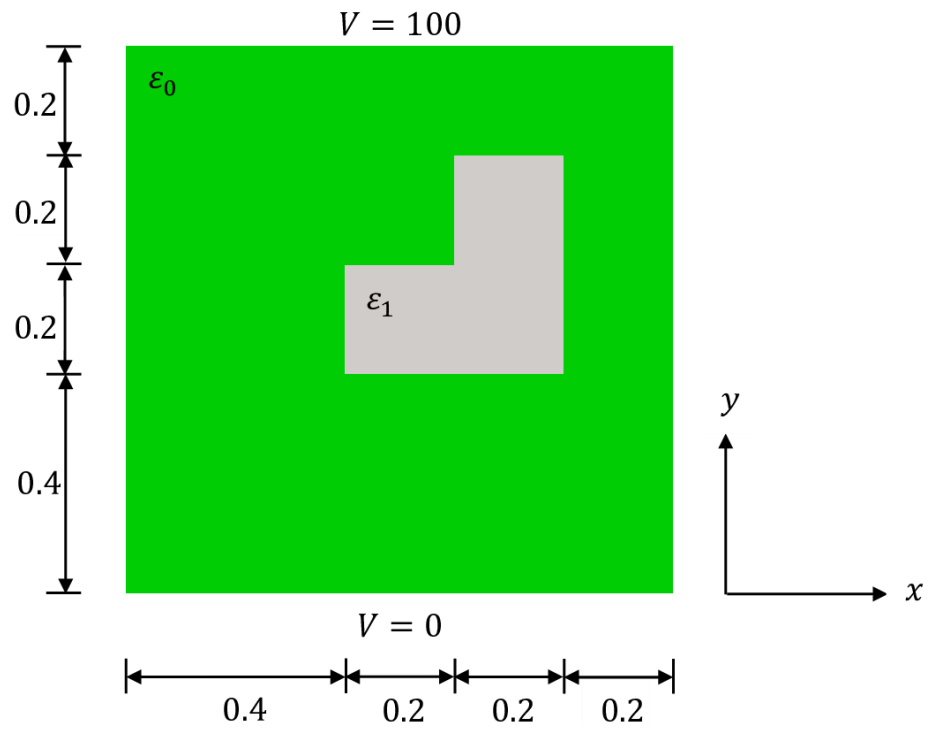


Figura 4.20. Dimensiones geométricas de modelo electrostático, dado en mm.

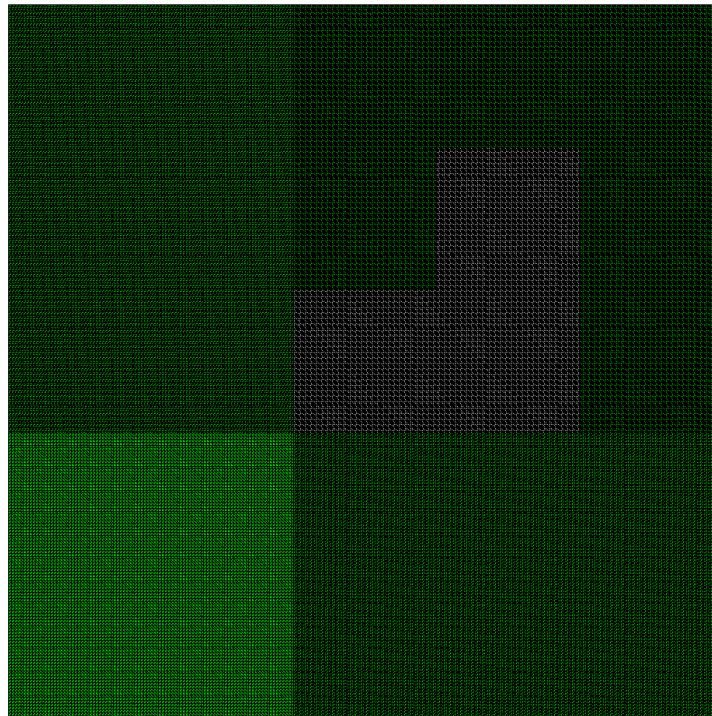


Figura 4.21. Malla de modelo electrostático conformada por elementos triangulares.

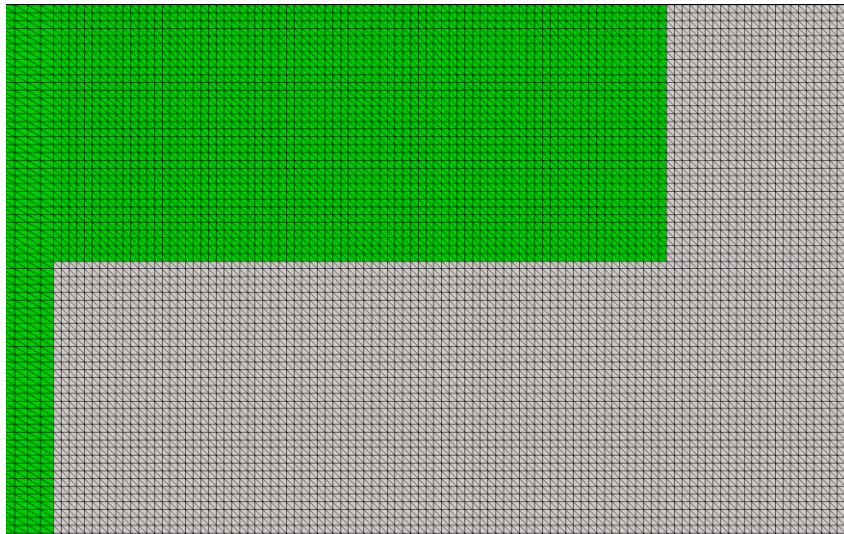


Figura 4.22. Acercamiento de malla para visualización de los elementos triangulares que lo conforman.

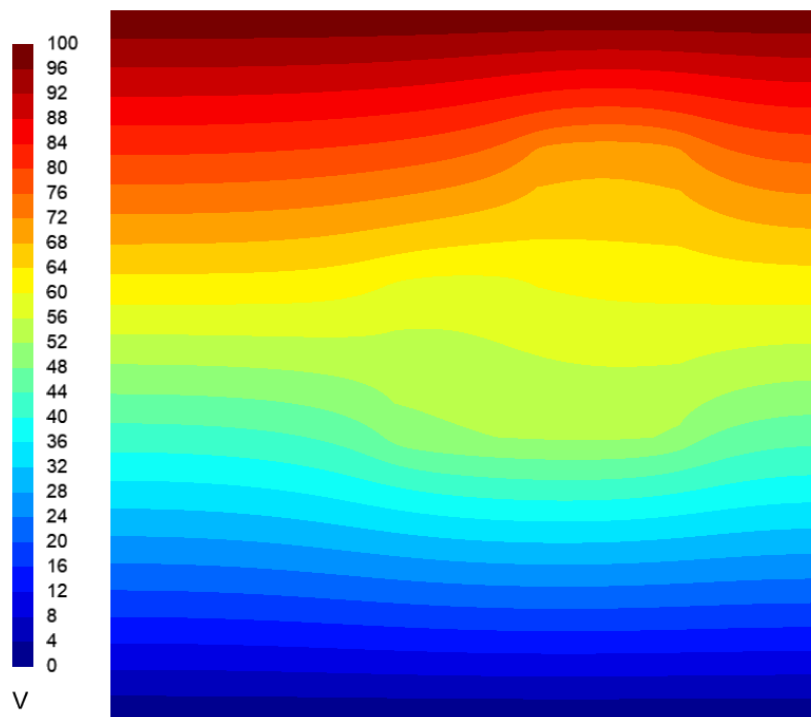


Figura 4.23. Distribución de los potenciales eléctricos (V) en modelo electrostático.

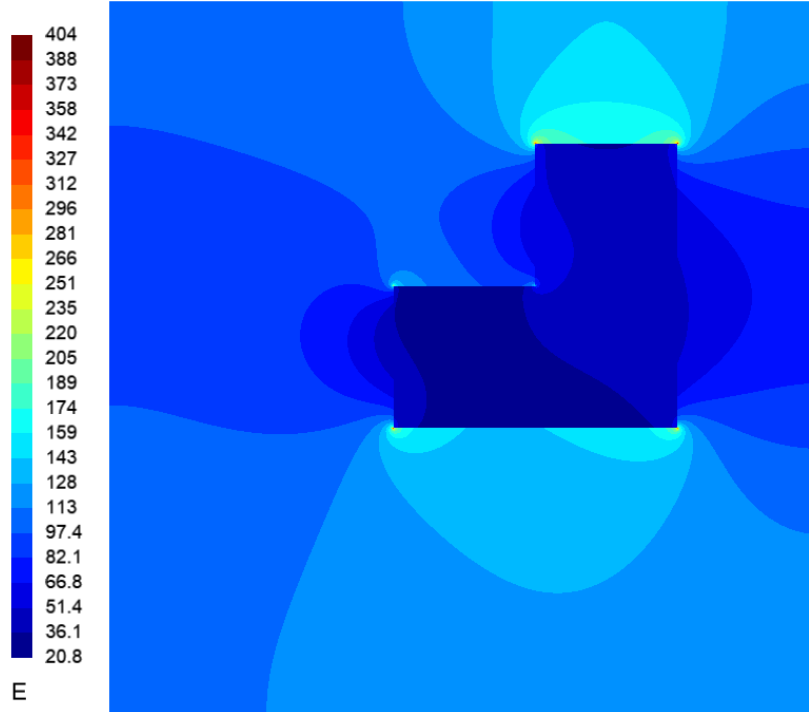


Figura 4.24. Distribución de la magnitud del campo eléctrico (V/m) en modelo electrostático.

4.2.2 Caso 2

Desarrollando el modelo electrostático tomado de la referencia [47], el cual consta de un conductor de longitud infinita con una sección transversal cuadrada, como se muestra en la Figura 4.25. La pared del canal en $y = 1$ está conectada a 100 V, mientras que las otras paredes se encuentran aterrizadas.

La malla del modelo conductor elaborado en Gmsh, contiene 1,225 nodos y 578 elementos triangulares de segundo orden, como se muestra en la Figura 4.26.

La distribución de los potenciales eléctricos obtenidos por el software de elementos finitos se muestra en la Figura 4.27.

Para determinar la distribución de los potenciales eléctricos en el conductor, es utilizada la solución analítica desarrollada en [47]:

$$V(x, y) = \frac{400}{\pi} \sum_{n=0}^{\infty} \frac{\sin(k\pi x) \sinh(k\pi y)}{k \sinh(k\pi)} \quad k = 2n + 1 \quad (4.2)$$

Utilizando Octave, es realizada la resolución de la ecuación (4.2) y los resultados obtenidos se muestran en la Figura 4.28. Posteriormente, es realizada su comparación con los potenciales obtenidos con el software de elementos finitos, como se muestra en la Figura 4.29.

Con la finalidad de determinar la exactitud del software de elementos finitos, es determinado el porcentaje de error entre ambos resultados, como se muestra en la Figura 4.30.

El tiempo requerido de simulación utilizando elementos triangulares, se muestra en la Tabla 4.7.

Para la simulación del modelo conductor mediante la malla conformada por elementos cuadráticos, es utilizada una malla que contiene 936 nodos y 289 elementos. En la Tabla 4.8, se muestra el tiempo requerido de simulación del modelo.

Tabla 4.7. Tiempo requerido de simulación de modelo conductor, utilizando elementos triangulares.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	6,538	1
ICCG CUDA	11,851	1

Tabla 4.8. Tiempo requerido de simulación de modelo conductor, utilizando elementos triangulares.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	6344	1
ICCG CUDA	11,752	1

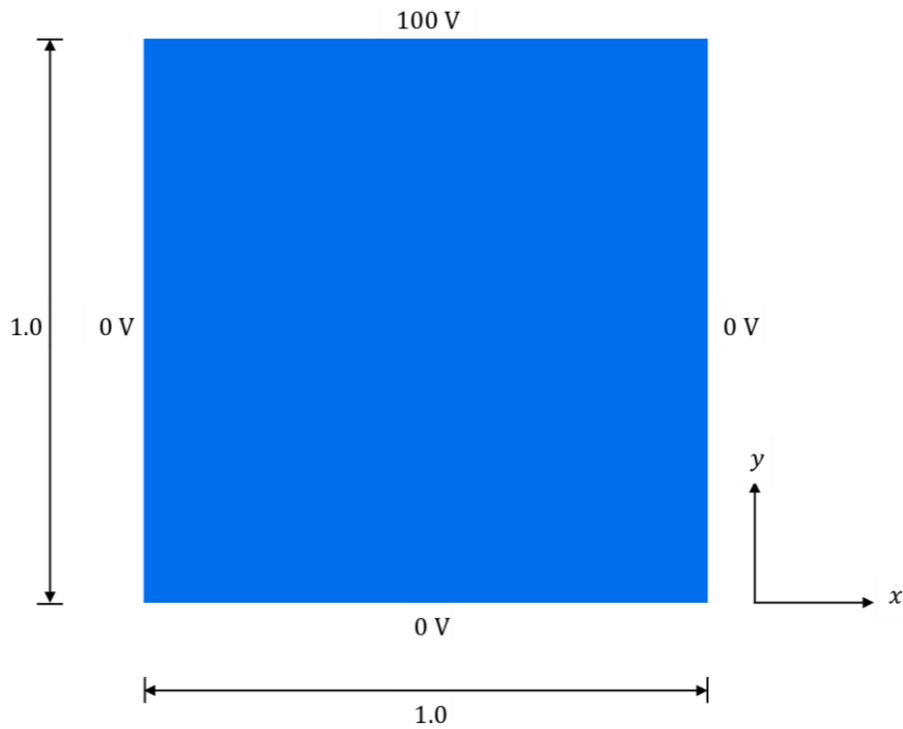


Figura 4.25. Dimensiones geométricas de conductor, dada en m.

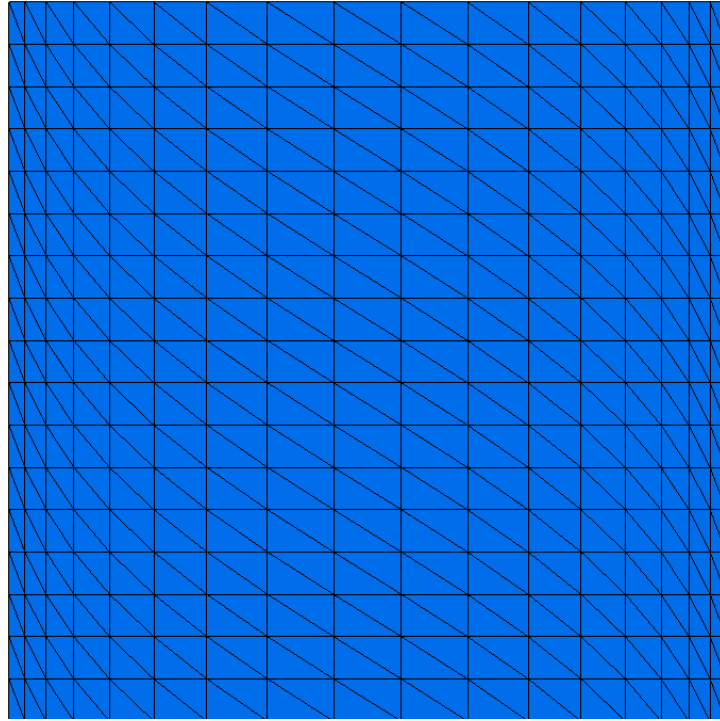


Figura 4.26. Malla de modelo conductor conformada por elementos triangulares.

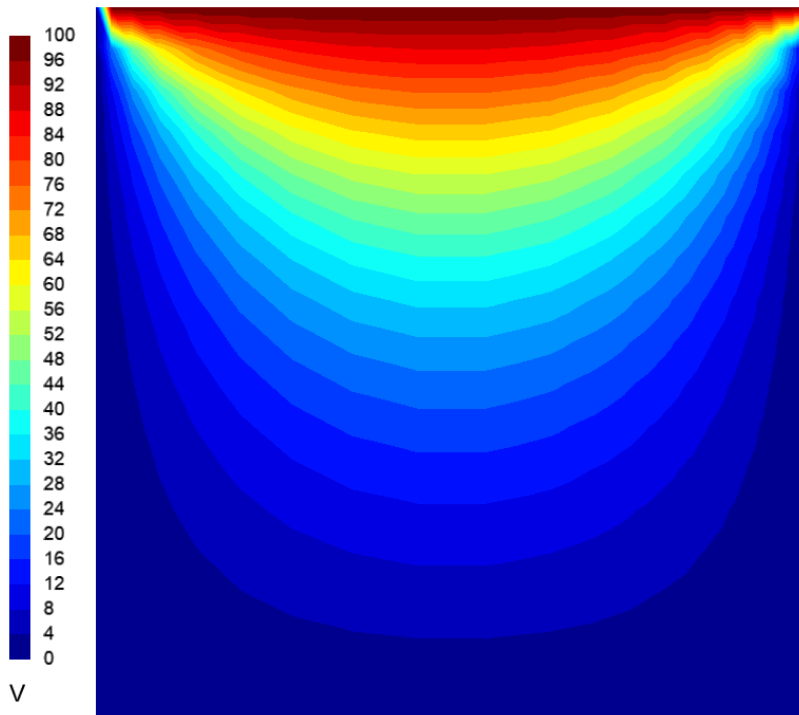


Figura 4.27. Distribución de los potenciales eléctricos (V) en conductor, obtenidas por software de elementos finitos.

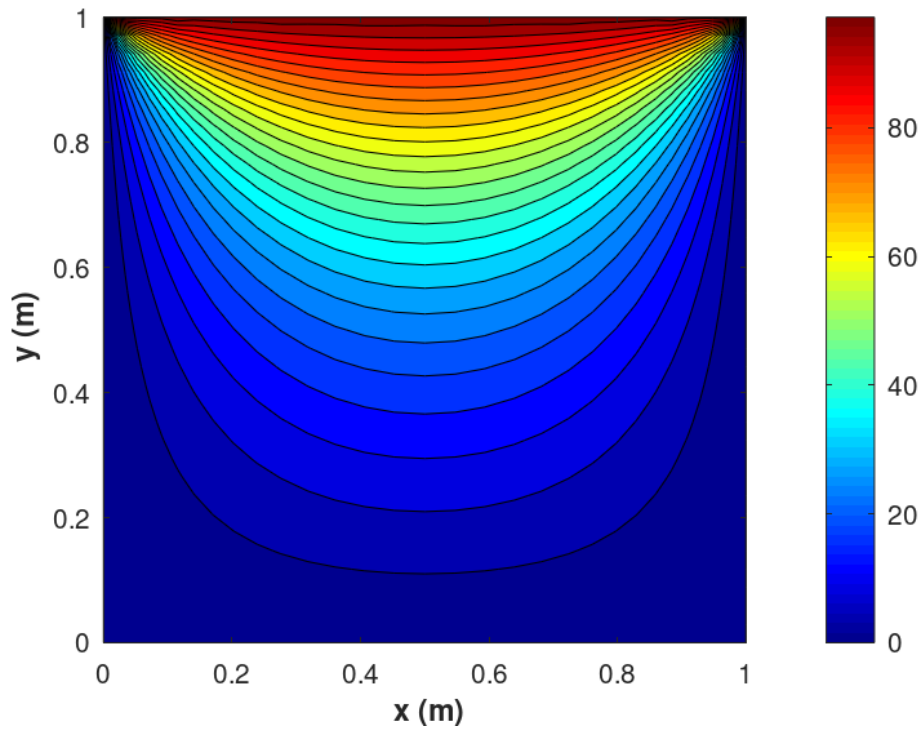


Figura 4.28. Distribución de los potenciales eléctricos (V) en conductor, obtenida por solución analítica.

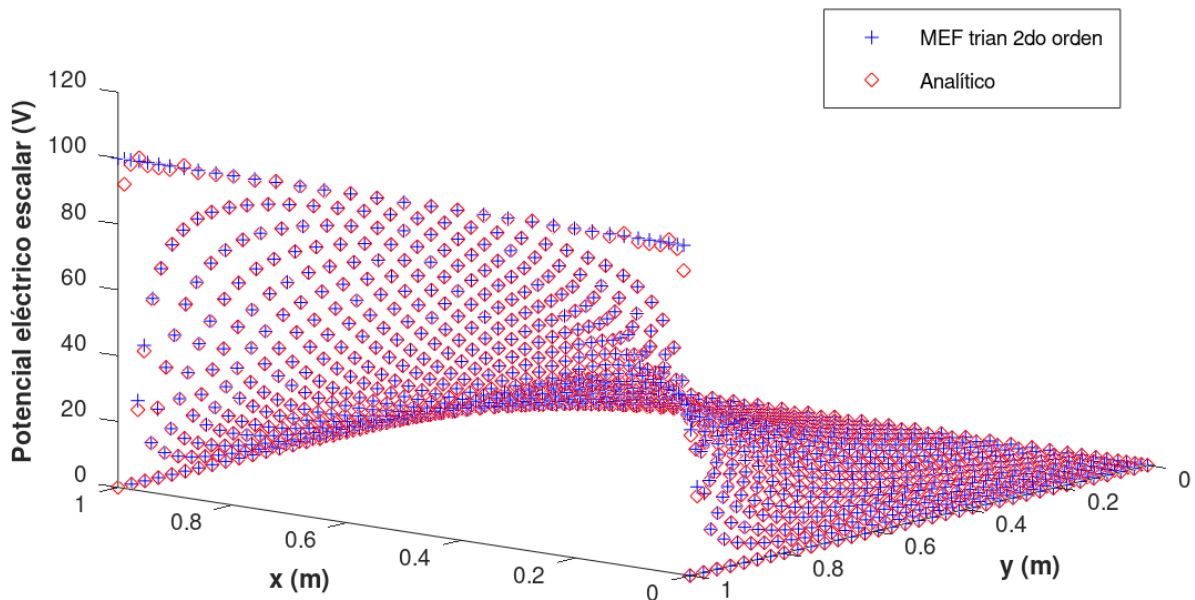


Figura 4.29. Comparación entre los potenciales obtenidos en la solución analítica y el software de elementos finitos.

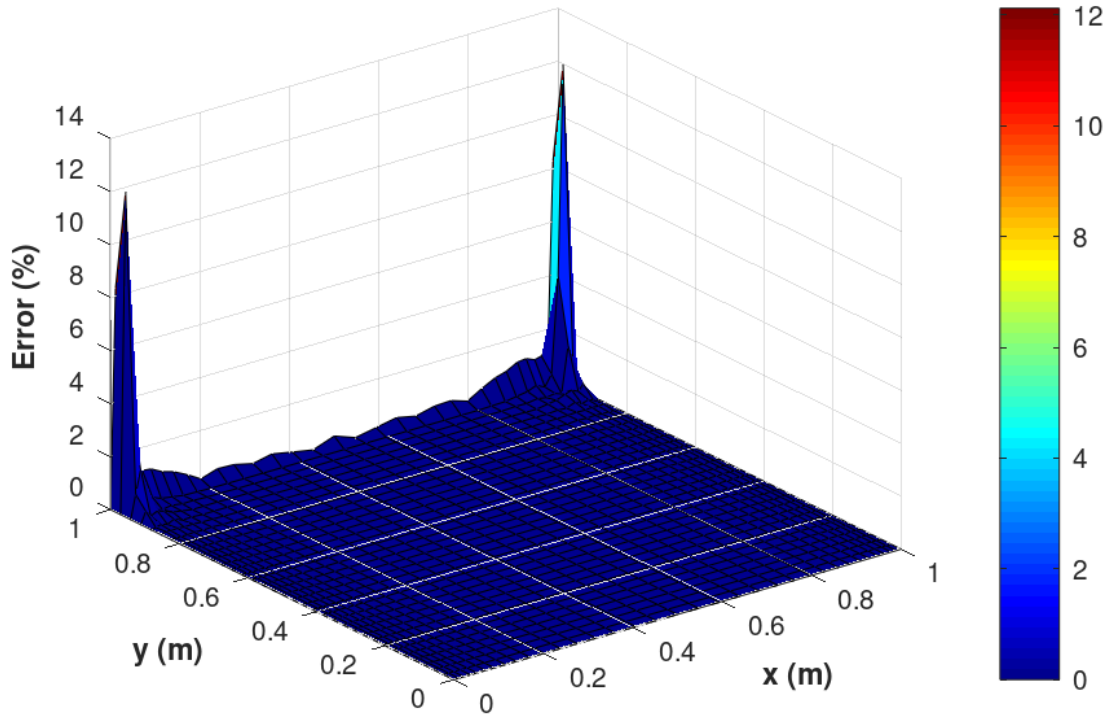


Figura 4.30. Porcentaje de error entre los potenciales de la solución analítica y el software de elementos finitos.

4.2.3 Caso 3

En el proceso de refinación del petróleo para la obtención de diversos productos como hidrocarburos, gasolina, diésel, entre otros. La sustancia requiere pasar por diversos filtros que ocasionan un proceso de electrificación y su conductividad eléctrica sea demasiada alta [48]-[49]. Al ser almacenado en un tanque metálico, el potencial eléctrico generado es demasiado elevado y al ser colocado en una atmosfera explosiva, puede originar una ignición.

El modelo bajo estudio tomado de la referencia [50], consiste en determinar el potencial y campo eléctrico generado en un tanque electrostático aterrizado. En su interior contiene un líquido electrostático y una porción de gas o aire como se muestra en la Figura 4.31. La región de color gris corresponde a la porción de gas o liquido el cual tiene un $\epsilon_0 = 8.8541878128 \times 10^{-5}$ F/m. La región de color azul corresponde al líquido electrostático el cual tiene una $\rho = 10 \times 10^{-5}$ C/m² y $\epsilon_r = 2.5\epsilon_0$.

Utilizando Gmsh, es generada la malla del tanque electrostático como se muestra en la Figura 4.32. La malla generada, contiene 24,649 nodos y 12,168 elementos triangulares de segundo orden.

Los resultados obtenidos por el software de elementos finitos utilizando el solver ICCG CUDA, son mostrados en las Figuras 4.33 y 4.34, en el cual es visualizado la distribución del potencial y campo eléctrico, respectivamente.

Para validar los potenciales eléctricos obtenidos en el software de elementos finitos, es utilizada la solución analítica desarrollada en la referencia [50], la cual describe la distribución de los potenciales eléctricos en el tanque:

$$V_g = \sum_{n=1}^{\infty} A_n \sin(\beta_x) \sinh(\beta[c-y]) \quad (4.3)$$

$$V_L = \sum_{n=1}^{\infty} \sin(\beta_x) [C_n \sinh(\beta_y) - F_p \{\cosh(\beta_y) - 1\}] \quad (4.4)$$

donde A y β son coeficientes dados por la solución, V_g corresponde al potencial en la región del gas, V_L corresponde al potencial en la región del líquido y el resto de las variables corresponden a:

$$F_p = \frac{4pa^2}{n^3\pi^3\epsilon} \quad (4.5)$$

$$C_n = F_p \left[\frac{\epsilon_r \sinh(\beta b) \sinh(\beta[c-b]) + \cosh(\beta[c-b]) \{\cosh(\beta b) - 1\}}{\epsilon_r \cosh(\beta b) \sinh(\beta[c-b]) + \sinh(\beta b) \cosh(\beta[c-b])} \right] \quad (4.6)$$

$$A_n = C_n \frac{\sinh(\beta b)}{\sinh(\beta[c-b])} - F_p \frac{\cosh(\beta b) - 1}{\sinh(\beta[c-b])} \quad (4.7)$$

$$\beta = \frac{n\pi}{a} \quad (4.8)$$

Al Derivar las ecuaciones (4.3) (4.4), es obtenido:

$$E_{x,g} = \sum_{n=1}^{\infty} A_n \beta \cos(\beta_x) \sinh(\beta[c-y]) \quad (4.9)$$

$$E_{y,g} = - \sum_{n=1}^{\infty} A_n \beta \sin(\beta_x) \cosh(\beta[c-y]) \quad (4.10)$$

$$E_{x,l} = \sum_{n=1}^{\infty} \beta \cos(\beta_x) [C_n \sinh(\beta_y) - F_p \{\cosh(\beta_y)\} + F_p] \quad (4.11)$$

$$E_{y,l} = \sum_{n=1}^{\infty} \beta \sin(\beta_x) [C_n \cosh(\beta_y) - F_p \sinh(\beta_y)] \quad (4.12)$$

donde:

$E_{x,g}$ = Componente x del campo eléctrico en la región del gas.

$E_{y,g}$ = Componente y del campo eléctrico en la región del gas.

$E_{x,l}$ = Componente x del campo eléctrico en la región líquida.

$E_{y,l}$ = Componente y del campo eléctrico en la región líquida.

Las ecuaciones (4.9) a (4.12) determinan el campo eléctrico generado en el tanque. Para determinar la magnitud del campo eléctrico es utilizado:

$$|\mathbf{E}| = \sqrt{E_x^2 + E_y^2} \quad (4.13)$$

La distribución de los potenciales y campo eléctrico obtenidos por la resolución de las soluciones analíticas en Octave, son mostradas en las Figuras 4.35 y 4.36.

Para validar los potenciales y campo eléctrico obtenidos por el software de elementos finitos, son comparados con los resultados obtenidos por las soluciones analíticas. En la Figura 4.37, se muestra la comparación a lo largo de $x = 0.4 \text{ m}$, $0 \leq y \leq 2 \text{ m}$. En la Figura 4.38, se muestra la comparación de los potenciales eléctricos generados en la interface entre el gas y el líquido contenidos en el tanque.

Para determinar la exactitud del software de elementos finitos, es determinado el porcentaje de error, como se muestran en las Figuras 4.39 y 4.40.

El tiempo de simulación requerido utilizando la malla conformada por elementos triangulares, es mostrado en la Tabla 4.7.

La malla conformada por elementos cuadriláteros de segundo orden, utilizado en la simulación del tanque electrostático, contiene 22,533 nodos y 7,396 elementos.

El tiempo de simulación requerido utilizando la malla conformada por elementos cuadriláteros, es mostrado en la Tabla 4.7.

Tabla 4.9. Tiempo requerido de simulación de modelo tanque, utilizando elementos triangulares.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	148,827	14
ICCG CUDA	273,005	9

Tabla 4.10. Tiempo requerido de simulación de modelo tanque, utilizando elementos cuadriláteros.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	180,288	17
ICCG CUDA	338,043	11

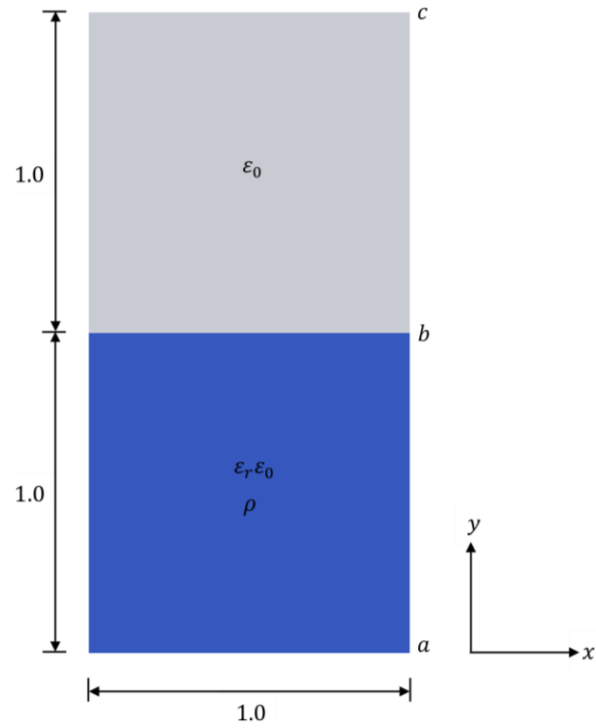


Figura 4.31. Dimensiones geométricas de tanque aterrizado que contiene un líquido con cargas electrostáticas, dada en m.

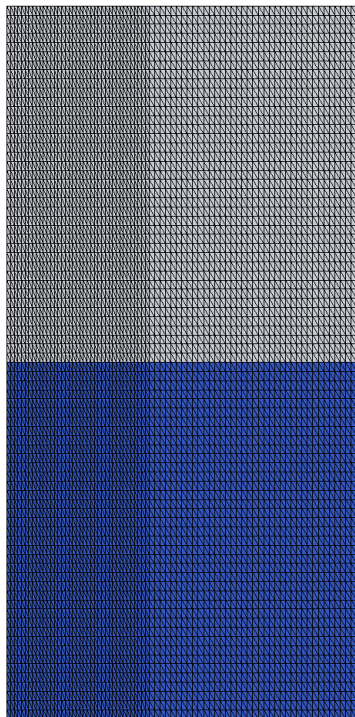


Figura 4.32. Malla de tanque aterrizado conformado por elementos triangulares.

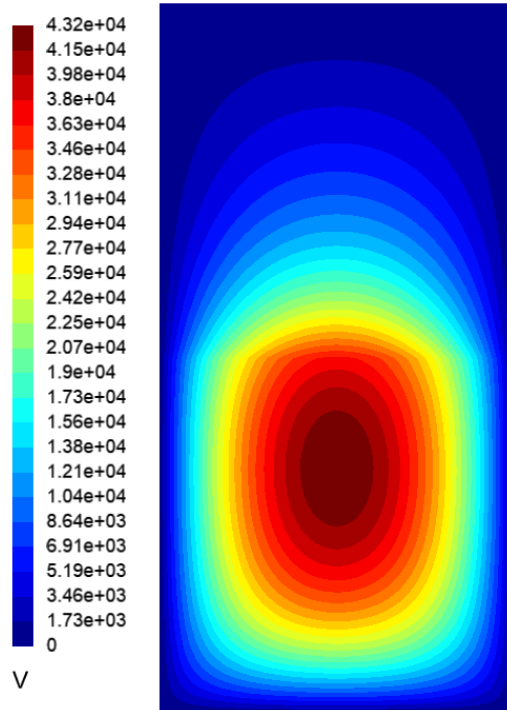


Figura 4.33. Distribución de los potenciales eléctricos (V) en tanque aterrizado (simulación).

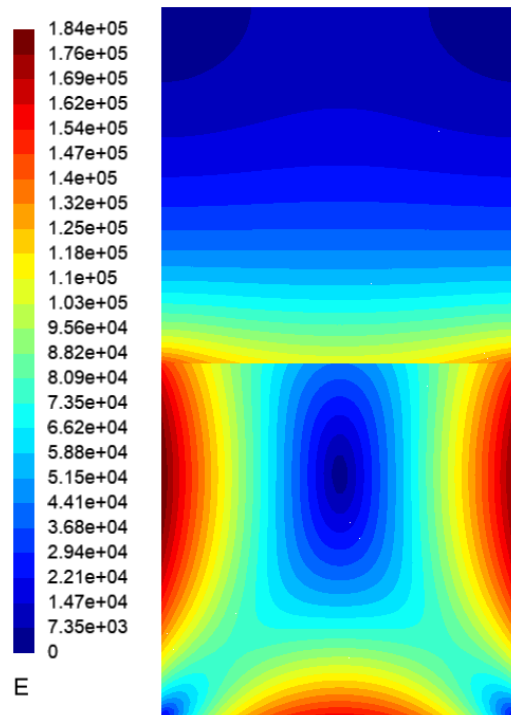


Figura 4.34. Distribución de la magnitud del campo eléctrico (V/m) en tanque aterrizado (simulación).

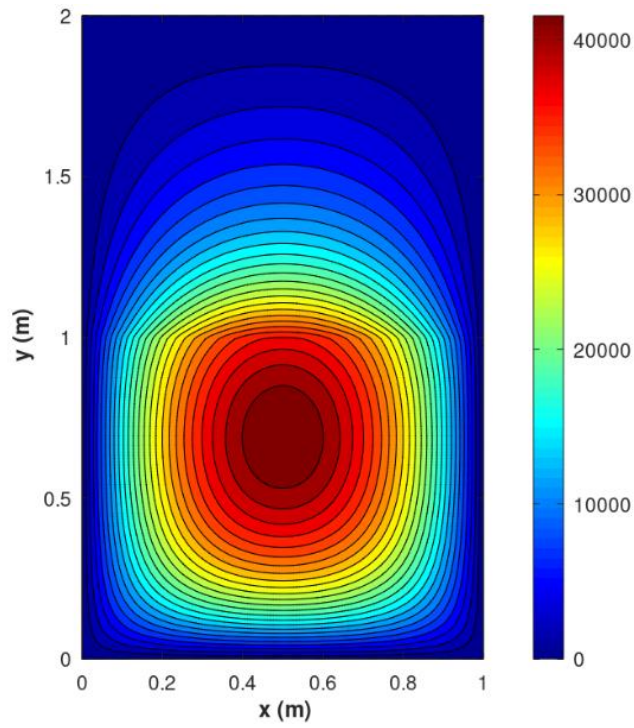


Figura 4.35. Distribución de los potenciales eléctricos (V) en tanque aterrizado (Solución analítica).

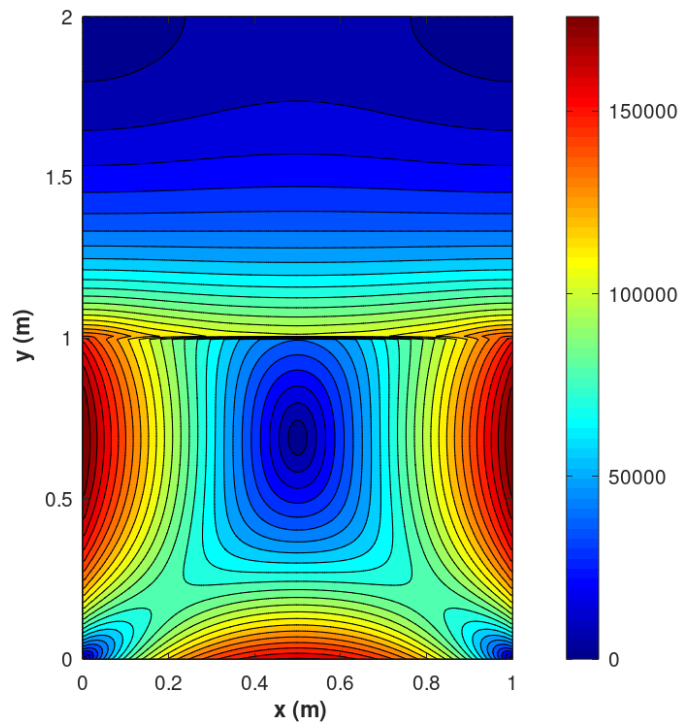


Figura 4.36. Distribución de la magnitud del campo eléctrico (V/m) en tanque aterrizado (Solución analítica).

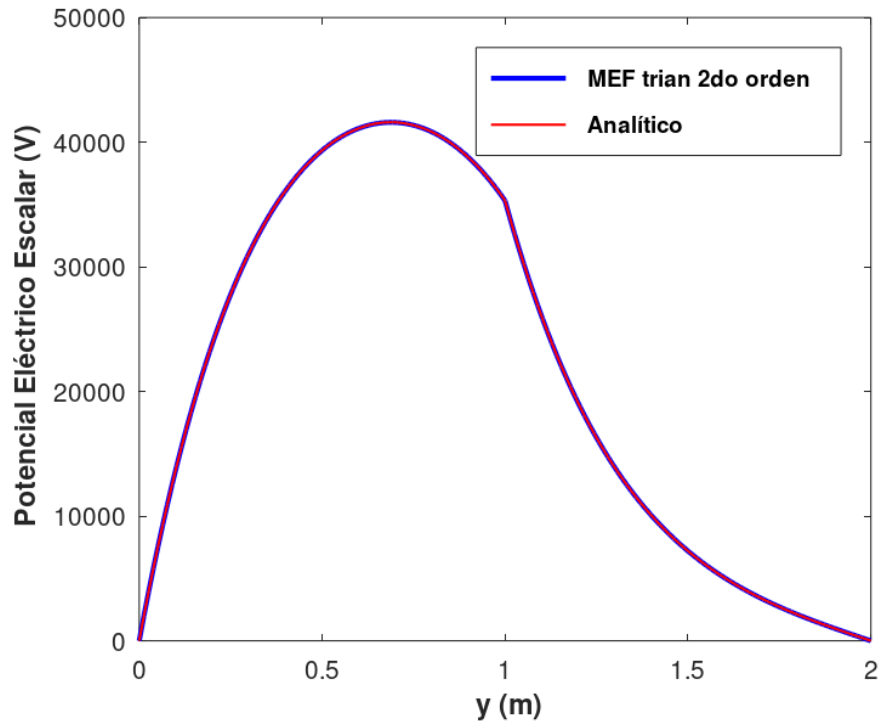


Figura 4.37. Comparación de la distribución de los potenciales eléctricos escalares a lo largo de $x = 0.4m, 0 \leq y \leq 2m$.

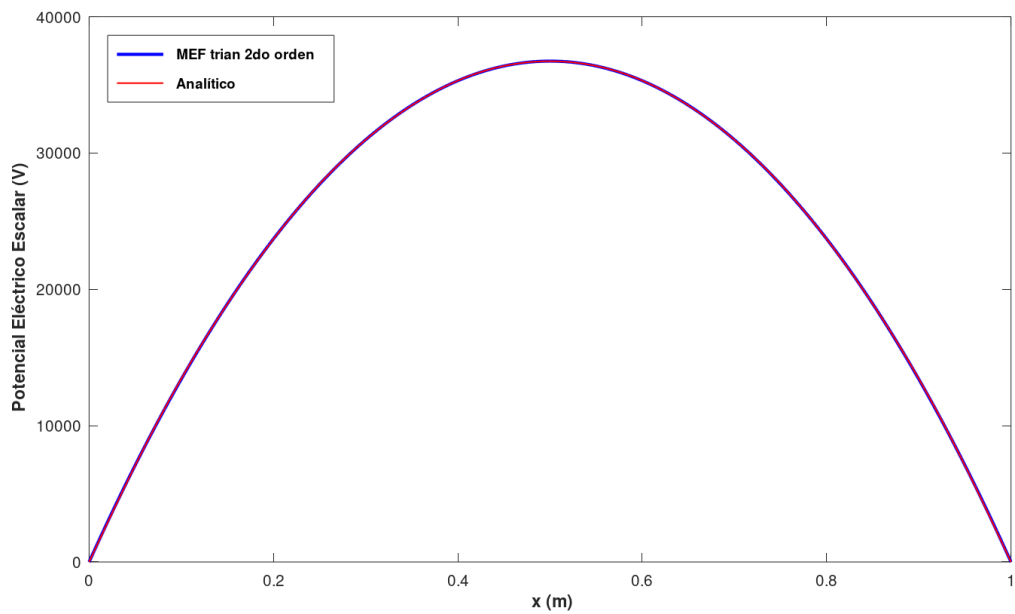


Figura 4.38. Comparación de la distribución de potenciales eléctricos escalares en la interface gas-líquido.

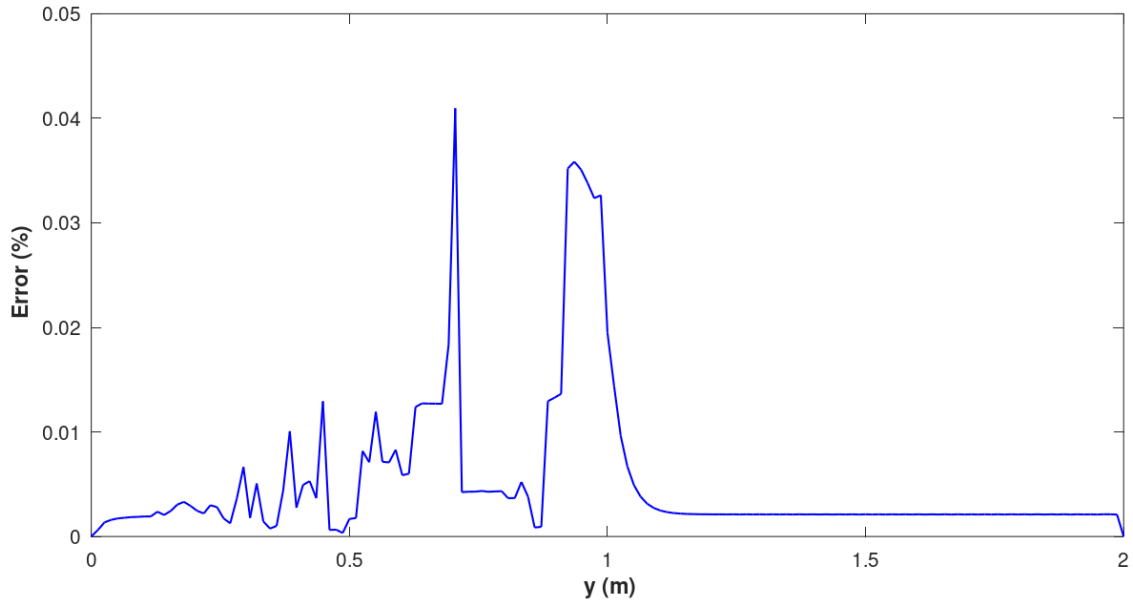


Figura 4.39. Porcentaje de error entre los potenciales eléctricos obtenidos en la simulación y la solución analítica, a lo largo de $x = 0.4m, 0 \leq y \leq 2m$.

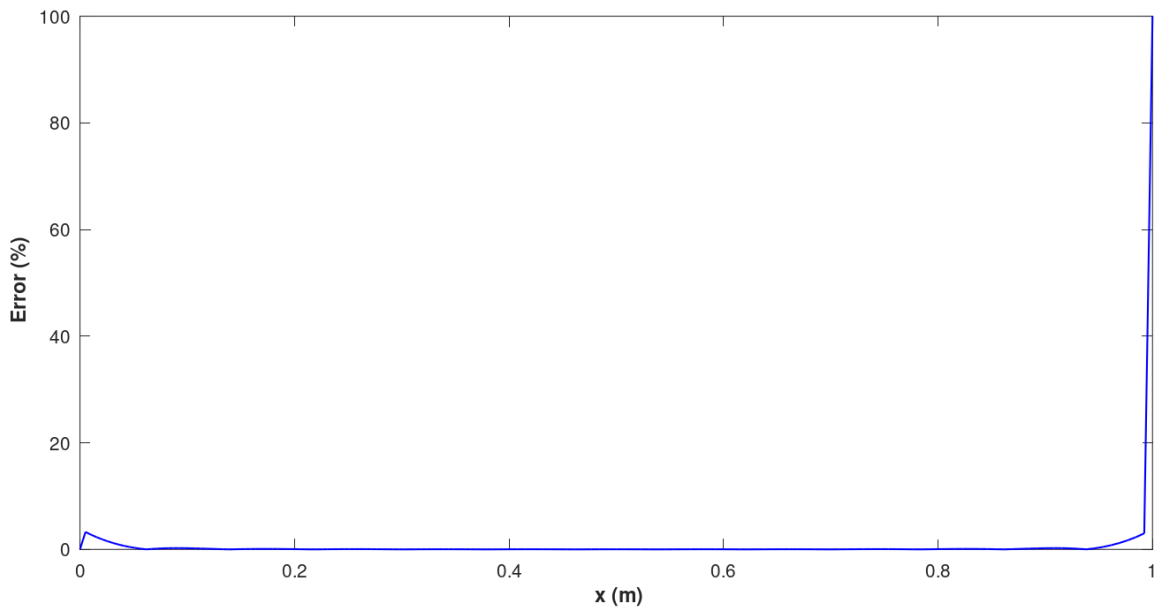


Figura 4.40. Porcentaje de error entre los potenciales eléctricos obtenidos en la simulación y la solución analítica en la interface gas- líquido.

4.3 Simulación de modelos magnetodinámicos

4.3.1 Caso 1

El modelo magnetodinámico tomado de la referencia [51], consiste en un bloque de cobre en el entrehierro de un circuito magnético, el cual está constituido por un núcleo de hierro laminado, con una permeabilidad relativa de 1000. Las bobinas trenzadas se encuentran envueltas alrededor de las ramas verticales centrales del núcleo de hierro y una corriente armónica dependiente del tiempo, fluye en las bobinas e impulsa el flujo alrededor del circuito. La frecuencia es tal que la profundidad del efecto piel en el cobre es de 0.5 cm. La estructura tiene una larga longitud en la dirección z, por lo tanto, no existe una variación significativa de los campos con z, en especial en casos 2D.

La geometría del bloque de cobre en el entrehierro del circuito magnético, es mostrada en la Figura 4.41, en el cual es descrita la cuarta parte de la sección transversal debido a su simetría.

La malla generada del circuito magnético en Gmsh, contiene 305,809 nodos y 152,352 elementos triangulares de segundo orden, como se muestra en la Figura 4.42. La Figura 4.43, muestra un acercamiento de la malla para una mejor visualización de los elementos triangulares.

Los resultados de los potenciales y campo eléctrico obtenidos en el software de elementos finitos utilizando el solver ICCG CUDA, son mostrados en las Figuras 4.44 y 4.45.

En la Tabla 4.11, se muestra el tiempo requerido de simulación utilizando la malla conformada por elementos triangulares.

En la Tabla 4.12, se muestra el tiempo requerido de simulación utilizando la malla conformada por elementos cuadriláteros, el cual contiene 229,633 nodos y 76,176 elementos.

Tabla 4.11. Tiempo requerido de simulación de modelo circuito magnético, utilizando elementos triangulares.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	1,896,361	306
ICCG CUDA	3,486,913	338

Tabla 4.12. Tiempo requerido de simulación de modelo circuito magnético, utilizando elementos cuadriláteros.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	1,893,907	315
ICCG CUDA	3,558,181	350

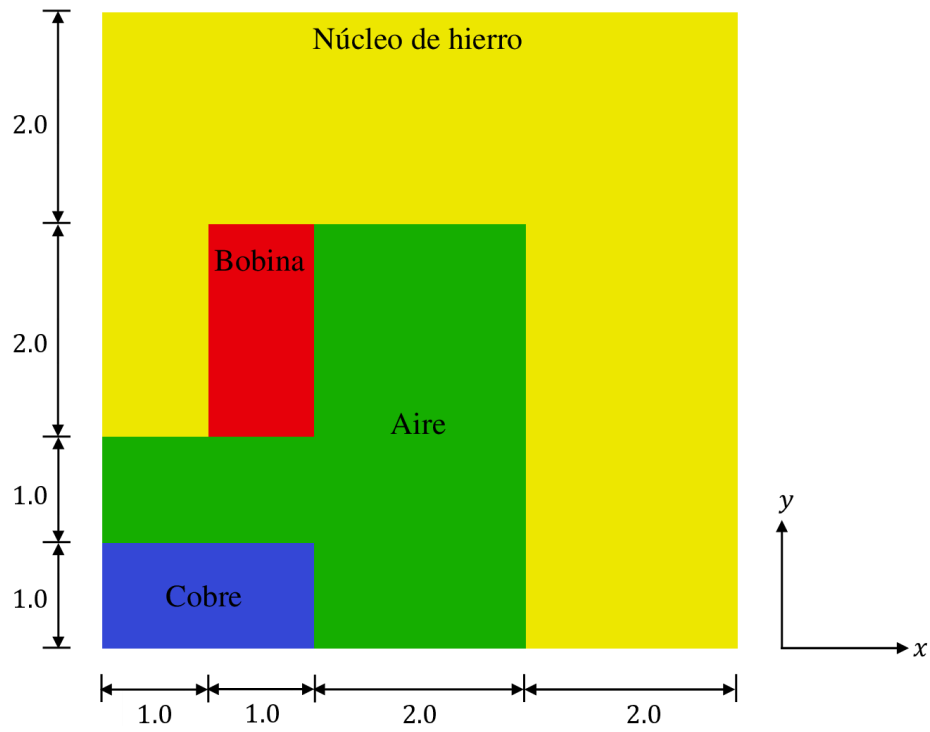


Figura 4.41. Geometría del bloque de cobre en el entrehierro del circuito magnético, dada en cm.

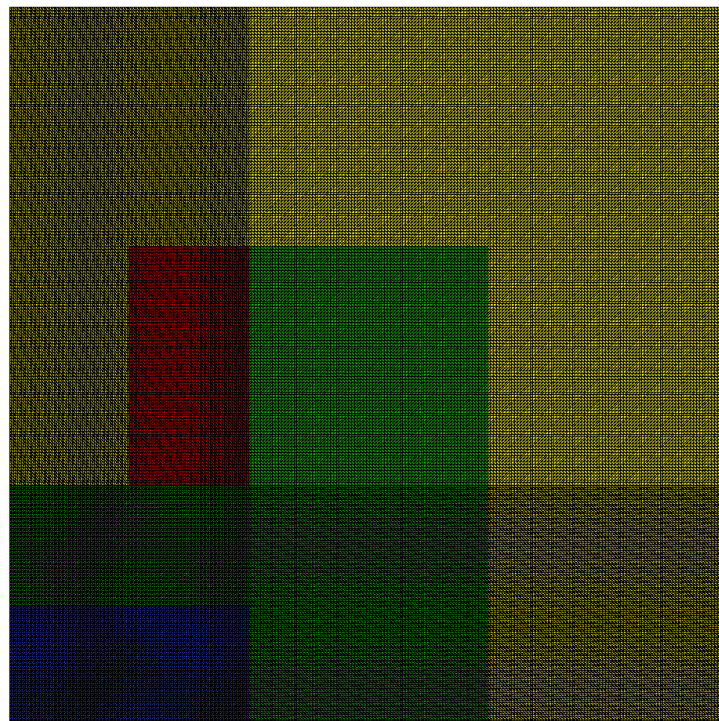


Figura 4.42. Malla del circuito magnético con elementos triangulares.

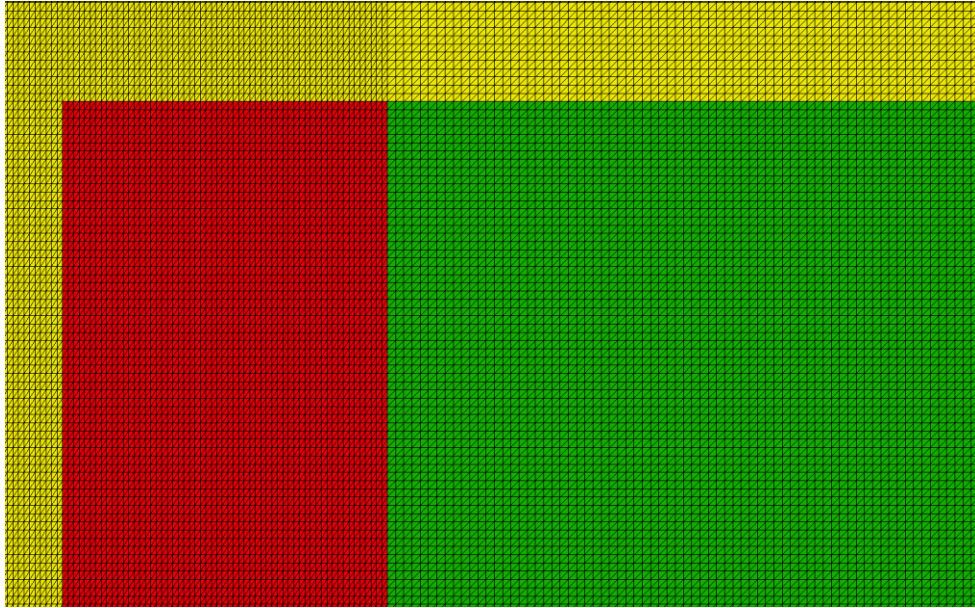


Figura 4.43. Visualización de elementos triangulares que conforman la malla del modelo circuito magnético.

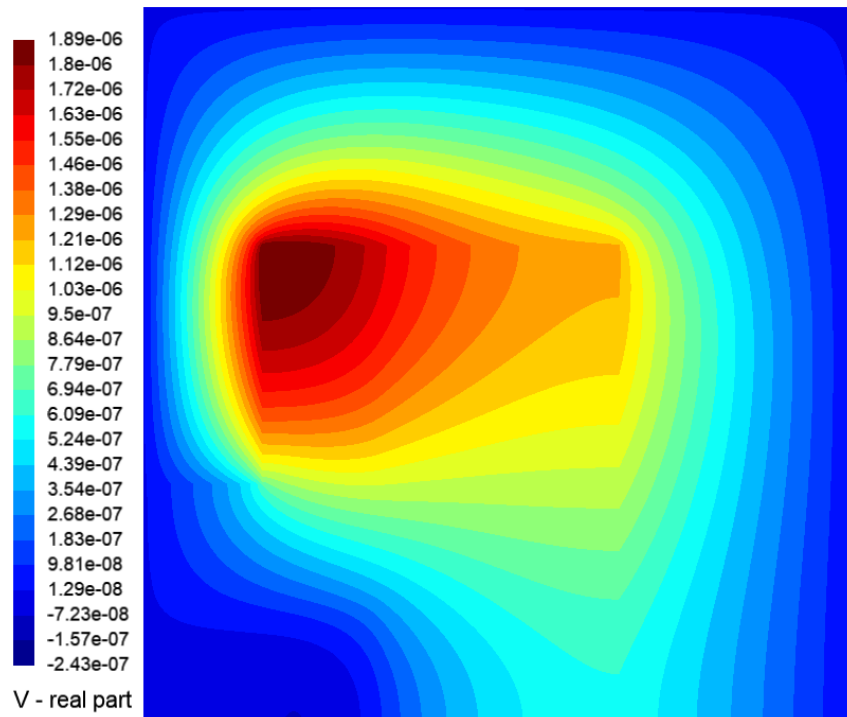


Figura 4.44. Distribución de los potenciales eléctricos (V) en el circuito magnético.

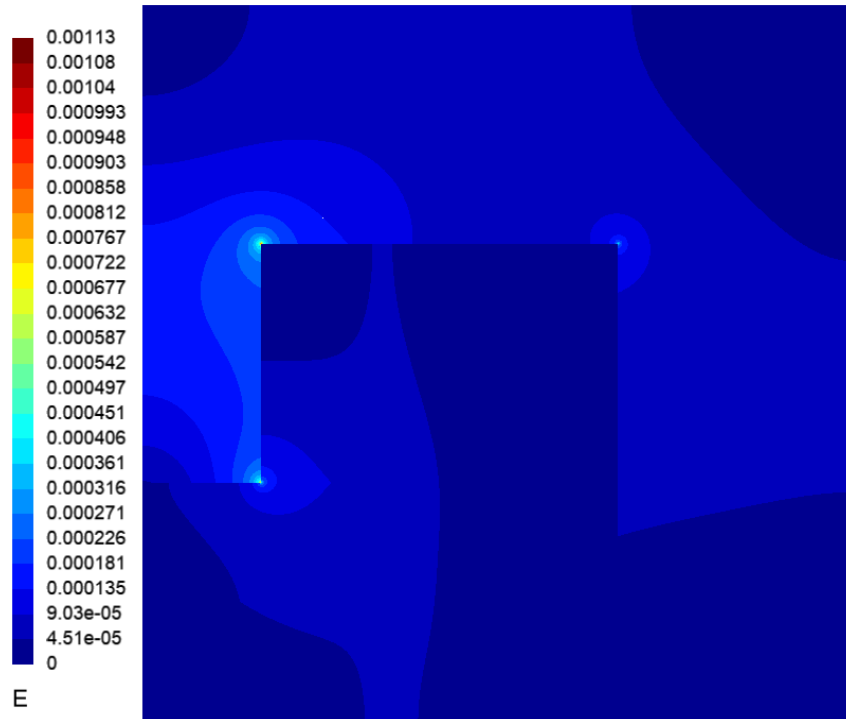


Figura 4.45. Distribución de la magnitud del campo eléctrico (V/m) en el circuito magnético.

4.4 Simulación de modelos electromagnéticos transitorios

4.4.1 Caso 1

El modelo transitorio bajo estudio tomado de las referencias [52]-[53], consiste en determinar la distribución del potencial magnético vectorial para su visualización en distintos tiempos. En la Figura 4.46, se muestra las dimensiones geométrías del modelo, la cual, está constituida por una región conductora (placa), una región de aire y una bobina alimentada por corriente continua.

La malla elaborada del modelo transitorio en Gmsh, contiene 10,769 nodos y 5,280 elementos triangulares de segundo orden, como se muestra en la Figura 4.47.

En la simulación del modelo transitorio utilizando el software de elementos finitos, es determinado la distribución del potencial magnético vectorial en distintos tiempo, como se muestran en las Figuras 4.48 a 4.50.

El tiempo requerido para la simulación del modelo al utilizar elementos triangulares, se muestra en la Tabla 4.13.

En la Tabla 4.14, se muestra el tiempo requerido para la simulación del modelo transitorio al utilizar una malla conformada por elementos cuadriláteros de segundo orden, el cual contiene 8,129 nodos y 2,640 elementos.

Tabla 4.13. Tiempo requerido de simulación de modelo transitorio, utilizando elementos triangulares.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	64,387	307
ICCG CUDA	118,005	160

Tabla 4.14. Tiempo requerido de simulación de modelo transitorio, utilizando elementos cuadriláteros.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	63,910	353
ICCG CUDA	119,691	183

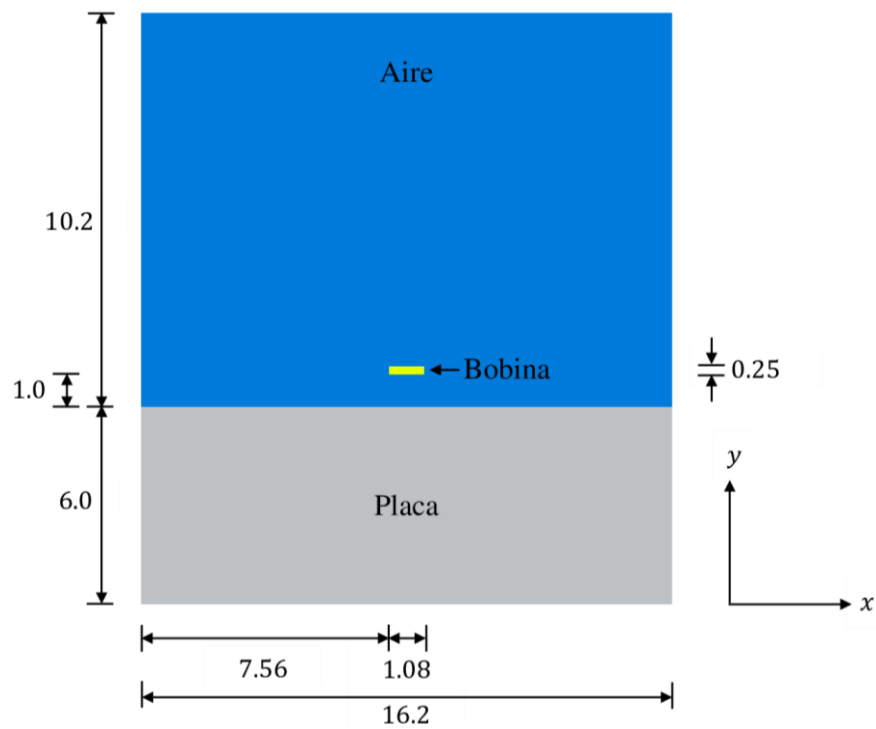


Figura 4.46. Dimensiones geométricas de modelo transitorio, dado en mm.

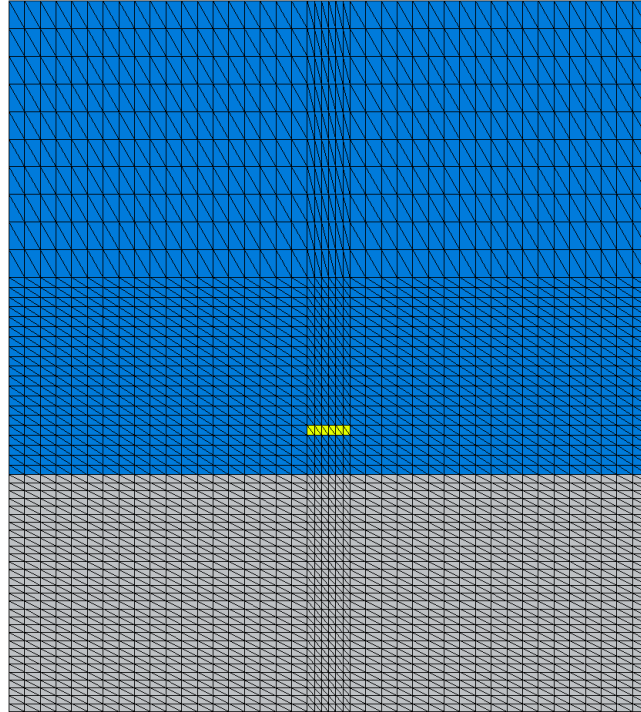


Figura 4.47. Malla de modelo transitorio con elementos triangulares en 2D.

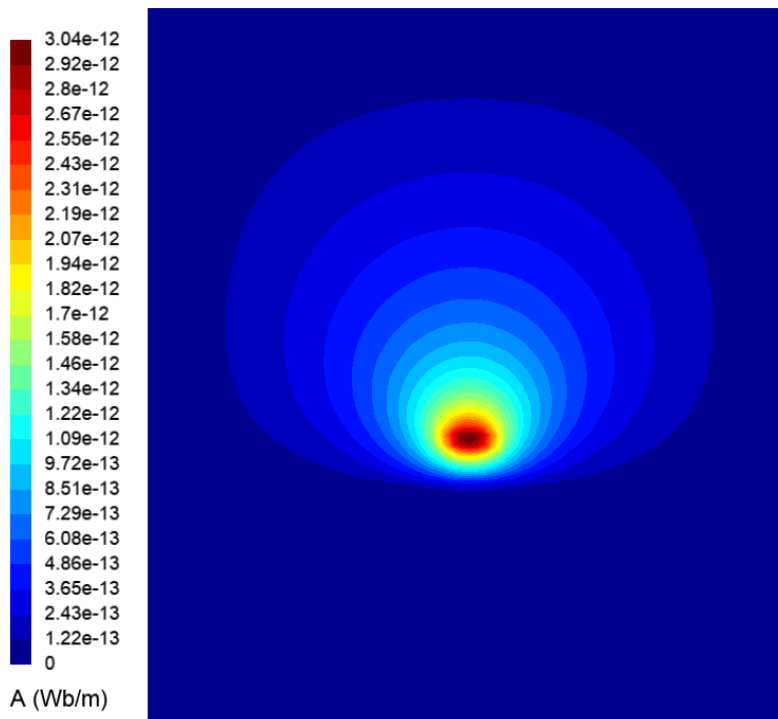


Figura 4.48. Distribución del potencial magnético vectorial en $1\mu s$.

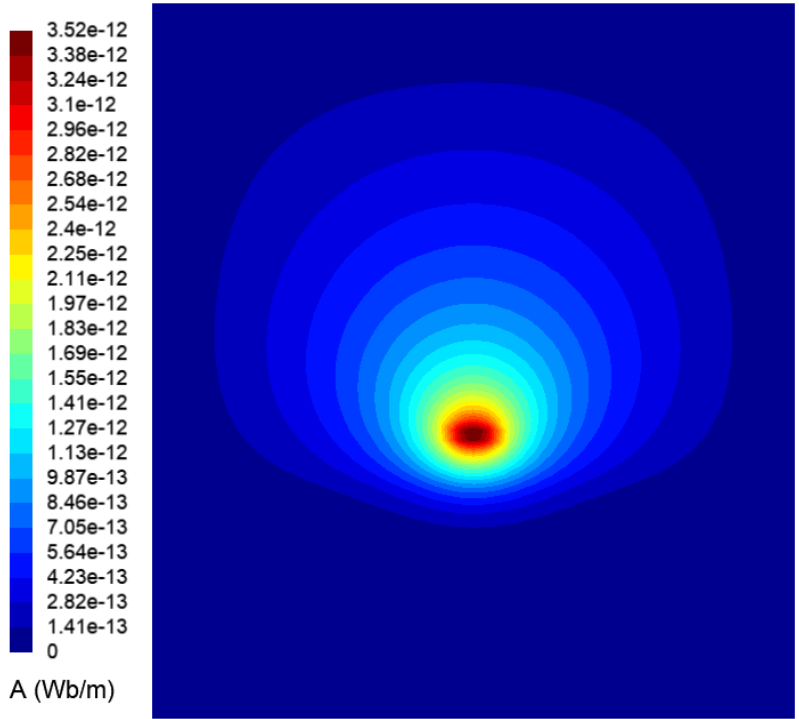


Figura 4.49. Distribución del potencial magnético vectorial en $20\mu s$.

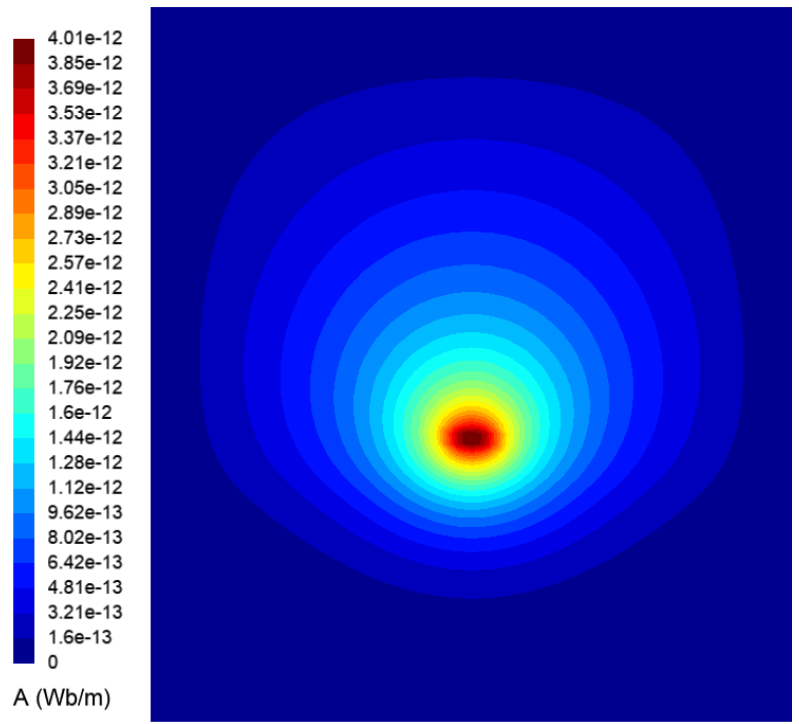


Figura 4.50. Distribución del potencial magnético vectorial.

4.5 Simulación de modelos de conducción de calor

4.5.1 Caso 1

En el proceso de diseño de máquinas eléctricas como transformadores, generadores o motores, es requerido identificar los puntos calientes que son generados en el proceso de transferencia de calor, debido al límite térmico de los materiales que lo componen. Si el límite térmico es superado, puede ocasionar un cortocircuito que compromete seriamente la seguridad del personal y la vida útil del equipo. Por lo tanto, el conocimiento previo del comportamiento permite implementar materiales y configuraciones que soporten las temperaturas generadas o incorporar sistemas de enfriamiento que disminuyan las temperaturas, aumentando la eficiencia y vida útil de los equipos eléctricos [54]-[57].

El primer caso tomado de la referencia [58], consiste en determinar la distribución de temperatura en un conductor, el cual tiene una conductividad térmica de 1.25 Btu / (hr m °F), una temperatura inicial de 30 °F y un tiempo de 1.2 hr. En la Figura 4.51, se muestra las dimensiones geométricas del conductor.

Utilizando Gmsh, es elaborada la malla del modelo conductor y se muestra en la Figura 4.52. La malla generada, contiene 57,121 nodos y 28,322 elementos triangulares de segundo orden.

En la simulación del modelo desarrollado en el software de elementos finitos con ICCG CUDA, es determinada la distribución de temperatura en distintos tiempos, como se muestran en las Figuras 4.53 a 4.55.

La solución analítica, mostrada en [58], está dada por las ecuaciones (4.14) y (4.15) y es solucionada en Octave para determinar la distribución de temperatura en el conductor, como se visualiza en la Figura 4.56.

$$T(x, y, t) = \sum_{n=1}^{\infty} \sum_{j=1}^{\infty} A_n \sin \left[\frac{n\pi x}{L_x} \right] \sin \left[\frac{j\pi y}{L_y} \right] \exp \left[- \left(\frac{k_x n^2 \pi^2}{L_x^2} + \frac{k_y j^2 \pi^2}{L_y^2} \right) t \right] \quad (4.14)$$

donde:

$$A_n = \frac{4(30)}{nj\pi^2} [(-1)^n - 1][(-1)^j - 1] \quad (4.15)$$

Para validar los resultados obtenidos en el software de elementos finitos, se realiza la comparación del mismo con los resultados obtenidos en la solución analítica, como se muestra en la Figura 4.57. Para determinar su exactitud, es calculado el porcentaje de error entre ambos como se muestra en la Figura 4.58.

El tiempo requerido para la simulación del modelo al utilizar elementos triangulares, se muestra en la Tabla 4.15.

En la Tabla 4.16, se muestra el tiempo requerido para la simulación del modelo al utilizar una malla conformada por elementos cuadriláteros de segundo orden, el cual contiene 42,960 nodos y 14,161 elementos.

Tabla 4.15. Tiempo requerido de simulación de conductor (caso 1), utilizando elementos triangulares.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	349,054	3970
ICCG CUDA	640,987	2020

Tabla 4.16. Tiempo requerido de simulación de conductor (caso 1), utilizando elementos cuadriláteros.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	347,636	4080
ICCG CUDA	652,312	2175

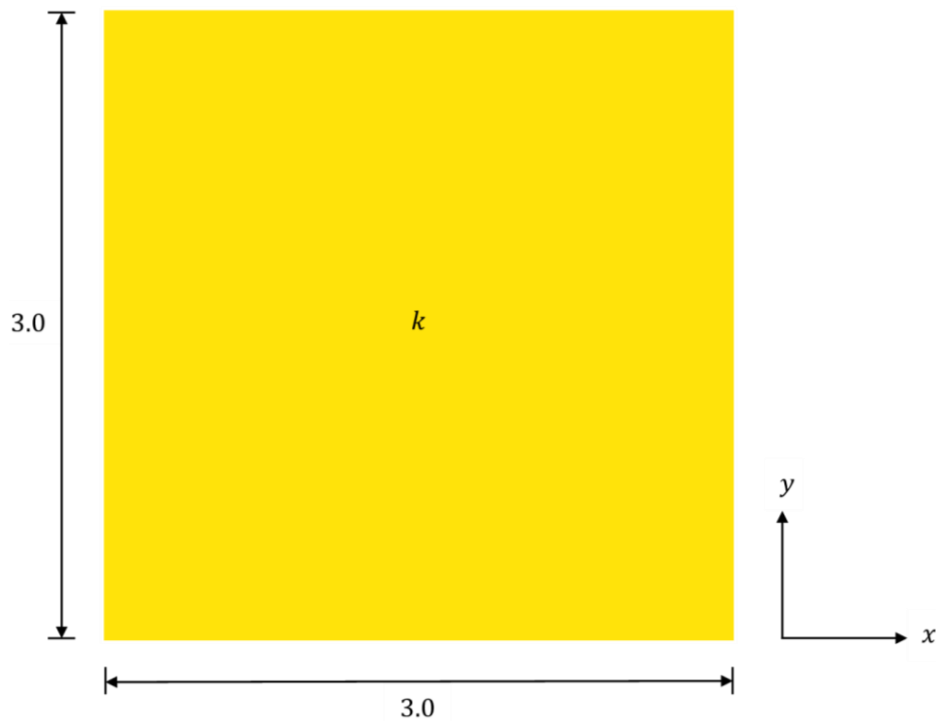


Figura 4.51. Dimensiones geométricas de conductor (caso 1), dada en m.

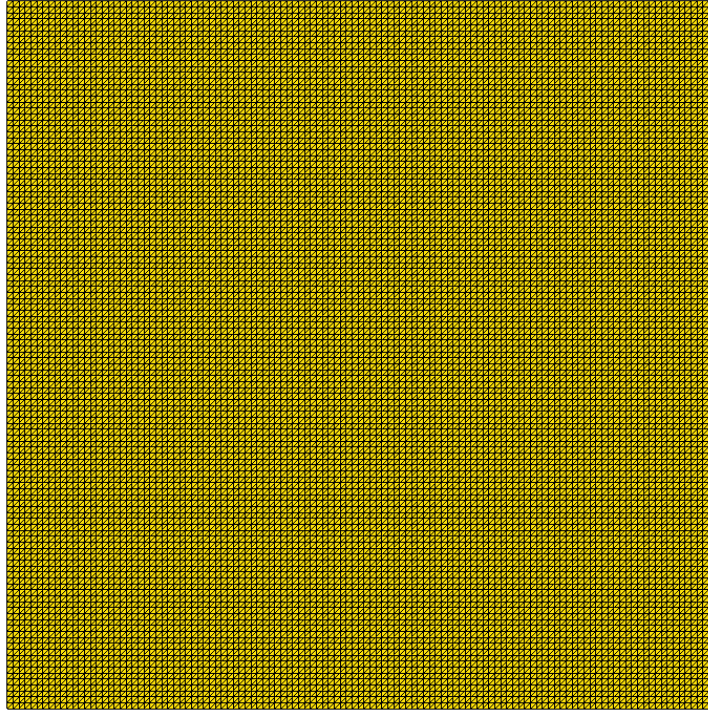


Figura 4.52. Malla del conductor (caso 1) conformada por elementos triangulares.

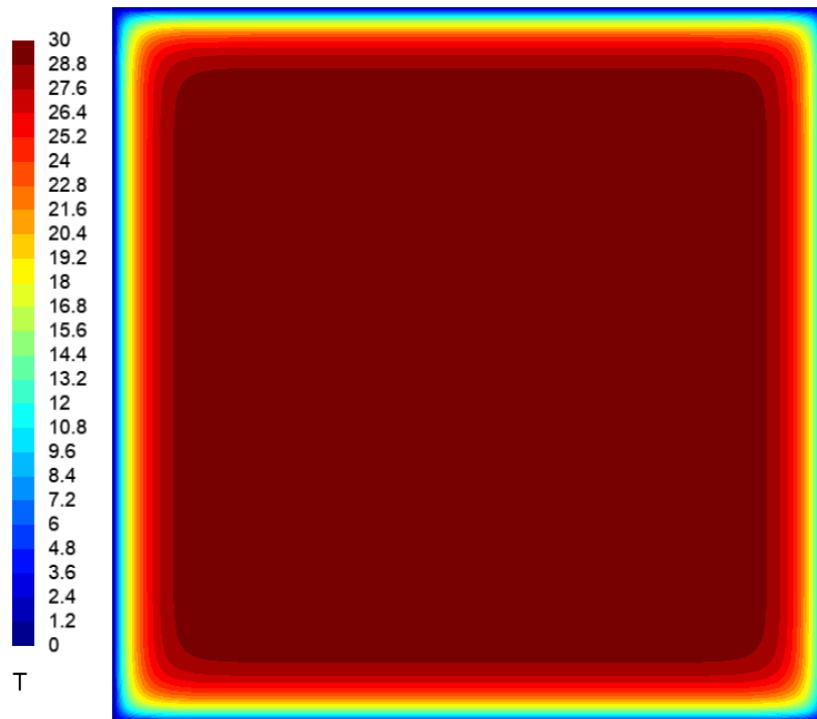


Figura 4.53. Distribución de temperatura ($^{\circ}\text{F}$) en conductor (caso 1), en 5 ms.

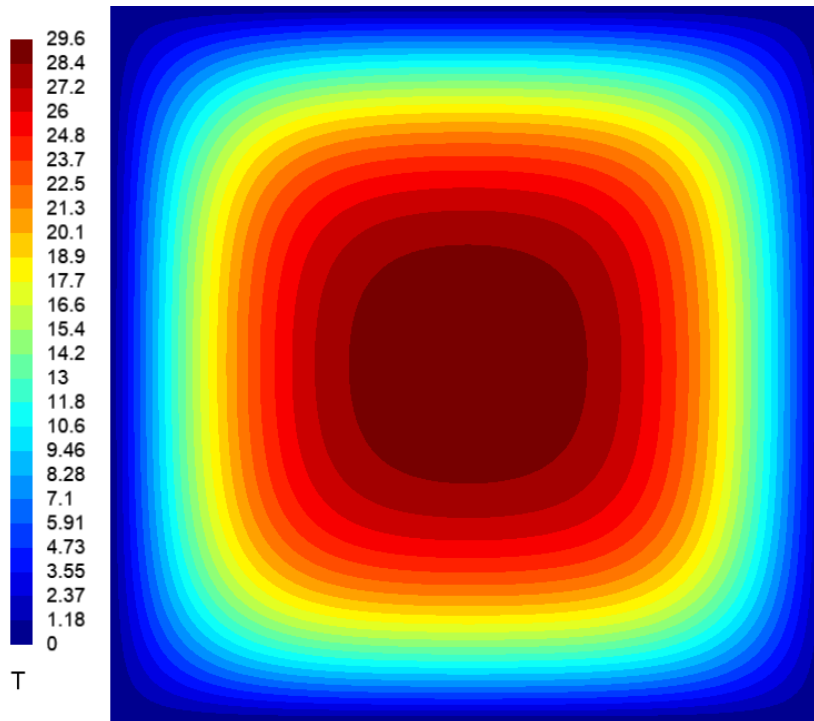


Figura 4.54. Distribución de temperatura (°F) en conductor (caso 1), en 0.1 s.

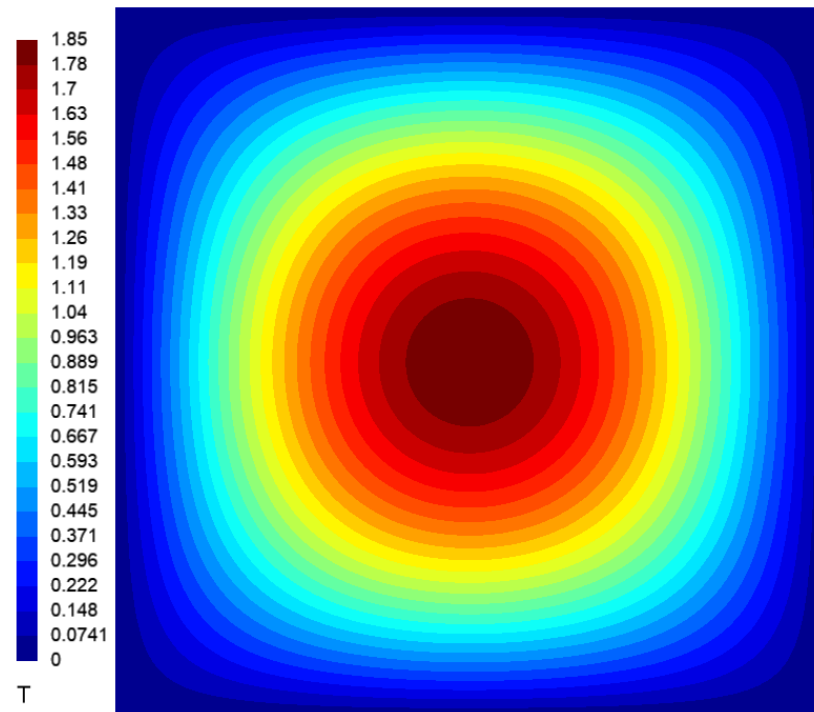


Figura 4.55. Distribución de temperatura (°F) en conductor (caso 1).

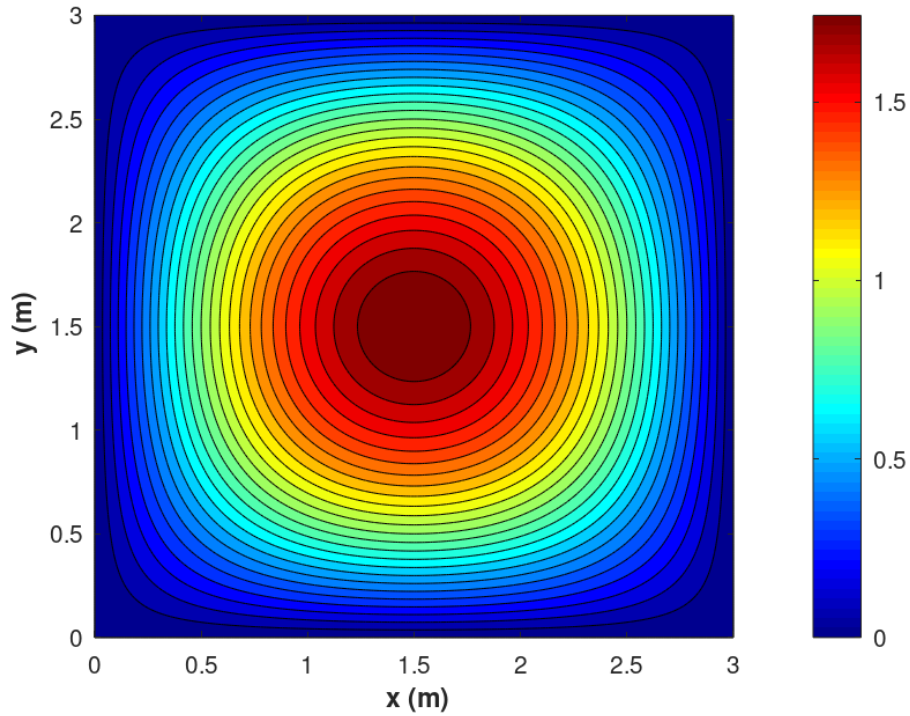


Figura 4.56. Distribución de temperatura ($^{\circ}\text{F}$) en conductor (caso 1), obtenido en solución analítica.

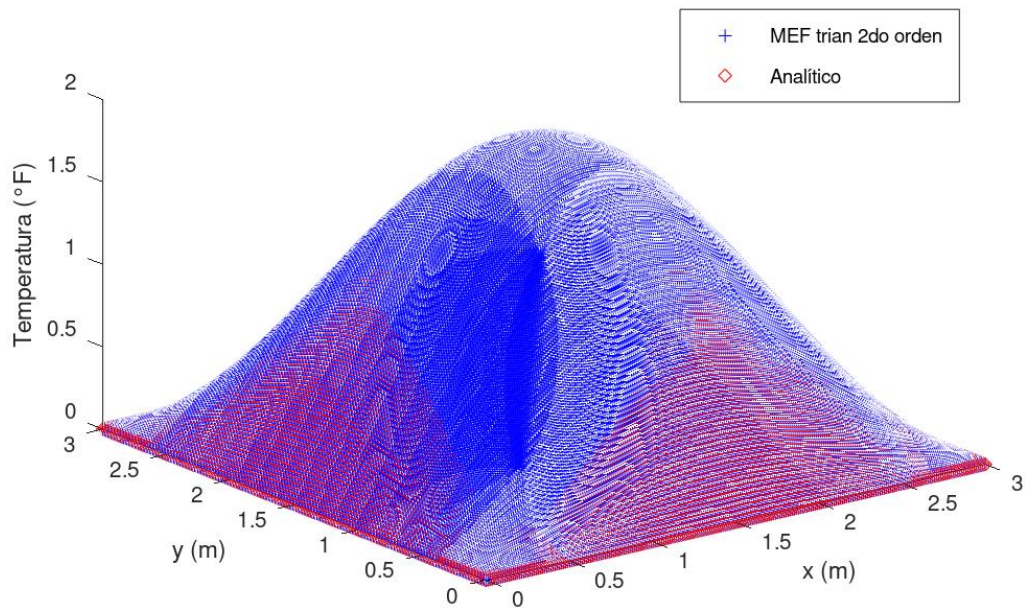


Figura 4.57. Comparación entre temperaturas obtenidas en solución analítica y simulación (caso 1).

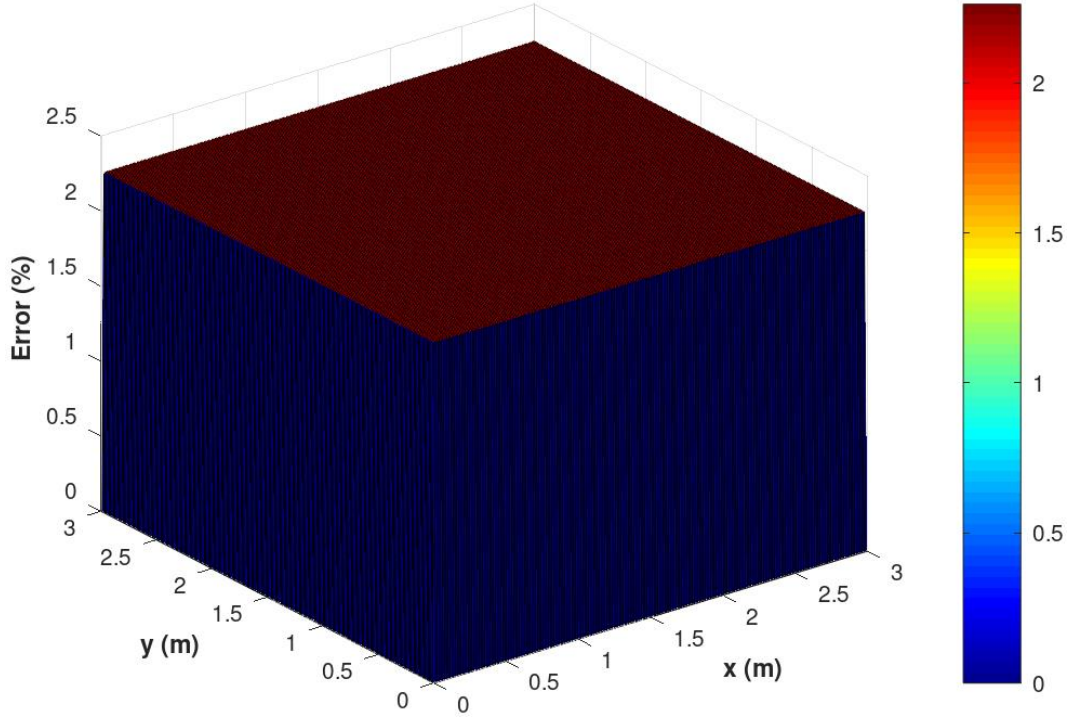


Figura 4.58. Porcentaje de error entre resultados de solución analítica y simulación (caso 1).

4.5.2 Caso 2

El segundo caso tomado de [58] consiste en determinar la distribución de temperatura en un conductor, el cual tiene una conductividad térmica de 1.25 Btu / (hr m °F), una temperatura inicial de 30 °F, un tiempo de 1.2 hr y una condición de frontera Neumann ubicado en una de sus paredes, como se muestra en la Figura 4.59.

Utilizando Gmsh, es elaborada la malla del modelo conductor y se muestra en la Figura 4.60. La malla generada, contiene 57,121 nodos y 28,322 elementos triangulares de segundo orden.

En la simulación del modelo desarrollado en el software de elementos finitos con ICCG CUDA, es determinada la distribución de temperatura en distintos tiempos, como se muestran en las Figuras 4.61 a 4.63.

La solución analítica, mostrada en [58], está dada por las ecuaciones (4.16) y (4.17) y es solucionada en Octave para determinar la distribución de temperatura en el conductor, como se visualiza en la Figura 4.64.

$$T(x, y, t) = \sum_{n=1}^{\infty} \sum_{j=1}^{\infty} B_n \cos \left[\frac{(2n-1)\pi x}{2L_x} \right] \sin \left[\frac{j\pi y}{L_y} \right] \exp \left[- \left(\frac{k_x (2n-1)^2 \pi^2}{4L_x^2} + \frac{k_y j^2 \pi^2}{L_y^2} \right) t \right] \quad (4.16)$$

donde:

$$B_n = \frac{8(30)}{\pi^2 j(2n-1)} (-1)^{n+2} [(-1)^j - 1] \quad (4.17)$$

Para validar los resultados obtenidos en el software de elementos finitos, se realiza la comparación del mismo con los resultados obtenidos en la solución analítica, como se muestra en la Figura 4.65. Para determinar su exactitud, es calculado el porcentaje de error entre ambos como se muestra en la Figura 4.66.

El tiempo requerido para la simulación del modelo al utilizar elementos triangulares, se muestra en la Tabla 4.17.

En la Tabla 4.18, se muestra el tiempo requerido para la simulación del modelo al utilizar una malla conformada por elementos cuadriláteros de segundo orden, la cual contiene 42,960 nodos y 14,161 elementos.

Tabla 4.17. Tiempo requerido de simulación de conductor (caso 2), utilizando elementos triangulares.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	350,586	4164
ICCG CUDA	644,051	2046

Tabla 4.18. Tiempo requerido de simulación de conductor (caso 2), utilizando elementos cuadriláteros.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	349,520	4525
ICCG CUDA	656,080	2380

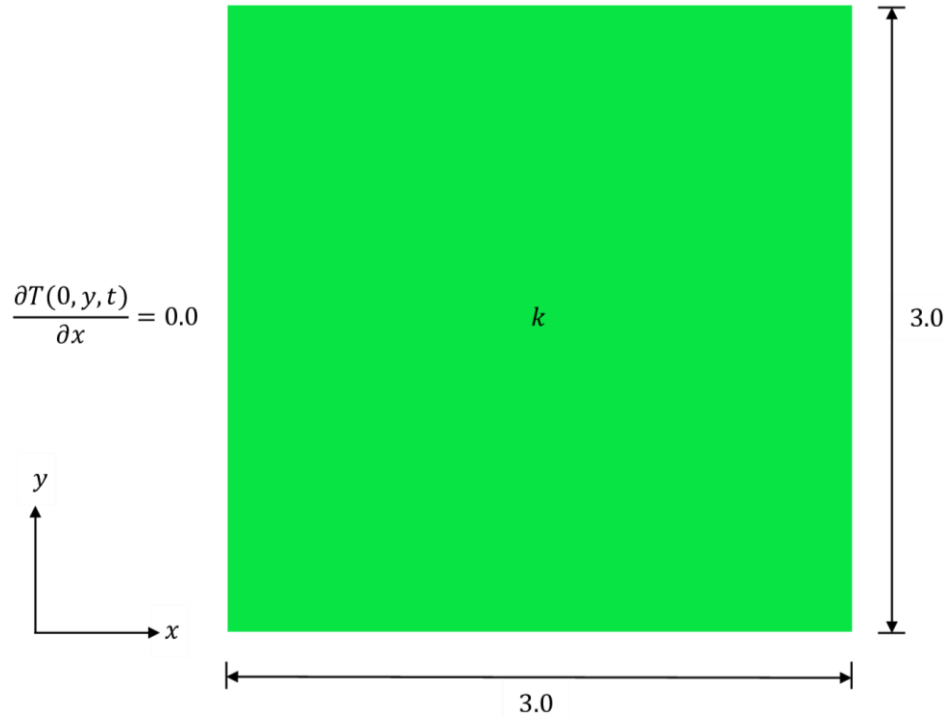


Figura 4.59. Dimensiones geométricas de conductor (caso 2), dada en m.

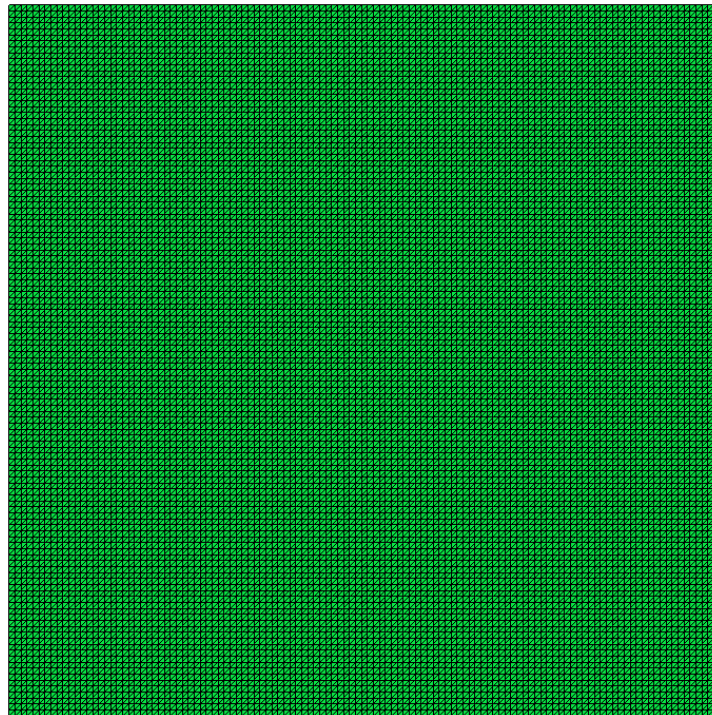


Figura 4.60. Malla del conductor (caso 2) conformada por elementos triangulares.

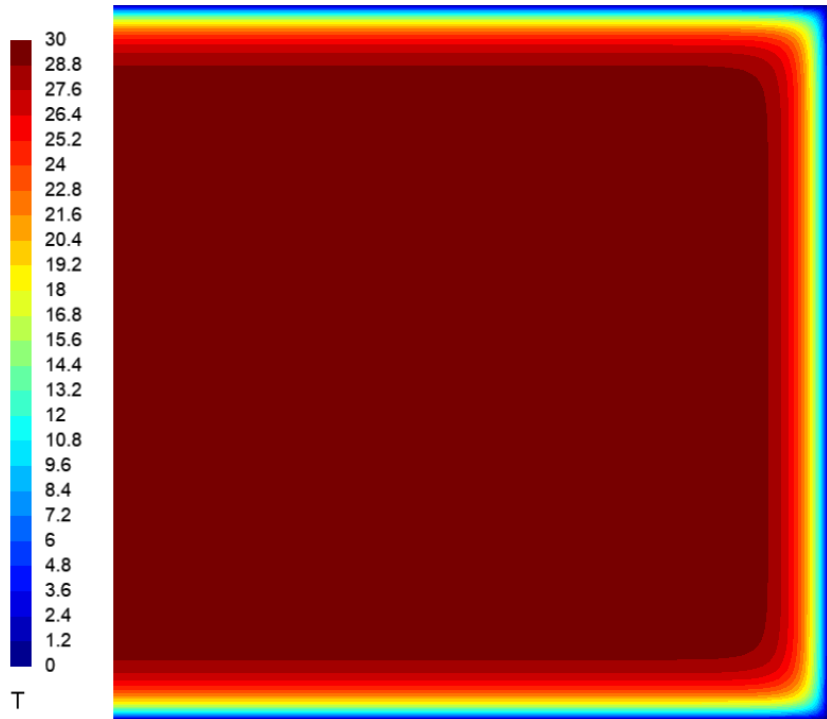


Figura 4.61. Distribución de temperatura (°F) en conductor (caso 2), en 5 ms.

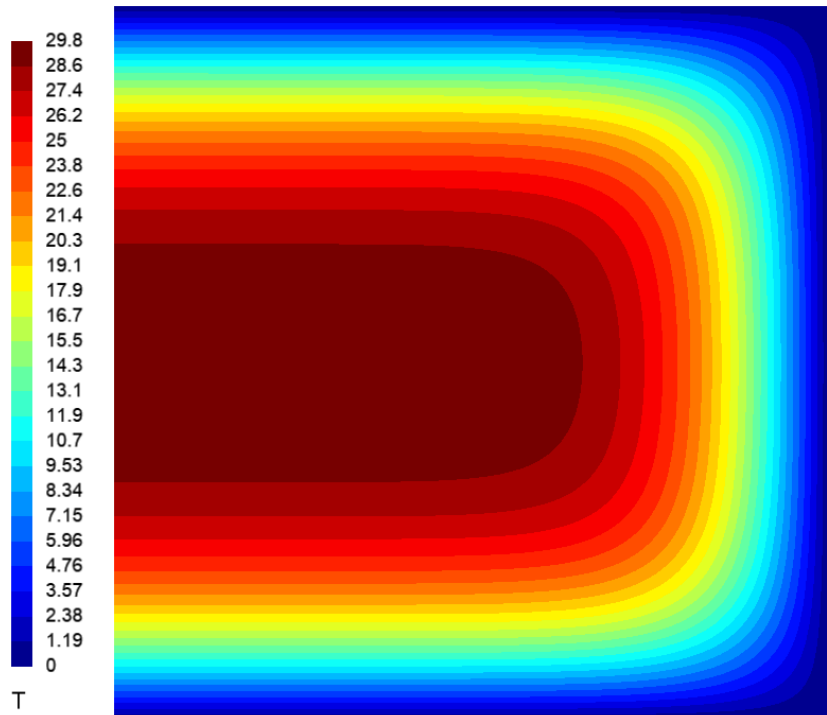


Figura 4.62. Distribución de temperatura (°F) en conductor (caso 2), en 0.1 s.

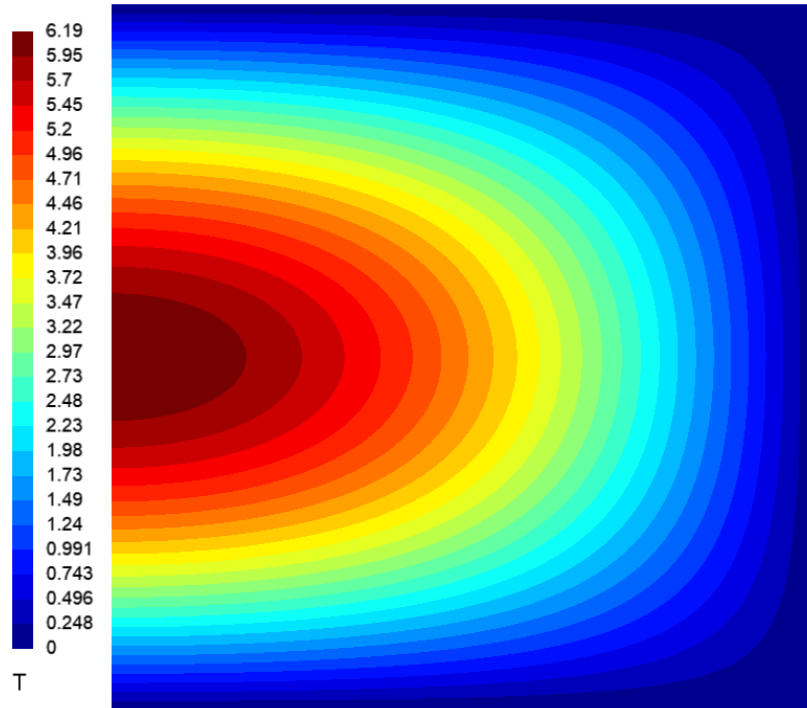


Figura 4.63. Distribución de temperatura (°F) en conductor (caso 2).

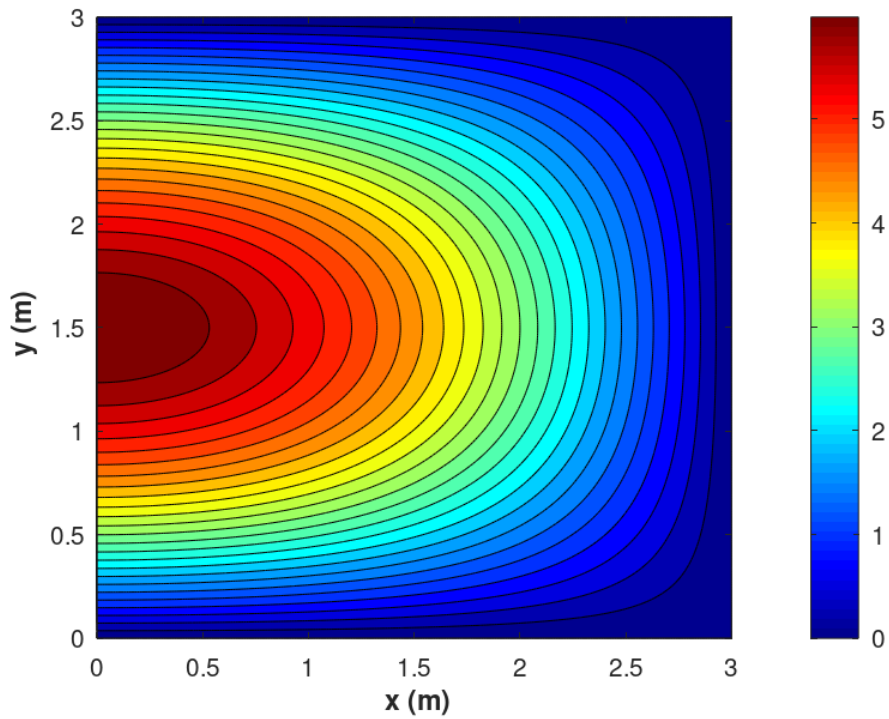


Figura 4.64. Distribución de temperatura (°F) en conductor (caso 2), obtenido en solución analítica.

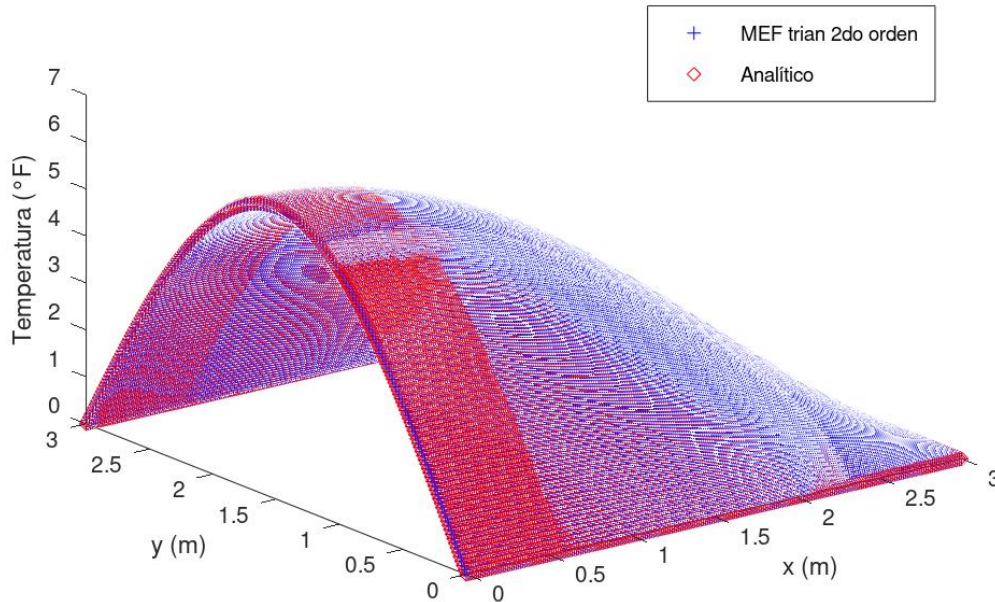


Figura 4.65. Comparación entre temperaturas obtenidas en solución analítica y simulación (caso 2).

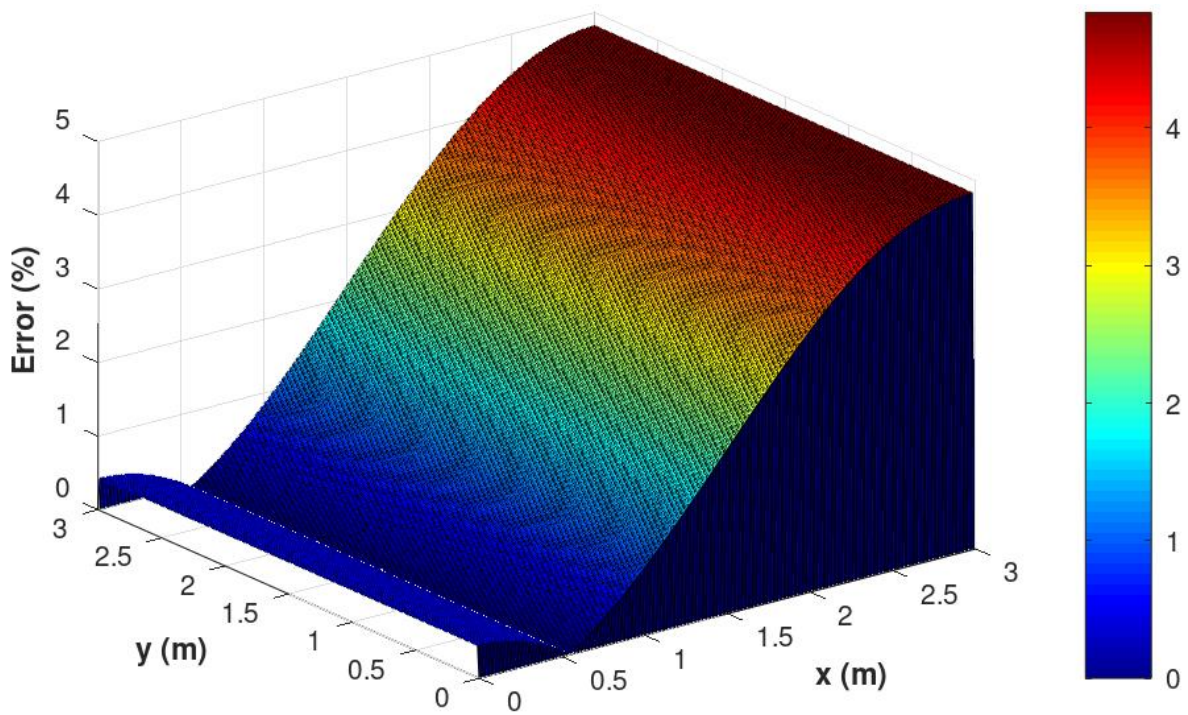


Figura 4.66. Porcentaje de error entre resultados de solución analítica y simulación (caso 2).

4.5.3 Caso 3

El tercer caso tomado de [58], consiste en determinar la distribución de temperatura en un conductor, el cual tiene una conductividad térmica de 1.0 Btu / (hr m °F), un tiempo de 0.75 hr y condiciones de frontera Dirichlet y Neumann, los cuales se visualizan en la Figura 4.67.

Utilizando Gmsh, es elaborada la malla del modelo conductor, la cual se visualiza en la Figura 4.68. La malla generada, contiene 57,121 nodos y 28,322 elementos triangulares de segundo orden.

En la simulación del modelo desarrollado en el software de elementos finitos con ICCG CUDA, es determinada la distribución de temperatura en distintos tiempos, como se muestran en las Figuras 4.69 a 4.71.

La solución analítica, mostrada en [58], está dada por las ecuaciones (4.18) y (4.19) y es solucionada en Octave para determinar la distribución de temperatura en el conductor, como se visualiza en la Figura 4.72.

$$T(x, y, t) = 1.0 + \sum_{n=1}^{\infty} \sum_{j=1}^{\infty} C_n \cos \left[\frac{(2n-1)\pi x}{2L_x} \right] \cos \left[\frac{(2j-1)\pi y}{2L_y} \right] \exp \left[- \left(\frac{k_x(2n-1)^2\pi^2}{4L_x^2} + \frac{k_y(2j-1)^2\pi^2}{4L_y^2} \right) t \right] \quad (4.18)$$

donde:

$$C_n = \frac{16(-1)(-1)^{n+1}(-1)^{j+1}}{\pi^2(2n-1)(2j-1)} \quad (4.19)$$

Para determinar su exactitud, es calculado el porcentaje de error entre ambos como se muestra en la Figura 4.73.

El tiempo requerido para la simulación del modelo al utilizar elementos triangulares, se muestra en la Tabla 4.19.

En la Tabla 4.20, se muestra el tiempo requerido para la simulación del modelo al utilizar una malla conformada por elementos cuadriláteros de segundo orden, la cual contiene 42,960 nodos y 14,161 elementos.

Tabla 4.19. Tiempo requerido de simulación de conductor (caso 3), utilizando elementos triangulares.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	352,126	3,860
ICCG CUDA	647,131	2,494

Tabla 4.20. Tiempo requerido de simulación de conductor (caso 3), utilizando elementos cuadriláteros.

Solver	Elementos diferentes de cero	Tiempo (s)
INTEL MKL PARDISO	351,415	4,217
ICCG CUDA	659,870	2,810

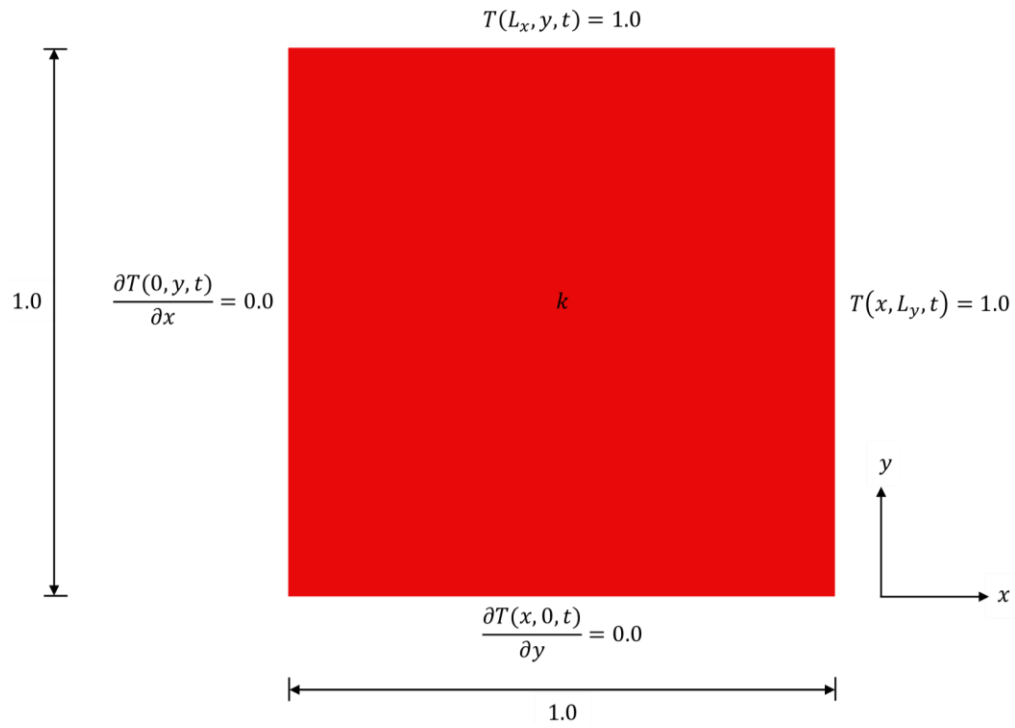


Figura 4.67. Dimensiones geométricas de conductor (caso 3), dada en m.

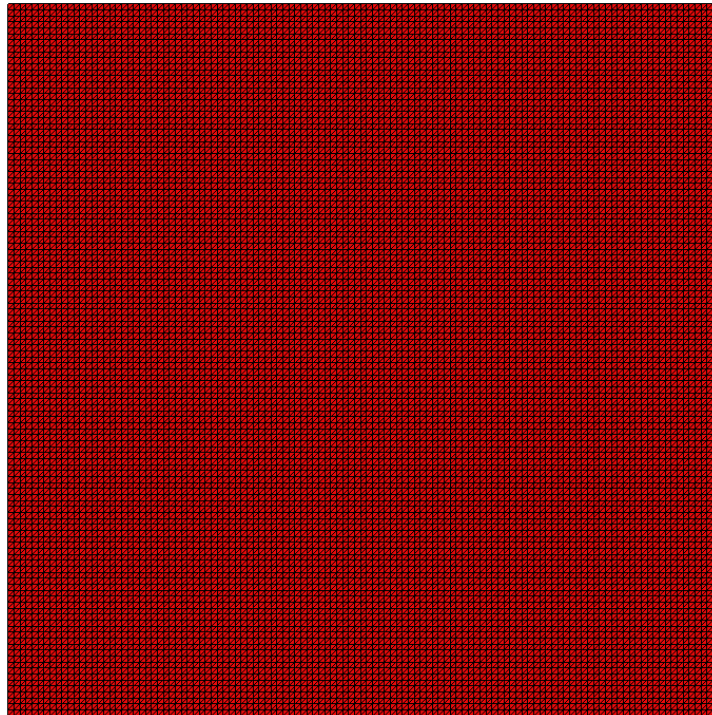


Figura 4.68. Malla del conductor (caso 3) conformada por elementos triangulares.

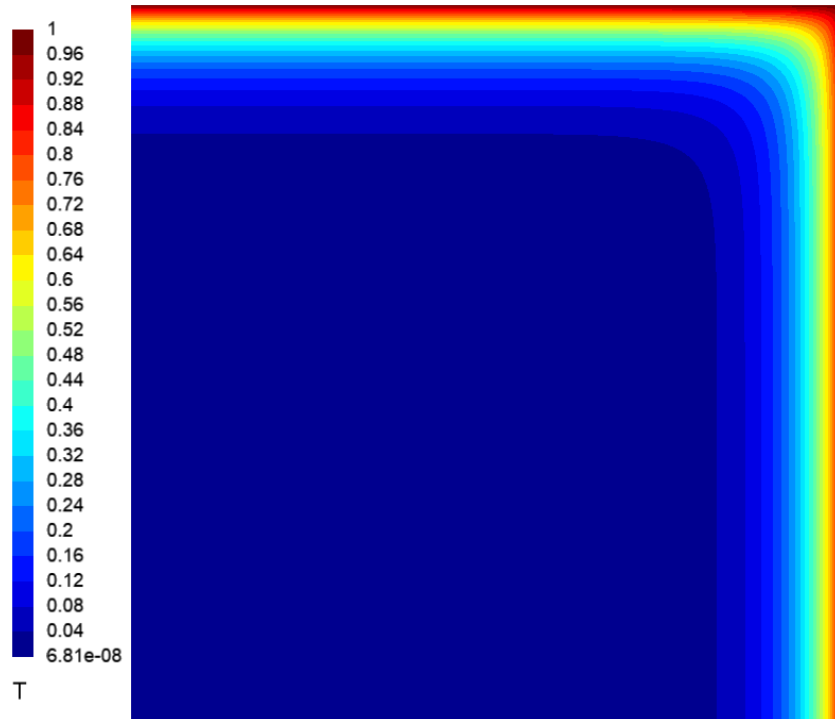


Figura 4.69. Distribución de temperatura (°F) en conductor (caso 3), en 3.125 ms.

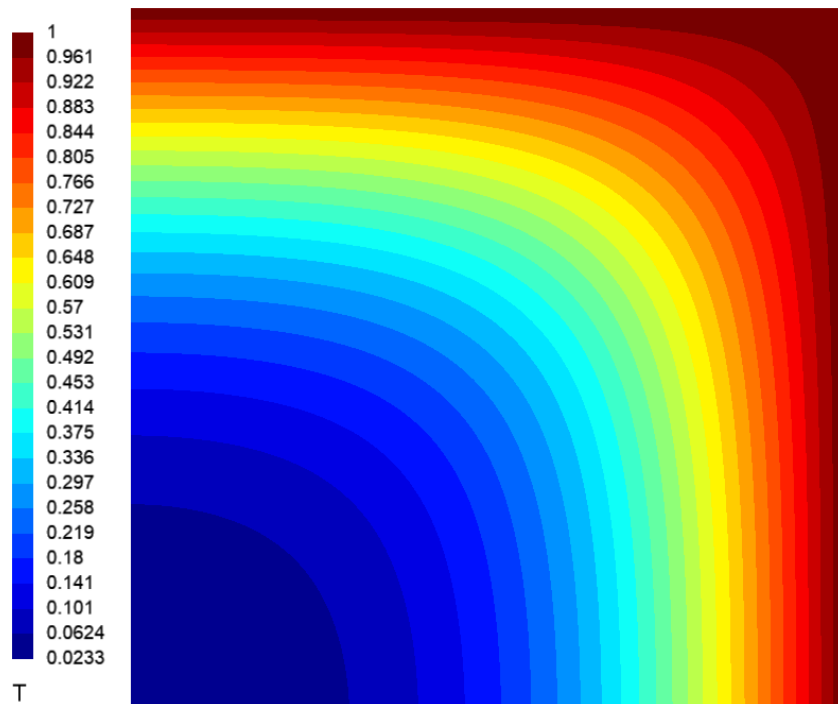


Figura 4.70. Distribución de temperatura (°F) en conductor (caso 3), en 0.0625 s.

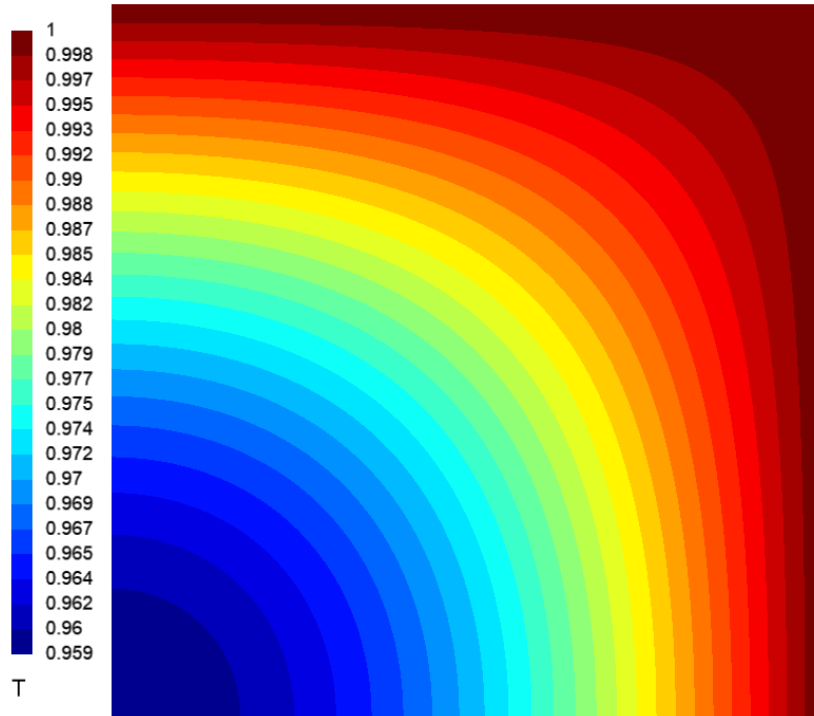


Figura 4.71. Distribución de temperatura ($^{\circ}\text{F}$) en conductor (caso 3).

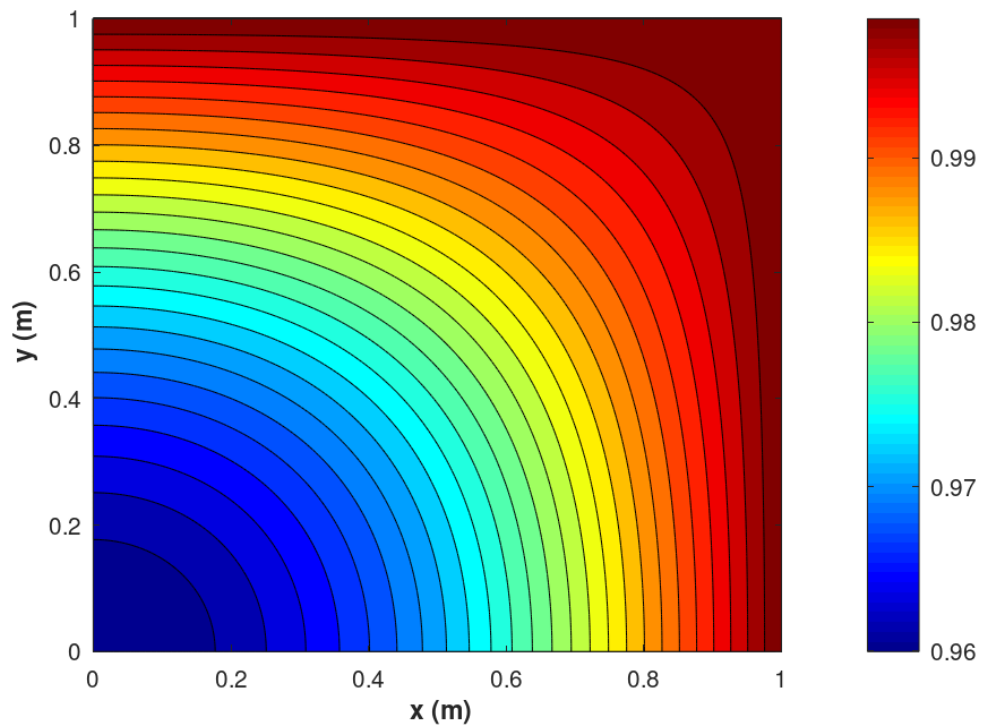


Figura 4.72. Distribución de temperatura ($^{\circ}\text{F}$) en conductor (caso 3), obtenido en solución analítica.

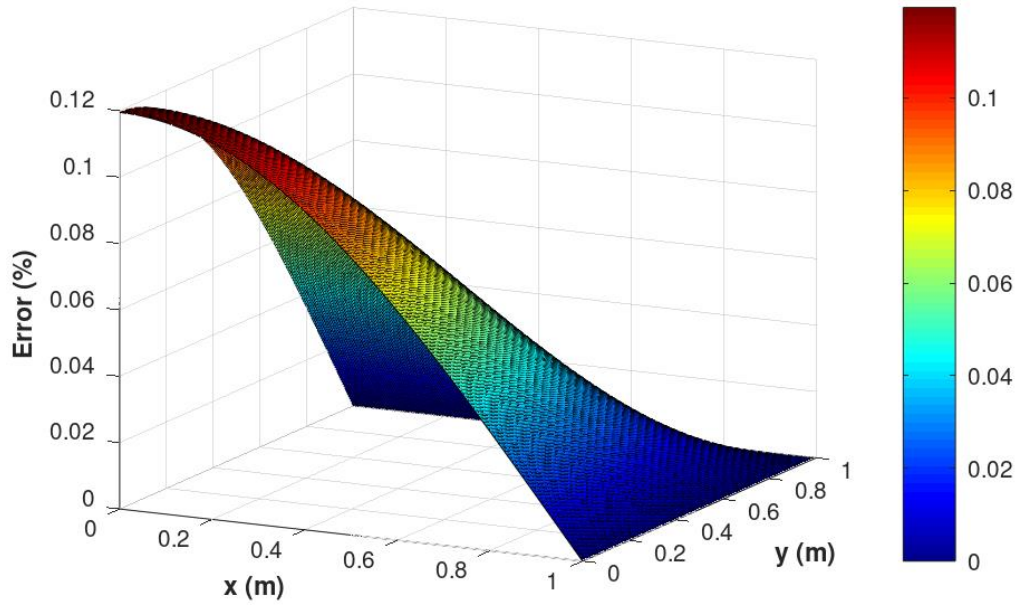


Figura 4.73. Porcentaje de error entre resultados de solución analítica y simulación (caso 3).

Capítulo 5

Conclusiones y trabajo futuro

5.1 Conclusiones

Como parte final de la tesis presentada se han obtenido diversas conclusiones, las cuales se mencionan a continuación.

El uso de elementos triangulares y cuadriláteros de segundo orden para la discretización de modelos, provee resultados aceptables y a medida que se aumenta la cantidad de elementos que conforma la malla, los resultados presentan mayor precisión.

La implementación de elementos isoparamétricos en modelos conformados por geometrías curvas, provee una mejor discretización en la región. En la simulación del modelo magnetostático denominado “caso 2”, al utilizar elementos isoparamétricos triangulares para la discretización del modelo, los resultados obtenidos en la resolución utilizando el solver CUDA ICCG, presentan un error menor al 3% al realizar la comparación con su respectiva solución analítica.

Con la simulación del modelo magnetostático denominado “caso 1” utilizando elementos triangulares y cuadriláteros de segundo orden, es realizada la comparativa del solver desarrollado CUDA ICCG utilizando las librerías de CUDA contra el solver comercial INTEL MKL PARDISO, donde se demuestra que los resultados obtenidos por ambos solver's en la resolución del modelo son idénticos.

En la simulación de modelos que contienen su respectiva solución analítica, al realizar un análisis comparativo entre los resultados obtenidos por el solver CUDA ICCG y los de la solución analítica, se determina que el porcentaje de error entre ambos resultados es menor al 5%, indicando un elevado grado de confiabilidad.

Al determinar el tiempo requerido para la resolución de los diversos modelos electromagnéticos utilizando ambos solver's, es apreciable que en diversos casos requieren el mismo tiempo de solución y en otros el solver CUDA ICCG presenta una ligera ventaja, a pesar de que es requerido ingresar la matriz completa del modelo en comparación del solver INTEL MKL PARDISO que permite la implementación de matrices simétricas. Sin embargo, en el caso de modelos transitorios el solver ICCG requiere un 50% menor de tiempo para la resolución del modelo.

La interfaz gráfica desarrollada permite la inserción y selección de diversos parámetros del modelo deseado a desarrollar, sin embargo, es requerido la implementación de una lógica de programación orientada a objetos para una eficiente ejecución. Además, es requerida la implementación de otras librerías o software que permitan un mejor desarrollo de interfaz para la presentación de un mejor diseño para el usuario final.

5.2 Trabajo futuro

Como trabajo futuro existe la posibilidad de realizar una mejora en el diseño de la interfaz gráfica con el objetivo de mejorar la experiencia del usuario final en la selección del modelo a simular, así como sus parámetros correspondientes para su solución.

Se plantea la posibilidad de mejorar la lógica de programación de los diversos modelos mediante la implementación de la programación orientada a objetos, con el objetivo de elevar la eficiencia del software de elementos finitos desarrollado.

El desarrollo de librerías que permitan la descripción de la geometría y el tipo de mallado de los modelos, así como la visualización de los resultados, generados en la etapa del post-procesamiento.

Referencias

- [1] R. Courant, “Variational methods for the solution of problems of equilibrium and vibrations,” *Bull. Am. Math. Soc.*, vol. 49, pp. 1–23, Jan. 1943.
- [2] L. C. Turner, M. J., Clough, R. W., Martin, H. C., Topp, “Stiffness and deflection analysis of complex structures,” *J. Aeronaut. Sci.*, vol. 23, no. 9, pp. 805–823, Sep. 1956.
- [3] R. W. Clough, “The finite element method in plane stress analysis,” *Proceedings of the Second ASCE Conference on Electronic Computation.*, Pittsburgh, PA, 1960.
- [4] O. C. Zienkiewicz and Y. K. Cheung, “Finite elements in the solution of field problems,” *The Engineer*, vol. 220, 1965.
- [5] “MSC Software.” [Online]. Available: <http://www.mscsoftware.com/page/msc-software>.
- [6] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method: The Basis*, Butterworth., vol. 1, Oxford, 2000.
- [7] S. Ahmed, “Finite-element method for waveguide problems,” *Electron. Lett.*, vol. 4, no. 18, pp. 387–389, 1968.
- [8] J. P. Webb, “Application of the finite-element method to electromagnetic and electrical topics,” *Reports Prog. Phys.*, vol. 58, no. 12, pp. 1673–1712, Dec. 1995.
- [9] P. Silvester, “High-order polynomial triangular finite elements for potential problems,” *Int. J. Eng. Sci.*, vol. 7, no. 8, pp. 849–861, Aug. 1969.
- [10] P. P. Silvester, “Finite-element solution of homogeneous waveguide problems,” *Alta Freq.*, vol. 38, pp. 313–317, 1969.
- [11] “ANSYS.” [Online]. Available: <https://www.ansys.com/about-ansys>.
- [12] “OpenGL.” [Online]. Available: <https://www.opengl.org/about/>.
- [13] “The COMSOL Group - The Origin of Multiphysics Software.” [Online]. Available: <https://www.comsol.com/company>.
- [14] J. Sanders and E. Kandrot, “CUDA by Example: An Introduction To General-Purpose GPU Programming”, Addison-Wesley, 2010.
- [15] “CUDA C++ Programming Guide”, NVIDIA, 2021.
- [16] J. Jin, “The Finite Element Method in Electromagnetics,” John Wiley & Sons, Inc., 2002.
- [17] M. N. O. Sadiku, “Computational Electromagnetics with MATLAB,” CRC Press, 2018.
- [18] J. P. A. Bastos and N. Sadowski, “Magnetic Materials and 3D Finite Element Modeling,” Boca Raton: CRC Press, 2014.
- [19] J. Smajic, C. Jäger, S. Neubauer, A. Bauer, D. J. Cheng and M. Widenhorn, “Coupled Electromagnetic-Mechanical Dynamic Analysis of Generator Circuit Breakers,” *IEEE Transactions on magnetics*, vol. 50, no. 2, 2014.
- [20] J. Li, P. Liu and V. Dinavahi, “Massively Parallel Computation for 3-D Nonlinear Finite Edge Element Problem With Transmission Line Decoupling Technique,” *IEEE Transactions on magnetics*, vol. 55, no. 10, 2019.
- [21] A. Capozzoli, O. Kilic, C. Curcio and A. Liseno, “The Success of GPU Computing in Applied Electromagnetics,” *Aces Journal*, vol. 33, no. 2, 2018.
- [22] A. Altinkaynak, “An efficient sparse matrix-vector multiplication on CUDA-enabled graphic processing units for finite element method simulations,” *Int. J. Numer. Methods Eng.*, vol. 110, no. 1, pp. 57–78, 2017.

- [23] V. Kindratenko, “Numerical Computations with GPUs,” Springer, 2014.
- [24] R. Couturier, “Designing Scientific Applications on GPUs,” CRC Press, 2014.
- [25] J.-C. Sabonnadiere and A. Konrad, “Engineering software-computing EM fields,” *IEEE Spectr.*, vol. 29, no. 11, pp. 52–56, 1992.
- [26] “Elmer, Open Source Finite Element Software for Multiphysical Problems.” [Online]. Available: <http://www.elmerfem.org/>.
- [27] “SALOME, the Open Source Integration Platform for Numerical Simulation.” [Online]. Available: <http://www.salome-platform.org/>.
- [28] “Agros Suite – A multiplatform application for the solution of physical problems.” [Online]. Available: <https://www.agros2d.org>.
- [29] “ONELAB: Open Numerical Engineering LABORatory.” [Online]. Available: <https://www.onelab.info>.
- [30] R. Anderson *et al.*, “MFEM: A modular finite element methods library”, *Computers & Mathematics with Applications*, vol. 81, pp. 42-74, 2021, doi: 10.1016/j.camwa.2020.06.009.
- [31] A. Díaz Díaz, R. Castañeda Balderas, A. F. Domínguez Alvarado, and C. I. Martínez Morfín, “Soluciones de Ecuaciones Diferenciales por Elemento Finito (SEDEF)”, *IIT*, vol. 21, no. 1, 2020, doi: 10.22201/fi.25940732e.2020.21n1.002.
- [32] Ö. Özgün and M. Kuzuoğlu, “MATLAB-based Finite Element Programming in Electromagnetic Modeling”, 1st ed. CRC Press, 2019.
- [33] P. Nithiarasu, R. W. Lewis and K. N. Seetharamu, “Fundamentals of The Finite Element Method for Heat and Mass Transfer”. 2nd ed. Wiley, 2016.
- [34] F. Smailbegovic, G. N. Gaydadjiev and S. Vassiliadis, “Sparse Matrix Storage Format”, 2005.
- [35] Intel, “Intel Math Kernel Library”, 2020.
- [36] NVIDIA (2019). CuSPARSE Library User Guide.
- [37] NVIDIA (2019). cuBLAS Library User Guide.
- [38] M. R. Hestenes and E. Stiefel, “Methods of Conjugate Gradients for Solving Linear Systems,” *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, 1952.
- [39] P. Nithiarasu, R. W. Lewis and K. N. Seetharamu, “Fundamentals of The Finite Element Method for Heat and Mass Transfer”. 2nd ed., Wiley, 2016.
- [40] J. Smart and K. Hock, “Cross-platform GUI Programming With wxWidgets”. Prentice Hall, 2005.
- [41] J. P. A. Bastos and N. Sadowski, “Electromagnetic Modeling by Finite Element Methods”, 1st ed. CRC Press, 2003.
- [42] C. Geuzaine and J.-F. Remacle, “Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities”. *International Journal for Numerical Methods in Engineering*, vol. 79, no.11, pp. 1309-1331, 2009.
- [43] A. Bermúdez, D. Gomez and P. Salgado “Mathematical Models and Numerical Simulation in Electromagnetism.” New York, 2014.
- [44] A. Bermúdez, R. Rodríguez and P. Salgado “A finite element method for the magnetostatic problem in terms of scalar potentials.” *SIAM J. Numer. Anal.*, vol. 46, no. 3, pp. 1338-1363, 2008.

- [45] W. H. Hayt, Jr and J. A. Buck “Engineering Electromagnetics”, New York: McGraw-Hill, 2012.
- [46] J. W. E., D. B., S. H. y Wehbring, R. (2020). GNU Octave version 6.1.0 manual: a high-level interactive language for numerical computations. [online]. Disponible en: <https://www.gnu.org/software/octave/doc/v6.1.0/>.
- [47] Matthew N. O. Sadiku, “Numerical Techniques in Electromagnetics”, 2nd ed. CRC Press, 2000.
- [48] K. Asano, “Electrostatic potential and field in a cylindrical tank containing charged liquid”, *Proc. IEEE*, vol. 124, no. 12, pp. 1277-1281, 1977, doi: 10.1049/piee.1977.0269.
- [49] Y. Matsubara, R. J. Pazda, and T. B. Jones, “Effect of fill pipe on transient decay of surface potential in a cylindrical vessel containing charged liquid”, *IEEE Trans. on Ind. Applicat.*, vol. 30, no. 1, pp. 205-209, feb. 1994, doi: 10.1109/28.273647.
- [50] M. N. O. Sadiku, A. Z. Makki, and L. C. Agba, “A further introduction to finite element analysis of electromagnetic problems”, *IEEE Trans. Educ.*, vol. 34, no. 4, pp. 322-329, nov. 1991, doi: 10.1109/13.102849.
- [51] J. P. Webb and B. Forghani “ T - Ω method using hierarchal edge elements”, *IEE Proc.-Sci. Meas. Technol.*, vol.142, no.2, 1995.
- [52] R. B. B. Ovando-Martínez, M. A. Arjona López and C. Hernández Flores “A Finite-Element Variable Time-Stepping Algorithm for Solving the Electromagnetic Diffusion Equation”. *IEEE Transactions on Magnetics*, vol. 48, no.2, pp. 647-650, 2012.
- [53] R. Ludwig “Numerical and analytical modeling of pulsed eddy currents in a conducting half-space”. *IEEE Transactions on Magnetics*, vol. 26, no. 1, pp.299-307, 1990.
- [54] A. K. Naskar and D. Sarkar “Numerical analysis of three dimensional steady state heat conduction in the rotor of an induction motor by finite element method”. *Proceedings of the 2014 International Conference on Control, instrumentation, Energy and Communication (CIEC)*, vol. 45, no.4, pp. 686-690, 2014.
- [55] M. chrittwieser, O. Bíró, E. Farnleitner and G. Kastner “Analysis of temperature distribution in the stator of large synchronous machines considering heat conduction and heat convection”. *IEEE Transactions on Magnetics*, vol. 51, no.3, pp. 20-23, 2015.
- [56] D. A. Staton and A. Cavagnino “Convection heat transfer and flow calculations suitable for electric machines thermal models”. *IEEE Transactions on Industrial Electronics*, vol. 55, no.10, pp. 3509–3516, 2008.
- [57] Y. Yang, B. Bilgin, M. Kasprzak, S. Nalakath, H. Sadek, M. Preindl, J. Cotton, N. Schofield and A. Emadi “Thermal management of electric machines”. *IET Electrical Systems in Transportation*, vol. 7, no. 2, pp. 104-116, 2017.
- [58] C. John, JR. Bruch and G. Zvoloski “Transient two-dimensional heat conduction problems solved by the finite element method”. *International Journal for Numerical Methods in Engineering*, vol. 8, pp. 481-494, 1974.