CAMPO ELÉCTRICO INDUCIDO EN OBJETOS CERCANOS A UNA LÍNEA DE TRANSMISIÓN DE ALTA TENSIÓN

JHONATHAN EDUARDO CORREA CABALLERO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA
2017

CAMPO ELÉCTRICO INDUCIDO EN OBJETOS CERCANOS A UNA LÍNEA DE TRANSMISIÓN DE ALTA TENSIÓN

JHONATHAN EDUARDO CORREA CABALLERO

Trabajo de grado para optar al título de Ingeniero Electricista

Director
JULIO CÉSAR CHACÓN VELASCO
Magíster en Potencia Eléctrica

Codirector
ERNESTO AGUILERA BERMÚDEZ
Doctor en Ingeniería

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA
2017

AGRADECIMIENTOS

A mi familia por la confianza y la motivación durante mis estudios universitarios y a lo largo de toda mi vida.

A mi director de proyecto, el profesor Julio Cesar Chacón, y mi codirector, el profesor Ernesto Aguilera Bermúdez, por sus consejos y dirección durante la realización de este proyecto.

A todos mis amigos, compañeros y profesores, que de alguna mañera han acompañado e influido en mi formación a través de estos años.

Jhonathan Eduardo Correa Caballero

CONTENIDO

Pág.

INTRODUCCIÓN 14

1. FUNDAMENTACIÓN TEÓRICA 16
1.1 NORMATIVAS INTERNACIONALES..... 16
1.2 LÍMITES DE CAMPO Y CORRIENTE INDUCIDA EN COLOMBIA..... 19
1.3 ZONAS DE SERVIDUMBR 20
1.4 PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACIÓ 20

2. MÉTODO DE LOS ELEMENTOS FINITOS Y CORRIENTE DE CORTOCIRCUITO INDUCIDA..... 24
2.1 MÉTODO DE ELEMENTOS FINITOS APLICADO A LA ELECTROSTÁTICA 25
2.1.1 Discretización de la región..... 25
2.1.2 Ecuaciones aplicadas a cada elemento..... 26
2.1.3 Ensamble de los elementos..... 28
2.1.4 Solución de las ecuaciones resultantes..... 29
2.2 CORRIENTE DE CORTOCIRCUITO INDUCIDA 30
2.2.1 Cálculo del área equivalente 30

3. HERRAMIENTA DE SIMULACIÓN DE CAMPOS ELÉCTRICOS CON OBJETOS BAJO LÍNEAS DE TRANSMISIÓN..... 35
3.1 HERRAMIENTA COMPUTACIONAL CELTIO 35
3.1.1 Parámetros del modelo 36
3.1.1.1 Configuración de la línea..... 36
3.1.1.2 Nivel de tensión..... 37
3.1.1.3 Objeto bajo la línea..... 38
3.1.2 Solución del modelo 39
3.1.2.1 Representación gráfica..... 39

3.1.2.2 Magnitud de campo eléctrico y corriente de cortocircuito 42
3.1.3 Recomendaciones de uso del Software 43

4. RESULTADOS DE SIMULACIÓN OBTENIDOS 44
4.1 VALIDACIÓN DEL MÉTODO 44
4.2 CASOS DE SIMULACIÓN 48
4.2.1 Caso 1. Línea de 115 kV con vivienda típica bajo la línea 48
4.2.2 Caso 2. Línea de 500 kV en delta con persona bajo la línea 49
4.2.3 Caso 3. Línea de 400 kV en configuración vertical doble circuito con camión bajo la línea 52

CONCLUSIONES 56

BIBLIOGRAFIA 58

ANEXOS 60

LISTA DE FIGURAS

Pág.

Figura 1. Magnitudes de campo máximo recomendadas 16

Figura 2. Porcentaje de personas que perciben la descarga en diferentes campos eléctricos. 17

Figura 3.Niveles de referencia para la exposición a campos eléctricos 18

Figura 4. Ancho de la zona de servidumbre. 21

Figura 5. Ejemplo de red de alta tensión sobre viviendas. 22

Figura 6. Líneas de Alta tensión en barrio Santa María. 23

Figura 7. Líneas de Alta tensión en San Luisito. 23

Figura 8. Elemento triangular típico. 26

Figura 9. Factor de forma para una caja. 31

Figura 10. Factor de forma para una vivienda con techo conductor triangular. 31

Figura 11. Factor de forma para placa horizontal. 32

Figura 12. Área equivalente usando el método de los 45º. 34

Figura 13. Interfaz gráfica programa CELTIO. 36

Figura 14. Configuración de la línea en CELTIO. 37

Figura 15. Panel Nivel de tensión de la línea. 37

Figura 16. Geometría del modelo para una línea en configuración horizontal con un camión bajo la línea. 39

Figura 17. Geometría del modelo para una persona ubica bajo una línea de 120 kV en configuración triangular (delta). 40

Figura 18. Malla del modelo para una persona ubica bajo una linea de 120 kV en configuración triangular (delta). 41

Figura 19. Intensidad de campo eléctrico y Corriente de cortocircuito inducida. 41

Figura 20. Solución al problema para una persona ubica bajo una línea de 120 kV en configuración triangular (delta). 42

Figura 21. Configuración de la línea Juanchito-Candelaria de 115 kV. 45

Figura 22. Perfil de campo valor rms. 47

Figura 23. Línea Juanchito-Candelaria de 115 kV en el software de simulación. 48

Figura 24. Configuración en delta línea de 500 kV. 50

Figura 25. Línea de 500 kV en el software de simulación. 52

Figura 26. Configuración en delta línea de 500 kV. 53
Figura 27. Línea de 400 kV en el software de simulación..... 54

LISTA DE TABLAS

Pág.

Tabla 1. Valores límites de exposición a campos electromagnéticos. 19
Tabla 2. Ancho de la zona de servidumbre de líneas de transmisión 21
Tabla 3. Área equivalente para algunas formas comunes. 33
Tabla 4. Valores medidos de Capacitancia a tierra, Isc/E, S, y altura efectiva para algunos objetos comunes. 34
Tabla 5. Área equivalente para los objetos utilizados. 43
Tabla 6. Parámetros de la línea Juanchito-Candelaria de 115 kV. 45
Tabla 7. Comparación Campo valor rms. 46
Tabla 8. Campo eléctrico y corriente inducidos sobre el objeto. 49
Tabla 9. Parámetros de línea de 500 kV. 50
Tabla 10. Campo eléctrico y corriente inducidos sobre el objeto. 51
Tabla 11. Parámetros de línea de 500 kV. 53
Tabla 12. Campo eléctrico y corriente inducidos sobre el objeto. 55

RESUMEN

TÍTULO: CAMPO ELÉCTRICO INDUCIDO EN OBJETOS CERCANOS A UNA LÍNEA DE TRANSMISIÓN DE ALTA TENSIÓN.*

AUTOR: JHONATHAN EDUARDO CORREA CABALLERO**

PALABRAS CLAVES: Métodos numéricos, Campo eléctrico inducido, Líneas de transmisión, Método de elementos finitos.

DESCRIPCIÓN:

Los efectos de líneas de transmisión aéreas han sido objeto de preocupación en la ingeniería eléctrica desde hace más de tres décadas. A niveles de alta tensión, se ha buscado establecer un método de medición de la tensión electrostáticamente inducida en los seres humanos o en vehículos, cercas y configuraciones típicas con las que los seres humanos puedan llegar a tener contacto. Gracias a los avances computacionales en las últimas décadas es posible generar una simulación basada en métodos numéricos que permita estimar los niveles de campo a los que puede estar sometido un cuerpo bajo líneas de alta tensión.

En este trabajo se realiza un programa computacional en 2D que permite simular el campo eléctrico que una línea de transmisión causa sobre algunos objetos que se encuentren bajo la línea, en la zona de servidumbre. Este programa permite estimar la magnitud de campo eléctrico inducido en algunos de los objetos que comúnmente se encuentran bajo la línea a partir de la solución obtenida del problema usando el método de elementos finitos aplicado a la geometría del problema en el software MATLAB.

Se toma como punto de partida la revisión de la literatura técnica internacional, los límites de campo eléctrico inducido definidos por la regulación colombiana y las posibles consecuencias que estos pueden generar. Luego se describe la aplicación del método de elementos finitos a un problema electrostático bidimensional y las ecuaciones y consideraciones necesarias para su aplicación en este proyecto.

Finalmente el programa se realiza en MATLAB usando los beneficios que la herramienta PDEtool ofrece para la programación y visualización de la solución obtenida, y la representación de los objetos típicos, como personas, vehículos o viviendas mediante figuras geométricas simples. También se analizan los valores de campo eléctrico obtenidos en el programa para distintos casos simulados en el mismo.

* Trabajo de investigación

** Facultad de Ingenierías Físico-mecánicas. Escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones. Director: Julio Cesar Chacón Velasco. Codirector: Ernesto Aguilera Bermúdez

ABSTRACT

TITLE: ELECTRIC FIELD INDUCED ON OBJECTS AROUND TRANSMISSION LINES*

AUTHOR: JHONATHAN EDUARDO CORREA CABALLERO**

KEYWORDS: Numerical methods, Electric field induction, Transmission lines, Finite Element method.

DESCRIPTION:

The effects caused by transmission lines have been object of study in electrical engineering research since at least three decades. At high voltage levels, in particular, there have been developed a number of methods to stablish the electrostatic voltage induced on human beings or objects which anyone can come into contact with. Thanks to computational development in the las decades it is possible today generate a numerical simulation that allow us to estimate field values at which a human body can be expose under typical transmission lines.

In this work it is made a 2D program that allows to simulate electric field induced on some objects under a transmission line, in the easement zone. This program estimates the magnitude of electric field induced from the solution obtained through the application of finite element method (FEM) to the geometry of each problem case.

This work takes, as starting point, a review of technical international papers, Colombian regulated limits to electric field exposure and the finite element method and some of the viable consequences that those fields could generate. Then it is described how the finite element method is applied to a electrostatic 2D problem and the equations that describe it.

The final software is written in MATLAB platform, using the benefits that PDEtool offers to programming, showing graphically the final solution, and representing objects as a person, a vehicle or a house through simple geometric shapes. Finally, electric field values obtained in the problems solved are compared.

* Research work

** Faculty of Physical-Mechanic Engineering. School of Electrical, Electronical and Telecommunications Engineering. Director: Julio Cesar Chacón Velasco. Co-director: Ernesto Aguilera Bermúdez.

NOMENCLATURA

FEM	Método de los elementos finitos, por sus siglas en Ingles.
PDE	Ecuación en derivadas parciales, por sus siglas en Ingles.
GUIDE	Editor de diseño de interfaz gráfica de usuario.
RETIE	Reglamento Técnico de Instalaciones Eléctricas.
EMF	Campo electromagnético, por sus siglas en Ingles.
ICNIRP	Comisión Internacional sobre Protección Frente a Radiaciones No Ionizantes.
ANSI	American National Standards Institute.
RMS	Valor eficaz, por sus siglas en Ingles.
OMS	Organización Mundial de la Salud.
STR	Sistema de transmisión regional.
SIN	Sistema Interconectado Nacional.

INTRODUCCIÓN

Varios métodos analíticos han sido desarrollados para describir los campos electromagnéticos alrededor de una línea de transmisión. Estos métodos permiten definir los niveles de campo, a diferentes alturas y distancias de las líneas, según la configuración de la línea de transmisión [1].

Para los objetos que se encuentran alrededor de la línea, el campo eléctrico sobre del objeto es usualmente representado a partir del campo promedio no perturbado. Esta consideración permite evaluar la inducción sin valorar la geometría del objeto, teniendo en cuenta que la presencia de un objeto tiene un efecto insignificante en las cargas eléctricas sobre el conductor que generan el campo eléctrico y de esta manera se simplifica el cálculo [2]. Actualmente, el cálculo de los campos electromagnéticos se puede estimar con relativa rapidez y mayor precisión usando métodos numéricos, como el método de elementos finitos (FEM, por sus siglas en inglés) dadas las ventajas que este método ofrece para resolver ecuaciones diferenciales en espacios complejos.

El presente trabajo de grado tiene como objetivo crear un programa computacional que permita estimar el campo eléctrico alrededor de algunos objetos alrededor de una línea de transmisión de alta tensión, aprovechando las facilidades que el método de elementos finitos ofrece para este tipo de problemas y las herramientas, como PDEtool y GUIDE, que el software MATLAB brinda para la programación de dicho programa.

Con base en los aspectos mencionados anteriormente la estructura del documento está organizada de la siguiente manera; En el capítulo 1 se presentan las consideraciones de seguridad y regulación que algunas normativas internacionales y la normativa colombiana tienen sobre los efectos que la exposición a campos electromagnéticos pueden generar en la salud humana. También se presentan

algunos casos donde la regulación colombiana no es respetada que representan la principal motivación para el desarrollo del software objeto de este trabajo.

En el capítulo 2 se describe brevemente como el método de elementos finitos es aplicado a problemas electrostáticos y como este método permite obtener el campo eléctrico inducido bajo una línea de transmisión eléctrica con objetos abajo de la línea, teniendo en cuenta ciertas consideraciones y simplificaciones. Además, se presenta un método para obtener la corriente de cortocircuito inducida en el objeto a partir del campo eléctrico inducido.

En el capítulo 3 se describe el programa de computación, llamado CELTIO, que permite simular el campo eléctrico inducido bajo líneas de transmisión en dos dimensiones para algunos objetos que pueden llegar a encontrarse bajo las líneas. Aquí se describe el alcance del programa, los elementos que lo componen y algunas recomendaciones para la correcta utilización del mismo.

Finalmente en el capítulo 4 se comienza comparando los resultados obtenidos por el programa con un caso particular donde se tienen valores de campo medidos bajo la línea como método de verificación de la exactitud del programa. Luego se simulan tres casos de objetos ubicados bajo líneas de transmisión y se observa si los resultados obtenidos superan o no los límites establecidos por la regulación colombiana.

1. FUNDAMENTACIÓN TEÓRICA

1.1 NORMATIVAS INTERNACIONALES

Varias guías de seguridad y salud, y regulaciones de múltiples fuentes, relacionadas con las líneas de transmisión de alta tensión son publicadas periódicamente alrededor del mundo. Entre ellas se encuentran el código nacional de seguridad Eléctrica (NESC), la Conferencia Americana de Higienistas Industriales Gubernamentales (ACGIH), el Consejo Nacional de Protección Radiológica de Inglaterra, y la Comisión Internacional sobre Protección Frente a Radiaciones No Ionizantes (ICNIRP), cuyos límites de campo son tomados como criterio en Colombia por el Reglamento Técnico de Instalaciones Eléctricas (RETIE) [3]. Los límites de algunas de estas entidades se resumen en la figura 1.

Figura 1. Magnitudes de campo máximo recomendadas

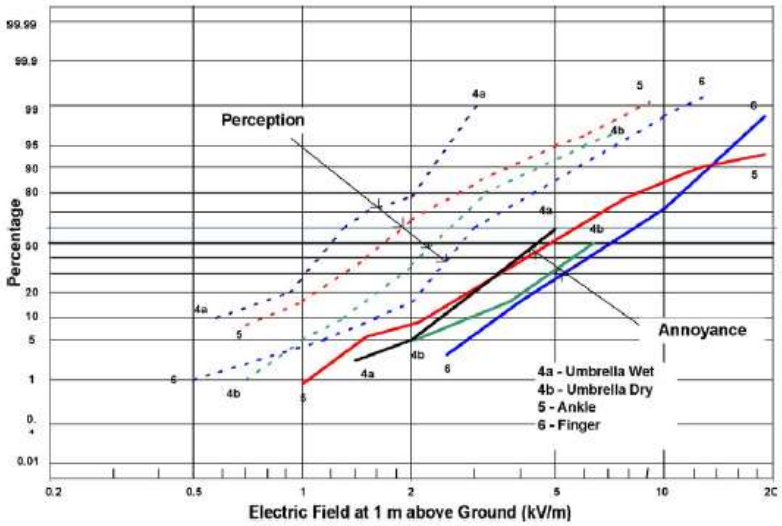
Valores guía de campos electromagnéticos a 60 Hz establecidos por organizaciones de Salud y Seguridad		
Organización	Campo Magnético	Campo Eléctrico
American Conference of Governmental and Industrial Hygienists (ACGIH) (Ocupacional y General)	10 mG 1000 mG	25 kV/m 1 kV/m
International Commision on Non-Ionizing Radiation Protection (ICNIRP)(Publico general, exposición continua)	2000 mG	4.2 kV/m
Non-Ionizing Radiation (NIR) Commitee of the American Industrial Hygiene Assoc. (AIHA) endorse in 2003 as ICNIRP's occupational levels for workers	4170 mG	8.3 kV/m
Institute of Electrical and Electronics Engineers (IEEE) Standard c95.6 (Publico general, exposición continua)	9040 mG	5 kV/m
UK National Radiological Protection Board (NRPB) (now Health Protection Agency [HPA])	2000 mG	4.2 kV/m
Australian Radiation and Nuclear Safety Agency (ARPANSA), Dec. 2006.	3000 mG	4.2 kV/m

Fuente: Electromagnetic Waves (EMF and RF) and Health Effect. Peter Valberg. 2015.

Como se puede observar existen en la figura 1, existen múltiples criterios sobre los valores de campo que son perjudícales para la salud. En 2002, la Agencia Internacional para la Investigación del Cáncer clasificó a las líneas de transmisión como “posiblemente cancerígenas para los seres humanos”, llamando así a la prevención a la exposición prolongada, pero teniendo en cuenta que no se ha encontrado aún un mecanismo biológico que soporte tales efectos. Se entiende que dado que las líneas de 50 Hz o de 60 Hz, como en el caso de Colombia, tienen un ancho de banda mucho mayor al tamaño del cuerpo humano y que la exposición a los campos no es constante, la interacción resulta en bajos niveles de energía depositados en el cuerpo y su peligrosidad no es evidente [2].

Las descargas inducidas por los campos eléctricos inducidos no tienen la energía suficiente para causar choques eléctricos, sin embargo, pueden causar sensaciones molestas perceptibles a una persona que está tocando un objeto aislado inmerso o a una persona aislada tocando un objeto con conexión a tierra. La sensibilidad a este fenómeno depende del punto de contacto, donde los puntos más sensibles son las manos, tobillos y dedos [4]. Este se ve reflejado en la figura 2, donde se muestra el porcentaje de personas que experimenta sensaciones en diferentes situaciones de descarga.

Figura 2. Porcentaje de personas que perciben la descarga en diferentes situaciones.



Fuente: IEEE 1978; EPRI 1982.

A la fecha, no existe un efecto de campos eléctricos en alterna del tipo y magnitud producidos en las líneas de transmisión que haya sido ampliamente detectado y aceptado por la comunidad científica. En 1972 se reportaron quejas de problemas en la salud de trabajadores en la Unión Soviética pero tanto exámenes médicos previamente realizados en trabajadores en los Estados Unidos en 1967 [5], así como otros estudios posteriores realizados en Suecia por Knave en 1980 y en Canadá por Roberge en 1976, no lograron atribuir ningún problema de salud a este tipo de campos eléctricos.

En cuanto a la corriente inducida en objetos, actualmente existe un estándar en Estados Unidos para objetos bajo una línea de transmisión, buscando minimizar posibles choques eléctricos y movimientos involuntarios que podrían causar un accidente (ANSI 2002). Esta corriente es la corriente de estado estacionario debido a efectos electroestáticos, en caso de que el objeto fuera cortocircuitado a tierra. El estándar limita la corriente de fuga a 5 mA bajo líneas de transmisión mayores a 98 kV AC [6].

Figura 3.Niveles de referencia para la exposición a campos eléctricos

Niveles de referencia de exposición al público general para campos eléctricos y magnéticos variables en el tiempo (valor rms)			
Rango de frecuencia	Intensidad de campo	Campo magnético	Densidad de flujo magnético
Hz	E (kV/m)	H (A/m)	B (T)
1-8	5	$3.2X10^4/f^2$	$4X10^{-2}/f^2$
8-25	5	$4X10^3/f$	$5X10^{-3}/f$
25-50	5	$1.6X10^2$	$2X10^{-4}$
50-400	$2.5X10^2/f$	$1.6X10^2$	$2X10^{-4}$
400-3k	$2.5X10^2/f$	$6.4X10^4/f$	$8X10^{-2}/f$
3k-10M	0.083	21	$2.7X10^{-5}$

Fuente: International Commission on non-ionizing radiation protection ICNIRP. 2010.

1.2 LÍMITES DE CAMPO Y CORRIENTE INDUCIDA EN COLOMBIA

Los valores máximos de intensidad de campo eléctrico y densidad de flujo magnético en baja frecuencia permitidos en Colombia están reglamentados en el RETIE en el artículo 14 siguiendo los criterios de la OMS y la ICNIRP en su revisión del año 2013 [3]. La sección 14.3 expone los valores límites de exposición según la actividad de las personas expuestas, ya sea por una exposición ocupacional o para el público en general, como se muestra en la tabla 1.

Tabla 1. Valores límites de exposición a campos electromagnéticos.

Tipo de exposición	Intensidad de campo eléctrico [kV/m]	Densidad de flujo magnético [μT]
Exposición ocupacional en un día de trabajo de ocho horas.	8,3	1000
Exposición del público en general hasta ocho horas continúas.	4,16	200

Fuente: RETIE. 2013.

Además de los límites de campo eléctrico y magnético a los que se puede ver expuesto una persona directamente, en Colombia existe un estándar para la corriente de cortocircuito inducida sobre un objeto que se encuentre bajo una línea de transmisión. Este estándar para la corriente de cortocircuito inducida está considerado en el artículo 13.2 del RETIE, para los casos en los que se requiera usar una distancia mayor a la recomendada por el reglamento, se debe garantizar que este valor de corriente no supere los 5 mA. Esta corriente corresponde a la corriente inducida de estado estacionaria descrita en la norma ANSI.

En este artículo en la nota agregada se puede citar: "*En el caso de tensiones línea – tierra que superen 98 kV, se podrán aumentar las distancias de la Tabla 13.2 o disminuir*

el campo eléctrico, considerando que el vehículo o equipo más grande esperado bajo la línea fuera conectado a tierra para limitar a 5 mA rms la corriente de estado estacionario debida a los efectos electrostáticos. Para calcular esta condición los conductores deben estar desenergizados y la flecha a 50 ºC".

1.3 ZONAS DE SERVIDUMBRE

En el reglamento Colombiano, específicamente en el RETIE, en el artículo 3, se definen la zona de servidumbre como una "*franja de terreno que se deja sin obstáculos a lo largo de una línea de transporte o distribución de energía eléctrica, como margen de seguridad para la construcción, operación y mantenimiento de dicha línea, así como para tener una interrelación segura con el entorno*."

En el artículo 22.2 del reglamento se indica que no se deben construir edificios, edificaciones, viviendas, casetas o cualquier tipo de estructuras para albergar personas o animales. Tampoco se debe permitir alta concentración de personas en estas áreas de servidumbre, o la presencia permanente de trabajadores o personas ajenas a la operación o mantenimiento de la línea, ni el uso permanente de estos espacios como lugares de parqueo, o reparación de vehículos o para el desarrollo de actividades comerciales o recreacionales.

Para las tensiones normalizadas en el país, en la Tabla 22.1 del RETIE se fijan los valores mínimos requeridos en el ancho de la zona de servidumbre, cuyo centro es el eje de la línea. Estos valores se muestran a continuación en la tabla 2.

1.4 PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACIÓN

Como se ha descrito anteriormente, los efectos que los campos inducidos bajo una línea de transmisión de alta de tensión tienen sobre la salud humana son aún un campo de investigación bastante amplio. Aun cuando no se tiene total certeza de las magnitudes de campo eléctrico máximas que puede soportar una persona que se ve expuesta a la presencia de tales campos, varias entidades internacionales han definido límites de

magnitud y tiempo de exposición bajo los cuales se han podido realizar labores de mantenimiento de las líneas sin que se registren afectaciones a la salud de sus operarios.

Figura 4. Ancho de la zona de servidumbre.

Fuente: RETIE. 2013.

Tabla 2. Ancho de la zona de servidumbre de líneas de transmisión

TIPO DE ESTRUCTURA	TENSIÓN (kV)	ANCHO MÍNIMO (m)
Torres/postes	500 (2 Ctos.)	65
	500 (1 Cto.)	60
Torres/postes	400 (2 Ctos.)	55
	400 (1 Cto.)	50
Torres	220/230 (2 Ctos.)	32
	220/230 (1 Cto.)	30
Postes	220/230 (2 Ctos.)	30
	220/230 (1 Cto.)	28
Torres	110/115 (2 Ctos.)	20
	110/115 (1 Cto.)	20
Postes	110/115 (2 Ctos.)	15
	110/115 (1 Cto.)	15
Torres/postes	57,5/66 (1 o 2 Ctos.)	15

Fuente: RETIE. 2013.

En Colombia, se limitan los posibles efectos causados por los campos eléctricos mediante los valores límites de exposición de campos electromagnéticos y la corriente electroestática máxima inducida en objetos que se encuentren bajo la línea de transmisión. Estos valores están reglamentados por el Ministerio de Minas y Energía y están prescritos en el RETIE, se basan en las indicaciones de la Organización Mundial de la Salud OMS y la Comisión Internacional sobre Protección Frente a Radiaciones No Ionizantes (ICNIRP ; por sus siglas en inglés).

A pesar de ello, se han presentado casos en el territorio colombiano donde estos límites no se han respetado y las persona, civiles u operarios de la red, se ven expuestas a valores que sobrepasan lo permitido por la ley y aún más importante, que podrían poner en riesgo la integridad humana.

Durante los últimos años en Colombia algunos medios de comunicación han registrado continuas quejas de ciudadanos que consideran estar siendo afectados por la presencia de redes eléctricas de alta tensión que no cumplen con las distancias de seguridad establecidas en Colombia en el RETIE, como se muestra a continuación:

Figura 5. Ejemplo de red de alta tensión sobre viviendas.

Fuente: Diario la opinión, Ocaña, 28 de abril de 2016.

Figura 6. Líneas de Alta tensión en barrio Santa María.

Fuente: Diario El Heraldo, Barranquilla, 2 de Octubre de 2012.

Figura 7. Líneas de Alta tensión en San Luisito.

RIESGO / COMUNIDAD DE DOS BARRIOS ALERTA SOBRE GRAVE PELIGRO

Alarma por zona energizada

El presidente de la JAC de San Luisito, José Ómar Salazar, vio morir electrocutado a un perro por la corriente de una cuerda de 115 kilovatios. Vecinos se quejan de dolores de cabeza.

Algunos vecinos consideran que el nombre de 'zona energizada' es demasiado suave para un sector donde han muerto electrocutados varios perros y un número no determinado de personas ha sufrido corrientazos.

El presidente de la Junta de Acción comunal del barrio San

Fuente: Grupo de investigación en Alta tensión GRALTA, Universidad del Valle.

Por tal motivo este proyecto, al simular el campo eléctrico inducido en objetos cercanos a una línea de transmisión de alta tensión, busca estimar si los valores de campo eléctrico inducido y de corriente de corto para los casos evaluados representan valores de campo y corriente considerados como peligrosos, para que luego las entidades o personas encargadas de la seguridad y protección de los ciudadanos puedan estimar las medidas necesarias para corregir estos casos que se encuentran fuera de la lo estipulado por el reglamento eléctrico colombiano.

2. MÉTODO DE LOS ELEMENTOS FINITOS Y CORRIENTE DE CORTOCIRCUITO INDUCIDA

El método de elementos finitos fue desarrollado inicialmente para aplicaciones en el campo de la ingeniería mecánica y la ingeniería civil [7]. La idea principal de este método consiste en que aunque una función puede ser compleja aplicada en una región de gran tamaño, una aproximación simple de esta función puede ser suficiente para una pequeña subregión.

En este método, la región total del problema se divide en un número finito de subregiones llamadas elementos finitos. Estos elementos no se superponen unos a otros y en el caso de aplicaciones en dos dimensiones, suelen usarse polígonos simples como triángulos y rectángulos para subdividir la región. Los triángulos son el elemento más común y es el elemento usado en el programa desarrollado en este trabajo, ya que cualquier región arbitrariamente de su forma, puede ser fácilmente aproximada mediante un conjunto de triángulos. Las ecuaciones a resolver son a menudo representadas en términos de integrales, en términos de energía y no como ecuaciones de campo. Estas ecuaciones son resueltas computacionalmente para cada uno de los elementos y la solución total es la suma integral de las soluciones de todos los elementos.

El método de elementos finitos ha tenido múltiple aplicaciones en el campo del electromagnetismo desde los años 70, gracias a que permite resolver ecuaciones diferenciales, como las ecuaciones que definen los campos electromagnéticos, y a diferencia de otros métodos, como son el método de diferencias finitas y el método de

los momentos, permite manejar más fácilmente aquellos problemas donde las regiones y fronteras tienen formas irregulares [8]. También permite que un único programa pueda ser usado para resolver diferentes problemas que estén descritos por las mismas ecuaciones diferenciales parciales cambiando únicamente los parámetros de entrada, que en el caso de este trabajo de investigación, facilita que el programa desarrollado sea de utilidad para diferentes niveles de tensión y distintos objetos que se encuentre bajo una línea de transmisión.

2.1 MÉTODO DE ELEMENTOS FINITOS APLICADO A LA ELECTROSTÁTICA

El análisis de un problema de elementos finitos involucra básicamente cuatro pasos: primero se discretiza la región en la que se va a solucionar el campo mediante un número finito de elementos, después se procede a definir las ecuaciones que gobiernan a uno de los elementos, llamado elemento típico, se ensamblan todos los elementos en la región especificada y luego se resuelven las ecuaciones obtenidas mediante iteraciones.

2.1.1 Discretización de la región

En esta parte se divide inicialmente la región en un número finito de elementos, en este caso triangulares, para los que se busca obtener una solución a la tensión Ve dentro de cada elemento *e* .Luego se interrelacionan la tensión de todos los elementos de tal manera que sea continúo a través de las fronteras que interconectan unos elementos con otros.

La solución aproximada para la región total es,

$V(x,y) = \sum_{e=1}^{N} Ve(x,y)$ (2.1)

Donde *N* es el número de elementos triangulares en los que se subdivide la región. Para cada uno de los elementos la tensión Ve es aproximada mediante la siguiente ecuación:

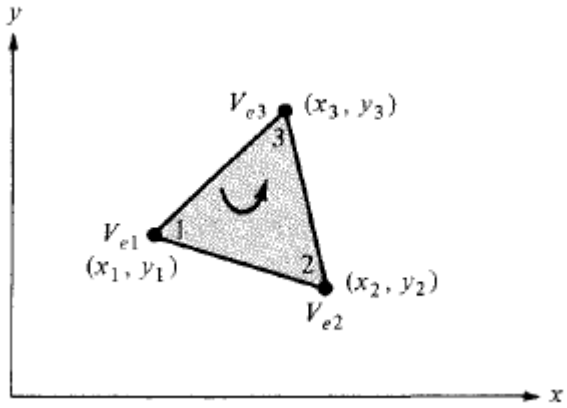
$Ve(x,y) = a + bx + cy$ (2.2)

$Ee = -\nabla Ve = -(ba_x + ca_y)$ (2.3)

Es decir, que para cada elemento triangular se asume que la variación de la tensión es lineal, por tanto se está asumiendo que el campo eléctrico Ee es uniforme dentro de cada elemento.

2.1.2 Ecuaciones aplicadas a cada elemento

Figura 8. Elemento triangular típico.



Fuente: Elements of Electromagnetics. Sadiku 3th Edition. 2000.

Para un elemento triangular que se encuentra en el plano cartesiano como el que se muestra en la figura 8, las tensiones $Ve1, Ve2\ y\ Ve3$ en los nodos 1,2 y 3 son:

$$\begin{bmatrix} Ve1 \\ Ve2 \\ Ve3 \end{bmatrix} = \begin{bmatrix} 1 & x1 & y1 \\ 1 & x2 & y2 \\ 1 & x3 & y3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \text{ (2.4)}$$

Sustituyendo esta ecuación en la ecuación 2.2 se tiene que,

$$Ve = [1 \quad x \quad y]\frac{1}{2A}\begin{bmatrix}(x2y3 - x3y2) & (x3y1 - x1y3) & (x1y2 - x2y1)\\ (y2 - y3) & (y3 - y1) & (y1 - y2)\\ (x3 - x2) & (x1 - x3) & (x2 - x1)\end{bmatrix}\begin{bmatrix}Ve1\\ Ve2\\ Ve3\end{bmatrix} \quad (2.5)$$

Donde,

$$A = \frac{1}{2}\,[(x2 - x1)(y3 - y1) - (x3 - x1)(y2 - y1)] \quad (2.6)$$

La variable *A* representa el área del elemento *e,* y si los nodos son numerados en dirección anti horaria iniciando desde cualquier nodo, el resultado es positivo. Esta ecuación define la tensión de en cualquier punto de coordenadas x,y que se encuentre dentro del elemento si se conocen inicialmente los valores de tensión en los nodos.

La energía por unidad de longitud We asociada al elemento e, asumiendo que la región está libre de carga ($\rho_s = 0$) está dada por,

$$We = \frac{1}{2}\int \varepsilon|E|^2 dS = \frac{1}{2}\int \varepsilon|\nabla Ve|^2 dS \quad (2.7)$$

Que escrito en forma matricial es igual a,

$$We = \frac{1}{2}\varepsilon[Ve]^T[C^e][Ve] \quad (2.8)$$

Donde $[Ve]^T$ denota a la transpuesta de la matriz Ve y C^e es usualmente la matriz de coeficientes o matriz de rigidez. El elemento ${C_{ij}}^e$ Puede ser considerado como el acoplamiento entre los nodos *i* y *j* del elemento *e*.

Una manera práctica de calcular los elementos de la matriz de coeficientes C^e consiste en definir,

$$P1 = (y2 - y3) \quad (2.9)$$

$$P2 = (y3 - y1) \quad (2.10)$$

$$P3 = (y1 - y2) \quad (2.11)$$

$$Q1 = (x3 - x2) \quad (2.12)$$

$$Q2 = (x1 - x3) \quad (2.13)$$

$$Q3 = (x2 - x1) \quad (2.14)$$

Luego con *Pi* y *Qi*, (donde *i*=1,2,3 son los nodos locales del elementos), cada término de la matriz de coeficientes es equivalente a:

$$C_{ij}{}^{e} = \frac{1}{4A}[PiPj + QiQj] \quad (2.15)$$

Donde,

$$A = \frac{1}{2}(P2Q3 - P3Q2) \quad (2.16)$$

$$\sum_{i=1}^{3} C_{ij}{}^{e} = 0 = \sum_{j=1}^{3} C_{ij}{}^{e} \quad (2.17)$$

2.1.3 Ensamble de los elementos

La energía asociada con el ensamblaje de todos los elementos en la malla está dada por,

$$W = \sum_{e=1}^{N} We = \frac{1}{2}\varepsilon[V]^{T}[C][V] \quad (2.18)$$

Donde,

$$[V] = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} \quad (2.19)$$

n es el número total de nodos, *N* es el número de elementos, y $[C]$ es llamada la matriz total de coeficientes, que representa el ensamblaje de las matrices individuales de coeficientes de los elementos.

Para obtener C_{ij} se debe considerar el hecho de que la distribución del potencial debe ser continua a través de las fronteras que unen un elemento con otro. La contribución a la posición *i,j* en $[C]$ proviene únicamente de los elementos que contienen los nodos *i* y *j*. La numeración de los nodos en la matriz global $[C]$ no necesariamente corresponde con la numeración de las matrices $[C^e]$ de cada elemento *e* y no necesariamente debe hacerse en sentido anti horario. La matriz de coeficientes es una matriz de *nxn,* simétrica y singular.

2.1.4 Solución de las ecuaciones resultantes

Según la teoría de cálculo variacional, la ecuación de Laplace que describe el campo eléctrico a solucionar se satisface cuando la energía en la región de la solución sea mínima. [7]. Esto implica que las derivadas parciales de *W* con respecto a cada nodo sea cero, esto es,

$$0 = \frac{\partial W}{\partial V_1} = \frac{\partial W}{\partial V_2} = \cdots = \frac{\partial W}{\partial V_n} = \sum_{i=1}^{n} V_i C_{ik} \quad (2.20)$$

Donde *n* es el número de nodos en la malla. De aquí se observa que si se quiere encontrar el potencial en un nodo libre iterativamente se debe empezar por definir la tensión en los nodos fijos, y asignar un valor de cero o un valor promedio a los nodos libres, de esta manera, el potencial en un nodo libre *k* es igual a:

$$V_k = \frac{1}{C_{kk}} \sum_{i=1, i\neq k}^{n} V_i C_{ik} \quad (2.21)$$

Al finalizar la primera iteración los nuevos valores de tensión calculados para los nodos libres son asignados para la segunda iteración. Este proceso se repite hasta que el cambio en la tensión en dos iteraciones subsecuentes en un mismo nodo pueda ser

considerado como despreciable.

2.2 CORRIENTE DE CORTOCIRCUITO INDUCIDA

La corriente considerada por la normativa nacional y algunas normativas internacionales como parámetro para medir el efecto que produce el campo eléctrico inducido sobre un objeto bajo una línea de transmisión eléctrica es la corriente de corto circuito inducida. Esta corriente representa la corriente que conduciría un cortocircuito entre el objeto y tierra u otro objeto aterrizado.

El cálculo de la corriente inducida por una línea de transmisión en un objeto involucra la solución de un problema tridimensional. Por tanto, la corriente de cortocircuito I_{sc} para un objeto que se encuentra puesto a tierra puede ser calculada simplificadamente de la siguiente manera,

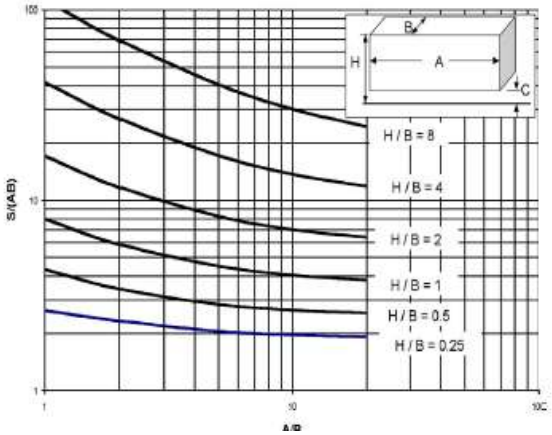
$$I_{sc} = j\omega\epsilon_0 ES \quad (2.22)$$

Donde,
ω es la frecuencia angular.
E es el campo normal promedio a la superficie del objeto.
S es el área equivalente al objeto que almacenaría la misma corriente que la corriente almacenada por el objeto.

2.2.1 Cálculo del área equivalente

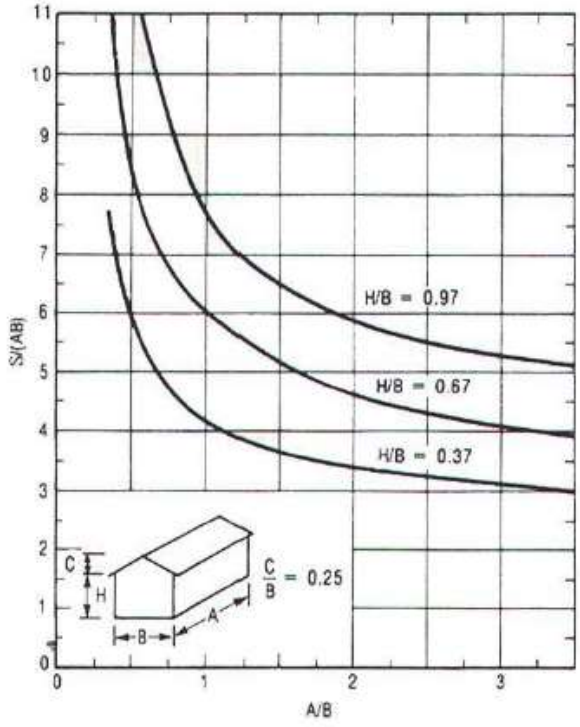
El cálculo del área equivalente ha sido estimado y medido para varios objetos comunes desde los años 70 [9]. Para algunas formas típicas, como cajas, casas o placas se han generado curvas que entregan un factor de forma (*S/AB*) donde A representa la longitud, B la profundidad y H la altura sobre el plano de tierra. Algunas curvas se muestran a continuación,

Figura 9. Factor de forma para una caja.



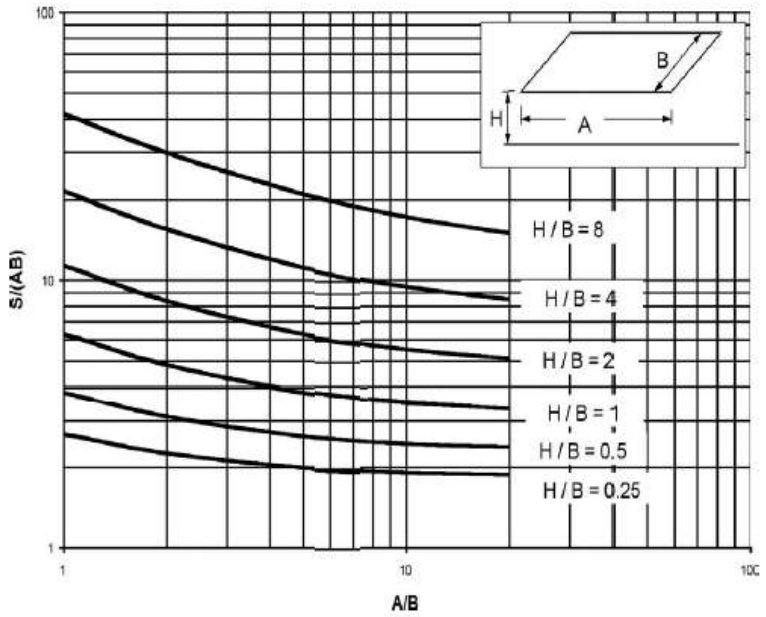
Fuente: EPRI 2005.

Figura 10. Factor de forma para una vivienda con techo conductor triangular.



Fuente: EPRI 2005.

Figura 11. Factor de forma para placa horizontal.



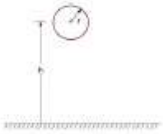
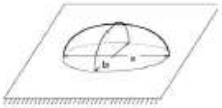
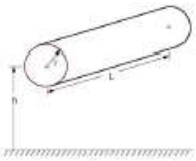
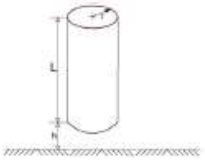
Fuente: EPRI 2005.

Para algunas formas existen fórmulas que permiten determinar el área equivalente de forma analítica. El EPRI resume algunas de las formas y la ecuación que define el área equivalente de las mismas en la tabla 3.

La exactitud de los resultados en el cálculo de la corriente de cortocircuito depende en gran medida de las suposiciones que se hagan sobre la forma del objeto. Cuando la forma de un objeto es complicada o poco simétrica, por razones prácticas, su área equivalente es estimada encerrando el objeto en una caja de dimensiones similares que represente al objeto (método de la caja), o metiendo el método de los 45º.

El método de los 45º consiste en proyectar las esquinas superiores del objeto en la superficie usando conos de 45º. El área total proyectada obtenida de esta forma es considerada como el área equivalente del objeto y se determina mediante la formula 2.23.

Tabla 3. Área equivalente para algunas formas comunes.

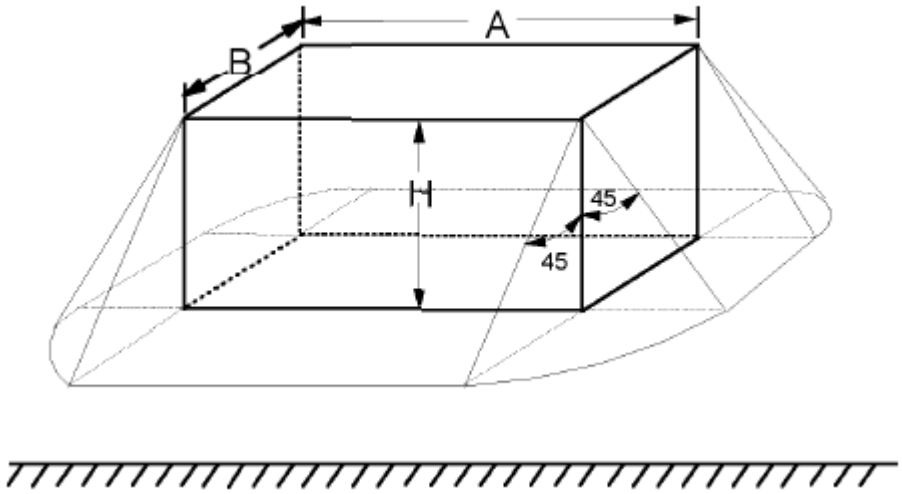
Objeto		Área Equivalente	Capacitancia de carga disruptiva
Esfera sobre el suelo			$C_s = 4\pi\varepsilon r$
Hemisferio sobre suelo		$S = 3\pi r^2$	
Medio elipsoide sobre el suelo		$S = 3\pi ab$	
Cilindro sobre el suelo		$S = \frac{2\pi h}{\ln(2h/r)} \cdot L + \frac{4\pi h}{\frac{1}{r} - \frac{1}{2h}}$	$C_s = \frac{2\pi\varepsilon}{\ln(2h/r)} L$
Medio cilindro sobre el suelo		$S = 4r(L + r)$	
Cilindro vertical		$S = \frac{\pi L^2}{\ln\left[\frac{L}{r}\sqrt{\frac{4h+L}{4h+3L}}\right]}$	

Fuente: EPRI 2005.

$S = AB + 2H(A + B) + \pi H^2$ (2.23)

Para algunos objetos con dimensiones estándar se han realizado mediciones de sus características eléctricas como son la corriente de cortocircuito, el área equivalente, la capacitancia a tierra o la altura efectiva. En la tabla 4 se muestran estos valores para varios objetos que comúnmente se encuentran bajo una línea de transmisión a una frecuencia de 60 Hz. La altura efectiva o equivalente es la altura que tendría una placa equivalente bajo un campo uniforme que produzca la misma corriente de cortocircuito que el objeto en estudio.

Figura 12. Área equivalente usando el método de los 45º.



Fuente: EPRI 2005.

Tabla 4. Valores medidos de Capacitancia a tierra, Isc/E, S, y altura efectiva para algunos objetos comunes.

60 Hz				
Objeto	Capacitancia a tierra ρF	$I_{sc}/E\ \mu A$/(V/m)	Área equivalente S m^2	Altura efectiva h
Persona (1.75 m, 68 kg)	100	0.018	5.395	0.477
Carro compacto	800	0.088	26.4	0.292
Valla (50 pies)	118	0.0445	13.3	1
Vagón	1200	0.13	39	0.287
Camión (4 ejes)	1000	0.34	101.9	0.9
Techo metálico (2.44m x 1.22 m)	105	0.028	8.39	0.707

Fuente: Radio-Frequency Hazards in the VLF to MF Band. Ghandi 1982.

3. HERRAMIENTA DE SIMULACIÓN DE CAMPOS ELÉCTRICOS CON OBJETOS BAJO LÍNEAS DE TRANSMISIÓN

A pesar de que la regulación eléctrica colombiana señala que no debería haber objetos ubicados bajo líneas de alta tensión en su zona de servidumbre, se han registrado casos donde esta situación no se cumple, y en tales situaciones es conveniente saber si las personas en contacto con estos campos están expuestas a altos niveles de peligro directa o indirectamente. Por lo tanto, el objetivo principal de este trabajo es crear una herramienta computacional de simulación que permita estimar con un cierto nivel de exactitud, el campo eléctrico generado por una línea de transmisión eléctrica bajo la presencia de algunos objetos comunes ubicados bajo la línea como en tales situaciones, y observar a partir de la simulación de campo el nivel de peligro.

3.1 HERRAMIENTA COMPUTACIONAL CELTIO[1]

La herramienta computacional está desarrollada con una interfaz gráfica en la herramienta de software matemático MATLAB. Dentro del software MATLAB el programa hace uso de una herramienta para solucionar ecuaciones parciales diferenciales mediante el método de elementos finitos llamada PDE Toolbox. Esta herramienta facilita la generación de la malla del problema, la formulación de las condiciones de frontera, y el trazado de geometrías complejas.

Este programa permite simular el campo eléctrico en 2D bajo líneas de transmisión de circuito sencillo o doble circuito para cinco casos típicos de objetos bajo la línea para cualquier nivel de tensión, y a partir de esta simulación obtener el campo eléctrico en cualquier punto del espacio y la corriente de cortocircuito inducida en el objeto. La interfaz gráfica del programa se muestra a continuación en la figura 13.

[1] La herramienta computacional CELTIO es la herramienta de simulación objetivo principal de este trabajo. Las siglas CELTIO provienen de los términos “Campo Eléctrico bajo Líneas de Transmisión Inducido en Objetos”.

Figura 13. Interfaz gráfica programa CELTIO.

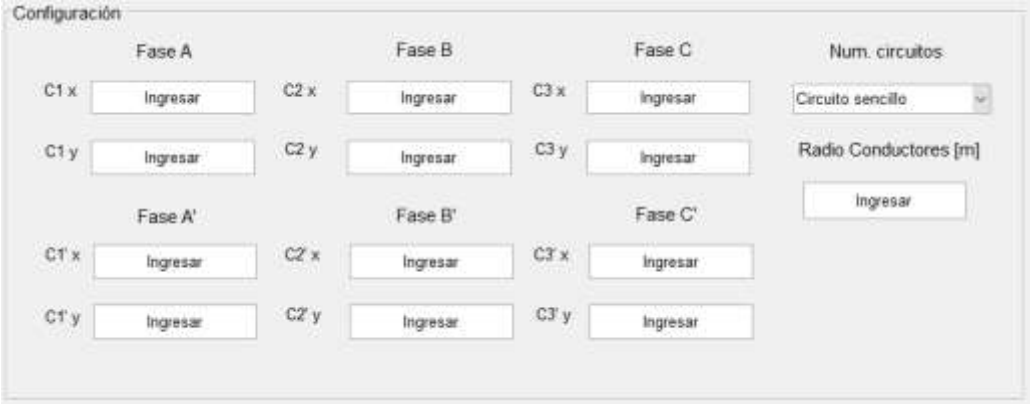
3.1.1 Parámetros del modelo

3.1.1.1 Configuración de la línea

En el panel llamado “Configuración” se ingresan los parámetros que definen la configuración de la línea de transmisión eléctrica a partir de las posiciones de los conductores, el radio de los conductores y si la línea es doble circuito o circuito sencillo.

En la lista de nombre “Num. Circuitos” se designa si la línea es de un solo circuito (circuito sencillo) o de doble circuito. Si se selecciona circuito sencillo solo las fases A, B y C harán parte del problema, si selecciona la opción de doble circuito, las fases A’, B’ y C’ harán parte del problema además de las fases A, B y C.

Figura 14. Configuración de la línea en CELTIO.

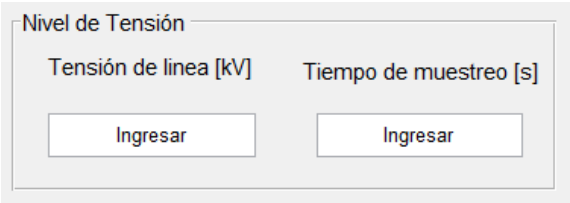


Las cajas de ingreso “C1x” y “C1y” representa las posiciones x,y en metros del conductor C1 asignado a la fase A. De la misma manera se asignan las posiciones de los demás conductores del problema a solucionar. En la caja de nombre “Radio conductores” se asigna el mismo radio en metros a todos los conductores de la configuración. En este programa la posición de los conductores siempre debe ser mayor a cero y no debe sobreponerse con los objetos para la correcta utilización del programa.

La selección de la posición de los conductores también es importante para la ubicación del objeto seleccionado, ya que este estará colocado en el punto medio entre el conductor más lejano y el conductor más cercano al origen de la coordenada x del plano cartesiano.

3.1.1.2 Nivel de tensión

Figura 15. Panel Nivel de tensión de la línea.



Las fases A, B y C representan las tres fases de la línea de transmisión trifásica donde a cada una se les asigna el nivel de tensión eléctrica en el conductor de la siguiente manera,

$V_A = V_{A'} = \frac{V_L}{\sqrt{3}} * \cos(\omega t)$ (3.1) [kV]

$V_B = V_{B'} = \frac{V_L}{\sqrt{3}} * \cos(\omega t - 120)$ (3.2) [kV]

$V_C = V_{C'} = \frac{V_L}{\sqrt{3}} * \cos(\omega t + 120)$ (3.3) [kV]

Donde,

V_A , V_B y V_c son los niveles de tensión en cada una de las fases A, B y C de la configuración. De la misma manera se asignan $V_{A'}$, $V_{B'}$ y $V_{c'}$ a las fases A', B' Y C'.
V_L es la tensión de línea en kV asignada a la línea. Esta tensión es ingresada en la caja de nombre "Tensión de línea" en el panel "Nivel de Tensión".
ω es la frecuencia angular en radianes sobre segundos para una frecuencia de 60 Hz típica en la transmisión de energía eléctrica en alterna en Colombia.
t es el instante en segundos que determina las tensiones en los conductores al hallar la solución de campo eléctrico. Este tiempo es ingresado en la caja "Tiempo de muestreo" en el panel "Nivel de Tensión".

3.1.1.3 Objeto bajo la línea

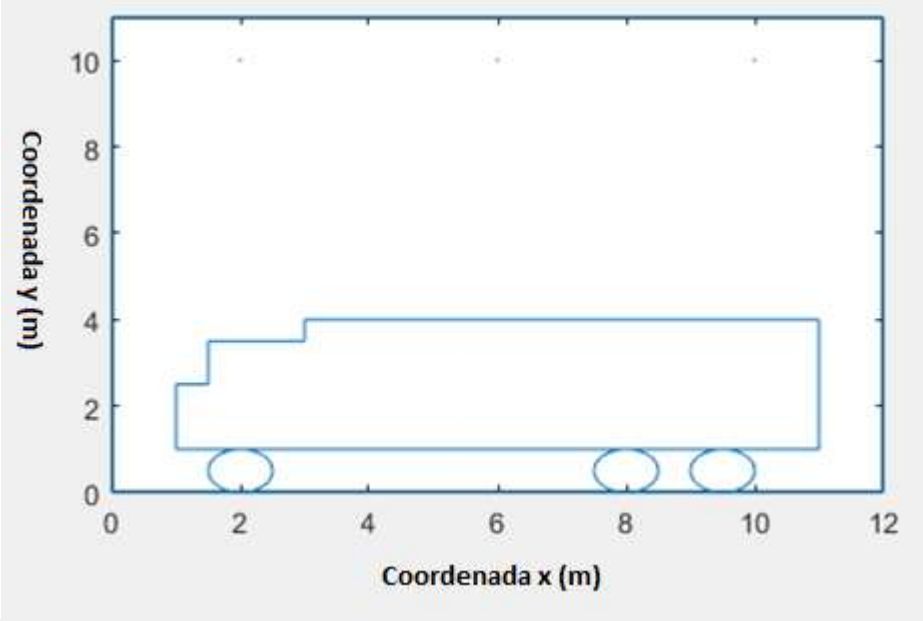
En la lista que lleva por título "Objeto bajo la línea" se selecciona el objeto que se va a ubicar bajo la línea de transmisión. Esta lista contiene cinco opciones descritas a continuación:

1. Una vivienda típica representada en una figura rectangular de dimensiones 10x5 metros (6 metros de profundidad) [10].
2. Una persona de altura promedio de 1,75 metros. (80 cm de ancho, 30 de

profundidad).

3. Un camión típico de 2 ejes con llantas de radio 20'' de dimensiones 10x4 metros (2,2 metros de profundidad).
4. Una edificación representada en una figura rectangular de 6x12,5 metros (10 metros de profundidad).
5. No hay ningún objeto ubicado bajo la línea.

Figura 16. Geometría del modelo para una línea en configuración horizontal con un camión bajo la línea.



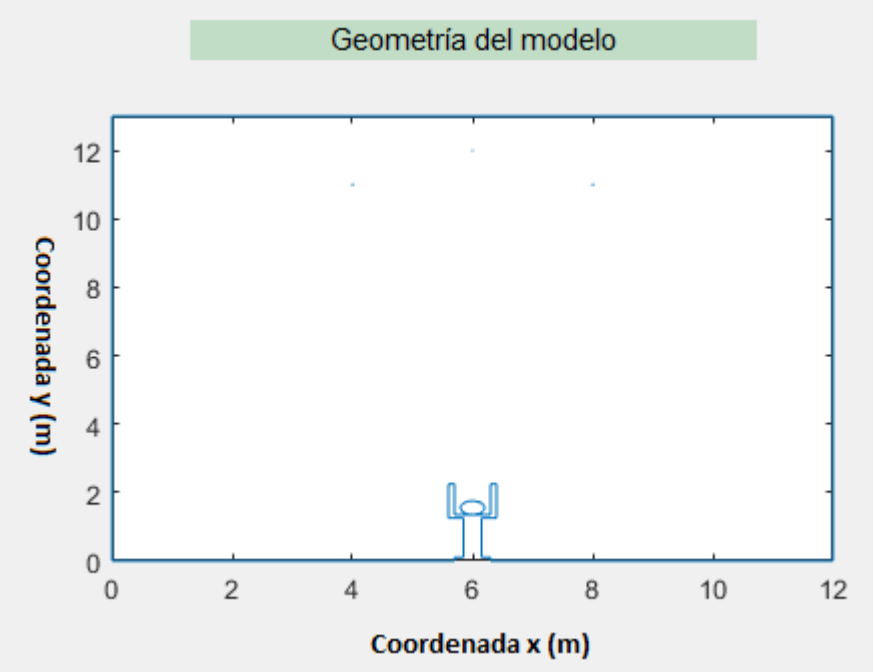
Todos los objetos se encuentran al nivel del suelo y están aterrizados (V=0 en la superficie del objeto). La posición del objeto está determinada por la posición de los conductores, donde el centro del objeto se ubicará en el punto medio entre el conductor más cercano al origen y en conductor más lejano al origen del eje horizontal (eje x).

3.1.2 Solución del modelo

3.1.2.1 Representación gráfica

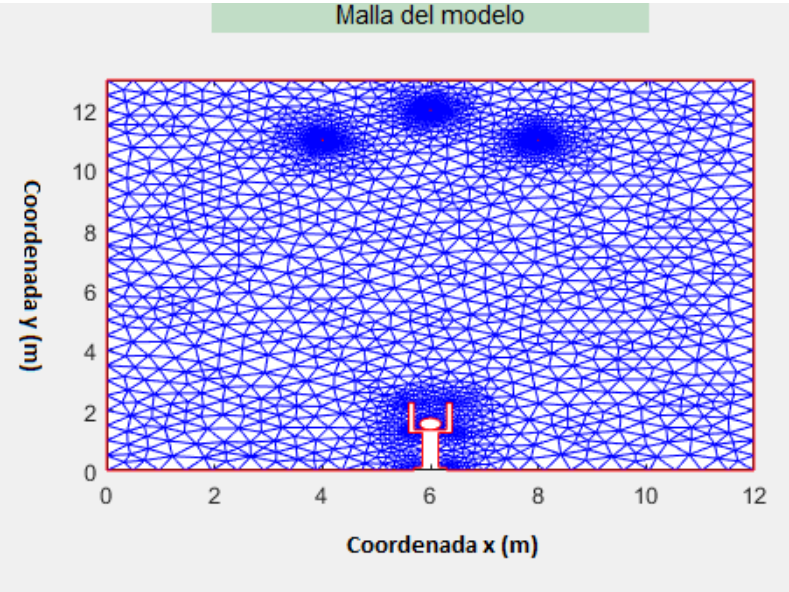
En la sección media del programa, despues de haber ingresado los parámetros a utilizar y haber presionado el boton Modelar, se muestra una representacion gráfica del problema en tres gráficos diferentes. En el primer gráfico que lleva por titulo “Geometría del modelo” se muestra el espacio del problema, con todos los elementos que lo componen (conductores y objeto).

Figura 17. Geometría del modelo para una persona ubicada bajo una línea de 120 kV en configuración triangular (delta).



En la figura central en el grafico que lleva por nombre “Malla del modelo” se muestra la malla desarrollada en el espacio del modelo para encontrar la solución de campo eléctrico mediante el método de elementos finitos. Esta malla está compuesta de elementos triangulares donde la longitud máxima de cualquier lado de cualquier elemento ha sido limitada a un máximo de 0,5 m.

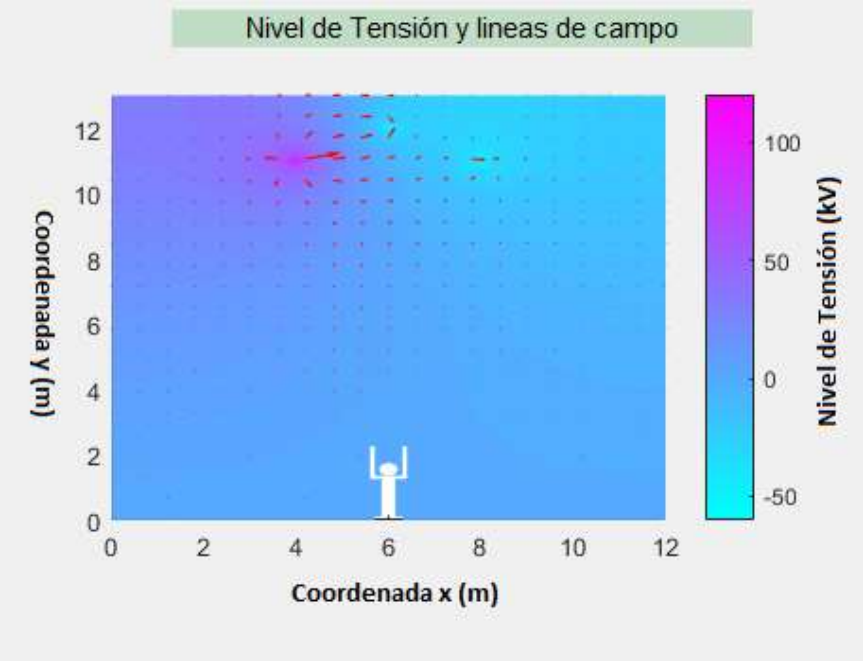
Figura 18. Malla del modelo para una persona ubicada bajo una linea de 120 kV en configuración triangular (delta).



En la última figura llamada “Nivel de Tensión y líneas de campo” se muestra la solución de dos formas para la geometría y malla mostradas en las dos primeras figuras. Mediante un mapa de colores de los diferentes niveles de tensión, donde al lado de la figura se encuentra una barra de colores que indica en valor en kV que representa cada color, y mediante vectores de campo donde la magnitud del vector es proporcional a la magnitud de campo en la posición y dirección del vector. Un ejemplo de este grafico se muestra en la figura 20.

Figura 19. Intensidad de campo eléctrico y corriente de cortocircuito inducida.

Figura 20. Solución al problema para una persona ubicada bajo una línea de 120 kV en configuración triangular (delta).



3.1.2.2 Magnitud de campo eléctrico y corriente de cortocircuito

En el panel llamado “Intensidad de Campo Eléctrico” mostrado en la figura 19, se puede hallar la magnitud de Intensidad de campo eléctrico en cualquier punto del espacio, mostrado en la figura 17. En este panel se ingresan las coordenadas de la posición x,y en metros en las cajas “Coord x” y “Coord y” respectivamente, y se obtiene la magnitud en kV/m en la posición seleccionada.

En el panel llamado “Corriente de corto” se muestra la corriente de cortocircuito inducida para el objeto seleccionado y el campo eléctrico obtenido al modelar el problema.
El área equivalente en metros cuadros, necesaria para el cálculo de esta corriente, se muestra a continuación en la tabla 5 para cada uno de los objetos. Este panel se muestra en la figura 19.

Tabla 5. Área equivalente para los objetos utilizados.

Objeto	Área equivalente (m^2)
Vivienda	345
Persona	5,395
Camión	184
Edificio	750

El área equivalente para los objetos Camión, vivienda y edificio, necesaria para el cálculo de la corriente de cortocircuito, fue determinado usando la aproximación a una caja con los factores de forma mostrados en la figura 9. Para una persona se usó el valor experimental descrito en la tabla 4 obtenidos del estudio *"Radio-Frequency Hazards in the VLF to MF Band"* realizado por O. P. Gandhi y I. Chatterjee [11].

3.1.3 Recomendaciones de uso del Software

- Todos los valores introducidos en la herramienta computacional deben ser introducidos en números enteros o decimales. No deben introducirse en notación científica o cualquier otra simplificación semejante.
- La representación decimal debe hacerse con un punto "." y no con coma ",".
- Es recomendable ingresar las posiciones de las fases de cada circuito de izquierda a derecha, de arriba abajo. Es decir, la fase a del circuito es la fase en la posición más a la izquierda y más arriba.
- Las posiciones de los conductores de fase deben ser mayores a 0, tanto en el eje "y" como en el eje "x". Es importante tener en cuenta las dimensiones del objeto bajo la línea al determinar las posiciones de los conductores.
- Todos los parámetros de distancia deben ser ingresados en metros [m], la tensión en kilo volts [kV] y el tiempo en segundos [s].
- El objeto siempre será ubicado en el medio entre el conductor más a la

izquierda y el conductor más a la derecha en el plano. Si los conductores se encuentren muy lejos del origen del plano, la malla del problema será muy grande, aumentado el tiempo de computación de la herramienta.

4. RESULTADOS DE SIMULACIÓN OBTENIDOS

4.1 VALIDACIÓN DEL MÉTODO

Con el fin de validar los valores obtenidos mediante las simulaciones realizadas en la herramienta computacional desarrollada, se comparan los resultados obtenidos de una línea de transmisión existente de la que se tienen valores medidos bajo la línea con la simulaciones de la herramienta. Estos valores fueron medidos para el caso en el que no se encuentra ningún objeto bajo la línea.

La línea a simular es la línea de 115 kV de Juanchito a Candelaria que pertenece a EPSA y hace parte del Sistema de Transmisión Regional (STR) de Colombia. Los datos fueron obtenidos de la tesis de pregrado titulada "*Medición y simulación de campos eléctricos y magnéticos de extremada baja frecuencia en líneas y subestaciones*" [12] realizada por Rodrigo Pérez Barco para la Universidad del Valle. Los parámetros de la línea se muestran a continuación en la tabla 6.

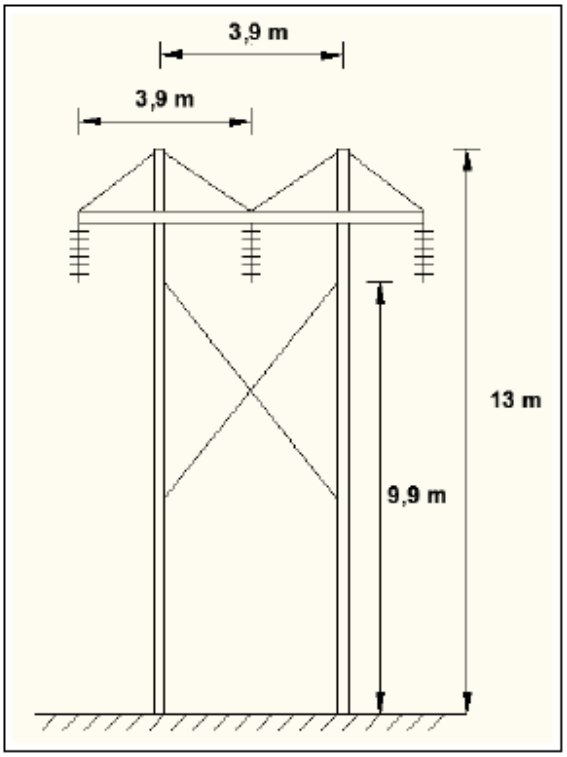
Usando la herramienta CELTIO para calcular el valor rms del Campo eléctrico a una altura de 1m para diferentes posiciones horizontales bajo la línea (la posición cero es donde se encuentra el conductor del medio de la configuración horizontal), se comparan dichos valores con los valores medidos en la tesis referenciada [12] y se calcula el error relativo, como se muestra la tabla 7.

Tabla 6. Parámetros de la línea Juanchito-Candelaria de 115 kV.

Parámetros Línea Juanchito-Candelaria de 115 kV	
Tensión [kV]	119,7
Configuración de la línea	Horizontal
Tipo de línea	Circuito sencillo
Configuración de los conductores	Un solo conductor
Calibre Conductor de fase	336,4 MCM
Diámetro Conductor de fase[mm]	18,821
Calibre Conductor de guarda	33,185 MCM
Diámetro Conductor de guarda [mm]	8,03
Altura línea en punto de medida [m]	6,8

Fuente: PÉREZ, Rodrigo. Medición y simulación de campos eléctricos y magnéticos de extremada baja frecuencia en líneas y subestaciones. 2001.

Figura 21. Configuración de la línea Juanchito-Candelaria de 115 kV.



Fuente: PÉREZ, Rodrigo. Medición y simulación de campos eléctricos y magnéticos de extremada baja frecuencia en líneas y subestaciones. 2001.

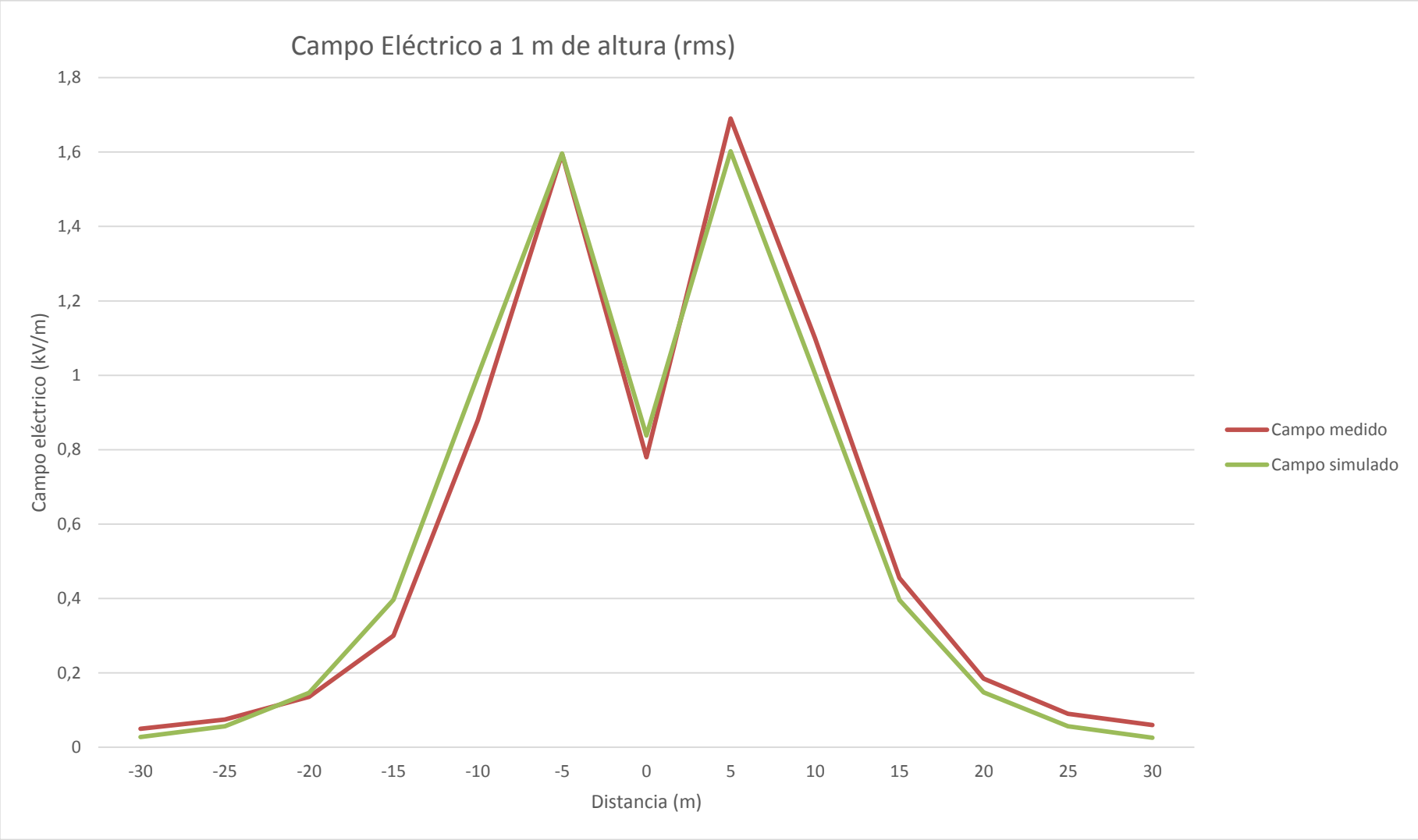
Para el cálculo del valor rms se simula en el programa la solución con desfases de las ondas de tensión de 0º a 360º, en intervalos de 30º. Los cables de guarda no son tenidos en cuenta en la simulación.

Tabla 7. Comparación Campo valor rms.

Campo Eléctrico (rms)			
Distancia	**V. medido**	**V. simulado**	**Error relativo (%)**
-30	0,05	0,0278	-44,3866
-25	0,075	0,0566	-24,5662
-20	0,1356	0,1468	8,2660
-15	0,3	0,3966	32,1850
-10	0,88	0,9999	13,6196
-5	1,595	1,5963	0,0833
0	0,78	0,8384	7,4924
5	1,69	1,6019	-5,2138
10	1,1	1,0037	-8,7540
15	0,455	0,3964	-12,8833
20	0,185	0,1482	-19,8976
25	0,09	0,0565	-37,2570
30	0,06	0,0259	-56,8170

De la tabla 7 y la figura 22 se puede observar que el error absoluto promedio del campo simulado en CELTIO comparado con el valor de campo medido es de 0,052, con un error relativo máximo porcentual del 56,8% para este modelo. Dado que los objetos se encuentran en los primeros 10 metros al centro de la línea el campo, para la aplicación de esta herramienta en la determinación del campo eléctrico inducido en objetos que se encuentran bajo la línea de transmisión es importante que los campos simulados a estas distancias sean representativos. Para distancias menores a 10 m el error relativo máximo para el modelo de la línea simulado es de alrededor del 13,6%.

Figura 22. Perfil de campo valor rms.



Fuente: Autor. 2017

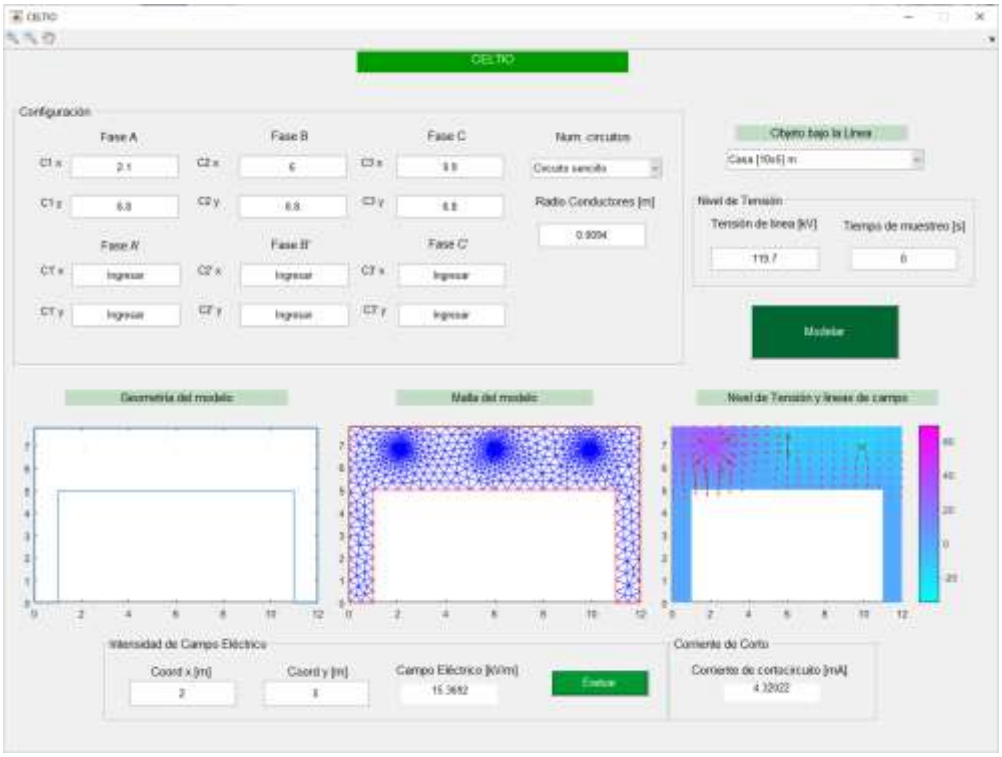
4.2 CASOS DE SIMULACIÓN

Con el propósito de observar si los niveles de campo eléctrico inducido y corriente inducido se superan bajo distintas situaciones, se simula el campo eléctrico para tres líneas de diferentes niveles de tensión con tres de los objetos escogidos ubicados bajo la línea.

4.2.1 Caso 1. Línea de 115 kV con vivienda típica bajo la línea

La primera línea a simular es la línea Juanchito-Candelaria de 115 kV, con una vivienda ubicada bajo la línea, cuyos parámetros están descritos en la tabla 5 y su configuración geométrica se muestra en la figura 21. Los resultados obtenidos en el programa CELTIO se muestran, para el instante t=0 y la flecha máxima, se muestran en la figura 23.

Figura 23. Línea Juanchito-Candelaria de 115 kV en el software de simulación.



Se obtiene el campo eléctrico en 10 posiciones sobre el objeto que se muestran en la tabla 8 para determinar el campo eléctrico rms sobre el objeto en diferentes partes del objeto y la corriente de cortocircuito rms inducida sobre el objeto. Los resultados obtenidos se muestran también en la tabla 8.

Tabla 8. Campo eléctrico y corriente inducidos sobre el objeto.

Campo y corriente sobre el objeto	
Posición (x y) m	E (kV/m) rms
(1 0)	0,010
(1 2)	0,0514
(1 4)	1,349
(2 5)	10,904
(4 5)	4,630
(8 5)	4,649
(10 5)	10,938
(11 4)	1,214
(11 2)	0,055
(11 0)	0,002
Isc (mA) rms	3,643

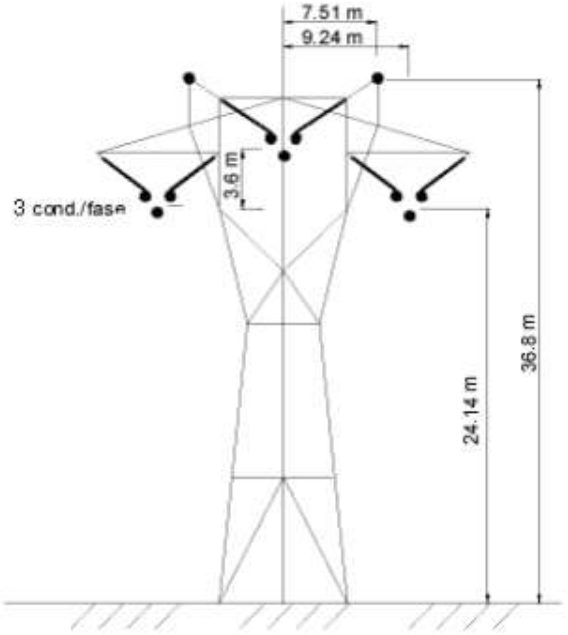
Los resultados mostrados en la tabla 8 muestran para el caso simulado, cuando la flecha es mayor (la altura de la línea es de 6,8 m), el límite de campo permitido ocupacional por la norma (8,3 kV/m) es superado en la superficie del objeto a una altura de 5 metros en dos posiciones y la corriente de cortocircuito inducido se mantiene bajo el límite de 5 mA rms.

4.2.2 Caso 2. Línea de 500 kV en delta con persona bajo la línea

Se simula el campo eléctrico inducido para una línea de configuración delta de 500 kV con una persona ubicada bajo la línea cuyos parámetros fueron obtenidos del artículo

"*To estimate the electric field near electric energy transmission systems by use simple software*" de la Universidad de Sao Paulo [13] y se muestran en la figura 24 y la tabla 9.

Figura 24. Configuración en delta línea de 500 kV.



Fuente: "To estimate the electric field near electric energy transmission systems by use simple software" Universidad de Sao Paulo. 2010.

Tabla 9. Parámetros de línea de 500 kV.

Parámetros Línea de 500 kV	
Tensión de línea	500 kV
Configuración de la línea	Delta
Tipo de línea	Circuito sencillo
Número de conductores por fase	Tres conductores
Distancia entre conductor del haz	0,475 m
Radio Conductor de fase	11,77 mm
Número Conductor de guarda	Dos conductores
Radio Conductor de guarda	8,03 mm
Altura línea en flecha máxima	14,64 m
Radio equivalente	0,1385 m

Fuente: "To estimate the electric field near electric energy transmission systems by use simple software" Universidad de Sao Paulo. 2010.

El radio equivalente se calculó mediante las siguientes ecuaciones,

$r_{eq} = R\ (\frac{nr}{R})^{\frac{1}{n}}$ (4.1)

$R = \frac{B}{2*sin(\frac{\pi}{n})}$ (4.2)

Donde,

r_{eq} es el radio equivalente de la fase.

r es el radio de cada conductor del haz.

n es el número de conductores en el haz.

R es el radio geométrico del haz.

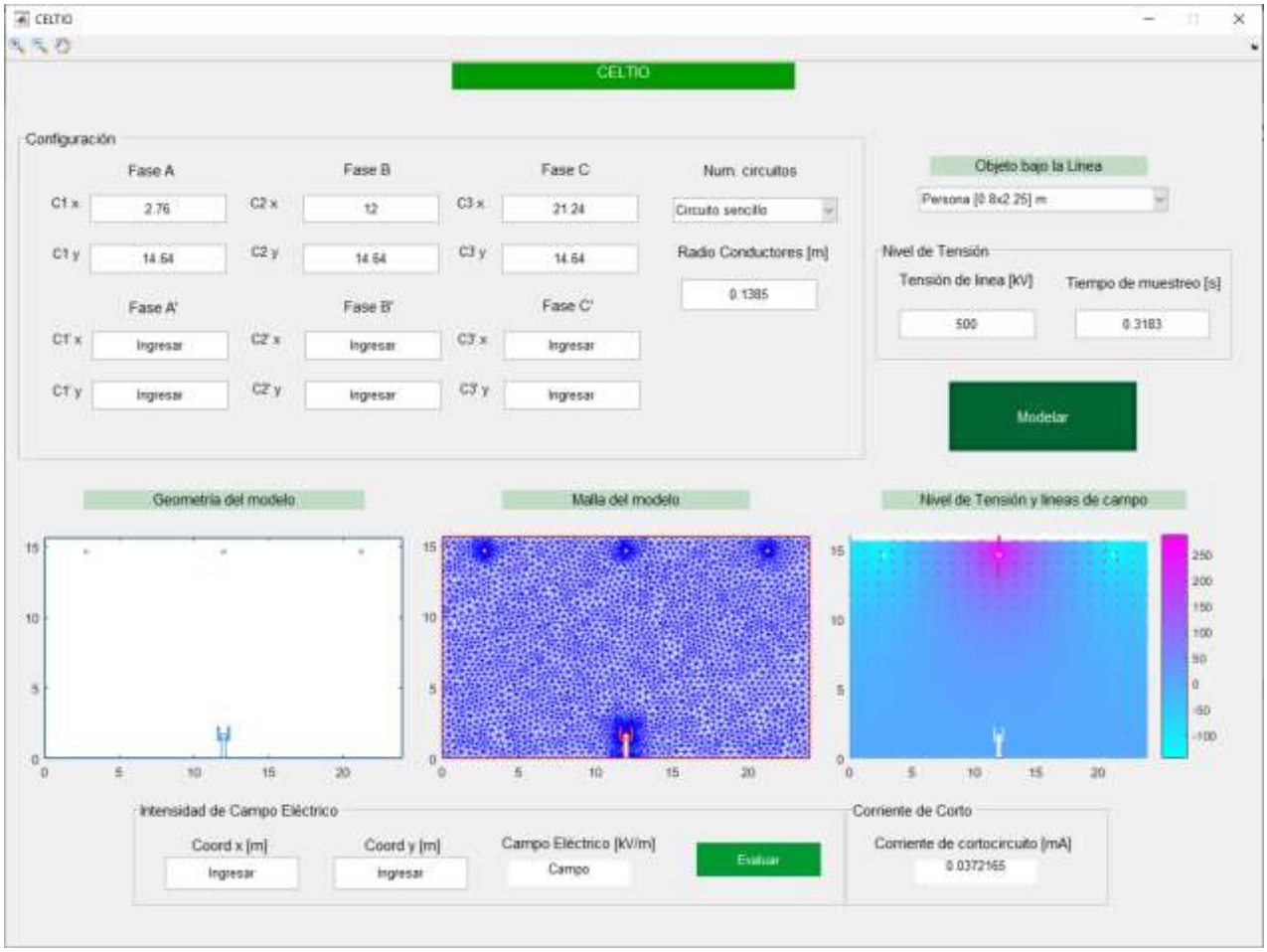
B es la distancia entre conductores del haz.

Introduciendo los parámetros de la línea en CELTIO para la posición de la línea donde la flecha es máxima, se obtuvo el campo eléctrico en diez posiciones sobre el objeto y la corriente inducida, estos valores se muestran en la tabla 10. Los resultados obtenidos en CELTIO se muestran en la figura 25.

Tabla 10. Campo eléctrico y corriente inducidos sobre el objeto.

Campo y corriente sobre el objeto	
posición (x y) m	E (kV/m) rms
(11,7 0,1)	0,1613
(11,85 1)	0,4397
(11,6 1,3)	1,4621
(11,6 2)	3,3430
(11,7 2)	0,8657
(12,3 2)	0,8837
(12,4 2)	3,2726
(11 4)	1,4389
(11 2)	0,4228
(11 0)	0,5535
Isc(mA) rms	0,0105

Figura 25. Línea de 500 kV en la herramienta de simulación.



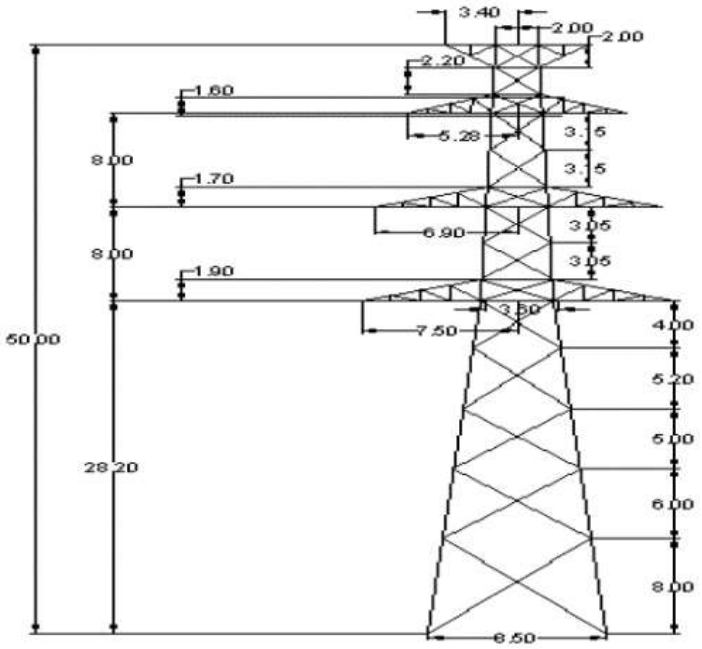
De la tabla 10 se observa que para la línea de 500 kV simulada con los parámetros descritos anteriormente con una persona bajo la línea en el punto de flecha máxima, el límite de campo permitido de exposición al público no se supera, ya que el valor rms es siempre menor 4,16 [kV/m] y el campo máximo obtenido en esas posiciones sobre el objeto es de 3,34 [kV/m].

4.2.3 Caso 3. Línea de 400 kV en configuración vertical doble circuito con camión bajo la línea

Se simula el campo eléctrico inducido para una línea de doble circuito en configuración

vertical de 400 kV con un camión ubicado bajo la línea. Los parámetros de la línea fueron obtenidos del artículo "*Optimization of transmission line towers: P-delta analysis"* de los autores R Vinay, A Ranjith, y A Bharath del Instituto Asiático de Tecnología [14] (AIT por sus siglas en inglés) y se muestran en la figura 26 y la tabla 11.

Figura 26. Configuración en delta línea de 500 kV.



Fuente: Optimization of transmission line towers: P-delta analysis. R Vinay, A Ranjith, A Bharath. AIT, 2014.

Tabla 11. Parámetros de línea de 500 kV.

Parámetros Línea de 500 Kv	
Tensión de línea	500 Kv
Configuración de la línea	Vertical
Tipo de línea	Doble circuito
Número de conductores por fase	Un conductor
Radio Conductor de fase	15.89 mm
Numero Conductor de guarda	Dos conductores
Radio Conductor de guarda	5,49 mm

Fuente: Optimization of transmission line towers: P-delta analysis. R Vinay, A Ranjith, A Bharath. AIT, 2014.

Introduciendo los parámetros generales de esta línea en la herramienta CELTIO, se obtuvo el campo eléctrico en diez posiciones sobre el objeto y la corriente inducida, estos valores se muestran en la tabla 12. Los resultados obtenidos en CELTIO se muestran en la figura 27.

Figura 27. Línea de 400 kV en el software de simulación.

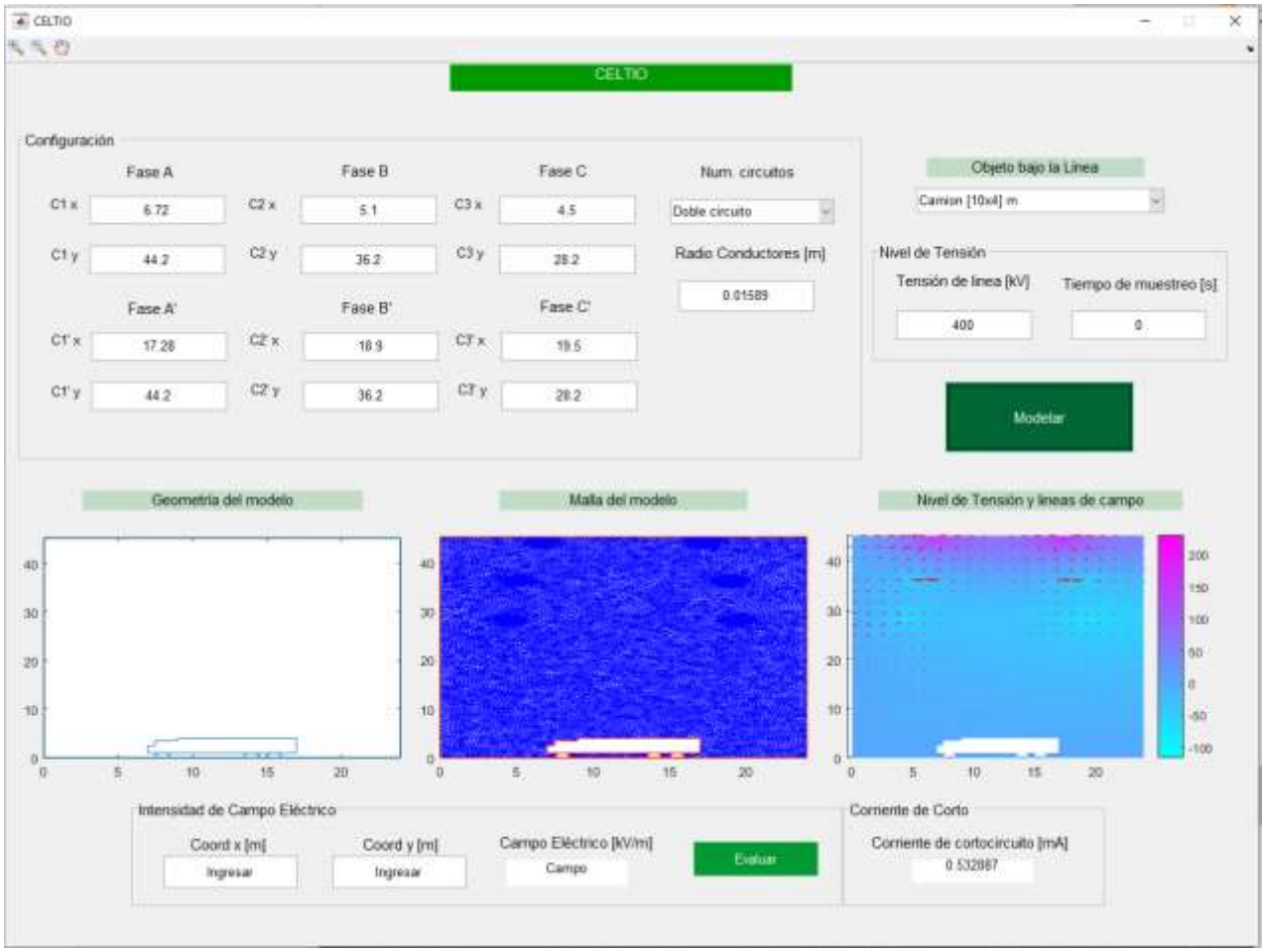


Tabla 12. Campo eléctrico y corriente inducidos sobre el objeto.

Campo y corriente sobre el objeto	
posicion (x,y) m	E (kV/m) rms
(7 1,5)	0,2305
(7,5 3)	0,5316
(9 4)	2,4378
(11 4)	0,9995
(13 4)	0,2554
(15 4)	0,7166
(17 4)	3,1289
(17 3)	1,4644
(17 2)	0,7440
(17 1)	0,2865
Isc(mA) rms	0,5487

En este último caso, según los valores obtenidos en las posiciones escogidas, se observa que el campo eléctrico inducido en la superficie del objeto no supera el límite de exposición de 4,16 [kV/m]. La corriente eléctrica inducida en el camión no supera tampoco el valor recomendado de 5 mA.

CONCLUSIONES

- Los efectos que los campos electromagnéticos producidos por líneas de transmisión ocasionan sobre la salud humana son aún un campo de estudio en desarrollo, donde la gravedad que conlleva superar los límites de campo, introducidos tanto en la normativa colombiana como en otros reglamentos internacionales, es aún un tema de discusión.

- El método de los elementos finitos permite calcular en un número relativamente corto de iteraciones el campo eléctrico producido bajo una línea de transmisión eléctrica cuando un objeto se encuentra bajo la línea. Este método numérico permite representar con facilidad objetos con formas no convencionales que representan un alto grado de dificultad en el análisis mediante otros métodos numéricos o analíticos.

- El software Matlab mediante la herramienta PDEtool facilita la implementación de este método en el análisis del campo inducido bajo una línea de transmisión cuando un objeto se encuentra en su zona de servidumbre, gracias a las funciones que permiten crear cómodamente la geometría y la malla aplicada al método.

- Para los valores obtenidos en los casos simulados donde las distancias recomendadas por el RETIE [3] fueron respetadas, los valores de campo inducido se encuentran bajo los límites recomendados y no representarían un problema inmediato a la salud de las personas expuestas. Existen situaciones anormales como el primer caso simulado en el capítulo 4 donde se observa que estos límites son superados ampliamente, y se debe garantizar que estas situaciones sean corregidas por el beneficio de la población expuesta.

- Es importante reconocer las ecuaciones y consideraciones tenidas en cuenta en la creación de la herramienta de computación desarrollada en este trabajo de

grado para una correcta aplicación de la misma. En la realidad los efectos que ocasiona un objeto ubicado bajo una línea de transmisión al campo eléctrico inducido en el mismo, representan un problema tridimensional con una serie de variables que han sido expresamente omitidas en este programa.

- Es importante realizar mediciones en campo que permitan contrastar los niveles de campo eléctrico bajo una línea de transmisión con objetos ubicados bajo la línea obtenidos en la herramienta de simulación con los valores obtenidos en la realidad, y de esta manera estimar apropiadamente el nivel de exactitud de la herramienta. Actualmente son pocas las referencias bibliográficas en las que estas comparaciones hayan sido medidas y verificadas.

BIBLIOGRAFIA

[1] G. S. Subcommittee, Electrostatic Effects OF Overhead Transmission Lines PART I-Hazards and Effects, En: IEEE Transactions on Power Apparatus and Systems, vol. PAS-91, 2 ed. 1972. p. 422-426.

[2] VALBERG, P. A. Electromagnetic Waves (EMF and RF) and Health Effects, En: Hamilton & Hardy's Industrial Toxicology. Hoboken, New Jersey. : John Wiley & Sons, Inc., 2015. p. 1069-1085.

[3] COLOMBIA. MINISTERIO DE MINAS Y ENERGÍA. Resolución 9-0708 (30, Agosto, 2013). Anexo general del RETIE con sus ajustes - RETIE. Bogotá D.C.: El Ministerio, 2013. 211 p.

[4] EPRI. EPRI AC Transmission Line Reference Book—200 kV and Above, 3 ed. Palo Alto, CA, EPRI, 2005.

[5] M.L.SINGEWALD, 0.R.LANGWORTH, H.B.KOUWENHOVEN, ll. Medical Follow-up Study of High Voltage Linemen Working in AC Electric Fields. 1967.

[6] NATIONAL ELECTRIC SAFETY CODE. Part 2. Safety rules for overhead lines. NTC-ISO 9001. Bogotá D.C.: ANSI, C2-2002. 324 p.

[7] RICHARD C. Booton. Computational Methods for Electromagnetics and Microwaves. Wiley, 1992. 192 p.

[8] SADIKU, Matthew N. O. Elements of Electromagnetics. 2 ed. New York: Oxford University Press, 1994.

[9] DENO, D.W. Electrostatic induction formulae, En: IEEE Transactions on Power Apparatus and Systems, PAS-94: 1975. p. 1524-1536.

[10] PLAN DE ORDENAMIENTO TERRITORIAL. Ficha normativa urbana: Normas generales para la edificación. Santiago de Cali.2000. Ac.026. p. 1.

[11] GANDHI. O. P., CHATTERJEE I., Radio-frequency hazards in the VLF to MF band," En: Proceedings of the IEEE, no. 12, vol. 70, 1982. p. 1462-1464.

[12] PÉREZ, Rodrigo. Medición y simulación de campos eléctricos y magnéticos de extremada baja frecuencia en líneas y subestaciones. Trabajo de grado Ingeniero Electricista. Santiago de Cali: Universidad del Valle. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica y Electrónica, 2001.

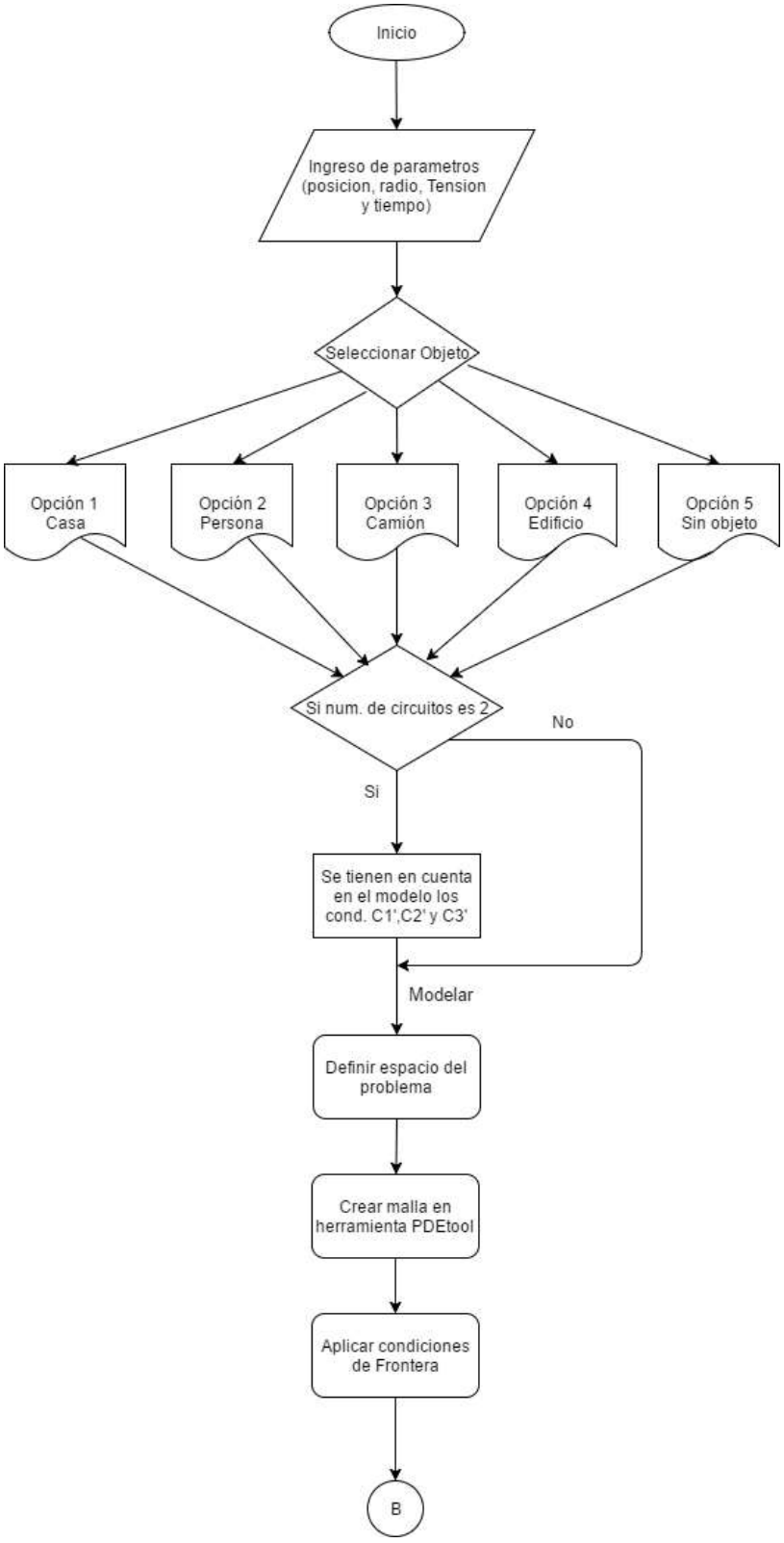
[13] RODRIGUES, J. F, *et al*. "To estimate the electric field near electric energy transmission systems by use simple software," En: 2010 9th IEEE/IAS International Conference on Industry Applications INDUSCON 2010. Sao Paulo, 2010, p. 1-6.

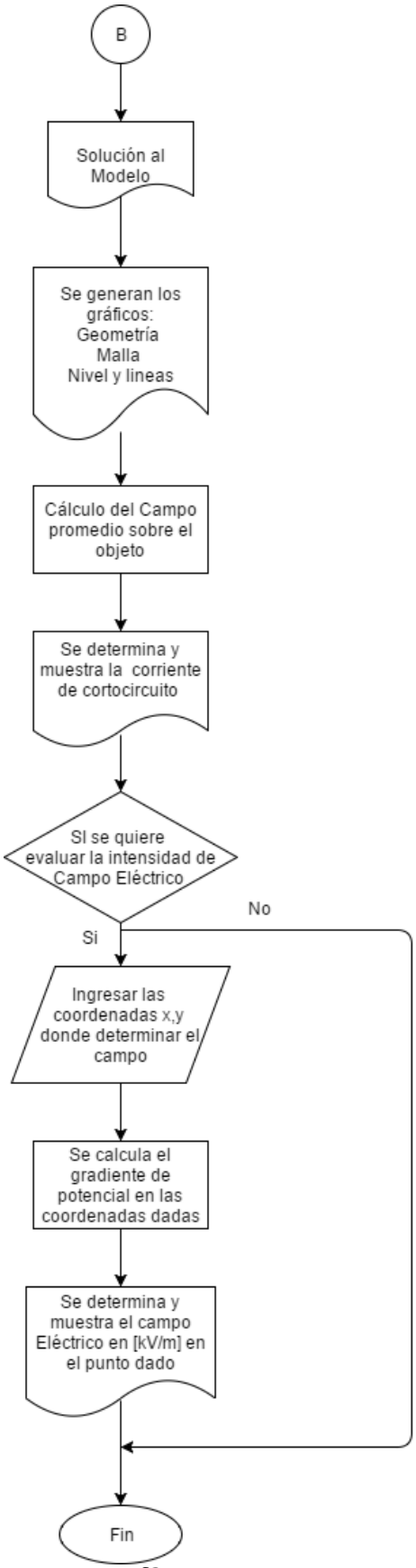
[14] J. VINAY R, RANJITH A, BHARATH A. Optimization of transmission line towers: P-delta analysis. Eng. Technol. 2014. p.14563-14569.

ANEXOS

Anexo A

Diagrama de flujo del programa




B
Solución al Modelo
Se generan los gráficos: Geometría Malla Nivel y lineas
Cálculo del Campo promedio sobre el objeto
Se determina y muestra la corriente de cortocircuito
SI se quiere evaluar la intensidad de Campo Eléctrico
No
Si
Ingresar las coordenadas x,y donde determinar el campo
Se calcula el gradiente de potencial en las coordenadas dadas
Se determina y muestra el campo Eléctrico en [kV/m] en el punto dado
Fin

Anexo B

Código de la herramienta de simulación CELTIO.m

```
function varargout = CELTIO(varargin)
% CELTIO MATLAB code for CELTIO.fig
%      CELTIO, by itself, creates a new CELTIO or raises the existing
%      singleton*.
%
%      H = CELTIO returns the handle to a new CELTIO or the handle to
%      the existing singleton*.
%
%      CELTIO('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in CELTIO.M with the given input arguments.
%
%      CELTIO('Property','Value',...) creates a new CELTIO or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before CELTIO_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to CELTIO_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help CELTIO

% Last Modified by GUIDE v2.5 03-Apr-2017 00:19:06

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @CELTIO_OpeningFcn, ...
                   'gui_OutputFcn',  @CELTIO_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before CELTIO is made visible.
function CELTIO_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to CELTIO (see VARARGIN)

% Choose default command line output for CELTIO
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes CELTIO wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = CELTIO_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a
double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a
double


% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a
double


% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a
double
```

```
% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5 as a
double


% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6 as a
double


% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%        str2double(get(hObject,'String')) returns contents of edit7 as a
double


% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
rstr=get(handles.edit7,'String'); %radio como texto
c1xstr=get(handles.edit1,'String'); %posicion x del conductor 1 como texto
c1ystr=get(handles.edit2,'String'); %posicion y del conductor 1 como texto
c2xstr=get(handles.edit3,'String'); %posicion x del conductor 2 como texto
c2ystr=get(handles.edit4,'String'); %posicion y del conductor 2 como texto
c3xstr=get(handles.edit5,'String'); %posicion x del conductor 3 como texto
c3ystr=get(handles.edit6,'String'); %posicion y del conductor 3 como texto
c12xstr=get(handles.edit16,'String'); %posicion x del conductor 1 como texto
c12ystr=get(handles.edit17,'String'); %posicion y del conductor 1 como texto
c22xstr=get(handles.edit18,'String'); %posicion x del conductor 2 como texto
c22ystr=get(handles.edit19,'String'); %posicion y del conductor 2 como texto
c32xstr=get(handles.edit20,'String'); %posicion x del conductor 3 como texto
c32ystr=get(handles.edit21,'String'); %posicion y del conductor 3 como texto
```

```
tc1str=get(handles.edit12,'String'); %tension de linea conductor como texto
tc2str=get(handles.edit13,'String'); %tiempo de muestreo como texto
r=str2double(rstr);
c1x=str2double(c1xstr);
c1y=str2double(c1ystr);
c2x=str2double(c2xstr);
c2y=str2double(c2ystr);
c3x=str2double(c3xstr);
c3y=str2double(c3ystr);
c12x=str2double(c12xstr);
c12y=str2double(c12ystr);
c22x=str2double(c22xstr);
c22y=str2double(c22ystr);
c32x=str2double(c32xstr);
c32y=str2double(c32ystr);
tc1=str2double(tc1str);
tc2=str2double(tc2str);

vl1=(tc1/sqrt(3))*cosd(2*pi*60*tc2); %tension conductor 1 al tiempo t
vl2=(tc1/sqrt(3))*cosd(2*pi*60*tc2-120);
vl3=(tc1/sqrt(3))*cosd(2*pi*60*tc2+120);

tipo=get(handles.popupmenu1,'Value');
ncirc=get(handles.popupmenu3,'Value');
% Conductores
C1 = [1 % circulo
    c1x % coordinada x del centro
    c1y % coordinada y del centro
    r]; % radio
C2 = [1
    c2x
    c2y
    r];
C3 = [1
    c3x
    c3y
    r];
C12 = [1 % circulo
    c12x % coordinada x del centro
    c12y % coordinada y del centro
    r]; % radio
C22 = [1
    c22x
    c22y
    r];
C32 = [1
    c32x
    c32y
    r];

switch tipo
    case 1 %Tipo Casa
 switch ncirc
 case 1
 mediax=((max([c1x,c2x,c3x])+min([c1x,c2x,c3x]))/2);
% Forma casa
```

```
casa = [3 % rectangulo
    4 % numero de lados
    mediax-5 % vertices x
    mediax+5
    mediax+5
    mediax-5
    0 % vertices y
    0
    5
    5];
%espacio

esp=[3 % rectangulo
    4 % numero de lados
    0 % vertices x
    2*mediax
    2*mediax
    0
    0 % vertices y
    0
    max([C1(3),C2(3),C3(3),casa(9)])+1
    max([C1(3),C2(3),C3(3),casa(9)])+1];
% Anexar extra ceros a los circulos para que tengan el mismo numero de
elementos de los rectangulos
C1 = [C1;zeros(length(casa) - length(C1),1)];
C2 = [C2;zeros(length(casa) - length(C2),1)];
C3 = [C3;zeros(length(casa) - length(C3),1)];
% Combinar las formas en una sola matrix
gd = [casa,esp,C1,C2,C3];
% Asignar nombres a las formas
ns = char('casa','esp','C1','C2','C3');
ns = ns';
%Crea la formula que define la zona a analizar
sf = '(esp)-(casa+C1+C2+C3)';
[dl,bt] = decsg(gd,sf,ns); % combina las formas usando la formula
[dl2,bt2] = csgdel(dl,bt); % removes face boundaries (boundary removal)

%crear modelo PDE e incluir geometria en el modelo
modelo = createpde;
geometryFromEdges(modelo,dl2);%with  boundary removal

%View the geometry
%%{
axes(handles.axes2);
pdegplot(modelo)
axis tight
%}

%aplicar condiciones de frontera
applyBoundaryCondition(modelo,'Edge',1,'u',0);
applyBoundaryCondition(modelo,'Edge',5:8,'u',0);
applyBoundaryCondition(modelo,'Edge',9:12,'r',vl1,'h',1);
applyBoundaryCondition(modelo,'Edge',13:16,'r',vl2,'h',1);
applyBoundaryCondition(modelo,'Edge',17:20,'r',vl3,'h',1);

case 2
```

```
mediax=((max([c1x,c2x,c3x,c12x,c22x,c32x])+min([c1x,c2x,c3x,c12x,c22x,c32x]))/2
);
    % Forma casa
casa = [3 % rectangulo
    4 % numero de lados
    mediax-5 % vertices x
    mediax+5
    mediax+5
    mediax-5
    0 % vertices y
    0
    5
    5];
%espacio
esp=[3 % rectangulo
    4 % numero de lados
    0 % vertices x
    2*mediax
    2*mediax
    0
    0 % vertices y
    0
    max([C1(3),C2(3),C3(3),C12(3),C22(3),C32(3),casa(9)])+1
    max([C1(3),C2(3),C3(3),C12(3),C22(3),C32(3),casa(9)])+1];
% Anexar extra ceros a los circulos para que tengan el mismo numero de
elementos de los rectangulos
C1 = [C1;zeros(length(casa) - length(C1),1)];
C2 = [C2;zeros(length(casa) - length(C2),1)];
C3 = [C3;zeros(length(casa) - length(C3),1)];
C12 = [C12;zeros(length(casa) - length(C12),1)];
C22 = [C22;zeros(length(casa) - length(C22),1)];
C32 = [C32;zeros(length(casa) - length(C32),1)];
% Combinar las formas en una sola matrix
gd = [casa,esp,C1,C2,C3,C12,C22,C32];
% Asignar nombres a las formas
ns = char('casa','esp','C1','C2','C3','C12','C22','C32');
ns = ns';
%Crea la formula que define la zona a analizar
sf = '(esp)-(casa+C1+C2+C3+C12+C22+C32)';
[dl,bt] = decsg(gd,sf,ns); % combina las formas usando la formula
[dl2,bt2] = csgdel(dl,bt); % removes face boundaries (boundary removal)

%crear modelo PDE e incluir geometria en el modelo
modelo = createpde;
geometryFromEdges(modelo,dl2);%with  boundary removal

%View the geometry
%%{
axes(handles.axes2);
pdegplot(modelo)
axis tight
%}

%aplicar condiciones de frontera
applyBoundaryCondition(modelo,'Edge',1,'u',0);
```

```
applyBoundaryCondition(modelo,'Edge',5:8,'u',0);
applyBoundaryCondition(modelo,'Edge',9:12,'r',vl1,'h',1);
applyBoundaryCondition(modelo,'Edge',13:16,'r',vl2,'h',1);
applyBoundaryCondition(modelo,'Edge',17:20,'r',vl3,'h',1);
applyBoundaryCondition(modelo,'Edge',21:24,'r',vl1,'h',1);
applyBoundaryCondition(modelo,'Edge',25:28,'r',vl2,'h',1);
applyBoundaryCondition(modelo,'Edge',29:32,'r',vl3,'h',1);


end

    case 2 %Tipo Persona
      switch ncirc
 case 1
mediax=((max([c1x,c2x,c3x])+min([c1x,c2x,c3x]))/2);
% Forma persona
pies = [3 % rectangulo
    4 % numero de lados
    mediax-0.3 % vertices x
    mediax+0.3
    mediax+0.3
    mediax-0.3
    0 % vertices y
    0
    0.1
    0.1];
cuerpo = [3 % rectangulo
    4 % numero de lados
    mediax-0.15 % vertices x
    mediax+0.15
    mediax+0.15
    mediax-0.15
    0 % vertices y
    0
    1.35
    1.35];
antebrazos = [3 % rectangulo
    4 % numero de lados
    mediax-0.4 % vertices x
    mediax+0.4
    mediax+0.4
    mediax-0.4
    1.25 % vertices y
    1.25
    1.35
    1.35];
brazos1 = [3 % rectangulo
    4 % numero de lados
    mediax-0.4 % vertices x
    mediax-0.3
    mediax-0.3
    mediax-0.4
    1.25 % vertices y
    1.25
    2.25
    2.25];
```

```
brazos2 = [3 % rectangulo
    4 % numero de lados
    mediax+0.3 % vertices x
    mediax+0.4
    mediax+0.4
    mediax+0.3
    1.25 % vertices y
    1.25
    2.25
    2.25];
cabeza = [1 % circulo
    mediax % coordinada x del centro
    1.55 % coordinada y del centro
    0.2]; % radio
%espacio
esp=[3 % rectangulo
    4 % numero de lados
    0 % vertices x
    mediax*2
    mediax*2
    0
    0 % vertices y
    0
    max([C1(3),C2(3),C3(3),brazos1(9)])+1
    max([C1(3),C2(3),C3(3),brazos1(9)])+1];
% Anexar extra ceros a los circulos para que tengan el mismo numero de
elementos de los rectangulos
C1 = [C1;zeros(length(pies) - length(C1),1)];
C2 = [C2;zeros(length(pies) - length(C2),1)];
C3 = [C3;zeros(length(pies) - length(C3),1)];
cabeza = [cabeza;zeros(length(pies) - length(cabeza),1)];
% Combinar las formas en una sola matrix
gd = [pies,cuerpo,antebrazos,brazos1,brazos2,cabeza,esp,C1,C2,C3];
% Asignar nombres a las formas
ns =
char('pies','cuerpo','antebrazos','brazos1','brazos2','cabeza','esp','C1','C2',
'C3');
ns = ns';
%Crea la formula que define la zona a analizar
sf = '(esp)-(pies+cuerpo+antebrazos+brazos1+brazos2+cabeza+C1+C2+C3)';
[dl,bt] = decsg(gd,sf,ns); % combina las formas usando la formula
[dl2,bt2] = csgdel(dl,bt); % removes face boundaries (boundary removal)

%crear modelo PDE e incluir geometria en el modelo
modelo = createpde;
geometryFromEdges(modelo,dl2);%with  boundary removal
%geometryFromEdges(modelo,dl);%without  boundary removal

%View the geometry
%%{
axes(handles.axes2);
pdegplot(modelo)
axis tight
%}
%aplicar condiciones de frontera
```

```
applyBoundaryCondition(modelo,'Edge',3:24,'u',0);
applyBoundaryCondition(modelo,'Edge',26:27,'u',0);
applyBoundaryCondition(modelo,'Edge',32:35,'r',vl1,'h',1);
applyBoundaryCondition(modelo,'Edge',36:39,'r',vl2,'h',1);
applyBoundaryCondition(modelo,'Edge',40:43,'r',vl3,'h',1);
     case 2
mediax=((max([c1x,c2x,c3x,c12x,c22x,c32x])+min([c1x,c2x,c3x,c12x,c22x,c32x]))/2
);
  % Forma persona
pies = [3 % rectangulo
    4 % numero de lados
    mediax-0.3 % vertices x
    mediax+0.3
    mediax+0.3
    mediax-0.3
    0 % vertices y
    0
    0.1
    0.1];
cuerpo = [3 % rectangulo
    4 % numero de lados
    mediax-0.15 % vertices x
    mediax+0.15
    mediax+0.15
    mediax-0.15
    0 % vertices y
    0
    1.35
    1.35];
antebrazos = [3 % rectangulo
    4 % numero de lados
    mediax-0.4 % vertices x
    mediax+0.4
    mediax+0.4
    mediax-0.4
    1.25 % vertices y
    1.25
    1.35
    1.35];
brazos1 = [3 % rectangulo
    4 % numero de lados
    mediax-0.4 % vertices x
    mediax-0.3
    mediax-0.3
    mediax-0.4
    1.25 % vertices y
    1.25
    2.25
    2.25];
brazos2 = [3 % rectangulo
    4 % numero de lados
    mediax+0.3 % vertices x
    mediax+0.4
    mediax+0.4
    mediax+0.3
    1.25 % vertices y
    1.25
```

```
    2.25
    2.25];
cabeza = [1 % circulo
    mediax % coordinada x del centro
    1.55 % coordinada y del centro
    0.2]; % radio
%espacio
esp=[3 % rectangulo
    4 % numero de lados
    0 % vertices x
    mediax*2
    mediax*2
    0
    0 % vertices y
    0
    max([C1(3),C2(3),C3(3),C12(3),C22(3),C32(3),brazos1(9)])+1
    max([C1(3),C2(3),C3(3),C12(3),C22(3),C32(3),brazos1(9)])+1];
% Anexar extra ceros a los circulos para que tengan el mismo numero de
elementos de los rectangulos
C1 = [C1;zeros(length(pies) - length(C1),1)];
C2 = [C2;zeros(length(pies) - length(C2),1)];
C3 = [C3;zeros(length(pies) - length(C3),1)];
C12 = [C12;zeros(length(pies) - length(C12),1)];
C22 = [C22;zeros(length(pies) - length(C22),1)];
C32 = [C32;zeros(length(pies) - length(C32),1)];
cabeza = [cabeza;zeros(length(pies) - length(cabeza),1)];
% Combinar las formas en una sola matrix
gd = [pies,cuerpo,antebrazos,brazos1,brazos2,cabeza,esp,C1,C2,C3,C12,C22,C32];
% Asignar nombres a las formas
ns =
char('pies','cuerpo','antebrazos','brazos1','brazos2','cabeza','esp','C1','C2',
'C3','C12','C22','C32');
ns = ns';
%Crea la formula que define la zona a analizar
sf = '(esp)-
(pies+cuerpo+antebrazos+brazos1+brazos2+cabeza+C1+C2+C3+C12+C22+C32)';
[dl,bt] = decsg(gd,sf,ns); % combina las formas usando la formula
[dl2,bt2] = csgdel(dl,bt); % removes face boundaries (boundary removal)

%crear modelo PDE e incluir geometria en el modelo
modelo = createpde;
geometryFromEdges(modelo,dl2);%with  boundary removal
%geometryFromEdges(modelo,dl);%without  boundary removal

%View the geometry
%%{
axes(handles.axes2);
pdegplot(modelo)
axis tight
%}
%aplicar condiciones de frontera

applyBoundaryCondition(modelo,'Edge',3:24,'u',0);
applyBoundaryCondition(modelo,'Edge',26:27,'u',0);
applyBoundaryCondition(modelo,'Edge',32:35,'r',vl1,'h',1);
applyBoundaryCondition(modelo,'Edge',36:39,'r',vl2,'h',1);
```

```
applyBoundaryCondition(modelo,'Edge',40:43,'r',vl3,'h',1);
applyBoundaryCondition(modelo,'Edge',44:47,'r',vl1,'h',1);
applyBoundaryCondition(modelo,'Edge',48:51,'r',vl2,'h',1);
applyBoundaryCondition(modelo,'Edge',52:55,'r',vl3,'h',1);

     end

    case 3 %Tipo Camion
   switch ncirc
 case 1
mediax=((max([c1x,c2x,c3x])+min([c1x,c2x,c3x]))/2);
% Forma camion
punta = [3 % rectangulo
    4 % numero de lados
    mediax-5 % vertices x
    mediax-3
    mediax-3
    mediax-5
    1 % vertices y
    1
    2.5
    2.5];
frente = [3 % rectangulo
    4 % numero de lados
    mediax-4.5 % vertices x
    mediax-3
    mediax-3
    mediax-4.5
    2.5 % vertices y
    2.5
    3.5
    3.5];
vagon = [3 % rectangulo
    4 % numero de lados
    mediax-3 % vertices x
    mediax+5
    mediax+5
    mediax-3
    1 % vertices y
    1
    4
    4];
llanta1 = [1 % circulo
    mediax-4 % coordinada x del centro
    0.5 % coordinada y del centro
    0.5]; % radio
llanta2 = [1 % circulo
    mediax+2 % coordinada x del centro
    0.5 % coordinada y del centro
    0.5]; % radio
llanta3 = [1 % circulo
    mediax+3.5 % coordinada x del centro
    0.5 % coordinada y del centro
    0.5]; % radio
%espacio
esp=[3 % rectangulo
```

```
    4 % numero de lados
    0 % vertices x
    2*mediax
    2*mediax
    0
    0 % vertices y
    0
    max([C1(3),C2(3),C3(3),vagon(9)])+1
    max([C1(3),C2(3),C3(3),vagon(9)])+1];
% Anexar extra ceros a los circulos para que tengan el mismo numero de
elementos de los rectangulos
C1 = [C1;zeros(length(vagon) - length(C1),1)];
C2 = [C2;zeros(length(vagon) - length(C2),1)];
C3 = [C3;zeros(length(vagon) - length(C3),1)];
llanta1 = [llanta1;zeros(length(vagon) - length(llanta1),1)];
llanta2 = [llanta2;zeros(length(vagon) - length(llanta3),1)];
llanta3 = [llanta3;zeros(length(vagon) - length(llanta3),1)];
% Combinar las formas en una sola matrix
gd = [punta,frente,vagon,llanta1,llanta2,llanta3,esp,C1,C2,C3];
% Asignar nombres a las formas
ns =
char('punta','frente','vagon','llanta1','llanta2','llanta3','esp','C1','C2','C3
');
ns = ns';
%Crea la formula que define la zona a analizar
sf = '(esp)-(punta+frente+vagon+llanta1+llanta2+llanta3+C1+C2+C3)';
[dl,bt] = decsg(gd,sf,ns); % combina las formas usando la formula
[dl2,bt2] = csgdel(dl,bt); % removes face boundaries (boundary removal)

%crear modelo PDE e incluir geometria en el modelo
modelo = createpde;
geometryFromEdges(modelo,dl2);%with  boundary removal
%geometryFromEdges(modelo,dl);%without  boundary removal

%View the geometry
%%{
axes(handles.axes2);
pdegplot(modelo)
axis tight
%}

%aplicar condiciones de frontera

applyBoundaryCondition(modelo,'Edge',1:3,'u',0);
applyBoundaryCondition(modelo,'Edge',7:31,'u',0);
applyBoundaryCondition(modelo,'Edge',32:35,'r',vl1,'h',1);
applyBoundaryCondition(modelo,'Edge',36:39,'r',vl2,'h',1);
applyBoundaryCondition(modelo,'Edge',40:43,'r',vl3,'h',1);
case 2
mediax=((max([c1x,c2x,c3x,c12x,c22x,c32x])+min([c1x,c2x,c3x,c12x,c22x,c32x]))/2
);
% Forma camion
punta = [3 % rectangulo
    4 % numero de lados
    mediax-5 % vertices x
    mediax-3
```

```
    mediax-3
    mediax-5
    1 % vertices y
    1
    2.5
    2.5];
frente = [3 % rectangulo
    4 % numero de lados
    mediax-4.5 % vertices x
    mediax-3
    mediax-3
    mediax-4.5
    2.5 % vertices y
    2.5
    3.5
    3.5];
vagon = [3 % rectangulo
    4 % numero de lados
    mediax-3 % vertices x
    mediax+5
    mediax+5
    mediax-3
    1 % vertices y
    1
    4
    4];
llanta1 = [1 % circulo
    mediax-4 % coordinada x del centro
    0.5 % coordinada y del centro
    0.5]; % radio
llanta2 = [1 % circulo
    mediax+2 % coordinada x del centro
    0.5 % coordinada y del centro
    0.5]; % radio
llanta3 = [1 % circulo
    mediax+3.5 % coordinada x del centro
    0.5 % coordinada y del centro
    0.5]; % radio
%espacio
esp=[3 % rectangulo
    4 % numero de lados
    0 % vertices x
    2*mediax
    2*mediax
    0
    0 % vertices y
    0
    max([C1(3),C2(3),C3(3),C12(3),C22(3),C32(3),vagon(9)])+1
    max([C1(3),C2(3),C3(3),C12(3),C22(3),C32(3),vagon(9)])+1];
% Anexar extra ceros a los circulos para que tengan el mismo numero de
elementos de los rectangulos
C1 = [C1;zeros(length(vagon) - length(C1),1)];
C2 = [C2;zeros(length(vagon) - length(C2),1)];
C3 = [C3;zeros(length(vagon) - length(C3),1)];
C12 = [C12;zeros(length(vagon) - length(C12),1)];
C22 = [C22;zeros(length(vagon) - length(C22),1)];
C32 = [C32;zeros(length(vagon) - length(C32),1)];
```

```
llanta1 = [llanta1;zeros(length(vagon) - length(llanta1),1)];
llanta2 = [llanta2;zeros(length(vagon) - length(llanta3),1)];
llanta3 = [llanta3;zeros(length(vagon) - length(llanta3),1)];
% Combinar las formas en una sola matrix
gd = [punta,frente,vagon,llanta1,llanta2,llanta3,esp,C1,C2,C3,C12,C22,C32];
% Asignar nombres a las formas
ns =
char('punta','frente','vagon','llanta1','llanta2','llanta3','esp','C1','C2','C3
','C12','C22','C32');
ns = ns';
%Crea la formula que define la zona a analizar
sf = '(esp)-(punta+frente+vagon+llanta1+llanta2+llanta3+C1+C2+C3+C12+C22+C32)';
[dl,bt] = decsg(gd,sf,ns); % combina las formas usando la formula
[dl2,bt2] = csgdel(dl,bt); % removes face boundaries (boundary removal)

%crear modelo PDE e incluir geometria en el modelo
modelo = createpde;
geometryFromEdges(modelo,dl2);%with  boundary removal
%geometryFromEdges(modelo,dl);%without  boundary removal

%View the geometry
%%{
axes(handles.axes2);
pdegplot(modelo)
axis tight
%}

%aplicar condiciones de frontera

applyBoundaryCondition(modelo,'Edge',1:3,'u',0);
applyBoundaryCondition(modelo,'Edge',7:31,'u',0);
applyBoundaryCondition(modelo,'Edge',32:35,'r',vl1,'h',1);
applyBoundaryCondition(modelo,'Edge',36:39,'r',vl2,'h',1);
applyBoundaryCondition(modelo,'Edge',40:43,'r',vl3,'h',1);
applyBoundaryCondition(modelo,'Edge',44:47,'r',vl1,'h',1);
applyBoundaryCondition(modelo,'Edge',48:51,'r',vl2,'h',1);
applyBoundaryCondition(modelo,'Edge',52:55,'r',vl3,'h',1);

   end

 case 4 %Tipo Edificio
   switch ncirc
 case 1
mediax=((max([c1x,c2x,c3x])+min([c1x,c2x,c3x]))/2);
% Forma edificio
edif = [3 % rectangulo
    4 % numero de lados
    mediax-3 % vertices x
    mediax+3
    mediax+3
    mediax-3
    0 % vertices y
    0
    12.5
    12.5];
%espacio
```

```
esp=[3 % rectangulo
    4 % numero de lados
    0 % vertices x
    2*mediax
    2*mediax
    0
    0 % vertices y
    0
    max([C1(3),C2(3),C3(3),edif(9)])+1
    max([C1(3),C2(3),C3(3),edif(9)])+1];
% Anexar extra ceros a los circulos para que tengan el mismo numero de
elementos de los rectangulos
C1 = [C1;zeros(length(edif) - length(C1),1)];
C2 = [C2;zeros(length(edif) - length(C2),1)];
C3 = [C3;zeros(length(edif) - length(C3),1)];
% Combinar las formas en una sola matrix
gd = [edif,esp,C1,C2,C3];
% Asignar nombres a las formas
ns = char('edif','esp','C1','C2','C3');
ns = ns';
%Crea la formula que define la zona a analizar
sf = '(esp)-(edif+C1+C2+C3)';
[dl,bt] = decsg(gd,sf,ns); % combina las formas usando la formula
[dl2,bt2] = csgdel(dl,bt); % removes face boundaries (boundary removal)

%crear modelo PDE e incluir geometria en el modelo
modelo = createpde;
geometryFromEdges(modelo,dl2);%with  boundary removal

%View the geometry
%%{
axes(handles.axes2);
pdegplot(modelo)
axis tight
%}

%aplicar condiciones de frontera
applyBoundaryCondition(modelo,'Edge',1:2,'u',0);
applyBoundaryCondition(modelo,'Edge',6:8,'u',0);
applyBoundaryCondition(modelo,'Edge',9:12,'r',vl1,'h',1);
applyBoundaryCondition(modelo,'Edge',13:16,'r',vl2,'h',1);
applyBoundaryCondition(modelo,'Edge',17:20,'r',vl3,'h',1);
case 2
mediax=((max([c1x,c2x,c3x,c12x,c22x,c32x])+min([c1x,c2x,c3x,c12x,c22x,c32x]))/2
);
% Forma edificio
edif = [3 % rectangulo
    4 % numero de lados
    mediax-3 % vertices x
    mediax+3
    mediax+3
    mediax-3
    0 % vertices y
    0
    12.5
    12.5];
```

```
%espacio
esp=[3 % rectangulo
    4 % numero de lados
    0 % vertices x
    2*mediax
    2*mediax
    0
    0 % vertices y
    0
    max([C1(3),C2(3),C3(3),C12(3),C22(3),C32(3),edif(9)])+1
    max([C1(3),C2(3),C3(3),C12(3),C22(3),C32(3),edif(9)])+1];
% Anexar extra ceros a los circulos para que tengan el mismo numero de
elementos de los rectangulos
C1 = [C1;zeros(length(edif) - length(C1),1)];
C2 = [C2;zeros(length(edif) - length(C2),1)];
C3 = [C3;zeros(length(edif) - length(C3),1)];
C12 = [C12;zeros(length(edif) - length(C12),1)];
C22 = [C22;zeros(length(edif) - length(C22),1)];
C32 = [C32;zeros(length(edif) - length(C32),1)];
% Combinar las formas en una sola matrix
gd = [edif,esp,C1,C2,C3,C12,C22,C32];
% Asignar nombres a las formas
ns = char('edif','esp','C1','C2','C3','C12','C22','C32');
ns = ns';
%Crea la formula que define la zona a analizar
sf = '(esp)-(edif+C1+C2+C3+C12+C22+C32)';
[dl,bt] = decsg(gd,sf,ns); % combina las formas usando la formula
[dl2,bt2] = csgdel(dl,bt); % removes face boundaries (boundary removal)

%crear modelo PDE e incluir geometria en el modelo
modelo = createpde;
geometryFromEdges(modelo,dl2);%with  boundary removal

%View the geometry
%%{
axes(handles.axes2);
pdegplot(modelo)
axis tight
%}

%aplicar condiciones de frontera
applyBoundaryCondition(modelo,'Edge',1:2,'u',0);
applyBoundaryCondition(modelo,'Edge',6:8,'u',0);
applyBoundaryCondition(modelo,'Edge',9:12,'r',vl1,'h',1);
applyBoundaryCondition(modelo,'Edge',13:16,'r',vl2,'h',1);
applyBoundaryCondition(modelo,'Edge',17:20,'r',vl3,'h',1);
applyBoundaryCondition(modelo,'Edge',21:24,'r',vl1,'h',1);
applyBoundaryCondition(modelo,'Edge',25:28,'r',vl2,'h',1);
applyBoundaryCondition(modelo,'Edge',29:32,'r',vl3,'h',1);

   end
case 5 %Sin objeto
  switch ncirc
 case 1
mediax=((max([c1x,c2x,c3x])+min([c1x,c2x,c3x]))/2);
%espacio
```

```
esp=[3 % rectangulo
    4 % numero de lados
    0 % vertices x
    2*mediax
    2*mediax
    0
    0 % vertices y
    0
    max([C1(3),C2(3),C3(3)])+1
    max([C1(3),C2(3),C3(3)])+1];
% Anexar extra ceros a los circulos para que tengan el mismo numero de
elementos de los rectangulos
C1 = [C1;zeros(length(esp) - length(C1),1)];
C2 = [C2;zeros(length(esp) - length(C2),1)];
C3 = [C3;zeros(length(esp) - length(C3),1)];
% Combinar las formas en una sola matrix
gd = [esp,C1,C2,C3];
% Asignar nombres a las formas
ns = char('esp','C1','C2','C3');
ns = ns';
%Crea la formula que define la zona a analizar
sf = '(esp)-(C1+C2+C3)';
[dl,bt] = decsg(gd,sf,ns); % combina las formas usando la formula
[dl2,bt2] = csgdel(dl,bt); % removes face boundaries (boundary removal)

%crear modelo PDE e incluir geometria en el modelo
modelo = createpde;
geometryFromEdges(modelo,dl2);%with  boundary removal

%View the geometry
%%{
axes(handles.axes2);
pdegplot(modelo)
axis tight
%}

%aplicar condiciones de frontera
applyBoundaryCondition(modelo,'Edge',4,'u',0);
applyBoundaryCondition(modelo,'Edge',5:8,'r',vl1,'h',1);
applyBoundaryCondition(modelo,'Edge',9:12,'r',vl2,'h',1);
applyBoundaryCondition(modelo,'Edge',13:16,'r',vl3,'h',1);
case 2
mediax=((max([c1x,c2x,c3x,c12x,c22x,c32x])+min([c1x,c2x,c3x,c12x,c22x,c32x]))/2
);
%espacio
esp=[3 % rectangulo
    4 % numero de lados
    0 % vertices x
    2*mediax
    2*mediax
    0
    0 % vertices y
    0
    max([C1(3),C2(3),C3(3),C12(3),C22(3),C32(3)])+1
    max([C1(3),C2(3),C3(3),C12(3),C22(3),C32(3)])+1];
```

```
% Anexar extra ceros a los circulos para que tengan el mismo numero de
elementos de los rectangulos
C1 = [C1;zeros(length(esp) - length(C1),1)];
C2 = [C2;zeros(length(esp) - length(C2),1)];
C3 = [C3;zeros(length(esp) - length(C3),1)];
C12 = [C12;zeros(length(esp) - length(C12),1)];
C22 = [C22;zeros(length(esp) - length(C22),1)];
C32 = [C32;zeros(length(esp) - length(C32),1)];
% Combinar las formas en una sola matrix
gd = [esp,C1,C2,C3,C12,C22,C32];
% Asignar nombres a las formas
ns = char('esp','C1','C2','C3','C12','C22','C32');
ns = ns';
%Crea la formula que define la zona a analizar
sf = '(esp)-(C1+C2+C3+C12+C22+C32)';
[dl,bt] = decsg(gd,sf,ns); % combina las formas usando la formula
[dl2,bt2] = csgdel(dl,bt); % removes face boundaries (boundary removal)

%crear modelo PDE e incluir geometria en el modelo
modelo = createpde;
geometryFromEdges(modelo,dl2);%with  boundary removal

%View the geometry
%%{
axes(handles.axes2);
pdegplot(modelo)
axis tight
%}

%aplicar condiciones de frontera
applyBoundaryCondition(modelo,'Edge',4,'u',0);
applyBoundaryCondition(modelo,'Edge',5:8,'r',vl1,'h',1);
applyBoundaryCondition(modelo,'Edge',9:12,'r',vl2,'h',1);
applyBoundaryCondition(modelo,'Edge',13:16,'r',vl3,'h',1);
applyBoundaryCondition(modelo,'Edge',17:20,'r',vl1,'h',1);
applyBoundaryCondition(modelo,'Edge',21:24,'r',vl2,'h',1);
applyBoundaryCondition(modelo,'Edge',25:28,'r',vl3,'h',1);

  end
end

%Coeficientes basados en d2V=0 laplace sin carga en el aire
specifyCoefficients(modelo,'m',0,'d',0,'c',1,'a',0,'f',0);

%generar la malla(mesh)
generateMesh(modelo,'Hmax',0.5);

%View the mesh
%%{
axes(handles.axes3);
pdeplot(modelo)
axis tight
%}

%solve the PDE
```

```
results = solvepde(modelo);
%to use guidata, make your variable part of the handles structure, that is,
handles.results=results;
guidata(hObject, handles);
%Ver los resultados
%%{
axes(handles.axes4);
%pdeplot(modelo,'XYData',results.NodalSolution) %solucion tipica
pdeplot(modelo,'XYData',results.NodalSolution,'FlowData',[-results.XGradients,-
results.YGradients]) %solucion con lineas de campo
axis tight
%}
%ejemplo graficar
%x=linspace(0,10,100);
%plot(handles.axes2,x,cos(x));
%o tambien
%axes(handles.axes2);
%plot(x,cos(x));
%
%Corriente de cortocircuito inducida
airp=8.8541878176e-12;
switch ncirc
    case 1
medcas=((max([c1x,c2x,c3x])+min([c1x,c2x,c3x]))/2);
    case 2
medcas=((max([c1x,c2x,c3x,c12x,c22x,c32x])+min([c1x,c2x,c3x,c12x,c22x,c32x]))/2
);
end
switch tipo
    case 1

[Gsolx1,Gsoly1]=evaluateGradient(results,medcas-5,0);%[kVolts/m]
Eabs1=sqrt((Gsolx1^2));%[kVolts/m] en y
Em1=sqrt((Gsolx1^2)+(Gsoly1^2))
[Gsolx2,Gsoly2]=evaluateGradient(results,medcas-5,2);%[kVolts/m]
Eabs2=sqrt((Gsolx2^2));%[kVolts/m] en y
Em2=sqrt((Gsolx2^2)+(Gsoly2^2))
[Gsolx3,Gsoly3]=evaluateGradient(results,medcas-5,4);%[kVolts/m]
Eabs3=sqrt((Gsolx3^2));%[kVolts/m] en y
Em3=sqrt((Gsolx3^2)+(Gsoly3^2))
[Gsolx4,Gsoly4]=evaluateGradient(results,medcas-4,5);%[kVolts/m]
Eabs4=sqrt((Gsoly4^2));%[kVolts/m] en y
Em4=sqrt((Gsolx4^2)+(Gsoly4^2))
[Gsolx5,Gsoly5]=evaluateGradient(results,medcas-2,5);%[kVolts/m]
Eabs5=sqrt((Gsoly5^2));%[kVolts/m] en y
Em5=sqrt((Gsolx5^2)+(Gsoly5^2))
[Gsolx6,Gsoly6]=evaluateGradient(results,medcas+2,5);%[kVolts/m]
Eabs6=sqrt((Gsoly6^2));%[kVolts/m] en y
Em6=sqrt((Gsolx6^2)+(Gsoly6^2))
[Gsolx7,Gsoly7]=evaluateGradient(results,medcas+4,5);%[kVolts/m]
Eabs7=sqrt((Gsoly7^2));%[kVolts/m] en y
Em7=sqrt((Gsolx7^2)+(Gsoly7^2))
[Gsolx8,Gsoly8]=evaluateGradient(results,medcas+5,4);%[kVolts/m]
Eabs8=sqrt((Gsolx8^2));%[kVolts/m] en y
Em8=sqrt((Gsolx8^2)+(Gsoly8^2))
[Gsolx9,Gsoly9]=evaluateGradient(results,medcas+5,2);%[kVolts/m]
Eabs9=sqrt((Gsolx9^2));%[kVolts/m] en y
```

```
Em9=sqrt((Gsolx9^2)+(Gsoly9^2))
[Gsolx10,Gsoly10]=evaluateGradient(results,medcas+5,0);%[kVolts/m]
Eabs10=sqrt((Gsolx10^2));%[kVolts/m] en y
Em10=sqrt((Gsolx10^2)+(Gsoly10^2))


Eavg=(Eabs1+Eabs2+Eabs3+Eabs4+Eabs5+Eabs6+Eabs7+Eabs8+Eabs9+Eabs10)/10;
Seq=5.75*60;
Isc=2*pi*60*Eavg*Seq*airp*(1e6);

    case 2
[Gsolx1,Gsoly1]=evaluateGradient(results,medcas-0.3,0.1);%[kVolts/m]
Eabs1=sqrt((Gsolx1^2));%[kVolts/m] en y
Em1=sqrt((Gsolx1^2)+(Gsoly1^2))
[Gsolx2,Gsoly2]=evaluateGradient(results,medcas-0.15,1);%[kVolts/m]
Eabs2=sqrt((Gsolx2^2));%[kVolts/m] en y
Em2=sqrt((Gsolx2^2)+(Gsoly2^2))
[Gsolx3,Gsoly3]=evaluateGradient(results,medcas-0.4,1.3);%[kVolts/m]
Eabs3=sqrt((Gsolx3^2));%[kVolts/m] en y
Em3=sqrt((Gsolx3^2)+(Gsoly3^2))
[Gsolx4,Gsoly4]=evaluateGradient(results,medcas-0.4,2);%[kVolts/m]
Eabs4=sqrt((Gsoly4^2));%[kVolts/m] en y
Em4=sqrt((Gsolx4^2)+(Gsoly4^2))
[Gsolx5,Gsoly5]=evaluateGradient(results,medcas-0.3,2);%[kVolts/m]
Eabs5=sqrt((Gsolx5^2));%[kVolts/m] en y
Em5=sqrt((Gsolx5^2)+(Gsoly5^2))
[Gsolx6,Gsoly6]=evaluateGradient(results,medcas+0.3,2);%[kVolts/m]
Eabs6=sqrt((Gsolx6^2));%[kVolts/m] en y
Em6=sqrt((Gsolx6^2)+(Gsoly6^2))
[Gsolx7,Gsoly7]=evaluateGradient(results,medcas+0.4,2);%[kVolts/m]
Eabs7=sqrt((Gsoly7^2));%[kVolts/m] en y
Em7=sqrt((Gsolx7^2)+(Gsoly7^2))
[Gsolx8,Gsoly8]=evaluateGradient(results,medcas+0.4,1.3);%[kVolts/m]
Eabs8=sqrt((Gsolx8^2));%[kVolts/m] en y
Em8=sqrt((Gsolx8^2)+(Gsoly8^2))
[Gsolx9,Gsoly9]=evaluateGradient(results,medcas+0.15,1);%[kVolts/m]
Eabs9=sqrt((Gsolx9^2));%[kVolts/m] en y
Em9=sqrt((Gsolx9^2)+(Gsoly9^2))
[Gsolx10,Gsoly10]=evaluateGradient(results,medcas+0.3,1);%[kVolts/m]
Eabs10=sqrt((Gsolx10^2));%[kVolts/m] en y
Em10=sqrt((Gsolx10^2)+(Gsoly10^2))

Eavg=(Eabs1+Eabs2+Eabs3+Eabs4+Eabs5+Eabs6+Eabs7+Eabs8+Eabs9+Eabs10)/10;
Seq=5.395;
Isc=2*pi*60*Eavg*Seq*airp*(1e6);

    case 3
[Gsolx1,Gsoly1]=evaluateGradient(results,medcas-5,1.75);%[kVolts/m]
Eabs1=sqrt((Gsolx1^2));%[kVolts/m] en y
Em1=sqrt((Gsolx1^2)+(Gsoly1^2))
[Gsolx2,Gsoly2]=evaluateGradient(results,medcas-4.5,3);%[kVolts/m]
Eabs2=sqrt((Gsolx2^2));%[kVolts/m] en y
Em2=sqrt((Gsolx2^2)+(Gsoly2^2))
[Gsolx3,Gsoly3]=evaluateGradient(results,medcas-3,4);%[kVolts/m]
Eabs3=sqrt((Gsoly3^2));%[kVolts/m] en y
Em3=sqrt((Gsolx3^2)+(Gsoly3^2))
```

```
[Gsolx4,Gsoly4]=evaluateGradient(results,medcas-1,4);%[kVolts/m]
Eabs4=sqrt((Gsoly4^2));%[kVolts/m] en y
Em4=sqrt((Gsolx4^2)+(Gsoly4^2))
[Gsolx5,Gsoly5]=evaluateGradient(results,medcas+1,4);%[kVolts/m]
Eabs5=sqrt((Gsoly5^2));%[kVolts/m] en y
Em5=sqrt((Gsolx5^2)+(Gsoly5^2))
[Gsolx6,Gsoly6]=evaluateGradient(results,medcas+3,4);%[kVolts/m]
Eabs6=sqrt((Gsoly6^2));%[kVolts/m] en y
Em6=sqrt((Gsolx6^2)+(Gsoly6^2))
[Gsolx7,Gsoly7]=evaluateGradient(results,medcas+5,4);%[kVolts/m]
Eabs7=sqrt((Gsoly7^2));%[kVolts/m] en y
Em7=sqrt((Gsolx7^2)+(Gsoly7^2))
[Gsolx8,Gsoly8]=evaluateGradient(results,medcas+5,3);%[kVolts/m]
Eabs8=sqrt((Gsolx8^2));%[kVolts/m] en y
Em8=sqrt((Gsolx8^2)+(Gsoly8^2))
[Gsolx9,Gsoly9]=evaluateGradient(results,medcas+5,2);%[kVolts/m]
Eabs9=sqrt((Gsolx9^2));%[kVolts/m] en y
Em9=sqrt((Gsolx9^2)+(Gsoly9^2))
[Gsolx10,Gsoly10]=evaluateGradient(results,medcas+5,1);%[kVolts/m]
Eabs10=sqrt((Gsolx10^2));%[kVolts/m] en y
Em10=sqrt((Gsolx10^2)+(Gsoly10^2))

Eavg=(Eabs1+Eabs2+Eabs3+Eabs4+Eabs5+Eabs6+Eabs7+Eabs8+Eabs9+Eabs10)/10;
Seq=9.2*20;
Isc=2*pi*60*Eavg*Seq*airp*(1e6);

    case 4
[Gsolx1,Gsoly1]=evaluateGradient(results,medcas-3,0);%[kVolts/m]
Eabs1=sqrt((Gsolx1^2));%[kVolts/m] en y
Em1=sqrt((Gsolx1^2)+(Gsoly1^2))
[Gsolx2,Gsoly2]=evaluateGradient(results,medcas-3,2);%[kVolts/m]
Eabs2=sqrt((Gsolx2^2));%[kVolts/m] en y
Em2=sqrt((Gsolx2^2)+(Gsoly2^2))
[Gsolx3,Gsoly3]=evaluateGradient(results,medcas-3,4);%[kVolts/m]
Eabs3=sqrt((Gsolx3^2));%[kVolts/m] en y
Em3=sqrt((Gsolx3^2)+(Gsoly3^2))
[Gsolx4,Gsoly4]=evaluateGradient(results,medcas-2,12.5);%[kVolts/m]
Eabs4=sqrt((Gsoly4^2));%[kVolts/m] en y
Em4=sqrt((Gsolx4^2)+(Gsoly4^2))
[Gsolx5,Gsoly5]=evaluateGradient(results,medcas-1,12.5);%[kVolts/m]
Eabs5=sqrt((Gsoly5^2));%[kVolts/m] en y
Em5=sqrt((Gsolx5^2)+(Gsoly5^2))
[Gsolx6,Gsoly6]=evaluateGradient(results,medcas+1,12.5);%[kVolts/m]
Eabs6=sqrt((Gsoly6^2));%[kVolts/m] en y
Em6=sqrt((Gsolx6^2)+(Gsoly6^2))
[Gsolx7,Gsoly7]=evaluateGradient(results,medcas+2,12.5);%[kVolts/m]
Eabs7=sqrt((Gsoly7^2));%[kVolts/m] en y
Em7=sqrt((Gsolx7^2)+(Gsoly7^2))
[Gsolx8,Gsoly8]=evaluateGradient(results,medcas+3,4);%[kVolts/m]
Eabs8=sqrt((Gsolx8^2));%[kVolts/m] en y
Em8=sqrt((Gsolx8^2)+(Gsoly8^2))
[Gsolx9,Gsoly9]=evaluateGradient(results,medcas+3,2);%[kVolts/m]
Eabs9=sqrt((Gsolx9^2));%[kVolts/m] en y
Em9=sqrt((Gsolx9^2)+(Gsoly9^2))
[Gsolx10,Gsoly10]=evaluateGradient(results,medcas+3,0);%[kVolts/m]
Eabs10=sqrt((Gsolx10^2));%[kVolts/m] en y
```

```
Em10=sqrt((Gsolx10^2)+(Gsoly10^2))

Eavg=(Eabs1+Eabs2+Eabs3+Eabs4+Eabs5+Eabs6+Eabs7+Eabs8+Eabs9+Eabs10)/10;
Seq=12.5*60;
Isc=2*pi*60*Eavg*Seq*airp*(1e6);
    case 5
Isc='No aplica';
end
set(handles.text28,'String',Isc);
```

```
function edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%        str2double(get(hObject,'String')) returns contents of edit8 as a
double


% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%        str2double(get(hObject,'String')) returns contents of edit9 as a
double


% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit10_Callback(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%        str2double(get(hObject,'String')) returns contents of edit10 as a
double


% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit11_Callback(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text
%        str2double(get(hObject,'String')) returns contents of edit11 as a
double


% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
exstr=get(handles.edit10,'String'); %coordenada x de tension como texto
eystr=get(handles.edit11,'String'); %coordenada y del de tension como texto
ex=str2double(exstr);
ey=str2double(eystr);
results=handles.results;
[Gsolx,Gsoly]=evaluateGradient(results,ex,ey);%[kVolts/m]
Esolx=-Gsolx
Esoly=-Gsoly
Eabs=sqrt((Esolx^2)+(Esoly^2));%[kVolts/m] en y
set(handles.text24,'String',Eabs);

% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton1


% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton2


% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents
as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu1


% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit12_Callback(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit12 as text
%        str2double(get(hObject,'String')) returns contents of edit12 as a
double


% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit13_Callback(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit13 as text
%        str2double(get(hObject,'String')) returns contents of edit13 as a
double


% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit14_Callback(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit14 as text
%        str2double(get(hObject,'String')) returns contents of edit14 as a
double


% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit16_Callback(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit16 as text
%        str2double(get(hObject,'String')) returns contents of edit16 as a
double


% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end



function edit17_Callback(hObject, eventdata, handles)
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit17 as text
%        str2double(get(hObject,'String')) returns contents of edit17 as a
double


% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit18_Callback(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit18 as text
%        str2double(get(hObject,'String')) returns contents of edit18 as a
double


% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit19_Callback(hObject, eventdata, handles)
% hObject    handle to edit19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit19 as text
%        str2double(get(hObject,'String')) returns contents of edit19 as a
double


% --- Executes during object creation, after setting all properties.
function edit19_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit20_Callback(hObject, eventdata, handles)
% hObject    handle to edit20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit20 as text
%        str2double(get(hObject,'String')) returns contents of edit20 as a
double


% --- Executes during object creation, after setting all properties.
function edit20_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit21_Callback(hObject, eventdata, handles)
% hObject    handle to edit21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit21 as text
%        str2double(get(hObject,'String')) returns contents of edit21 as a
double


% --- Executes during object creation, after setting all properties.
function edit21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu3 contents
as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu3


% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```